Printed Version of PNA Help

Agilent Technologies
2-Port and 3-Port
PNA Series Network Analyzers

Manufacturing Part Number: N5230-90017

Printed in USA

Print Date: October 17, 2005

Supersedes: August 2, 2005

© Agilent Technologies, Inc. 2004 - 2005 All rights reserved.

Documentation Warranty

THE MATERIAL CONTAINED IN THIS DOCUMENT IS PROVIDED "AS IS," AND IS SUBJECT TO BEING CHANGED, WITHOUT NOTICE, IN FUTURE EDITIONS. FURTHER, TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, AGILENT DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED WITH REGARD TO THIS MANUAL AND ANY INFORMATION CONTAINED HEREIN, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. AGILENT SHALL NOT BE LIABLE FOR ERRORS OR FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH THE FURNISHING, USE, OR PERFORMANCE OF THIS DOCUMENT OR ANY INFORMATION CONTAINED HEREIN. SHOULD AGILENT AND THE USER HAVE A SEPARATE WRITTEN AGREEMENT WITH WARRANTY TERMS COVERING THE MATERIAL IN THIS DOCUMENT THAT CONFLICT WITH THESE TERMS, THE WARRANTY TERMS IN THE SEPARATE AGREEMENT WILL CONTROL.

DFARS/Restricted Rights Notice

If software is for use in the performance of a U.S. Government prime contract or subcontract, Software is delivered and licensed as “Commercial computer software” as defined in DFAR 252.227-7014 (June 1995), or as a “commercial item” as defined in FAR 2.101(a) or as “Restricted computer software” as defined in FAR 52.227-19 (June 1987) or any equivalent agency regulation or contract clause. Use, duplication or disclosure of Software is subject to Agilent Technologies’ standard commercial license terms, and non-DOD Departments and Agencies of the U.S. Government will receive no greater than Restricted Rights as defined in FAR 52.227-19(c)(1-2) (June 1987). U.S. Government users will receive no greater than Limited Rights as defined in FAR 52.227-14 (June 1987) or DFAR 252.227-7015 (b)(2) (November 1995), as applicable in any technical data.

Printing Copies of Documentation from the Web

To print copies of documentation from the Web, download the PDF file from the Agilent web site:

- Go to http://www.agilent.com.
- Enter the document’s part number (located on the title page) in the **Quick Search** box.
- Click GO.
- Click on the hyperlink for the document.
- Click the printer icon located in the tool bar.

Contacting Agilent

This information supersedes all prior HP contact information.			
Online assistance: www.agilent.com/find/assist			
Americas			
Brazil *(tel)* (+55) 11 3351 7012 *(fax)* (+55) 11 3351 7024	**Canada** *(tel)* +1 877 894 4414 *(alt)* +1 303 662 3369 *(fax)* +1 800 746 4866	**Mexico** *(tel)* 1800 254 2440 Ext 2703 *(alt) from USA* 18008374039 *(fax)* 1 800 254 422	**United States** *(tel)* 800 829 4444 *(alt)* (+1) 303 662 3998 *(fax)* 800 829 4433
Asia Pacific and Japan			
Australia *(tel)* 1 800 225 574 *(fax)* 1 800 681 776 *(fax)* 1 800 225 539	**China** *(tel)* 800 810 0189 *(fax)* 800 820 2816	**Hong Kong** *(tel)* 800 933 229 *(fax)* 800 900 701	**India** *(tel)* 1600 112 626 *(alt)* +65 6275 0800 *(fax)* 1600 113 040
Japan (Bench) *(tel)* 0120 421 345 *(alt)* (+81) 426 56 7832 *(fax)* 0120 01 2144	**Japan (On-Site)** *(tel)* 0120 421 345 *(alt)* (+81) 426 56 7832 *(fax)* 0120 012 114	**Malaysia** *(tel)* 1800 880 399 *(fax)* 1800 801 054	**New Zealand** *(tel)* +64 4 939 0635 *(alt)* 0800 738 378 *(fax)* +64 4 972 5364
Singapore *(tel)* 1 800 275 0880 *(fax)* (+65) 6755 1214	**South Korea** *(tel)* 080 778 0011 *(fax)* 080 778 0013	**Taiwan** *(tel)* 0800 047 669 *(fax)* 0800 047 667 *(fax)* +886 3492 0779	**Thailand** *(tel)* +66 2 267 5913 *(tel)* 1 800 2758 5822 *(fax)* 1 800 653 336
Europe			
Austria *(tel)* 0820 87 44 11* *(fax)* 0820 87 44 22	**Belgium** *(tel)* (+32) (0)2 404 9340 *(fax)* (+32) (0)2 404 9395	**Denmark** *(tel)* (+45) 7013 1515 *(fax)* (+45) 7013 1555	**Finland** *(tel)* (+358) (0) 10 855 2100 *(fax)* (+358) (0) 10 855 2923
France *(tel)* 0825 010 700* *(fax)* 0825 010 701*	**Germany** *(tel)* 01805 24 6333* *(fax)* 01805 24 6336*	**Ireland** *(tel)* (+353) 1 890 924 204 *(fax)* 1 890 924 024	**Israel** *(tel)* (+972) 3 9288 504 *(alt)* (+972) 3 9288 544 *(fax)* (+972) 3 9288 520
Italy *(tel)* (+39) (0)2 9260 8484 *(fax)* (+39) (0)2 9544 1175	**Luxemburg** *(tel)* (+32) (0)2 404 9340 *(fax)* (+32) (0)2 404 9395	**Netherlands** *(tel)* (+31) (0)20 547 2111 *(fax)* (+31) (0)20 547 2190	**Russia** *(tel)* (+7) 095 797 3963 *(alt)* (+7) 095 797 3900 *(fax)* (+7) 095 797 3901
Spain *(tel)* (+34) 91 631 3300 *(fax)* (+34) 91 631 3301	**Sweden** *(tel)* 0200 88 22 55* *(alt)* (+46) (0)8 5064 8686 *(fax)* 020 120 2266*	**Switzerland (French)** *(tel)* 0800 80 5353 opt. 2* *(fax)* (0) 22 567 5313	**Switzerland (German)** *(tel)* 0800 80 5353 opt. 1* *(fax)* 0 44 272 7373
Switzerland (Italian) *(tel)* 0800 80 5353 opt. 3* *(fax)* (0) 22 567 5314	**United Kingdom** *(tel)* (+44) (0)7004 666666 *(fax)* (+44) (0)7004 444555		
(tel) = primary telephone number; *(alt)* = alternate telephone number; *(fax)* = FAX number; * = in country number 8/10/05			

This page intentionally left blank.

Whats New 28
Administrative Tasks
PNA User Accounts and Passwords 33
Computer Properties 36
Operating System Recovery 39
Windows Considerations 42
Quick Start
Front Panel Tour 44
Rear Panel Tour 45
Powering the PNA ON and OFF 46
PNA User Interface 49
Traces, Channels, and Windows on the PNA 51
Basic Measurement Sequence 54
Using Help 55
1. Set Up a Measurement
Preset the PNA 63
Measurement Parameters 70
Frequency Range 75
Power Level 80
Sweep Settings 87
Trigger 96
Data Format and Scale 101
Preconfigured Measurement Setups 109
Customize Your Analyzer Screen 113
Copy Channels 121
2. Optimize
Dynamic Range 124
Dynamic Range-4 Jumpers 126
Dynamic Range-Test Set Option 127
Number of Data Points 128
Phase Accuracy 130
Electrically Long Devices 135
Reflection Accuracy 137
Measurement Stability 140
Noise Reduction Techniques 142
Crosstalk 148

Effects of Accessories 149
Fastest Sweep 150
Multiple State Measurements 152
Fastest Data Transfer 155
Using Macros 156
3. Calibrate
Calibration Overview 160
Select a Calibration 162
Calibration Wizard 166
Using Cal Sets 179
ECAL 186
Accurate Calibrations 204
Validity of a Cal 210
Measurement Errors 215
Modify Cal Kits 223
3-Port Measurements 238
Power Calibration 240
Calibration Standards 249
TRL Calibration 253
Test Fixture Macro 257
4. Analyze Data
Locate Data Using Markers 263
Math & Memory Operations 272
Use Limits to Test Devices 277
5. Output Data
Save and Recall Data 282
Drive Mapping 290
Print 292
Programming
Command Search
Command Search 297
COM
Commands
Objects
The Analyzer Object Model 298
Application Object 299
CalFactorSegments Collection 304

Calibrator Object 305
CalKit Object 309
CalManager Object 311
CalSet Object 313
CalSets Collection 317
CalStandard Object 318
Capabilities Object 320
Channel Object 322
Channels Collection 328
Gating Object 329
HWAuxIO Object 331
HWExternalTestSetIO Object 333
HWMaterialHandlerIO Object 334
IIFConfiguration Object 336
IMixer Interface 338
Limit Test Collection 341
LimitSegment Object 342
Marker Object 343
Measurement Object 346
Measurements Collection 352
NAWindow Object 353
NAWindows Collection 355
Port Extension Object 356
PowerLossSegment Object 357
PowerLossSegments Collection 359
PowerSensor Object 360
PowerSensorCalFactorSegment Object 362
PowerSensors Collection 363
Preferences Object 364
SCPIStringParser Object 365
Segment Object 366
Segments Collection 368
SMC Type Object 370
SourcePowerCalibrator Object 372
Trace Object 374
Traces Collection 376

Transform Object 377
TriggerSetup Object 379
VMC Type Object 380
Properties
AcceptTriggerBeforeArmed 382
AcquisitionDirection 383
Active Cal Kit 384
Active Channel 385
Active Marker 386
Active Measurement 387
Active NAWindow 388
ActiveXAxisRange 389
Active Trace 390
AllowArbitrarySegments 391
Alternate Sweep 392
Application 393
Arrange Windows 394
Attenuator Mode 395
Attenuator 396
AutoOrient 397
Averaging Count 398
Averaging Factor 399
Averaging ON/OFF 400
AvoidSpurs 401
Bandwidth Target 402
Bandwidth Tracking 403
Begin Response 404
Begin Stimulus 405
Bucket Number 406
C0 407
C1 408
C2 409
C3 410
Cal Factor 411
Cal Type (applied) 412
Calibration Name 413

Calibration Port 414
Calibration TypeID 415
Cal KitType 416
CalKitType (FCA) 417
Cal Power 418
Center 419
Center (Meas) 420
Center Frequency 421
Channel Number 422
CharacterizeMixerOnly 423
CharFileName 424
CharMixerReverse 425
CitiContents 426
CitiFormat 427
CompatibleCalKits 428
ConnectorType 429
Count 430
Couple Ports 431
CoupleChannelParams 432
Coupled Markers 433
CoupledParameters - Gate 434
CoupledParameters - Transform 435
CW Frequency 436
Delay 437
DeltaMarker 438
Description 439
Display Format 440
DisplayAutomationErrors 441
DisplayGlobalPassFail 442
Distance 443
DistanceMarkerMode 444
DistanceMarkerUnit 445
Do1PortEcal 446
Do2PortEcal 447
Domain 448
Dwell Time 449

ECALCharacterization	450
ECALCharacterization (smc)	451
ECALCharacterization (vmc)	452
ECALCharacterizationEx	453
ECALCharacterizationIndexList	454
ECALIsolation	455
ECALModuleNumberList	456
EcalOrientation	457
EcalOrientation1Port	459
EcalOrientation2Port	460
ECALPortMapEx	461
ElecDelay Medium	463
Electrical Delay	464
Error Correction	465
External ALC	466
ExternalTriggerConnectionBehavior	467
ExternalTriggerDelay	469
Filter BW	470
Filter CF	471
Filter Loss	472
Filter Q	473
FirmwareMajorRevision	474
FirmwareMinorRevision	475
FirmwareSeries	476
FootSwitch	477
Footswitch Mode	478
Format (Marker)	479
Format	440
Frequency	480
Frequency Span	481
Frequency Offset Divisor	482
Frequency Offset Frequency	483
Frequency Offset Multiplier	484
Frequency Offset Override To CW	485
Frequency Offset State	486
Gate Shape	487

Gate Type 488
GPIBAddress 489
GPIB Mode 490
IDString 491
IF Bandwidth Option 492
IF Bandwidth 493
IFDenominator 494
IFFilterSampleCount 495
IFFilterSamplePeriod 496
IFFilterSamplePeriodList 497
IFFilterSamplePeriodMode 498
IFFilterSource Property 499
IFGainLevel 500
IFGainMode 501
IFGateEnable 502
IFNumerator 503
IFSideband 504
IFSourcePath 505
IFStartFrequency 506
IFStopFrequency 507
Impulse Width 508
IndexState 509
Input A 510
Input B 511
Input C 512
InputDenominator 513
InputIsGreaterThanLO 514
InputNumerator 515
InputPower 516
InputStartFrequency 517
InputStopFrequency 518
InternalTestsetPortCount 519
Interpolate Correction 520
Interpolate Normalization 521
Interrupt 522
IsContinuous 523

IsECALModuleFoundEx	524
IsFrequencyOffsetPresent	525
IsHold	526
IsReceiverStepAttenuatorPresent	527
IsReferenceBypassSwitchPresent	528
IsSParameter	529
IterationsTolerance	530
Kaiser Beta	531
L0	532
L1	533
L2	534
L3	535
Label	536
LastModified	537
LimitTestFailed	538
Limit Line Begin Stimulus	405
Limit Line End Stimulus	539
Limit Line Begin Response	404
Limit Line End Response	540
Limit Type	541
Line Display	542
LoadCharFromFile	543
LoadPort	544
LocalLockoutState	545
LODenominator	546
LOFixedFrequency	547
LogMagnitudeOffset	548
LOName	549
LONumerator	550
LOPower	551
LORangeMode	552
Loss	553
Loss (sourceCal)	554
LOStage	555
LOStartFrequency	556
LOStopFrequency	557

MagnitudeOffset	558
MagnitudeSlopeOffset	559
Marker Bucket Number	406
Marker Format (all)	560
Marker Format (indiv)	479
Marker Interpolate(all)	561
Marker Interpolate (indiv)	562
Marker Number	563
MarkerReadout	564
MarkerReadoutSize	565
MarkerState	566
Marker Type	567
Marker X-axis Value	568
Marker Y-axis Value	569
Maximum Frequency	570
MaximumFrequency (capabilities)	571
MaximiumFrequency (sourceCal)	572
MaximumIFFilterSampleCount	573
MaximumIterationsPerPoint	574
MaximumNumberOfPoints	575
MaximumReceiverStepAttenuator	576
MaximumSourceALCPower	577
MaximumSourceStepAttenuator	578
Mean	579
Medium	580
Minimum Frequency	581
MinimumFrequency (capabilities)	582
MinimumFrequency (sourceCal)	583
MinimumIFFilterSampleCount	584
MinimumNumberOfPoints	585
MinimumReceiverStepAttenuator	586
MinimumSourceALCPower	587
Name (meas)	588
Name (trace)	589
Name (CalKit object)	590
NominalIncidentPowerState	591

Normalization	592
Number (meas)	593
Number of Points	594
Number of Points (Meas)	595
NumberOfPorts	596
OmitIsolation	597
OneReadoutPerTrace	598
Options	599
OrientECALModule	600
OutputFixedFrequency	601
OutputSideband	602
OutputStartFrequency	603
OutputStopFrequency	604
Parameter	605
Parent	606
PassFailLogic	608
PassFailMode	609
PassFailPolicy	610
PassFailScope	611
PassFailStatus	612
Peak Excursion	613
Peak Threshold	614
PeakTo Peak	615
Phase Offset	616
Port 1	617
Port 2	618
Port 3	619
PortCLogic	620
PortCMode	621
Port Label	622
PortLogic	623
PortMode	624
Power Slope	625
Power Acquisition Device	626
Power Meter Channel	627
Power Meter GPIBAddress	628

R1 Input Path	629
Readings Per Point	630
ReadingsTolerance	631
ReadyForTriggerState	632
Receiver Attenuator	633
ReceiverCount	634
ReceiverStepAttenuatorStepSize	635
Receive Port	636
ReduceIFBW	637
ReferenceCalFactor	638
Reference Marker State	639
Reference Level	640
Reference Position	641
Scope	642
Search Function	643
SecurityLevel	644
Segment Number	645
Show Statistics	646
SICL	647
SICLAddress	648
Simultaneous2PortAcquisition	649
Smoothing Aperture	650
Smoothing ON/OFF	651
SnPFormat	652
Sound On Fail	653
SourceCount	654
Source Port	655
Source	656
SourcePowerCalPowerOffset	657
Source Power Correction	658
Source Power Option	659
Source Power State	660
Span	661
Span (Meas)	662
Standard Deviation	663
Standard For Class	664

Start Frequency	666
Start Power	667
Start	668
Start (Meas)	669
State	670
Statistics Range	671
Step Rise Time	672
StimulusValues	673
Stop Frequency	674
Stop Power	675
Stop	676
Stop (Meas)	677
SweepEndMode	678
SweepHoldOff	679
Sweep Generation Mode	680
Sweep Time	681
Sweep Type	682
System Impedance Z0	683
SystemName	684
Target Value	685
Test Port Power	686
ThruCalMethod (FCA)	687
ThruCalMethod	688
ThruPortList	689
Title	690
Title State	691
Trace Math	692
Tracking	693
Transform Mode	694
Trigger Delay	695
TriggerOutputEnabled	696
Trigger Mode	697
Trigger Signal	698
Trigger Type	699
Type (calstd)	700
TZImag Property	701

TZReal Property 702
UnusedChannelNumbers 703
UsedChannelNumbers 704
Use Power Loss Segments 705
Use Power Sensor Frequency Limits 706
User Range 707
User Range Max 708
User Range Min 709
UserPresetEnable 710
ValidConnectorType 711
Velocity Factor 712
View 713
Visible 714
WGCutoffFreq 715
Window Number 716
Window State 717
XAxis Point Spacing 718
YScale 719
Z0 720
Methods
Abort 721
AbortPowerAcquisition 722
Acquire Cal Standard 723
Acquire Cal Standard2 725
AcquireCalConfidenceCheckECALEx 727
AcquirePowerReadings 728
AcquireStep 729
Activate 730
Activate Marker 731
Activate Window 732
Add (channels) 733
Add (measurement) 734
Add (naWindows) 735
Add (PowerLossSegment) 736
Add (PowerSensorCalFactorSegment) 737
Add (segments) 738

Allow All Events	739
Allow Event Category	740
Allow Event Message	741
Allow Event Severity	742
Apply	743
ApplyPowerCorrectionValues	744
Autoscale	745
Averaging Restart	746
Build Hybrid Kit	747
Calculate Error Coefficients	748
Calculate	749
Change Parameter	751
CheckPower	753
Close CalSet	754
ComputeErrorTerms	755
ConfigNarrowBand3	756
Continuous Sweep	758
Copy	759
CopyToChannel	760
Create SParameterEX	761
CreateCalSet	762
CreateCustomMeasurementEx	763
Create Custom Cal	765
Create Measurement	766
DataToDivisor	768
DataToMemory	769
Delete	770
Delete Marker	771
Delete All Markers	772
DeleteCalSet	773
Delete ShortCut	774
Delta Marker	438
Disallow All Events	775
Do Print	776
DoECAL1PortEx	777
DoECAL2PortEx	778

DoneCalConfidenceCheckECAL	779
Execute	780
Execute Shortcut	781
GenerateErrorTerms	782
GenerateSteps	783
Get AuxIO	784
Get Cal Standard	785
Get CalManager	786
Get CalSetByGUID	787
Get CalSetCatalog	788
Get CalSetUsageInfo	789
Get Cal Types	790
Get Complex	791
Get DataByString	792
Get Data	794
Get ECALModuleInfoEx	796
Get Error Term	797
Get ErrorTerm2	799
Get Error Term By String	801
Get Error Term Complex	802
Get Error Term Complex2	804
Get Error Term Complex By String	806
Get Error Term List	807
Get Error Term List2	809
Get ExternalTestSetIO	810
Get Filter Statistics	811
Get Guid	812
Get Input Voltage	813
Get Input1	814
Get MaterialHandlerIO	815
Get NAComplex	816
GetNumberOfGroups	818
Get Output	819
Get Output Voltage	820
Get OutputVoltage Mode	821
Get Paired Data	822

Get Port 824
Get PortC Data 825
Get Reference Marker 826
Get Required Eterm Names 827
Get Scalar 828
Get Shortcut 830
Get SnPData 831
Get SourcePowerCalDataEx 833
Get SourcePowerCalDataScalarEx 834
Get Standard 836
Get Standard By String 838
Get Standard Complex 839
Get Standard Complex By String 841
Get StandardsList 842
Get Standard List2 844
Get StandardsForClass 845
Get StepDescription 847
Get Test Result 848
Get Trace Statistics 849
Get X-Axis Values 850
GetXAxisValues (Meas) 851
Get X-axis Values Variant 852
Has CalType 853
Hold 855
Hold (All Chans) 856
Initialize 857
Item 858
LaunchCalWizard 860
LaunchPowerMeterSettingsDialog 861
LoadFile 862
Manual Trigger 863
MessageText 864
Next IF Bandwidth 865
Number of Groups 866
Open CalSet 867
Parse 869

Preset (app and chan) 870
Previous IF Bandwidth 871
Print To File 872
Put Complex 873
Put Data Complex 875
Put ErrorTerm 877
Put ErrorTerm2 879
Put Error Term By String 881
Put ErrorTerm Complex 882
Put ErrorTerm Complex2 884
Put Error Term Complex By String 885
Put Formatted Scalar Data 886
Put NAComplex 887
Put Output 889
Put Output Voltage 890
Put Output Voltage Mode 891
Put Port 892
Put PortCData 894
Put Scalar 895
Put Shortcut 897
Put SourcePowerCalDataEx 898
Put SourcePowerCalDataScalarEx 899
Put Standard 900
Put Standard By String 902
Put Standard Complex 903
Put Standard Complex By String 905
Quit 906
Read Data 907
Read Raw 908
Recall 910
Recall Kits 911
Remove 912
Reset 913
Restore Cal Kit Defaults 914
Restore Cal Kit Defaults All 915
Resume 916

Save	917
Save (CalSet)	919
Save CalSets	920
SaveCitiDataData	921
SaveCitiFormattedData	922
Save File	923
Save Kits	924
Search Filter Bandwidth	925
Search Max	926
Search Min	927
Search Next Peak	928
Search Peak Left	929
Search Peak Right	930
Search Target	931
Search Target Left	932
Search Target Right	933
SelectCalSet	934
Set All Segments	935
Set Cal Info	936
SetCalInfo2 (power)	937
Set Center	939
Set CW	940
Set Electrical Delay	941
Set FailOnOverRange	942
SetPowerAcquisitionDevice	943
Set Frequency LowPass	944
Set Reference Level	945
Set StandardsForClass	946
Set Start	948
Set Stop	949
Show Marker Readout	950
Show Status Bar	951
Show Stimulus	952
Show Table	953
Show Title Bars	954
Show Toolbar	955

Single 956
StringToNACalClass 957
StringtoNAErrorTerm2 959
UserPreset 960
UserPresetLoadFile 961
UserPresetSaveState 962
Write Data 963
Write Raw 964
Events
OnCalEvent 966
OnChannelEvent 967
OnDisplayEvent 968
OnHardwareEvent 969
OnMeasurementEvent 970
OnSCPIEvent 971
OnSystemEvent 972
OnUserEvent 973
Examples
CalSet_Examples 974
Getting Trace Data from the Analyzer 979
Perform a Source Power Cal 982
Upload a Source Power Cal 985
Upload Segment Table 987
Create and Cal an SMC Measurement 989
Create and Cal a VMC Measurement 991
Create an SMC Fixed Output Meas 994
Create a Pulsed Measurement 996
Perform an ECAL Confidence Check 998
Limit Line Testing Example with COM 1000
Events Example 1001
Concepts
Configure for COM-DCOM Programming 1002
COM Fundamentals 1007
Getting a Handle to an Object 1010
Collections in the Analyzer 1013
COM Data Types 1014

PNA Automation Interfaces 1016
Working with the Analyzer's Events 1017
Read and Write Calibration Data using COM 1021
C and the COM Interface 1023
Using .NET 1026
SCPI
Commands
List of SCPI Commands 1028
SCPI Command Tree 1040
Abort 1042
Calculate
Correction 1043
Custom 1048
Data 1051
Filter 1057
Format 1062
Function 1063
Limit 1067
Marker 1072
Math 1088
Mixer 1090
Normalize 1092
Parameter 1095
RData 1100
Smoothing 1101
Transform 1103
Common Commands 1112
Control 1115
Display 1120
Format 1135
Hardcopy 1137
Initiate 1138
Memory 1140
Output 1149
Route 1150
Sense

Average 1151
Bandwidth 1153
Correction
Correction 1155
Correction CKIT 1169
Collect CKIT 1174
Session 1197
SMC 1202
VMC 1208
Cset 1217
Guided 1222
Couple 1231
Frequency 1233
IF 1237
Mixer 1242
Offset 1259
Power 1263
Roscillator 1264
Segment 1265
Sweep 1275
XAxis 1282
Source 1283
Source Correction 1290
Status 1304
System 1317
Trigger 1327
Examples
Catalog Measurements 1330
Create a measurement 1331
Setup Sweep Parameters 1332
Setup the Display 1333
Channels, Windows, and Measurements using GPIB 1335
Perform Guided ECal or Mech Cal 1337
Perform Guided 1Port 1340
Perform Unguided ECAL 1342
Perform Unguided 2-Port Mech Cal 1343

Perform an Unguided 1-Port Cal on Port 2 1345
Perform a Source Power Cal 1347
Uploading a Source Power Cal 1352
Perform a Sliding Load Cal 1356
Perform an ECAL Confidence Check 1357
Create and Cal a VMC Measurement 991
Create and Cal an SMC Measurement 989
Create an SMC Fixed Output Measurement 1365
Getting and Putting Data using GPIB 1368
Establish a VISA Session 1370
Status Reporting 1372
Create New Cal Kit using SCPI 1374
Modify a Calibration Kit 1380
GPIB using Visual C 1382
Create a Custom Power Meter Driver 1386
GPIB Pass Through 1389
PNA as Controller and Talker/Listener 1390
Concepts
GP-IB Fundamentals 1392
The Rules and Syntax of SCPI Commands 1399
Calibrating the PNA Using SCPI 1403
Getting Data from the Analyzer 1405
Understanding Command Synchronization 1408
Reading the Analyzer's Status Registers 1413
Configure for VISA and SICL 1416
Data Access Map
Data Map 1419
Rear Panel IO Connectors
Auxiliary IO connector 1420
External TestSet IO Connector 1425
Material Handler IO Connector 1430
COM versus SCPI 1443
Using Macros 156
Code Translator 1445
Tutorials
Network Analyzer Basics 1446

Connector Care 1447
ESD Protection 1456
Impedance Matching Model 1457
Reflected Waves 1461
Reflection Measurements 1462
Phase Measurements 1466
Amplifier Measurements 1469
AM-PM Conversion 1472
Antenna Measurements 1476
Antenna Macro 1478
Complex Impedance 1482
Deviation from Linear Phase 1484
Group Delay 1487
Gain and Flatness 1492
Gain Compression 1495
Reverse Isolation 1499
Absolute Output Power 1501
Time Domain
Time Domain 1503
Frequency Offset (Opt 080)
Frequency Offset (Option 080) 1517
Frequency Converting Device Measurements 1524
Frequency Offset Calibration 1525
Conversion Loss 1527
Conversion Compression 1530
Isolation 1532
Harmonic Distortion 1535
Return Loss and VSWR 1537
Frequency Converter Application (Opt 083)
Known Issues 1539
Overview 1540
Using the Frequency Converter Application 1541
Calibrations 1550
Configure a Mixer 1564
Configure an External LO Source 1572
High Power Measurements

External Leveling Circuit 1577

High-Power Measurements 1578

E835x Standard

Diagram: Basic Config 1588

Diagram 1: Fwd Msmts 1589

Diagram 2: Fwd Msmts 1590

Diagram 1: Fwd & Rev Msmts 1591

Diagram 3: Fwd & Rev Msmts 1592

Diagram 1: Booster amp 1593

Diagram 2: Booster amp 1594

Diagram 3: 2 Booster amps 1595

Diagram 1: Ext level 1596

Diagram 2: Ext level 1597

Diagram 3: Ext level 1598

Diagram: Cal Power Levels 1599

E835x Opt 015

Diagram: Basic Config 1600

Diagram 1: Fwd Msmts 1601

Diagram 2: Fwd Msmts 1602

Diagram: Fwd & Rev Msmts 1603

Diagram 1: Booster amp 1604

Diagram 2: Booster amp 1605

Diagram 1: Ext level 1606

Diagram 2: Ext level 1607

Diagram: Cal Power levels 1608

E836x Standard

Diagram: Basic Config 1609

Diagram: Fwd Msmts 1610

Diagram: Fwd & Rev Msmts 1611

Diagram: Booster amp 1612

E836x Opt 014

Diagram: Basic Config 1613

Diagram 1: Fwd Msmts 1614

Diagram 2: Fwd Msmts 1615

Diagram 1: Booster amp 1617

Diagram 2: Booster amp 1619

Diagram: Fwd & Rev Msmts 1621

Diagram: Cal Power Levels 1623

E880x Standard

Diagram: Basic config 1624

Diagram: Fwd Msmts 1625

Diagram: Fwd & Rev Msmts 1626

Diagram: Booster amp 1627

Diagram: Ext Level 1628

E880x Opt 014

Diagram: Basic Config 1629

Diagram 1: Fwd Msmts 1630

Diagram 2: Fwd Msmts 1631

Diagram: Fwd & Rev Msmts 1632

Diagram 1: Booster amp 1633

Diagram 2: Booster amp 1634

Diagram 1: Ext level 1635

Diagram 2: Ext level 1636

Diagram: Cal Power Levels 1637

N338x Standard

Diagram: Basic Config 1638

Diagram: Fwd Msmts 1639

Diagram: Fwd & Rev Msmts 1640

Diagram: Booster amp 1641

Diagram: Ext level 1642

N338x Opt 014

Diagram: Basic Config 1643

Diagram 1: Fwd Msmts 1644

Diagram 2: Fwd Msmts 1645

Diagram: Fwd & Rev msmts 1646

Diagram 1: Booster amp 1647

Diagram 2: Booster amp 1648

Diagram 1: Ext level 1649

Diagram 2: Ext level 1650

Diagram: Cal Power Levels 1651

Networking the PNA

Drive Mapping 290

Easy vs Secure Configuration 1652

Changing Network Client 1653
Product Support
Troubleshoot the PNA 1654
List of Error Messages 1657
About Error Messages 1713
Accessories 1716
USB to GPIB Adapter 1722
Firmware Update 1725
PNA Configurations and Options 1730
Option Enable 1736
Other Resources 1739
SCPI Errors 1741
Technical Support 1752
Diagnostic Tools and Adjustments
3.8 GHz Frequency Adjust 1757
10 MHz Reference Adjust 1759
LO Power Adjust 1761
Offset LO Power Adjustment 1763
Operators Check 1764
Phase-Lock IF Gain Adjust 1766
System Verification 1767
Source Cal 1776
Receiver Cal 1779
Receiver Display 1782
Serial Bus Test 1783
Instrument Calibration 1784
IF Access Applications (Opt H11)
About the H11 Option 1785
External Test Head Configuration 1787
Pulsed Application 1791
Specifications
E8356A, 57A, 58A 1801
E8801A, 02A. 03A 1831
N3381A, 82A, 83A 1855
E8361A 1880
E8362A, 63A, 64A 1934

E8362B, 63B, 64B 1935

N5230A 1977

N5250A 2044

Glossary 2050

What's New in PNA Code Version 4.87

New 4.87 Programming Commands

To check your PNA code version, click **Help**, then **About Network Analyzer**

What's New in PNA Code Version 4.86

- Dedicated calibration window
- Option H11 Verification
- Calibration Class Label
- Data-Based Cal Kits can now be modified
- Safely shutdown the PNA without a mouse

What's New in PNA Code Version 4.83

- New PNA-L Models (N5230A Options 020 / 025 / 120 / 125)
- User Preset using SCPI and COM
- Magnitude Offset
- Calibrated Fixed Output **VMC** measurements
- New VMC **R1** and **B** absolute power measurements
- Material Handler Trigger Control Also available using SCPI and COM
- Global Pass / Fail Status Also available using SCPI and COM.
- **Changes to ECAL** - existing **SCPI** programs may be impacted
    - Default setting for Thru Method
    - Isolation disabled
    - Adapter Removal now performs a Full 2-port cal on each side of the adapter.
    - Unknown Thru and Flush Thru now use two, instead of three, connection steps.
- Sweep dialog now has Apply button
- Revised Op Check
- Procedure to Run VNC

See New 4.83 Programming Commands

What's New in PNA Code Version 4.25

- Time Domain Coupling
- Time Domain Distance Markers
- GPIB Pass-through
- Arbitrary Segment Sweep
- New External Trigger Capabilities
- Pulse Profiling and other Pulsed Application features
- Support for Windows© XP

See New 4.25 Programming Commands

What's New in PNA Code Version 4.0

- New PNA-L Model Number
- IF Access (Option H11)
- External Millimeter Module Support (Option H11)
- Pulsed Application (Option H08)
- Reduced IFBW at Low Frequencies
- CitiFile (*.cti) Format Support
- Enhanced SnP File Save
- New Source Cal Accuracy Settings
- Frequency Converter Application
  - Fixed Output Measurements
  - Enhanced External Source Control
- Import Legacy Cal Kits
- 8510 Data Processing Mode
- Help in Chinese

See New 4.0 Programming Commands

What's New in PNA Code Version 3.5

- Support for E8361A Options 014, 016, 081 (configurable test set, receiver attenuators, and reference switch).

- Frequency Blanking - Turn off the display of frequency information for security purposes.
- Enhanced Local Lockout for GPIB and COM
- New SCPI Edit Cal Kit capabilities with new example program. (Connector name, Kit description, Standard remove, standard description.)
- Support for up to eight ECAL modules connected to the PNA.
- Enhanced FCA mixer configuration - (Calculate Input)
- Change FCA measurements using SCPI

What's New in 3.1

- Support for new PNA model E8361A - 67 GHz
- Frequency Converter Application (Opt. 083)
- Expanded Math calibration algorithm

What's New in 3.0

- Frequency Offset Measurements - Stimulus and Response can be tuned to different frequencies.
- Support for new PNA models - E8362B, E8363B, E8364B.
- Increased Points (16,001), Channels (32), and Windows (16)
  See Note about processing speed.
- Fixturing Macro - Helps compensate for the effects of test fixturing.
- X-axis Point Spacing - Alternate display method for Segment Sweep.
- Copy Channels - Copies channel settings to a new or existing channel.
- Trigger IN / OUT - Support for edge-triggering on new PNA models.
- Calibration enhancements - Converged cal kits (no longer Version 1 and Version 2). Replaces Cal Kit Manager.

What's New in 2.6

- Arbitrary Impedance calibration standard selections
- Many improvements to printed Help documentation including Programming Command Finder.

What's New in 2.5

- ECAL - Basic support for 4-port module.
- ECAL User-Characterize - Use adapters with your ECAL module and store the characterization results in the module.
- Log Frequency Sweep - Logarithmic sweep and display.
- TRL Calibration - Added impedance and reference plane options.
- Calibration Wizard - Improved error reporting messages.
- Auto-Check - Automatically checks the Internet for new releases of the PNA.
- Cal Kit Manager - Manage your Version 1 Cal Kits (Removed in Rev 3.0).

What's New in 2.0

The following are new and modified features in PNA code version A.02 (from A.01.5).

New Features

- 3 port Measurements - includes selection of all 9 S-parameters, selection of receiver "C" and associated ratios, and power level control on port 3.
- Calibration Sets - concept similar to that developed for the 8510 Network Analyzer but with more capability and flexibility.
- User Defined PRESET - you can customize the front panel PRESET functionality.
- AUXILIARY I/O, EXTERNAL TEST SET I/O, and MATERIAL HANDLER I/O - New Rear Panel Connectors.
- ECAL Confidence Check - ensure confidence in your Electronic Calibration.
- Manage Files... under File menu provides a File Manager functionality within the PNA.
- Source Power Cal and Receiver Power Cal - Power level flatness correction using for accurate absolute power measurements.
- S3P ASCII - file format for data files.
- .jpg format for screen dumps - includes white background on direct draw surface.
- Minimize - View menu selection allows you to minimize the PNA application.

Modified Features

- Calibration Wizard - supports guided and unguided calibrations. Guided calibrations provide additional features of "adapter removal", "unknown thru" and "full 3 port cal type". Both guided and unguided cals support "sliding load" calibration standards and more than one standard for a given frequency span.
- Limit Lines -can now be viewed on multiple traces in a single window. Also Limit lines are now updated for non active traces.

- 16 independent channels - Increased from 4.

- Added several Marker Improvements:
  Coupled Markers
  Marker Readout
  Memory traces can now support markers.

- SYSTem:PRESet - this SCPI command will no longer restore the display if it is disabled due to DISP:ENAB OFF. The display is restored by either front panel Preset or *RST.

- **PNA Product Overview** - includes the new additions to the PNA Series analyzers. Launch from the PNA desktop.

PNA User Accounts and Passwords

When the PNA power is switched on, it automatically logs into Windows using the default user name and password. This gives anyone full access to the analyzer. The following steps can be taken to increase security of your PNA.

- Require users to logon when the PNA computer is turned ON - Learn how to enable this feature
- Setup individual accounts on the PNA with varying level of access - Learn how to Add or Change User Accounts and Passwords

Please read about Anti-virus protection for your PNA

Existing User Accounts

The following user accounts already exist on new PNAs:

- **Default User Account**
  Beginning in April 2004, PNAs were shipped from the factory with the default user name is **PNA-Admin** and the password is **agilent.**
  For PNAs shipped before that, the default user name is **Administrator** and the password is either **tsunami** or left blank.
  These accounts are created by Windows and cannot be deleted.
  We recommend you change the password and, if desired, the user name.
  DO NOT FORGET YOUR NEW PASSWORD. You will not be able to start your PNA without it.
- **Agilent Account** This Administrator account is created by Agilent for service purposes. Each PNA has a unique password for this account. Although allowed by Windows, please do not delete this account.
- **Guest Account** This account allows anyone to type in any name, without password, and gain limited access to the PNA files. This account is created by Windows and cannot be deleted. It can be renamed. This account is turned OFF when the PNA is shipped.

Notes

- Although allowed by Windows, do NOT setup an Administrator account without a password. Internet viruses look for, and exploit, this condition.
- You can create as many user accounts as you like.
- The user name is not case sensitive. The password IS case sensitive.
- The PNA local policies are set so that, if logon is required, you must retype the user name (and password) every time. Do not change the local policies on the PNA.

How to Require Users to Logon when the PNA Computer is turned ON.

How do I know which Operating System I have?

Windows 2000	Windows XP
On the Windows taskbar, click **Start**, then **Settings**, then **Control Panel**	On the Windows taskbar, click **Start**, then **Run**
Double click **Users and Passwords**	Type **control userpasswords2** then click **OK**
Check **Users must enter a user name and password to use this computer.**	Check **Users must enter a user name and password to use this computer.**

To turn this function OFF, perform the same procedure, but clear the checkbox. The account that is selected when the checkbox is cleared is the account that is automatically logged on when the PNA is turned ON.

Add or Change User Accounts and Passwords

If the analyzer is in a secure environment, you can setup PNA users by name and grant various levels of access. This is particularly important when the PNA is remotely controlled or accessed over LAN.

You can designate a person as the administrator and then configure the PNA to allow others to use it with reduced permissions. That is, other people can be signed on to use the analyzer but they will not have the ability to perform all of the administrative functions that you can as the administrator.

How to add or change a user account and password

How do I know which Operating System I have?

Windows 2000	Windows XP
In the analyzer **System** menu, point to **Configure**, and click **Control Panel**.	Click Start, then point to Settings, then click Control Panel
In the **Control Panel** window, scroll down and select the **Users and Passwords** application.	Click **User Accounts**
On the **Users** tab, if the **Add** button appears dimmed, select the **Users must enter a user name and password to use this computer** check box near the top of the window.	Follow the prompts to: • Change an account • Create a new account • Change the way users log on or off.
Click **Add** to enter the information for yourself or for another user.	**CAUTION:** Although allowed by Windows, do NOT allow an Administrator account without a password. Internet viruses look for, and exploit, this condition.
In the **User name** box, enter a user name for the user. In the **Full name** box, enter the full name of the user.	
In the **Description** box, enter a description for the user. Then, click **Next**.	

In the **Password** box, have the user type a password. Have the user retype the password in the **Confirm password** box. Then, click **Next**.	
Select the level of access that you wish to grant this user. **Note**: Standard users and restricted users are unable to switch GPIB modes and install firmware. There are several other levels of security that you may grant in the **Other** list. A description of each of these other levels is displayed beneath the **Other** box when it is selected. Then, click **Finish**.	
In the **Users for this computer** box, validate the user name and security level group of the user.	
If you want this user to be able to use the network analyzer without entering their password each use, clear the **Users must enter a user name and password to use this computer** check box. Click **OK**.	
When the **Automatically Log On** window is displayed, have the new user type their password in the **Password** box and have them retype the password in the **Confirm Password** box.	
Click **OK** to complete this user addition.	
In the **File** menu, click **Close** to close the Control Panel.	

PNA Computer Properties

The PNA uses a personal computer and a Windows operating system. The following are common tasks that you may need to perform on the PNA computer.

- **View or change Full Computer Name**
- **Check IP Address**
- **Check the amount of RAM**
- **Check CPU Speed**
- **Set Time and Date**

Other Administrative Task Topics

View or change Full Computer Name

Your PNA has a unique computer name that identifies it on a network. To view or change the computer name, you must first minimize the PNA application.

How do I know which Operating System I have?

Windows 2000	Windows XP
On the desktop, right-click **My Computer**	On the desktop, right-click **My computer** Icon
Click **Properties**	Click **Properties**
Click the **Network Identification** tab at the top of the dialog box	Click the **Computer Name** tab at the top of the dialog box
Click **Properties**	Click **Change** next to "**..rename this computer..**" message
Type your new **Computer Name**	Type your new **Computer Name**

Note: To add your computer to a domain, or to set up the networking configuration, contact your company's I.T. department. This setup is custom for each company.

To restore the PNA application, click **PNA Analyzer** in the task bar at the bottom of the screen.

Check IP Address

If your PNA is connected to a LAN, you can view the IP address and other networking information.

1. Minimize the PNA application
2. Click **Start**, then **Run**
3. Type **cmd**, then click **OK**

4. At a DOS prompt, type **ipconfig /all**

Check the amount of RAM

Random Access Memory (RAM) is the amount of working memory in your computer. The PNA application can require up to 512 MB of RAM depending on the settings you use. If your PNA is operating slowly when you have more than four windows open or if you routinely use more than 1601 data points, you may need to upgrade to 512 MB.

To view the amount of PNA RAM, you must first minimize the PNA application.

Windows 2000 or Windows XP

1. On the desktop, right-click **My Computer**
2. Click **Properties**
3. Click the **General** tab at the top of the dialog box
4. The amount of RAM appears at the bottom of the window.

To restore the PNA application, click **PNA Analyzer** in the task bar at the bottom of the screen.

Check CPU Speed

The speed of the PNA processor (CPU) is a factor in determining how quickly the PNA processes data. See PNA configurations to learn if you can upgrade your PNA CPU. To check your PNA CPU speed, you must first minimize the PNA application.

How do I know which Operating System I have?

Windows 2000	**Windows XP**
On the desktop, right-click **My Computer**	On the desktop, right-click **My computer** Icon
Click **Manage**	Click **Properties**
Open **System Tools** folder, then click **System Information.**	Click the **General** tab at the top of the dialog box
Click **System Summary**.	The CPU speed appears near the bottom of the window
After refreshing, the CPU speed appears at the end of the **Processor** entry.	

To restore the PNA application, click **PNA Analyzer** in the task bar at the bottom of the screen.

Set Time and Date

Both Windows 2000 and XP

To set the time and date on your PNA, you must first minimize the PNA application.

1. Move the cursor to the lower corner of the screen

2. When the taskbar appears, double-click on the displayed time. This opens the **Date/Time Properties** dialog box.

3. Change the date, time, and time zone as appropriate.

To restore the PNA application, click **PNA Analyzer** in the task bar at the bottom of the screen.

Recovering from PNA Hard Drive Problems

The leading cause of PNA failures is problems with the PNA Hard Disk Drive (HDD). These problems are usually preventable (see Preventing PNA HDD Problems), and in many cases, recoverable. The following could save you weeks of downtime and the cost of replacing your PNA HDD.

The following three symptoms indicate a probable HDD failure. First, attempt to perform an Operating System Recovery. If it fails, the HDD must be replaced.

1. Will not complete the boot process.
    - If ‘Couldn’t find NTLDR’ appears, remove if a disk is in the PNA floppy drive. If there is no disk in the floppy drive, proceed to Operating System Recovery.
    - If ‘Operating System not found’ appears, replace the HDD.
2. Continual blue screens or multiple errors.
3. Excessive clicking or a loud high-pitch whine.
    - Back up files and replace HDD.

The following three symptoms indicate missing or corrupt files. First, try the suggestions indicated by bullets. Then, perform an Operating System Recovery, which is likely to solve the problem.

1. Required system files are not found.
2. The operating system or PNA application does not function properly.
    - Uninstall the PNA application using Add/Remove Programs from the Control Panel. Then reinstall the PNA application by double-clicking on D:\Updates\Firmware\ *.msi. or download a firmware update using AgileUpdate.
3. The PNA functions, but is very slow.
    - Check for excessive clicking or a loud high-pitch whine (see #3 above).
    - Check RAM. For best performance, RAM must be at least 128MB for PNA firmware rev. 3.0 or below; 256MB for firmware rev. 3.1 or higher. See PNAHelp to learn more.
    - Scan for Virus and “spyware”.

Operating System Recovery

This procedure will completely restore the operating system of the PNA back to the original factory condition. It will not work if the hard drive is damaged.

Important note: This procedure will overwrite the entire C: drive and the following information will be lost:

- All changes and additions made to the analyzer since purchase (except options)
- All user accounts and passwords
- All added programs

- All saved instrument states
- All user configurations and Windows settings

You can save this information to a floppy drive or to a folder on the D:drive. The D:drive will not be affected by this restoration process. In addition, factory instrument calibration will not be affected if no changes have been made since the original factory calibration. If changes have been made through the service routines, copy all files that start with “mxcalfile_” in the C:\Program Files\Agilent\Network Analyzer directory, and save them to the D:\Calfiles directory.

Procedure

1. Click **Start, Shutdown, Restart**
2. After the Agilent logo, when ‘**Please select the operating system to start:**’ displays, quickly press the down arrow on the keyboard to select the System Recovery partition, (NOT the Recovery Console). Press **Enter**. **NOTE:** If you do not reach this screen during the boot process, you may be able to boot from a floppy disk. See <u>If you CANNOT boot from the Hard Drive</u> below.
3. The PNA will display a warning about proceeding and give you two chances to abort. After the second warning, the restoration process will begin immediately.
4. Restoration takes from 15 to 45 minutes. There is no user interaction required. Near the end of the process, the PNA may reboot several times. Do not attempt to use the analyzer until the PNA application is running and all activity has stopped for at least one minute.
5. Once the operating system is completely recovered, you can restore any files you may have saved back to their original locations. The mxcalfiles that were stored in D:\Calfiles will be restored automatically. Other files must be copied or moved manually.
6. If PNA application updates had previously been downloaded through AgileUpdate, they can now be re-installed. Double-click on D:\Updates\Firmware\ *.msi. or download a firmware update using AgileUpdate.

If you CANNOT boot from the Hard Drive

During the boot process, you may not reach the screen that displays ‘**Please select the operating system to start**’. You may be able to boot from a floppy disk if your PNA Hard drive is labeled ’S.02.xx’ and ‘S.05.50’. If labeled with a different code, the process will fail at step 2 below, but no further damage will be done.

Make this disk using a computer that is currently running DOS or Windows 95/98/ME/XP. Note: Windows NT or 2000 cannot be used.

To create this disk, right-click on the A: drive in Windows Explorer, select **Format**, then select **Make System Disk.** This may be called **copy system files**, or **Create an MS-DOS startup disk**.

To boot from this disk:

1. Insert the created system disk into the PNA and restart. The PNA should detect the floppy disk and boot from it. See below if it does NOT boot.
2. At the DOS prompt (A:\>)type CD C: and press Enter. You are really changing to the D: drive because a DOS operating system does not recognize the NTFS format and assumes the first valid drive is C:.
3. Type **C:\minint\bin\recovery.bat** and press **Enter**. If this returns an error, type **C:\Autoexec.bat**
4. Continue with Operating System Restoration <u>Procedure</u> (above).

If the PNA does not Boot from the floppy, then change the PNA Bios settings. Due to bios revisions, this procedure may vary slightly from the following:

1. Restart the PNA.

2. When the Agilent logo appears, press F2 (function key) on the keyboard a few times. The Bios menu will appear after several seconds.

3. Select **Boot** using the keyboard arrows. Read "Item Specific Help" to learn to move the **Removable Devices / floppy drive** to a boot position before the **Hard Drive**.

4. Select **Save** then **Exit**.

5. To prevent an error message when a floppy disk is inserted while powering up, use this procedure to change back to booting from the hard drive.

Microsoft Windows® XP / 2000 Considerations

Critical Notes

Windows Operating System

Note: Beginning in April 2004, the PNA is shipped from the factory with a modified version of Microsoft Windows XP operating system. Previously, the PNA was shipped with Windows 2000. The PNA application performs identically using these two operating systems.

To determine which Operating System is installed on your PNA:

1. Minimize the PNA application
2. On the PNA desktop, click Start.
3. Along the side of the Start menu appears one of the following:

   - **Windows 2000 Professional**
   - **Windows XP Professional**

VERY IMPORTANT information to protect your hard drive!

The leading cause of PNA failures is problems with the PNA Hard Disk Drive (HDD). These problems are usually preventable, and in many cases, recoverable. Learn more about protecting your PNA.

Using USB

The PNA has at least two USB ports for connecting devices: one on the front panel and at least one on the rear panel. The main advantages of USB are "hot" connects and disconnects and fast data transfer speeds. Electronic Calibration modules are now available with USB connections.

The first time you plug a device into a USB port there is some wait time. Windows reports it is identifying the hardware, then searching for the correct driver, then installing the driver (if it was found).

Connecting that same device back into that same port later is quick and easy, but if you move the device to a different USB port, you will have to wait through the hardware ID and driver search again.

Note: Certain USB devices (such as ECAL modules) require you be logged on with Administrator privileges the first time you plug them into the PNA. This must be done for each serial number. Click **Next** to choose the default settings when installing new USB devices.

Plug & Play Stability and Security

Plug & Play capabilities is similar to Win 95 and 98. It provides both a stable and secure operating environment. You may notice also that it greatly reduces the number of required reboots.

LAN Connections

Windows supports DHCP and fixed IP addressing. Also, "Hot" connect and disconnect of the LAN cable, as well as a visual indicator of LAN status in system tray area, makes LAN connections more intuitive. In addition, the Hardware Wizard helps users with system hardware configuration.

Single and Double Click option.

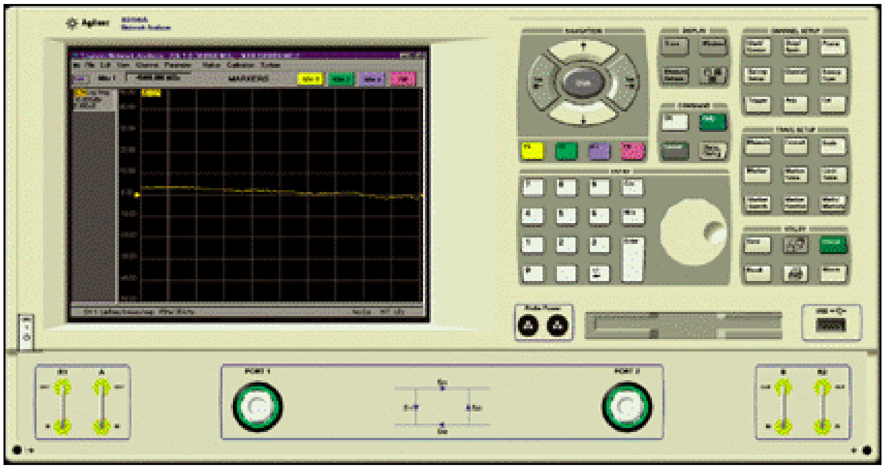
By default, Windows allows a single-click method of launching icons. To revert to double-clicking, click **Start**, then **Settings**, then **Control Panel,** then click **Mouse**. In the Mouse Properties dialog, select **Double-click to open an item**. Then click **OK**.

Printing

Adding a printer should be done outside of the PNA application. Learn more.

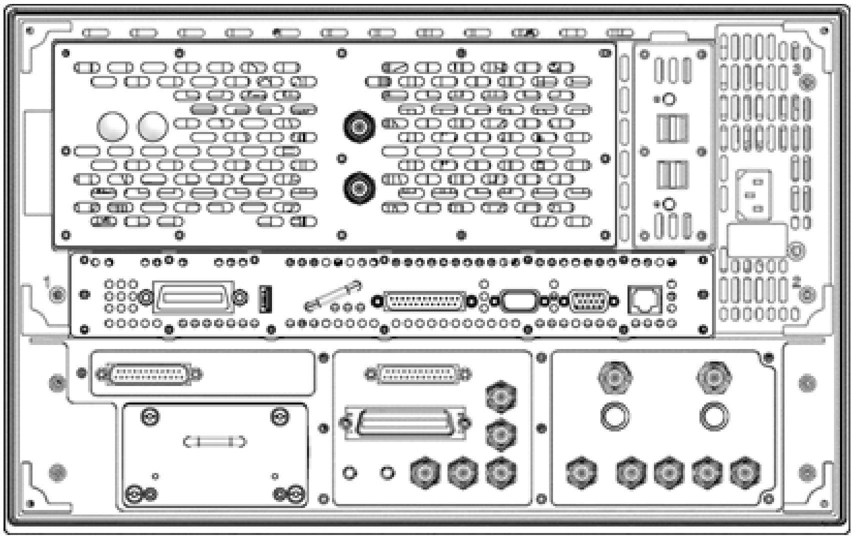
Front Panel Tour

Click on the sections of the front panel for information.



Rear Panel Tour

This image includes ALL rear-panel features.
Your PNA may not have this capability or look.
Click on a connector for detailed information.



Powering the PNA ON and OFF

There are three PNA operating modes:

Hibernate

ON

Shutdown

Note: If the PNA front-panel keypad or USB ports are not responding, SHUTDOWN or RESTART the PNA; do NOT Hibernate. This causes the PNA drivers to awaken from hibernation in the same corrupt state.

How to Log off, Shut down, Restart, or Hibernate the PNA.

WITH A MOUSE

1. On the PNA **System** menu, click **Windows Taskbar**
2. On the Windows Taskbar, click **Shutdown**
3. In the **What do you want the computer to do?** list, choose an action:
   - Log off (closes programs and disconnects from the network)
   - Shut down
   - Restart (shutdown and start)
   - Hibernate
4. Click **OK** to perform the action

WITHOUT A MOUSE

- To Hibernate, BRIEFLY press the green power button.
- To Shutdown - ONLY if the PNA is locked and you cannot operate the mouse or keypad - Press and hold the power button for at least four seconds. **This practice should be avoided!** Repeated shutdowns in this manner WILL damage the hard drive. Learn more about damaging the PNA hard drive.
- **Recommended -** To SAFELY shutdown the PNA without a mouse, configure the PNA so you can choose what to do when the power button is briefly pressed (as in Step 3 above). PNAs shipped after June 2005 are already configured this way:
  1. From Windows Control Panel, select **Power Options**
  2. Click **Advanced** Tab
  3. Under **Power buttons**, select **Ask me what to do.**
  4. Click **OK** to end configuration.

The next time the power button is pressed, a dialog box will ask **What do you want the computer to**

do? Use the PNA front panel **Tab** and **Enter** keys to choose an action.

Hibernate Mode

- In hibernate mode the current instrument state is automatically saved to the hard disk before the PNA is powered OFF.
- When the PNA is powered ON, this instrument state is loaded, thus saving time over a full system boot-up.
- A password is NOT required to resume PNA operation after Hibernate mode.
- The hibernation state is the normal OFF state. A small amount of standby power is supplied to the PNA when it is in the hibernation mode. This standby power only supplies the power switch circuits and the 10 MHz reference oscillator; no other CPU-related circuits are powered during hibernation. To guarantee that your measurements meet the PNA specified performance, allow the PNA to **warm-up for 90 minutes** after the power button light has changed from yellow back to green.

ON Mode

- To turn ON the PNA press the yellow power button.
- The power button will change to green when power is ON.

Shutdown Mode

- In shut down mode the current instrument state is NOT automatically saved before the PNA is powered OFF.
- When the PNA is again powered ON, a full system boot-up is performed and the PNA powers-up in the preset settings.
- A password is required to resume PNA operation after being in Shutdown mode.
- To guarantee that your measurements meet the PNA specified performance, allow the PNA to **warm-up for 90 minutes** after the power button light has changed from yellow back to green.
- The PNA should only be shut down for service or to provide security via password protection.
- The power button will change to yellow when power is OFF.

Note: If the PNA is locked and you cannot operate the mouse or keypad, shut down the PNA by pressing and holding the power button for at least four seconds.
This practice should be avoided! Repeated shutdowns in this manner WILL damage the hard drive. Learn more about damaging the PNA hard drive.

Unplugging the PNA

- Remove the power cord from the PNA ONLY when the power button is yellow, in either Hibernate or Shutdown mode.
  If the power cord is removed while the power button is green (PNA ON), damage to the hard drive is **likely**.

- The button will remain yellow for several seconds after the power cord has been removed.
- When plugged back in and the power button is pressed to ON, the PNA starts in the mode it was in when the power cord was unplugged, either Hibernate or Shutdown.

Front Panel Interface

There are three ways to use the front panel keys:

- **Active Entry Toolbar** (quickest)
- **Launch Dialog Boxes**
- **Navigate Menus** (most comprehensive)

Other Quick Start topics

Active Entry Toolbar

Not all settings can be made this way. For making ALL settings use Menus.

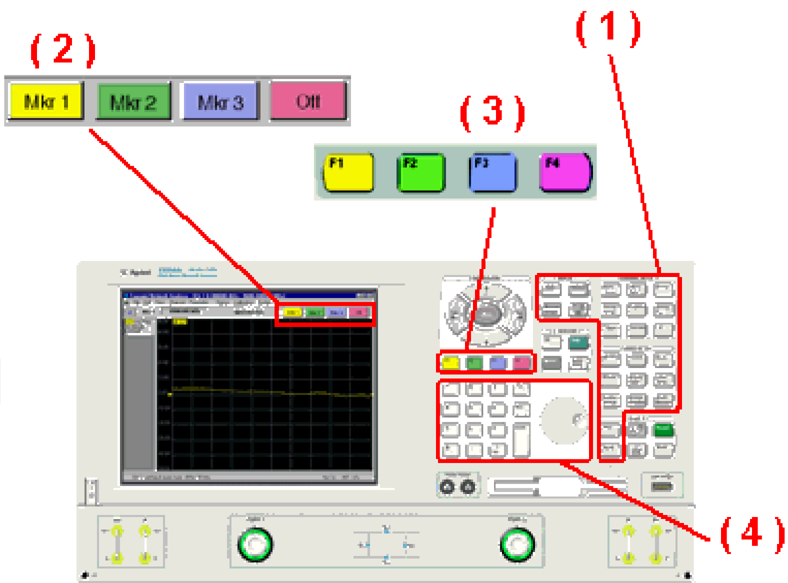
You can make settings quickly using this four step procedure.

(1) Press a key

(2) View active entry

(3) Select a function

(4) Enter a value (if necessary)

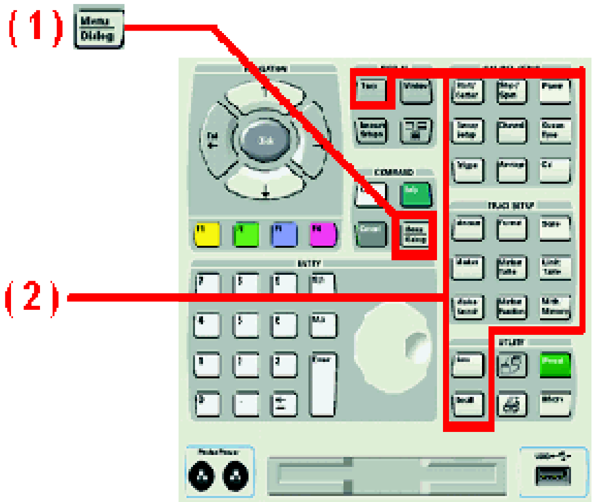


Launch Dialog Boxes,

To quickly launch MOST dialog boxes:

(1) Press the Menu/Dialog Key

(2) Select a function key



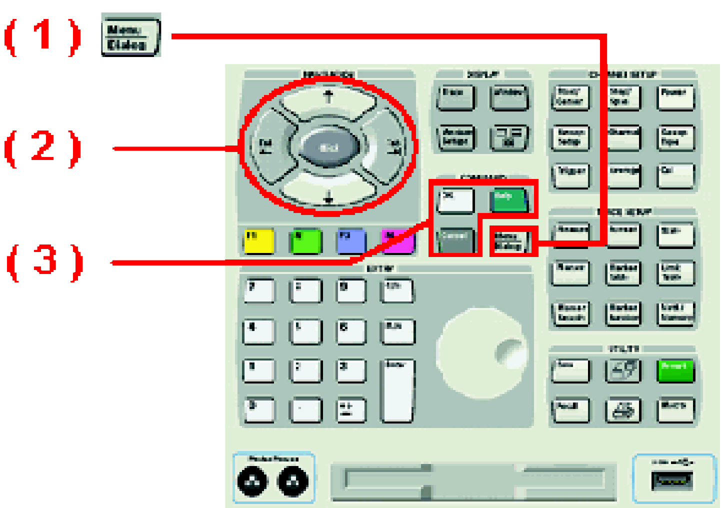
Navigate Menus

You can access ALL PNA functions using Menus:

(1) Press the Menu/Dialog Key

(2) Use the direction keys to navigate through the Menus. Use the "Click" key to make a selection.

(3) Other Command keys are available for cancelling or seeking Help (if necessary)



Traces, Channels, and Windows on the PNA

It is critical to understand the meaning of the following terms as they are used on the PNA.

Traces

Channels

Windows

Managing Windows

Note: You may experience a significant decrease in computer processing speed with combinations of the following: increased number of points, number of traces, and calibration error terms (full 2-port or 3-port). If this becomes a problem, you can increase the amount of RAM with PNA Option 022. To monitor the amount of PNA memory usage, press **Ctrl Alt Delete**, select **Task Manager**, then click on the **Performance** tab.

Other Quick Start topics

Traces are a series of measured data points.

Trace settings affect the mathematical operations and presentation of the measured data. A trace must be selected (active) to modify its settings. To select a trace, click the Trace Status button. The following are Trace settings.

- Parameter
- Format
- Scale
- Calibration ON / OFF
- Trace Math
- Markers
- Electrical Delay
- Phase Offset
- Smoothing
- Transform

Channels contain traces. The PNA can have up to **32 independent channels.**

Channel settings determine **how** the trace data is measured . All traces that are assigned to a channel share the same channel settings. A channel must be selected (active) to modify its settings. To select a channel, click the Trace Status button of a Trace in that channel. The following are channel settings:

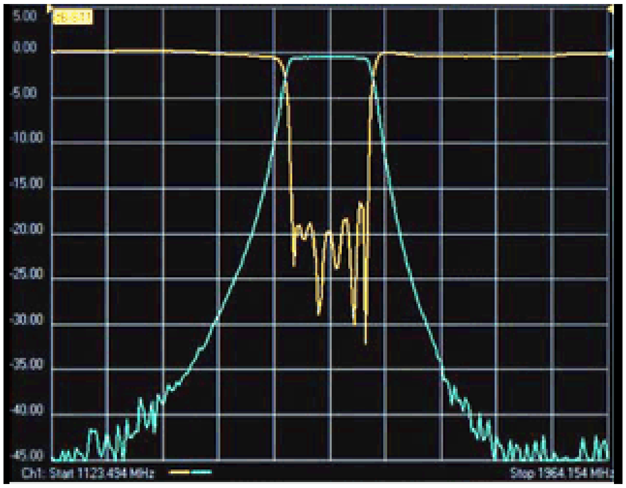
- Frequency span
- Power

- Calibration data
- IF Bandwidth
- Number of Points
- Sweep Settings
- Average
- Trigger (some settings)

Windows are used for viewing traces.

- The PNA can show up to **16 windows** on the screen.
- Each window can contain up to **4 traces**.
- Windows are completely independent of channels.
- Most Window settings are made from the **View** menu. See Customize the PNA screen for details.
- Learn to create and manage windows.

The following is a window containing two traces. Both traces use the same channel 1 settings as indicated by the annotation at the bottom of the window.

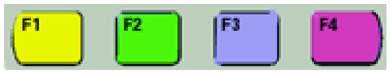


Managing Windows

How to create a new window

Use one of the following methods:

- Press 

 + 

- Click **Trace**, **New Trace.** Check **New Window**.
- Click **Window** then **New.** A new window appears with a default trace (S11) in the same channel as the active window.

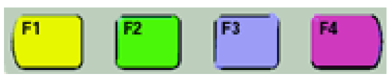
Learn more about using the front panel interface

How to view a window with the full screen view:

When more than four windows are on the screen, the Trace Status pane is hidden and resolution becomes poor. To view the window with the full screen:

Select the trace:

- Press 

 + 

Then press . Press again to minimize the window

Basic Measurement Sequence

The following process can be used to setup all PNA measurements:

Step 1. Set Up Measurements
Reset the analyzer, create a measurement state, and adjust the display.

Step 2. Optimize Measurements
Improve measurement accuracy and throughput using techniques and functions.

Step 3. Perform a Measurement Calibration
Reduce the measurement errors by performing a calibration.

Step 4. Analyze Data
Analyze the measurement results using markers, math operations, and limit tests.

Step 5. Print, Save or Recall Data
Save or print the measurement data.

Using Help

Help Rev. 2005-08-02
PNA Rev. 4.87
Copyright 2000-2005
Agilent Technologies, Inc.

This topic discusses the following:

- **Printing Help**
- **Copying Help to your PC**
- **Launching Help**
- **Navigating Help**
- **Help Languages**
- **Glossary**
- **Dialog Boxes**
- **About Network Analyzer**
- **Documentation Warranty**
- **Suggestions Please**

Other Quick Start Topics

Printing Help

Beginning in May 2005, a .pdf version of PNA Help is once again available for download at http://www.agilent.com/find/pna. Search for "PNA Help" and scroll to Manuals and Guides. You can still print individual PNA Help topics by clicking the Print icon at the top of the PNA Help window.

Copying Help to your PC

With the Help system on your PC, you can read about the PNA while away from it. You can also Copy and Paste programming code from this Help system directly into your programming environment.

The Help file is located on your PNA hard-drive at **C:\ Winnt\ Help\ PNAHelp.chm**. If both the PNA and PC are connected to LAN, you can map a drive and copy the file directly. The Help system is also on the CDROM that ships with the PNA.

Launching Help

The Help system can be launched in three ways:

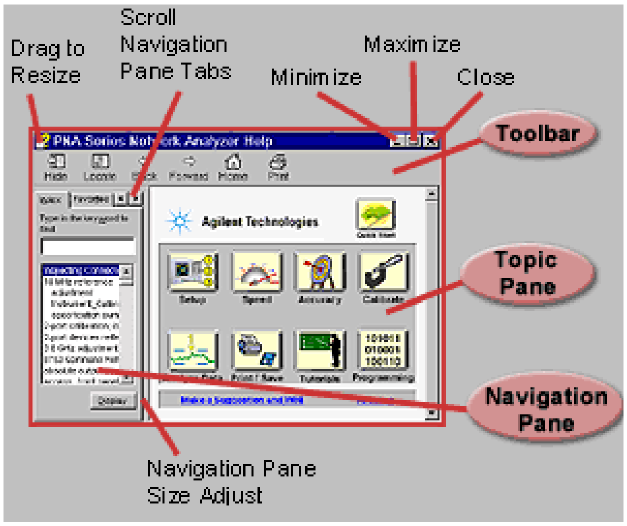
1. From the front panel Help button.
2. From the **Help** drop-down menu

3. From Dialog Box Help

Navigating Help

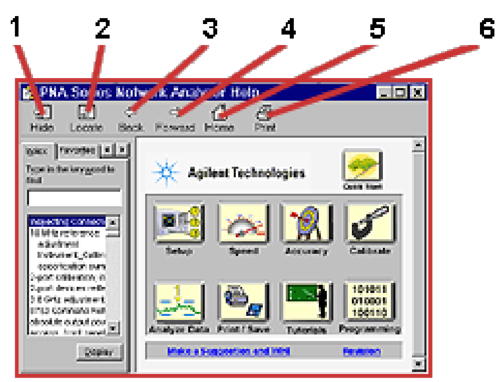
The Help Window contains 3 panes (regions):

1. Toolbar Pane
2. Topic Pane
3. Navigation Pane



Toolbar Pane

The Toolbar is at the top of all Help windows. It allows you to resize the window, browse and print the selected topic.



1. Hide or show the navigation pane
2. Locate the topic in the table of contents
3. Back to topic visited previously
4. Forward again if **Back** was clicked
5. Go to the Home page.
6. Print the topic pane.

Navigation Pane

Click the following tabs in the Navigation Pane to access information in the Help system:

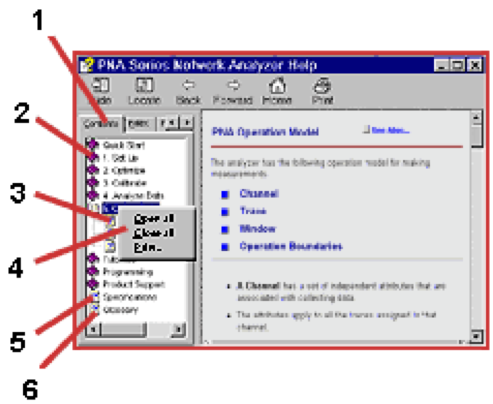
Table of Contents Tab

Index Tab

Search Tab

Favorites Tab

(Table of)Contents Tab



1. Click tab to select Table of Contents.
2. Click a book to access related topics.
3. Click to display a topic.
4. Right click to access menu.
5. Click to display specifications
6. Click to display glossary

Index Tab

The index tab allows you to type a keyword and go to only the most applicable topics.

1. Click tab to select index.
2. Type keyword to find topics of interest.
3. View suggested topics. (Double-click to display topic.)
4. Click to display topic.

Search Tab

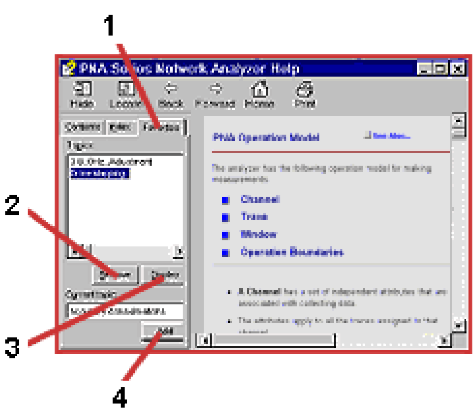
The following rules apply for using full-text search:

- Searches are not case-sensitive.
- You can search for any combination of letters (a-z) and numbers (0-9).
- Punctuation marks (period, colon, semicolon, comma, and hyphen) are ignored during a search.
- You can group the words of your search using double quotes or parentheses. Examples: "response calibration" or (response calibration). This requirement makes it impossible to search for quotation marks.
- Use Wildcard expressions:
  * To search for one undefined character use a question mark (?). For example, searching for **cal?** will find **calc** and **calf.**
  * To search for more than one undefined character use an asterisk (*). Searching for **Cal*** will find **calibration** and **calculate**
- Use Boolean operators to define a relationship between two or more search words.

Search for	Example	Results will show topics containing:
Two words in the same topic	response AND calibration	Both the words "response" and "calibration"
Either of two words in a topic	response OR calibration	Either the word "response" or the word "calibration" or both
The first word without the second word in a topic	response NOT calibration	The word "response" but not the word "calibration"
Both words in the same topic, close together.	response NEAR calibration	The word "response" within eight words of the word "callibration"

Favorites Tab

The favorites tab allows you to store (bookmark) the topics you refer to most often so that they can be recalled easily.



1. Click tab to view stored topics in Favorites.
2. Remove selected topic.
3. Display selected topic.
4. Add (store) current topic.

Topic Pane

The Topic pane allows you to view the contents of the selected topic.

Help Languages

The Help system is available in five languages:

- English
- French
- German
- Japanese
- Spanish
- Chinese

To upgrade to the latest version, or to make Chinese Help available, use AgileUpdate.

To select a different help language:

On the **Help** menu, point to **Help Language**. Then click the language.

Note: The Programming section is only available in English. Also, the most recent version of Help will be English.

Viewing Help in Japanese or Chinese

These two languages use a different character font set than the other languages. To view these languages, you must change the following settings:

1. From the PNA, click **View**
2. Click **Title Bars** (if not already checked)
3. Minimize the PNA screen
4. From the desktop of the PNA. click **Start / Settings / Control Panel / Regional Options**.
5. Under **Language settings for the system**, scroll to and check **Japanese or Chinese**
6. Under **Your locale (location)** select **Japanese or Chinese**

7. Click **Set default...**

8. Under **Select the appropriate locale**, select **Japanese or Chinese**

These changes may cause the Windows 2000 fonts to become small. To change them back to larger fonts, change the following settings:

1. From the PNA desktop click **Start / Settings / Control Panel**

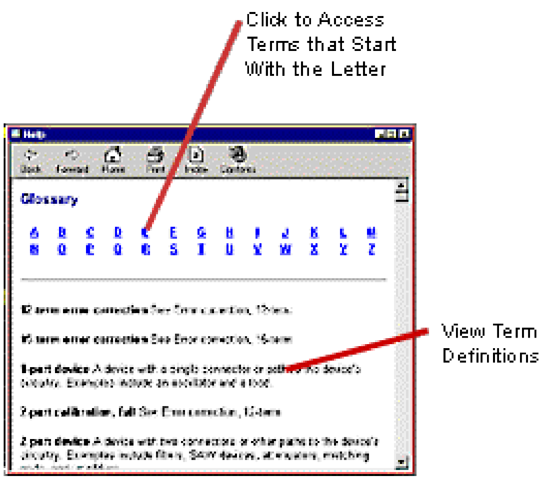
2. Click **Display / Appearance**

3. Under **Scheme** select **Windows Standard**

To restart the PNA application, click **PNA Analyzer** taskbar button at the bottom of the screen

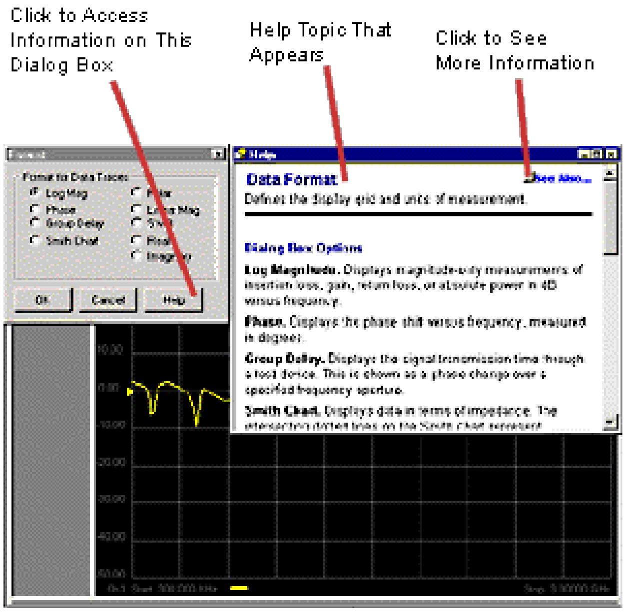
Glossary

The Glossary holds definitions of words, in alphabetical order.



Note: Click on a word in green text throughout Help to see the glossary definition.

Dialog Boxes



About Network Analyzer

To learn the following about your analyzer, click **Help**, then **About Network Analyzer:**

- Model number
- Frequency range
- Options installed on the analyzer
- Serial number of the analyzer
- Version of the installed analyzer software
- Version of hard drive in the analyzer

Documentation Warranty

THE MATERIAL CONTAINED IN THIS DOCUMENT IS PROVIDED "AS IS," AND IS SUBJECT TO BEING CHANGED, WITHOUT NOTICE, IN FUTURE EDITIONS. FURTHER, TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, AGILENT DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED WITH REGARD TO THIS MANUAL AND ANY INFORMATION CONTAINED HEREIN, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. AGILENT SHALL NOT BE LIABLE FOR ERRORS OR FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH THE FURNISHING, USE, OR PERFORMANCE OF THIS DOCUMENT OR ANY INFORMATION CONTAINED HEREIN. SHOULD AGILENT AND THE USER HAVE A SEPARATE WRITTEN AGREEMENT WITH WARRANTY TERMS COVERING THE MATERIAL IN THIS DOCUMENT THAT CONFLICT WITH THESE TERMS, THE WARRANTY TERMS IN THE SEPARATE AGREEMENT WILL CONTROL.

Suggestions Please!

Please let us know about your experience using PNA Help. Send your comments to: pna_help@am.exch.agilent.com. Comment about any aspect of the help system. Here are a few areas that you might consider:

- Does anything appear to be broken?
- Did you find what you were looking for?
- Was the information you found helpful?
- Any suggestions as to how we can improve the help system?

Your comments go directly to the help system authors. For help with technical questions, please refer to Technical Support.

Preset the PNA

When you Preset the PNA, it is set to known, or preset conditions. You can use the factory default preset conditions, or define your own User Preset conditions.

Preset (Default) Conditions

User Preset Conditions

See other 'Setup Measurements' topics

Preset Default Conditions

How to Preset the PNA
Use one of the following methods: •  Preset •  System Window Preset User Preset... Learn more about using the front panel interface

Click to view the **factory preset conditions**.

Frequency Settings

Power Settings

Sweep Settings

Segment Sweep Settings

Trigger Settings

Display Settings

Response Settings

Calibration Settings

Marker Settings

Limit Test Settings

Time Domain Settings (Option 010)

Global Display Settings

Frequency Settings:

Measurement Parameter	S11
Start Frequency	Minimum frequency of the PNA
Stop Frequency	Maximum frequency of the PNA
CW Frequency	1 GHz

See the PNA configurations for the minimum and maximum frequency of your PNA

Power Settings:

Test Port Power	0 dBm for E8356/7/8A 0 dBm for E8801/2/3A 0 dBm for N3381/2/3A -5 dBm for N5230A - 20 GHz -10 dBm for N5230 - 40 GHz -15 dBm for N5230 - 50 GHz -12 dBm for E8362/3/4 A or B, standard -15 dBm for E8361A -17 dBm for E8362/3/4 A or B with option UNL or 014
Port Power Coupled	On
Auto Attenuation	On
Attenuator Value	0 dB
Power Slope	Off
Slope Value	0 dB/GHz

Sweep Settings:

Type	Linear Frequency
Mode	Continuous
Generation	Analog
Auto Sweep Time	On
Number of Points	201

Segment Sweep Settings:

Active Segments	1
Start Frequency	PNA start frequency
Stop Frequency	1 MHz for E8356/7/8A 1 MHz for E8801/2/3A 1 MHz for N3381/2/3A 1 GHz for E836xA/B
Number of Points	21
Power	PNA preset test port power
IF Bandwidth	50 KHz for N5230A 35 kHz for all other models
Reduce IF BW at Low Frequencies	ON
Dwell Time	0

Trigger Settings

Source	Internal
Mode	Sweep

Display Settings:

Format	Log Mag

These settings apply for formats when selected:

Format	Scale	Reference Position	Reference Value
Log Mag	10 dB/	5	0 dB
Phase	45 degrees/	5	0 degrees
Group Delay	10 nsec/	5	0 s
Linear Mag	100 munits/	0	0 units
SWR	1 unit/	0	1 unit
Real	2 units/	5	0 units
Imaginary	2 units/	5	0 units
Polar	1 unit/	n/a	1 unit
Smith Chart	1 unit/	n/a	1 unit

Response Settings:	
Channel Number	1
IF Bandwidth	50 KHz for N5230A 35 kHz for all other models
Averaging	Off
Averaging Factor	1
Smoothing	Off
Smoothing Factor	1% of span
Electrical Delay	0 s
Velocity Factor	1.0
Phase Offset	0 degrees
Math/Memory Trace View	Data

Calibration Settings:	
Correction State	Off
Interpolation State	On
Calibration Type	None
Cal Kit Number	Current Cal Kit Number
System Z0	50 ohms
Port Extensions State	Off
Port Ext. Values Input A, B Port 1, 2	0

Marker Settings:

Initial Frequency	Current Center Frequency
Reference	None
Interpolation	On
Format	Trace Default
Type	Normal
Function	Max Value
Domain	Full Span
Table	Empty
Coupling	Always uncoupled

Limit Test Settings:

Limit Testing	Off
Line Display	ON
Sound on Fail	Off

Limit List Settings:

Type (OFF, MAX, MIN)	OFF
Begin Stimulus	0
End Stimulus	0
Begin Response	0
End Response	0

Time Domain Settings:

Transform State	Off
Transform Mode	Band Pass
Transform Start	-10 ns
Transform Stop	10 ns
Window	6.0 (Kaiser-Bessel factor)
Gating State	Off
Gating Start	-10 ns
Gating Stop	10 ns
Gate Type	Band Pass
Gate Shape	Normal

Global Display Settings:

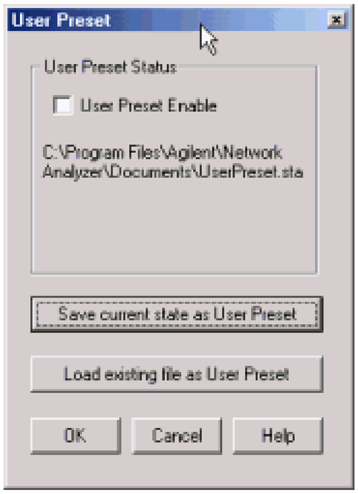
Trace Status	On
Frequency/ Stimulus	Off
Marker Readout	On (when a marker is activated)
Toolbars Shown	Active Entry
Status Bar State	ON

User Preset Conditions

The analyzer can be **preset** to either **default** conditions or **User Preset** conditions.

How to set User Preset

Learn more about using the front panel interface



User Preset dialog box help

Allows the selection of User Preset conditions and the storing and recovery of those conditions.

User Preset Enable
Check - The PNA is preset to **User Preset** conditions when the Preset button is pressed.
Clear - The PNA is preset to **Default** conditions when the Preset button is pressed.

Save current state as User Preset Click to store the current instrument state as the User Preset conditions. File is stored as C:\ Program Files\ Agilent\ Network Analyzer\ Documents\ UserPreset.sta.

Load existing file as User Preset Click to retrieve an instrument state to be used as the User Preset conditions.

Measurement Parameters

This topic contains the following information:

- **S-Parameters** (fixed ratios)
- **Arbitrary Ratio** (choose your own ratio)
- **Unratioed Power** (absolute power)
- **How to Select a Measurement Parameter**

See other 'Setup Measurements' topics

S-Parameters

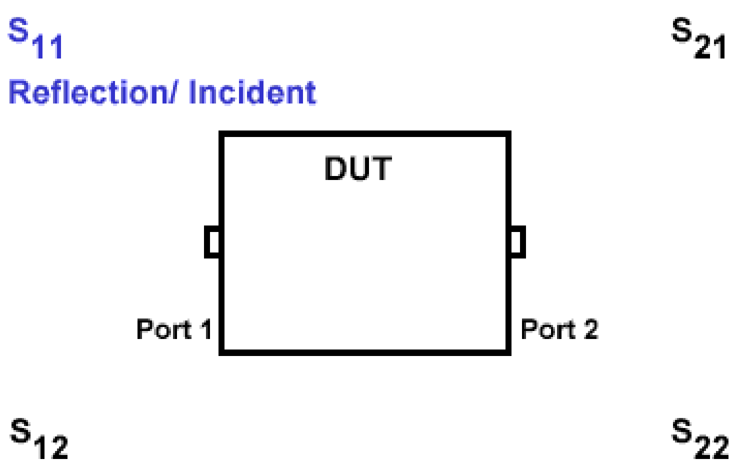
S-parameters (scattering parameters) are used to describe the way a device modifies a signal. For a two port device, there are **four S-Parameters.** (If you have a 3-port PNA model, see the topic, 3-port Measurements.) The syntax for each parameter is described by the following:

S out - in

out = port number where the signal output is measured (receiver)

in = port number where the signal is applied (source)

Move the mouse over each S-parameter to see the signal flow:



For two-port devices:

- When the source goes into port 1, the measurement is said to be in the **forward** direction.
- When the source goes into port 2, the measurement is said to be in the **reverse** direction.
- The analyzer automatically switches the source and receiver to make a forward or reverse measurement. Therefore, the analyzer can measure all four S-parameters for a two-port device with a single connection.

For PNAs with 3 measurement ports, see 3 port measurements

Common Measurements with S-Parameters

Reflection Measurements (S11 and S22)

- Return loss
- Standing wave ratio (SWR)
- Reflection coefficient
- Impedance
- S_{11}, S_{22}

Transmission Measurements (S21 and S12)

- Insertion loss
- Transmission coefficient
- Gain/Loss
- Group delay
- Deviation from linear phase
- Electrical delay
- S_{21}, S_{12}

Arbitrary Ratio

Arbitrary ratio allows you to choose your own ratio of input and reference signals using the A, B, R1 and R2 receivers. The following are common uses of arbitrary ratio:

- Comparing the phase between two paths of a device. An example could be something simple like a power splitter or more complicated like a dual-channel receiver.
- Measurements that require a higher dynamic range than the analyzer provides with S-parameters.

Your PNA **MAY** have front-panel jumper cables that go directly to the A, B, R1 and R2 receivers. Learn about the front-panel jumpers on your PNA.

Unratioed Power

The unratioed power parameter allows you to look at the absolute power of a device going into any of the receivers.

Note: Unratioed parameters cannot be used for phase or group delay measurements, or any measurement that has averaging activated.

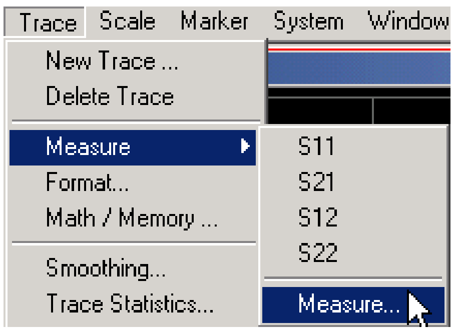
How to select a Measurement Parameter

Use either of the following methods to create a new trace:

- 

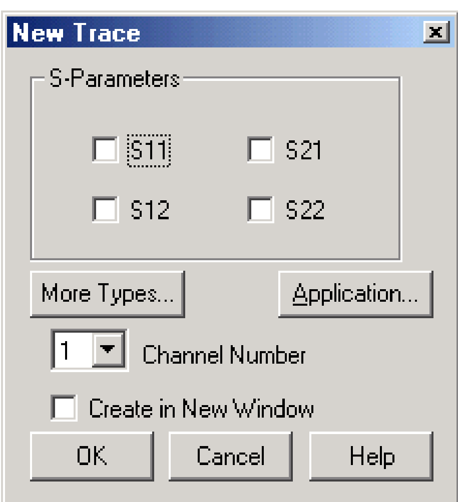
- 

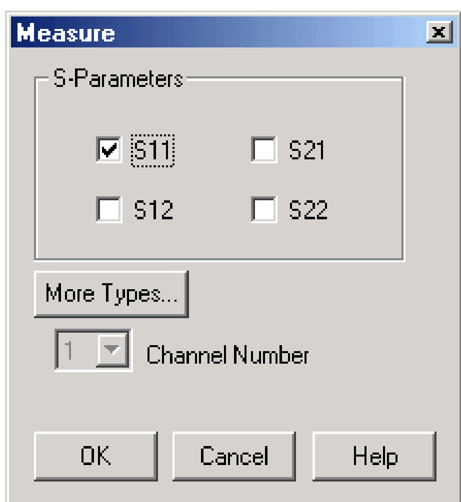
Use either of the following methods to **change** the **Active** trace:

- 

- 

Learn more about using the front panel interface





New Trace and Measure dialog box help

The following applies to both the **New Trace** and **Measure** dialog boxes

S-Parameters Check ONE to create or change the active measurement.

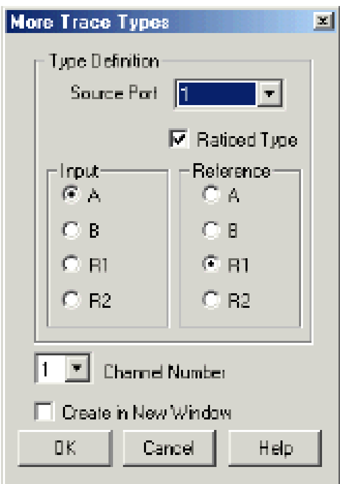
More Types... Press to open the More Types Dialog Box (for **Unratioed** or **Arbitrary Ratio** measurements)

Application... Press to list the measurement applications installed on the analyzer.

Channel Number (New Trace only) Select the channel to add the new trace to.

Create in New Window (New Trace only)
Check to create the trace in a new window
Clear to create the trace in the active window

About Measurement Parameters (top of page)



More Types dialog box help

Source Port Select the PNA port that will emit the source power

Ratioed Type Check to create a ratioed measurement (then select both Input and Reference receivers)
Clear to create an unratioed measurement (then select Input receiver only)

Input Select the PNA receiver to measure your DUT response. You can directly access the receivers ports by removing the front-panel jumper cables.

Reference Select the PNA reference receiver to ratio with the Input receiver (ratio measurement only). The reference level must be between 0 dBm and –35 dBm

Channel Number (New Trace only) Select the channel to add the new trace to.

Create in New Window (New Trace only) Check to create the trace in a new window.
Clear to create the trace in the active window

About Measurement Parameters (top of page)

Frequency Range

Frequency range is the span of frequencies you specify for making a device measurement.

- How to Set Frequency Range
- CW Frequencies
- Frequency Resolution
- Phase Lock Lost Indicator
- Frequency Band Crossings

See other 'Setup Measurements' topics

How to set Frequency Range

There are two ways to set the frequency range:

A. Specify the Start and Stop frequencies of the range.

B. Specify the Center frequency and desired span of the range.

The following will accomplish both methods:

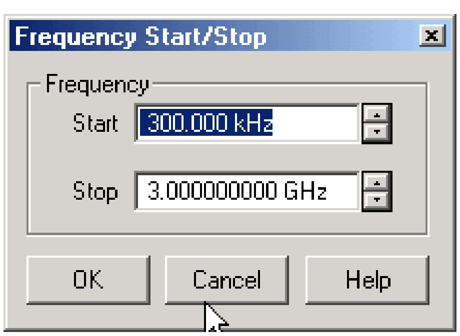
- 

The following settings will open dialog boxes:

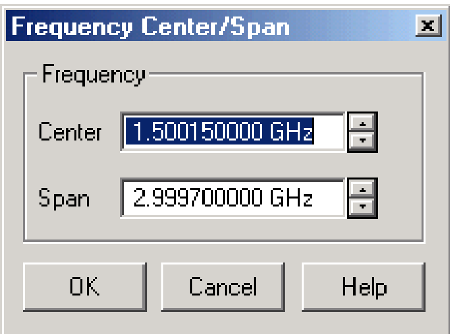
- 

Learn more about using the front panel interface

See the frequency ranges of all PNA models



Frequency Start/Stop dialog box help
Start Specifies the beginning frequency of the swept measurement range. **Stop** Specifies the end frequency of the swept measurement range.



Frequency Center/Span dialog box help
Center Specifies the value at the center of the frequency sweep. This value can be anywhere in the analyzer range. **Span** Specifies the span of frequency values measured to either side of the center frequency.

CW Frequencies

Measurements with a CW Time sweep or Power sweep as stimulus are made at a single frequency rather than over a range of frequencies.

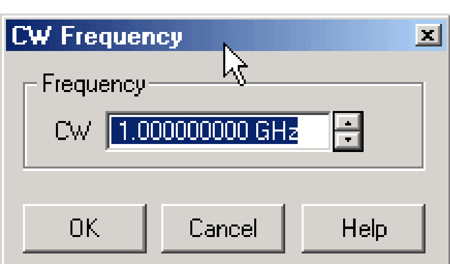
How to set CW Frequency

First specify CW Sweep Type

You can set CW frequency from within the Sweep Type dialog box **or**

After specifying CW from the Sweep type dialog box, change CW frequency by selecting the following::

Learn more about using the front panel interface



CW Frequency dialog box help
CW Type a value and the first letter of the suffix (k,m,or g) or use the up and down arrows to select any value within the range of the PNA.

Frequency Resolution

The resolution for setting frequency is 1 Hz.

Phase Lock Lost Indicator

Under normal operating conditions the analyzer stimulus frequency will be phase locked and within specifications (see frequency specs). If the analyzer is unable to create the stimulus frequency to specifications, a **Phase Lock Lost** message will appear on the screen (see Troubleshoot the PNA).

Frequency Band Crossings

The frequency range of the PNA covers several internal frequency bands. The higher the frequency range of the PNA, the larger the number of bands. The source power to your DUT turns off as the stimulus frequency is swept through these band crossings. To learn more, see Power ON and OFF during Sweep and Retrace and Stimulus-Sweep Recommendation for a DUT with an AGC Loop.

These band-crossings occur at the following frequencies:

For 3 GHz, 6 GHz, and 9 GHz PNA models:

Band	Frequency
1	10 MHz
2	748 MHz
3	1500 MHz
4	3000 MHz
5	4500 MHz
6	6500 MHz

For PNA models E8362 / 63/ 64 A/B
(A models do not have band 0)

Band	Freq (GHz)	Band	Freq (GHz)	Band	Freq (GHz)
0	.045				
1	0.748	**9**	7.60	**17**	25.60
2	1.500	**10**	10.00	**18**	30.00
3	3.00	**11**	12.00	**19**	32.00
4	3.80	**12**	12.8	**20**	36.00
5	4.50	**13**	15.20	**21**	38.40
6	4.80	**14**	16.00	**22**	40.00
7	6.00	**15**	20.00	**23**	45.60
8	6.40	**16**	22.80	**24**	48.00

For PNA model E8361A

Band	Freq (GHz)	Band	Freq (GHz)	Band	Freq (GHz)
0	.045	**9**	10.00	**18**	32.00
1	0.748	**10**	12.00	**19**	36.00
2	1.500	**11**	12.80	**20**	40.00
3	3.00	**12**	15.40	**21**	44.70
4	3.80	**13**	16.00	**22**	46.20
5	4.00	**14**	20.00	**23**	51.20
6	4.50	**15**	24.00	**24**	60.00
7	6.00	**16**	25.60	**25**	64.00
8	77.00	**17**	30.00		

For PNA model N5230A

Band	Freq (GHz)	Band	Freq (GHz)	Band	Freq (GHz)
1	.045	**11**	10.500	**21**	28.600
2	.748	**12**	12.500	**22**	31.250
3	1.5	**13**	15.750	**23**	31.500
4	3.125	**14**	16.667	**24**	33.333
5	4.167	**15**	18.750	**25**	37.000
6	5.250	**16**	21.000	**26**	40.500
7	6.250	**17**	22.500	**27**	41.667
8	7.875	**18**	25.000	**28**	42.000
9	8.333	**19**	26.250	**29**	46.800
10	9.375	**20**	26.500	**30**	

Power Level

Power level is the power of the analyzer source signal at the test port. The following items describe details about power levels.

- Power Settings
- Unleveled Indicator
- Power Coupling Between Ports
- Source Attenuation
- Attenuation with Uncoupled Ports
- Receiver Attenuation
- Power Slope
- Power ON and OFF during Sweep and Retrace

See other 'Setup Measurements' topics

Power Settings

The test port output power is specified over frequency (See the Power Range and Frequency Range for your PNA)

How to make Power settings

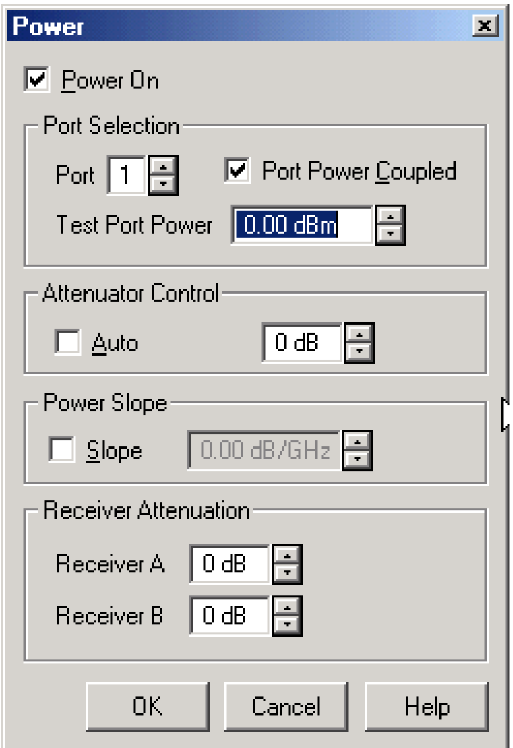
Use one of the following methods:

- 

- 

Note: If Power Sweep is selected, (from Sweep Type dialog box) clicking **Channel, Power** will invoke a Power (Sweep) dialog box.

Learn more about using the front panel interface



Power dialog box help

Defines and controls the analyzer source power.

Power On Toggles the source power ON and OFF.

Note: Power ON/OFF setting and Instrument State Save and Recall.

If power is OFF when an instrument state is saved, the power will be OFF when the state is recalled.

If power is ON when an instrument state is saved, then when recalled, the power setting will be the SAME as the current power setting. To protect your DUT, power will NOT be turned ON by an instrument state recall if the current power setting is OFF.

Port Selection

Port Specifies the port which the values apply.

Port Power Coupled Toggles between coupled and uncoupled test port power. With the ports coupled, the power levels are the same at each test port. With the ports uncoupled, you can set different power levels for each test port. Learn more about coupled ports.

Test Port Power Sets the value of the power at the output of the test port.

Attenuator Control

This area of the dialog box is unavailable if there is no attenuator option installed in the PNA. If available, the value of attenuation placed in the source path can be chosen automatically or manually. If done manually, type or select an attenuation value in 10 dB increments. See PNA Options to view the availability and range of source attenuation on your PNA.

Auto Select to allow automatic attenuator control. The attenuation value select box will be unavailable. Clear to manually set the attenuator. Type or select the attenuation value in the adjacent text box.

Learn more about Source Attenuation.

Power Slope

Compensates for a device's power loss over a frequency sweep by increasing the PNA output power over frequency.

Slope Select to set the power slope. Clear to set power slope OFF.

Learn more about Power Slope.

Receiver Attenuation

Available on PNA models as option 015 and option 016.

Receiver A and **Receiver B** Type or select independent attenuation values for each receiver.

Learn more about Receiver Attenuation.

Unleveled Indicator

The letters **LVL** appear on the status bar when the analyzer source is unleveled. To see the status bar, click **View**, then **Status Bar**.

LVL appears when the source power is **greater** than the maximum specified power. Resolve this by lowering the power level to return to a leveled condition (See your analyzer's Power Range).

If LVL continues to appear after lowering the source power, see Troubleshoot the PNA to determine the cause of the unleveled condition.

Power Coupling Between Ports

By default, the power level at each Port is **coupled to Port 1.** This means each Port will have the same output power level as Port 1. Some measurement applications require different power levels between ports. The analyzer allows you to **uncouple** the port powers so that you can set the power levels independently for each port.

For example, if you want to measure the gain and reverse-isolation of a high-gain amplifier, you must set the power for each port separately. This is because the power required for the input port of the amplifier is much lower than the power required for the output port. See Attenuation With Uncoupled Ports.

CAUTION! You can damage the analyzer receivers if the input power levels exceed the maximum values. See your analyzer's Technical Specifications for the maximum input power to a receiver.

Source Attenuation

The PNA uses a programmable step source attenuator to cover the full power range. The attenuator adjusts the power level of the test signal into the test device without changing the level of the incident power in the reference path. This provides:

- Better signal source accuracy and performance
- A more accurate source match

Source attenuation is available on all PNAs in various ranges, either as standard equipment or as an option. If attenuation is not installed, the attenuator control is unavailable in the Power dialog box and not shown in the Active Entry Toolbar. See PNA Options to see the availability and range of source attenuation on your PNA.

The value of source attenuation can be set automatically or manually through the Power dialog box or front panel Power key.

Automatic Attenuation

With the attenuator control set to **Auto** (default), the analyzer:

- Allows you to select **any power level** within the total operating range of the instrument
- Automatically selects the attenuator position and source power so that the **optimum range** of the total attenuator range is used.

Manual Attenuation

There are some measurement applications where you need to set the source power and attenuator manually. These include:

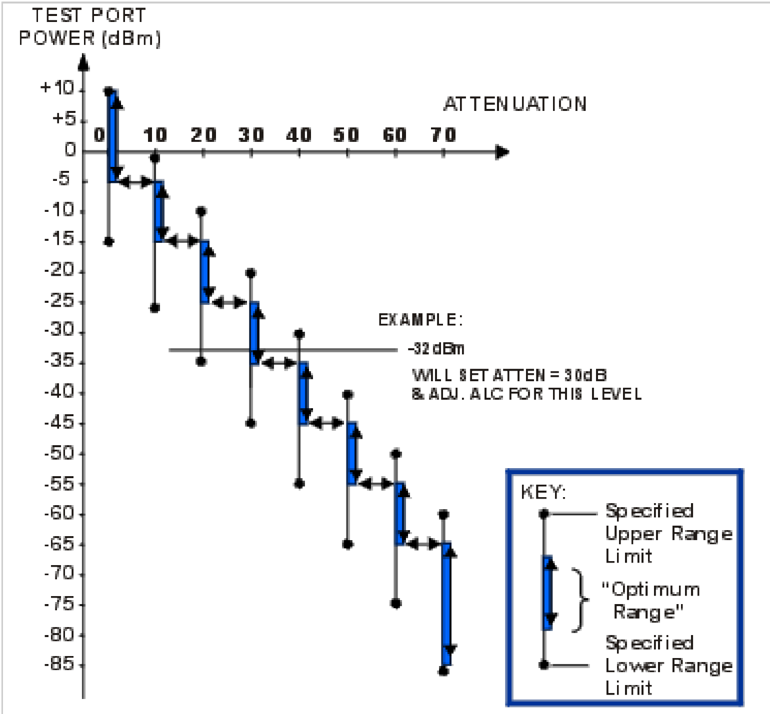
- **Testing Reflection Amplifiers** (oscillators or conditionally unstable amplifiers). This requires a very good impedance match with the source (20dB return loss) over a wide frequency range. Choose **Auto Off**, with an attenuation level of 10 or 20 dB (or more) to ensure the best source match.
- **Power Sweeps** The attenuator is not allowed to switch attenuator settings during a power sweep to avoid premature wear from continuous switching. The Auto setting may not allow full coverage of the desired power sweep range. Choose **Auto Off** and a combination of source power range and attenuator position to provide the full desired power sweep range required for the measurement.

Attenuation Ranges and Values

The attenuator has different positions, allowing power levels in many power ranges. The number of power ranges available is determined by the source attenuation installed in your PNA. See PNA Options to see the availability and range of source attenuation on your PNA.

- Each range has a total specified span (25 dB in the following Attenuation Values graphic).
- The optimum setting is the middle of the range. This range provides the best accuracy and performance of the source leveling system. The optimum ranges are the blue regions in the following graphic.
- An attenuator setting can be selected manually or automatically. If automatic is selected, the blue optimum ranges (shown in the following graphic) are used.

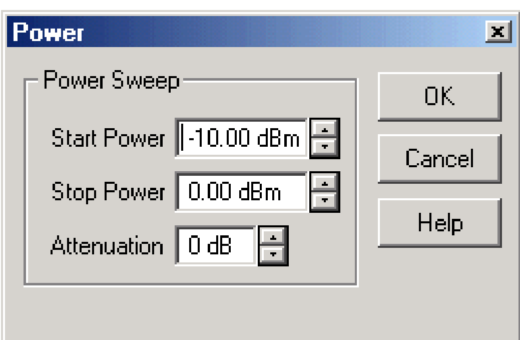
(Attenuator ranges vary, this particular range is 70 dB)



Note: Error correction is fully accurate only for the power level at which a measurement calibration was performed. However, when changing power within the same attenuator range at which the measurement calibration was performed, ratioed measurements can be made with nearly full accuracy (non-ratioed measurements with less accuracy).

Source Attenuation in Power Sweep

To set source attenuation in power sweep mode, first select Power Sweep type, then click **Channel, Power**



Power (Sweep) dialog box help
Start Power Sets the beginning value of the power sweep. **Stop Power** Sets the end value of the power sweep. **Attenuation** Sets the value of the source attenuator for both source ports. Learn more about Source Attenuation. Learn more about Power Sweep.

Attenuation with Uncoupled Ports

PNA models with Option UNL (and standard E8356/57/58A models) have a source attenuator with each test port. The test port power levels can be set independently.

PNA models with Option 1E1 have a single source attenuator that is switched between test ports. If the ports are uncoupled, the test port power levels can be set differently. If the power levels are far apart, the single source attenuator will have to change position when switching between ports. To avoid premature wear of the mechanical step attenuator, the PNA does the following:

- Prevents continuous switching between power ranges
- Automatically engages hold mode after measuring each channel one time

If averaging is on, the hold mode does not engage until the specified number of sweeps is completed for each channel.

Receiver Attenuation

To avoid damaging the PNA receivers, you may need to attenuate the output signal from the device under test.

PNA models with Option 015 or Option 016 provide internal attenuators to both port receivers. The receiver attenuator characteristics are:

- Range:
    - 0 to 50 dB (E8361A only)
    - 0 to 35 dB (all other PNA models)
- Resolution:
    - 10 dB (E8361A only)
    - 5 dB (all other PNA models)

Power Slope

Power slope helps compensate for cable and test fixture power losses at increased frequency.

- With the power slope feature enabled, the port output power increases (or decreases) as the sweep frequency increases.
- The units of power slope are dB/GHz.
- Power slope can only be set to values of 0.5, 1, 1.5, or 2 (positive or negative).

Power ON and OFF during Sweep and Retrace

The frequency range of the PNA covers several internal frequency bands. The higher the frequency range of the PNA, the larger the number of bands. For example, a 9 GHz PNA has 6 frequency bands, a 50 GHz PNA has 25 frequency bands. See the frequency band crossings.

Power to the DUT is set to minimum during band changes to avoid causing power spikes to the DUT. Minimum power is determined by the attenuator setting.

Retrace occurs when the source gets to the end of your selected frequency span and moves back to the start frequency. Power to the DUT is again set to minimum when retracing across frequency bands.

Therefore, the following occurs for various stimulus settings:

1. **Single band sweep** - The power to the DUT is always ON, even during retrace.

2. **Multi-band sweep** - The power to the DUT will go to minimum while sweeping across a band crossing. It will go to minimum again during retrace.

3. **Power sweep** - Because power sweep is always done at a single frequency, the frequency is always in a single band and the source power is always ON. At the end of a power sweep, power is immediately changed back to the start power. This helps prevent the DUT from overheating.

4. **Single sweep** - There are two methods of single sweep:

   - Manual trigger mode - At the end of a sweep the setup for the next sweep completes and the PNA waits for a trigger. During the setup of a multi-band sweep, the band swap completes and power to the DUT is turned back ON.

   - Hold mode - At the end of a sweep, power stays at minimum until the sweep is started.

Caution: Avoid expensive repairs to your PNA. Read Electrostatic Discharge Protection.

Sweep Settings

A sweep is a series of consecutive data point measurements taken over a specified sequence of stimulus values. You can make the following sweep settings:

Sweep Type

Sweep Time

Sweep Setup

See other 'Setup Measurements' topics

Sweep Type

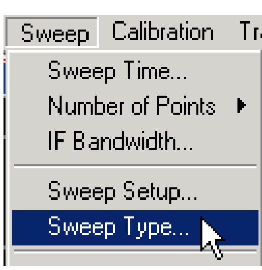
There are five sweep types in the PNA:

- **Linear Frequency** - Sets a linear frequency sweep that is displayed on a standard grid with ten equal horizontal divisions. This is the default sweep type.
- **Log Frequency -** Sets the source to step in logarithmic increments and the data is displayed on a logarithmic x-axis.
- **Power Sweep** - Activates a power sweep at a single frequency that you specify. You can set any power range < 25 dB. (Default -10 dBm to 0 dBm.) Power sweep is used to characterize power-sensitive circuits, with measurements such as gain compression or AGC (automatic gain control) slope. Learn more about power sweep
- **CW Time -** Sets the PNA to a single frequency, and the data is displayed versus time.
- **Segment Sweep -** Sets the PNA to sweep through up to 30 user-defined sweep segments. Learn more about Segment sweep

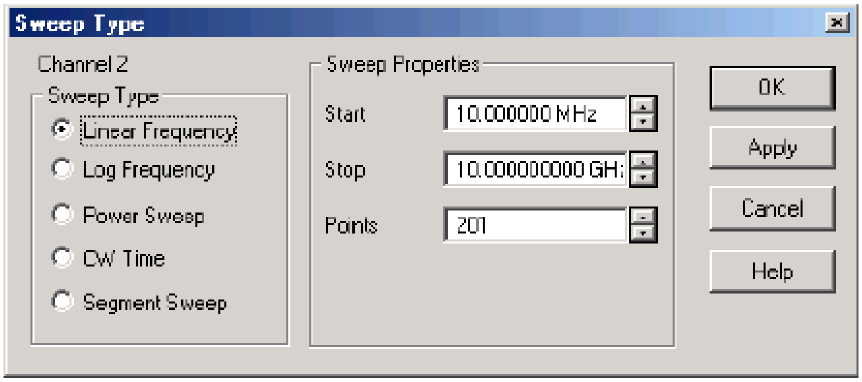
How to set Sweep Type

Use one of the following methods:

- 

- 

Learn more about using the front panel interface



Sweep Type dialog box help

Note: Sweep Settings are not applied until either **OK** or **Apply** is pressed.

Channel The active channel when Sweep Type was selected. Sweep settings will be applied to this channel.

Sweep Type

Linear Frequency Sets a linear frequency sweep that is displayed on a standard grid with ten equal horizontal divisions.

- **Start** Sets the beginning value of the frequency sweep.
- **Stop** Sets the end value of the frequency sweep.
- **Points** Sets the number of data points that the PNA measures during a sweep. Range: 2 to 16001.(Default is 201).

Log Frequency The source is stepped in logarithmic increments and the data is displayed on a logarithmic x-axis. This is usually slower than a continuous sweep with the same number of points.

- **Start** Sets the beginning value of the frequency sweep.
- **Stop** Sets the end value of the frequency sweep.
- **Points** Sets the number of data points that the PNA measures during a sweep. Range: 2 to 16001. (Default is 201).

Power Sweep Activates a power sweep at a single frequency that you specify. Learn about power sweep

- **Start** Sets the beginning value of the power sweep.
- **Stop** Sets the end value of the power sweep.
- **CW Frequency** Sets the single frequency where the PNA remains during the measurement sweep.

CW Time Sets the PNA to a single frequency, and the data is displayed versus time.

- **CW Frequency** Sets the frequency where the PNA remains during the measurement.
- **Sweep Time** Sets the duration of the measurement, which is displayed on the X-axis.
- **Points** Sets the number of data points that the PNA measures during a sweep. Range: 2 to 16001.(Default is 201).

Segment Sweep Sets the PNA to sweep through user-defined sweep segments. Learn more about Segment sweep

- **Independent Power Levels** Check to set the source power level for each segment.
- **Independent IF Bandwidth** Check to set the IF bandwidth for each segment.
- **Independent Sweep Time** Check to set the duration of the measurement for each segment.
- **X-Axis Point Spacing** Check to scale the X-Axis to include only the segments. Learn more.
- **Allow Arbitrary Segments** Check to allow arbitrary frequencies (overlapped or reverse sweeps). Learn more
- **Show Table** Shows the table that allows you to create and edit segments.
- **Hide Table** Hides the segment table from the screen.

OK Applies setting changes and closes the dialog box.

Apply Applies setting changes and leaves the dialog box open to make more setting changes.

Cancel Closes the dialog. Setting changes that have been made since the last **Apply** button click are NOT applied.

Power Sweep

Activates a power sweep at a single frequency that you specify. You can set any power range < 25 dB. (Default -10 dBm to 0 dBm.) Power sweep is used to characterize power-sensitive circuits, with measurements such as gain compression or AGC (automatic gain control) slope.

- Power is stepped from a start value to a stop value at a single frequency.
  - The source amplitude increases in discrete power steps at each data point across the power sweep range.
  - The number of data points and power range determine the size of these steps.
- Choose a combination of source power range and manual attenuator position (if the PNA has source attenuators) to obtain the power sweep range required for the measurement.

Note:To set the attenuation in power sweep, click **Channel, Power** after selecting Power Sweep.

In Power Sweep mode, the following power settings apply:

- Port Power Coupled setting (ON/ OFF) is ignored. Both test ports use the same Power Sweep settings.
- Test Port Power setting is ignored. The code uses the Start Power and Stop Power to determine the power levels present at the test ports.
- Attenuator Control is set to Manual
- Power Slope (dB/GHz) is ignored. The output frequency is CW.

Learn more about Power Settings.

Segment Sweep

Segment Sweep activates a sweep which consists of frequency sub-sweeps, called segments. For each segment you can define independent power levels, IF bandwidth, and sweep time.

Once a measurement calibration is performed on the entire sweep or across all segments, you can make calibrated measurements for one or more segments.

In segment sweep type, the analyzer does the following:

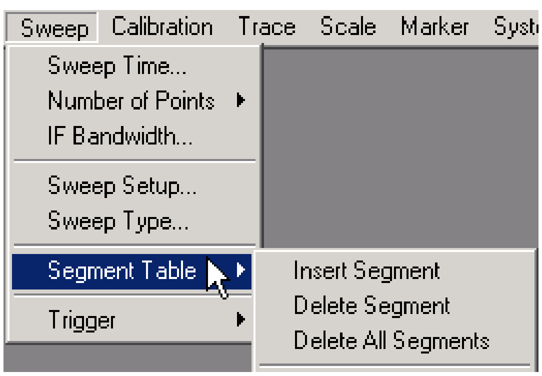
- Sorts all the defined segments in order of increasing frequency
- Measures each point
- Displays a single trace that is a composite of all data taken

Restrictions for segment sweep:

- The frequency range of a segment is not allowed to overlap the frequency range of any other segment.
- The number of segments is limited only by the combined number of data points for all segments in a sweep.
- The combined number of data points for all segments in a sweep cannot exceed 16001.
- All segments should have power levels within the same attenuator range to avoid premature wear of the mechanical step attenuator. See Power Level.

How to Insert, Delete, and Modify Segments

Insert and Delete Segments



Insert Segment - adds a sweep segment before the selected segment.

Note- You can also click the "down" arrow on your keyboard to quickly add many segments.

Delete Segment - removes the selected segment

Delete All Segments - removes all segments

To Modify an Existing Segment

To make the following menu settings available, you must first show the segment table. Click **View**, point to **Tables**, then click **Segment Table.**

	STATE	START	STOP	POI	IFBW	P1 PWR	P2 PWR	TIME
1	ON	20.000000 MHz	1.000000 GHz	21	10.0 kHz	17.00 dBm	0.00 dBm	2.474 msec
2	ON	1.000000 GHz	4.000000 GHz	21	35.0 kHz	17.00 dBm	0.00 dBm	630.000 µsec

The above graphic shows the Segment table with all independent settings selected, including source power uncoupled (two power settings).

STATE Click the box on the segment to be modified. Then use the up / down arrow to turn the segment ON or OFF.

START Sets start frequency for the segment. Click the box and type a value and the first letter of a suffix (**K**Hz, **M**hz, **G**Hz). Or double-click the box to select a value.

STOP Sets stop frequency for the segment. Click the box and type a value and the first letter of a suffix (**K**Hz, **M**hz, **G**Hz). Or double-click the box to select a value.

POINTS Sets number of data points for this segment. Type a value or double-click the box to select a value.

To set IFBW, Power, and Sweep Time independently for each segment:

On the **Sweep** menu, click **Sweep Type**, then **Segment Sweep**.
Check the appropriate **Sweep Properties** boxes

Then click the box and type a value or double-click the box and select a value.

Note: If the following are NOT set, the entire sweep uses the channel IFBW, Power, and Time settings.

IFBW Sets the IF Bandwidth for the segment.

POWER Sets the power level for the segment.

TIME Sets the Sweep time for the segment.

Learn more about using the front panel interface.

X-Axis Point Spacing

This feature applies to Segment Sweep only and affects how a trace is drawn on the screen.

How to select X-Axis Point Spacing

On the Sweep Type dialog box, click **Segment Sweep**

Then check **X-Axis Point Spacing**

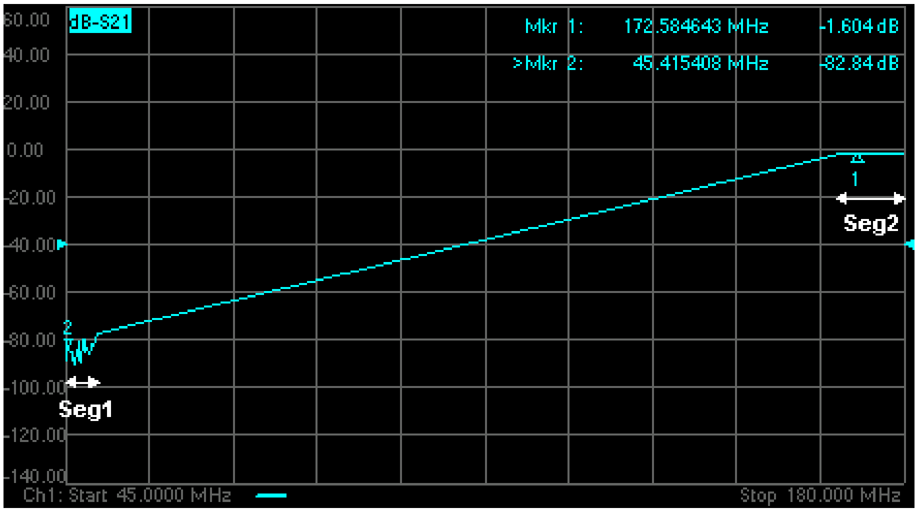
Learn more about using the front panel interface

Without X-axis point spacing, a multi-segment sweep trace can sometimes result in squeezing many measurement points into a narrow portion of the x-axis.

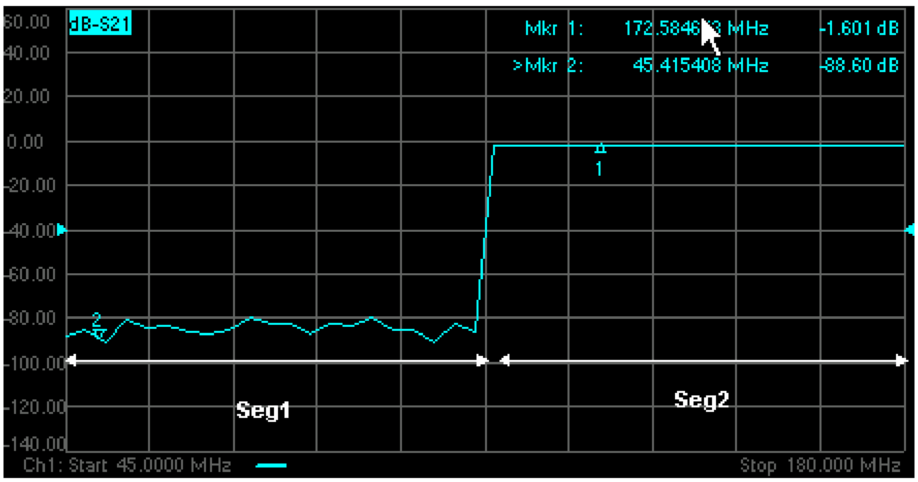
With X-Axis Point Spacing, the x-axis position of each point is chosen so that all measurement points are evenly spaced along the x-axis.

For example, given the following two segments:

	STATE	START	STOP	POINTS
1	ON	45.000000 MHz	50.000000 MHz	21
2	ON	170.000000 MHz	180.000000 MHz	21



Without X-Axis Point Spacing



With X-Axis Point Spacing

Arbitrary Segment Sweep

This feature enables arbitrary frequencies to be entered into the segment sweep table, which allows the following:

- segments can have overlapping frequencies.
- reverse sweep (set the stop frequency to be less than the start frequency).

How to select Arbitrary Segment Sweep
On the Sweep Type dialog box, click **Segment Sweep**
Then check **Allow Arbitrary Segment Sweep**
Learn more about using the front panel interface

Note: The markers, display, limit lines, formatting, and calibration features may produce unusual results for overlapped and reverse sweep segments.

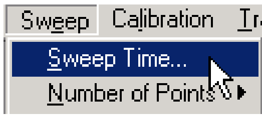
When Allow Arbitrary Segment is checked, X-axis point spacing is automatically turned ON.

Sweep Time

The PNA automatically maintains the fastest sweep time possible with the selected measurement settings. However, you can increase the sweep time.

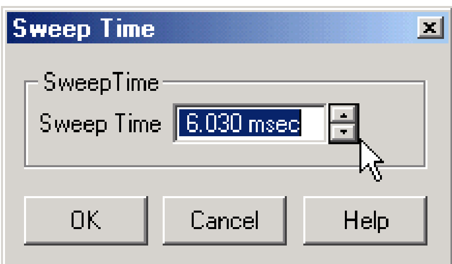
How to set Sweep Time

Use one of the following methods:

- 

- 

Learn more about using the front panel interface



Sweep Time dialog box help

Specifies the time the PNA takes to acquire data for a sweep. The maximum sweep time of the PNA is 86400 seconds (1 day).

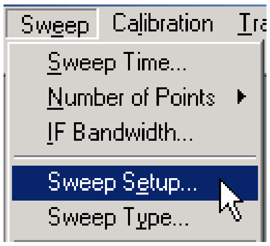
Note: If sweep time accuracy is critical, use ONLY the up and down arrows next to the sweep time entry box to select a value that has been calculated by the PNA. Do NOT type a sweep time value.

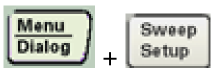
- The actual sweep time includes this acquisition time plus some "overhead" time.
- The PNA automatically maintains the fastest sweep time possible with the selected measurement settings. However, you can increase the sweep time using this setting.
- Enter **0** seconds to return the analyzer to the fastest possible sweep time.
- The sweep time setting is applied to the active channel.
- Stepped sweep is automatically selected if sweep time is at, or above, 8msec.
- A Sweep Indicator  appears on the data trace when the Sweep Time is 0.3 seconds or greater, or if trigger is set to Point Sweep Mode. The indicator is located on the last data point that was measured by the receiver. If the indicator is stopped (point sweep mode) the source has already stepped to the next data point.

Sweep Setup

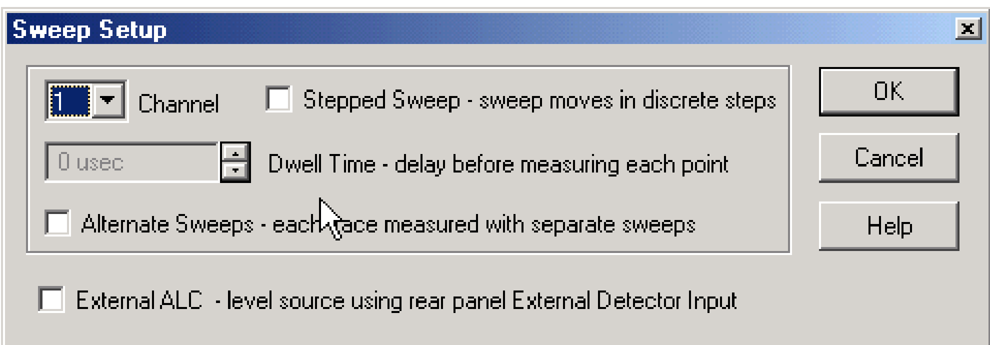
How to make Sweep Setup settings

Use one of the following methods:

- 

- 

Learn more about using the front panel interface



Sweep Setup dialog box help

Channel Specifies the channel that the settings apply to.

Stepped Sweep When checked (Stepped Sweep) the PNA source is tuned, then waits the specified Dwell time, then takes response data, then tunes the source to the next frequency point. This is slower than Analog Sweep, but is more accurate when testing electrically-long devices.

When cleared (Analog Sweep) the PNA takes response data AS the source is sweeping. The sweep time is faster than Stepped, but could cause measurement errors when testing electrically-long devices.

When the dialog checkbox is cleared, the PNA could be in either Analog or Step mode. Stepped sweep is automatically selected when any of the following occur:

- IF Bandwidth is at, or below, 100Hz
- Sweep time is at, or above, 8msec.
- Source Power Correction is ON.

Dwell Time Specifies the time the source stays at each measurement point before the analyzer takes the data. Only applies to stepped sweep. The maximum dwell time is 100 seconds. See also Electrically Long Devices.

Alternate Sweeps When checked, the PNA measures only one receiver per sweep.
When cleared, the PNA measures both the A and B receivers (if used) each sweep. See also Crosstalk.

External ALC When checked, the analyzer is enabled to receive an external signal that you provide for leveling the source output. The external ALC signal input connector is the External Detector Input on the rear panel. Available ONLY on 3 GHz, 6 GHz, and 9 GHz PNA models. Use this when using a booster amplifier in your test setup.

Trigger

A trigger is a signal that causes the PNA to make a measurement sweep. The PNA offers great flexibility in configuring the trigger function.

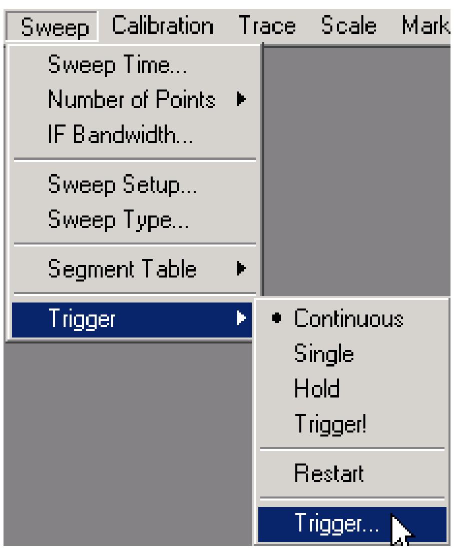
View the interactive Trigger Model animation to see how triggering works in the PNA.

- How to Set Trigger
- Source
- Scope
- Channel Settings
- Restart
- External Triggering

See other 'Setup Measurements' topics

How to Set Trigger

Use one of the following methods:

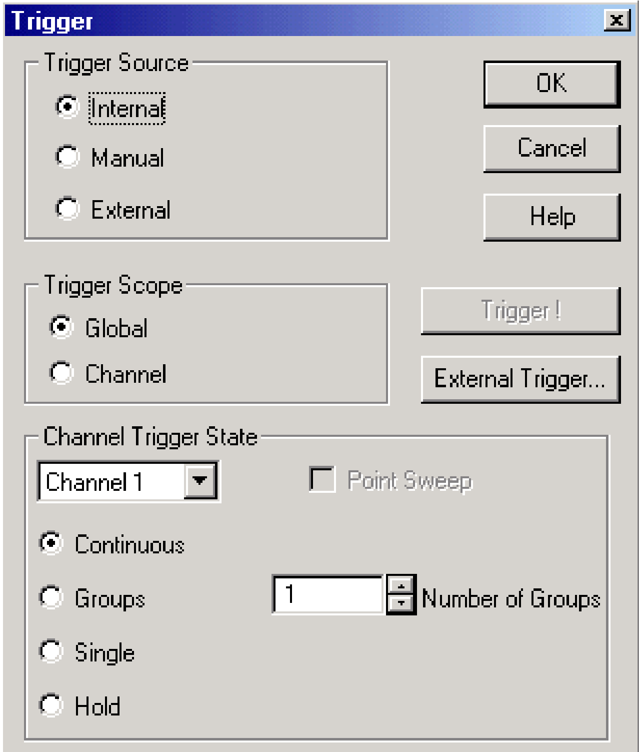
- 

- 

 + 

Learn more about using the front panel interface

Note: The settings Continuous, Single, Hold apply only to the active channel. These settings are available from the Sweep Trigger menu (shown above) and Active Entry keys.



Trigger dialog box help

View the interactive Trigger Model animation to see how triggering works in the PNA.

Trigger Source

These settings determine **where** the trigger signals originate for all existing channels. A valid trigger signal can be generated only when the PNA is not sweeping.

Internal Continuous trigger signals are sent by the PNA as soon as the previous measurement is complete.

Manual One trigger signal is sent when invoked by the Trigger button, the active toolbar, or a programming command.

External Trigger signals are sent through the Aux I/O connector (pin19) on the rear panel. The trigger is level sensitive and can be either active TTL high or active TTL low.

Trigger Scope

These settings determine **what** is triggered.

Global All channels not in Hold receive the trigger signal [Default setting]

Channel Only the next channel that is not in Hold receives the trigger signal. This is not very obvious or useful unless Trigger Source is set to Manual. This setting enables Point Sweep mode.

Channel Trigger State

These settings determine **how many** trigger signals the channel will accept.

Note: ONE trigger signal causes ALL measurements in the channel to be made; first the forward measurements (S11, S21) are made simultaneously - then reverse measurements (S12, S22). There are two exceptions to this:

1. When Alternate sweep is selected, only one receiver is measured per trigger.
2. When Point Sweep is selected, only one data point is measured per trigger.

Continuous The channel accepts an infinite number of trigger signals.

Groups The channel accepts only the number of trigger signals that is specified in the Number of Groups text box, then goes into Hold. Before selecting groups you must first increment the Number of Groups text box to greater than one.

Number of Groups Specify the number of trigger signals the channel will accept before going into Hold. First increment to desired number, then select Groups.

Single The channel accepts ONE trigger signal, then goes into Hold.

Hold The channel accepts NO trigger signals.

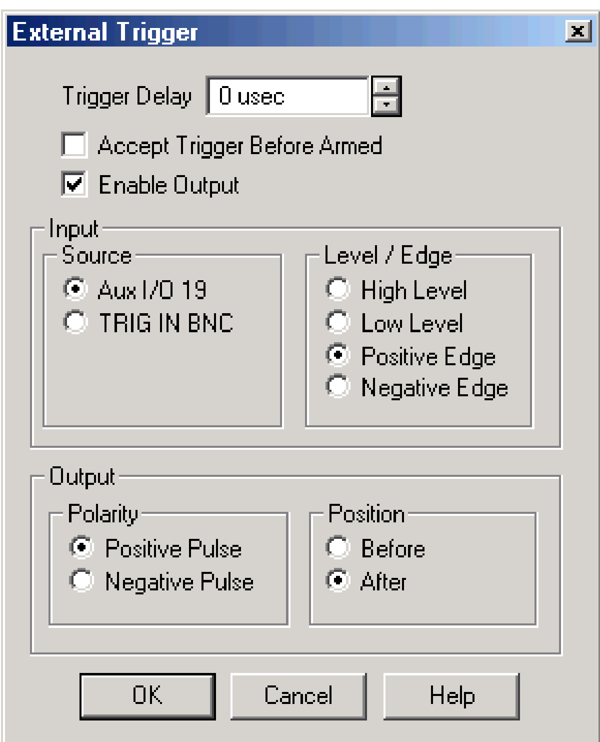
Point Sweep When triggered, the channel measures the next data point in the sweep. Subsequent trigger signals go to the same channel until all of its measurements are complete. Then the next channel that is not in Hold is triggered. Available only when both Trigger Scope = CHANNEL and Trigger Source = MANUAL. **Note:** Selecting Point Sweep does not change how sweep average is calculated. Learn more.

Trigger! - Manually sends one trigger signal to the PNA.

External Trigger - Invokes the External Trigger dialog box.

Restart - Channels in Hold are set to single trigger (the channel accepts a single trigger signal). All other settings are unaffected, including decrementing trigger counts.

View the interactive Trigger Model animation to see how triggering works in the PNA.



External Trigger dialog box help

Note: Although changes to settings in this dialog box affect the measurement immediately, you can click **Cancel** to revert to the previous settings.

Input

Trigger Delay After an external trigger is applied, the start of the sweep is held off for this specified amount of time plus any inherent latency.

Channel (Available only when Channel trigger is selected.) Select a channel to assume the specified trigger delay value. Each channel can have a different delay value.

Source The PNA accepts trigger input signals through one of the following input connectors:

- Aux I/O - pin 19
- Handler I/O - pin 18
- I/O 1 (TRIG IN) BNC

Level / Edge

High Level The PNA is triggered when it is armed (ready for trigger) and the TTL signal at the select input is HIGH.

Low Level The PNA is triggered when it is armed (ready for trigger) and the TTL signal at the select input is LOW.

Positive Edge After the PNA arms, it will trigger on the next positive edge. If Accept Trigger Before Armed is set, PNA will trigger as soon as it arms if a positive edge was received since the last data was taken.

Negative Edge After the PNA arms, it will trigger on the next negative edge. If Accept Trigger Before Armed is set, PNA will trigger as soon as it arms if a negative edge was received since the last data was taken.

Edge triggering is NOT available on the following PNA models: E835xA, E8362A, E8363A, E8364A.

Accept Trigger Before Armed When checked, as the PNA becomes armed (ready to be triggered), the PNA will immediately trigger if any triggers were received since the last taking of data. The PNA remembers only one trigger signal. All others are ignored.

When this checkbox is cleared, any trigger signal received before PNA is armed is ignored.

This feature is only available when positive or negative EDGE triggering is selected.

Output

Enable Output When checked, the PNA is enabled to send trigger signals out the rear-panel I/O (TRIG OUT) BNC connector.

Polarity The trigger pulse output from the PNA is either in the Positive or Negative direction.

Position The trigger pulse output is sent either BEFORE or AFTER a receiver measurement.

Configure external triggering remotely using CONTrol:SIGNal (SCPI) or ExternalTriggerConnectionBehavior (COM).

Data Format and Scale

A data format is the way the analyzer presents measurement data graphically. You should pick a data format appropriate to the information you want to learn about the test device.

- Rectangular (Cartesian) Display Formats
- Polar
- Smith Chart
- How to Change Format
- Scale, Reference Level and Position
- Magnitude Offset

See other 'Setup Measurements' topics

Rectangular Display Formats

Seven of the nine available data formats use a rectangular display to present your measurement data. This display is also known as Cartesian, X/Y, or rectilinear. The rectangular display is especially useful for clearly displaying frequency response information of your test device.

- Stimulus data (frequency, power, or time) appears on the X-axis, scaled linearly
- Measured data appears on the Y-Axis.

Log Mag (Logarithmic Magnitude) Format

- Displays Magnitude (no phase)
- Y-axis: dB
- Typical measurements:
  - Return Loss
  - Insertion Loss or Gain

Phase Format

- Displays Phase (no magnitude)
- Y-axis: Phase (degrees)
- Typical Measurements:
  - Deviation from Linear Phase

Group Delay Format

- Displays signal transmission (propagation) time through a device

- Y-axis: Time (seconds)
- Typical Measurements:
  - Group Delay

Linear Magnitude Format

- Displays positive values only
- Y-axis: Unitless (**U**) for ratioed measurements
  Watts (**W**) for unratioed measurements.
- Typical Measurements:
  - reflection and transmission coefficients (magnitude)
  - time domain transfer

SWR Format

- Displays reflection measurement data calculated from the formula (1+ρ)/ (1ρ) where ρ is reflection coefficient.
- Valid only for reflection measurements.
- Y axis: Unitless
- Typical Measurements:
  - SWR

Real Format

- Displays only the real (resistive) portion of the measured complex data.
- Can show both positive and negative values.
- Y axis: Unitless
- Typical Measurements:
  - time domain
  - auxiliary input voltage signal for service purposes

Imaginary Format

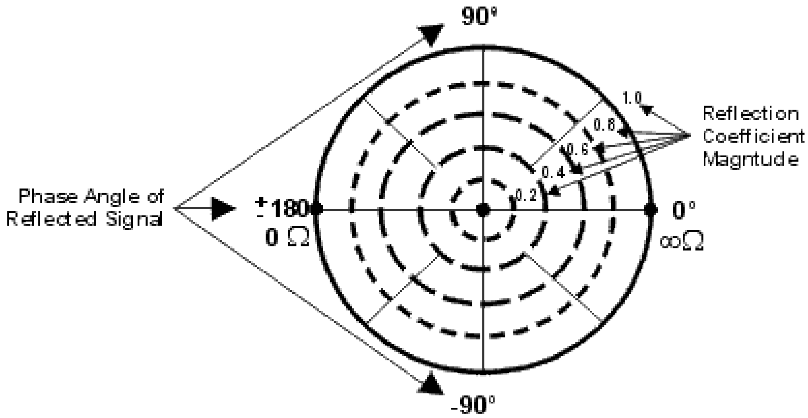
- Displays only the imaginary (reactive) portion of the measured data.
- Y - axis: Unitless
- Typical Measurements:
  - impedance for designing matching network

Polar Format

Polar format is used to view the magnitude and phase of the reflection coefficient (Γ) from your S_{11} or S_{22} measurement.

You can use Markers to display the following:

- Linear magnitude (in units) or log magnitude (in dB)
- Phase (in degrees)



- The dashed circles represent reflection coefficient. The outermost circle represents a reflection coefficient (Γ) of 1, or total reflected signal. The center of the circle represents a reflection coefficient (Γ) of 0, or no reflected signal.
- The radial lines show the phase angle of reflected signal. The right-most position corresponds to zero phase angle, (that is, the reflected signal is at the same phase as the incident signal). Phase differences of 90°, ±180°, and -90° correspond to the top, left-most, and bottom positions on the polar display, respectively.

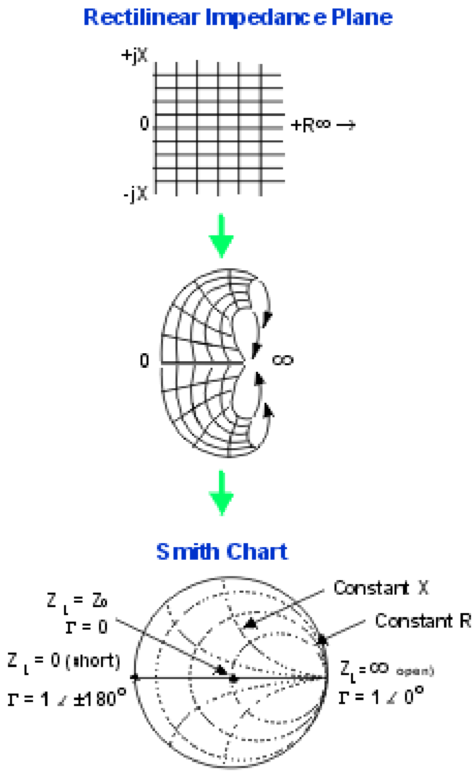
Smith Chart Format

The Smith chart is a tool that maps the complex reflection coefficient (Γ) to the test device's impedance.

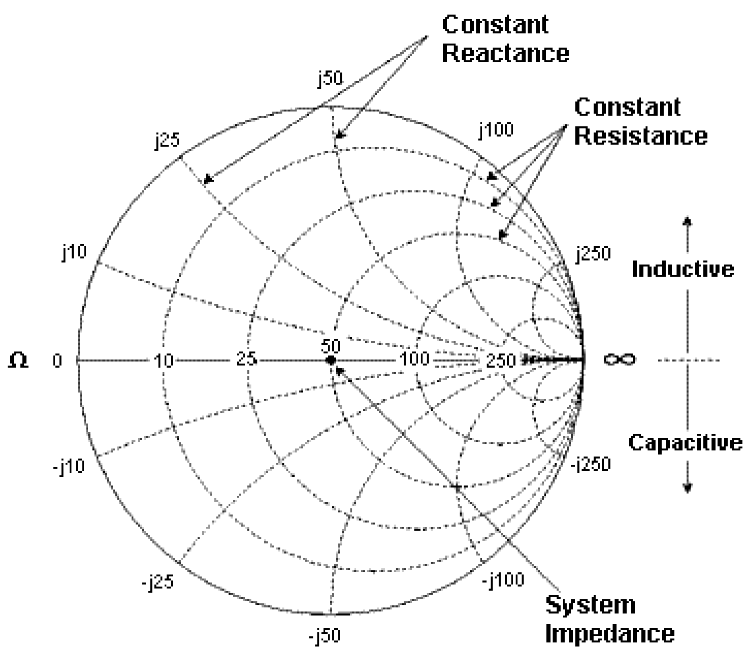
In a Smith chart, the rectilinear impedance plane is reshaped to form a circular grid, from which the series resistance and reactance can be read (R + jX).

You can use Markers to display the following:

- Resistance (in units of ohms)
- Reactance as an equivalent capacitance (in units of farads) or inductance (in units of henrys)



Interpreting the Smith Chart

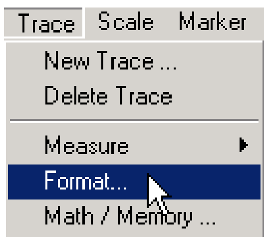


- Every point on the Smith Chart represents a complex impedance made up of a real resistance (r) and an imaginary reactance (r+-jX)
- The horizontal axis (the solid line) is the real portion of the impedance - the resistance. The center of the horizontal axis always represents the system impedance. To the far right, the value is infinite ohms (open). To the far left, the value is zero ohms (short)

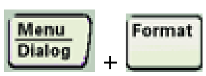
- The dashed circles that intersect the horizontal axis represent constant resistance.
- The dashed arcs that are tangent to the horizontal axis represent constant reactance.
- The upper half of the Smith chart is the area where the reactive component is positive and therefore inductive.
- The lower half is the area where the reactive component is negative and therefore capacitive.

How to set the Format

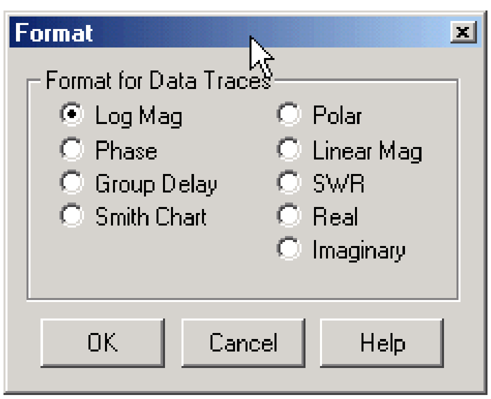
Use one of the following methods:

- 

- 

- 

Learn more about using the front panel interface



Format dialog box help

Log Mag

Phase

Group Delay

Smith Chart

Polar

Linear Mag

SWR

Real

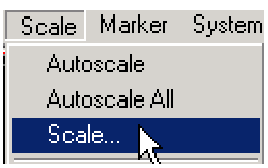
Imaginary

Scale, Reference Level and Position

The Scale, Reference Level and Reference Position settings (along with format) determine how the data trace appears on the PNA screen.

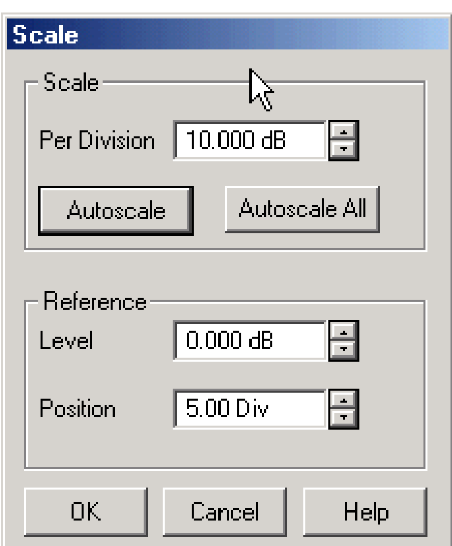
How to set Scale, Reference Level, and Position

Use any of the following methods:

- 

- 

Learn more about using the front panel interface



Scale dialog box help

Scale

Per Division Sets the value of the vertical divisions of a rectangular display format. In Polar and Smith Chart formats, scale sets the value of the outer circumference. Range: 0.001dB/div to 500 dB/div

Autoscale - Automatically sets value of the vertical divisions and reference value to fit the ACTIVE data trace within the grid area of the screen. The stimulus values and reference position are not affected.

The analyzer determines the smallest possible scale factor that will allow all the displayed data to fit onto 80 percent of the vertical grid.

The reference value is chosen to center the trace on the screen.

Autoscale All Automatically scales ALL data traces in the ACTIVE WINDOW to fit vertically within the grid area of the screen.

Reference

Level In rectangular formats, sets the value of the reference line, denoted by 

on the PNA screen. Range: -500 dB to 500 dB.

In Polar and Smith chart formats, reference level is not applicable.

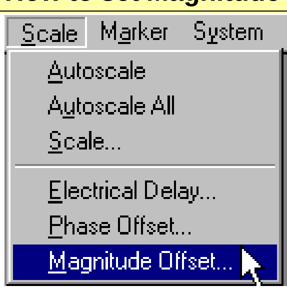
Position In rectangular formats, sets the position of the reference line. Zero is the bottom line of the screen and ten is the top line. Default position is five (middle).

In Polar and Smith chart formats, reference position is not applicable.

Magnitude Offset

Magnitude Offset allows you to offset the magnitude (not phase) data by a fixed and / or sloped value in dB. If the display format is Linear Magnitude or Real (unitless), the conversion from dB is performed and the correct amount of offset is implemented.

How to set Magnitude Offset



Learn more about using the front panel interface

Magnitude Offset dialog box help

The Magnitude offset setting affects only the active trace.

Offset Offsets the entire data trace by the specified value.

Slope Offsets the data trace by a value that changes with frequency. The offset slope begins at 0 Hz.

For your convenience, the offset value at the start frequency is calculated and displayed.

See where this operation is performed in the data processing chain.

Pre-configured Measurement Setups

- Pre-configured setups for NEW measurements
- Pre-configured arrangements for EXISTING measurements

Before reading this topic, it is important to understand Traces, Channels, and Windows in the PNA.

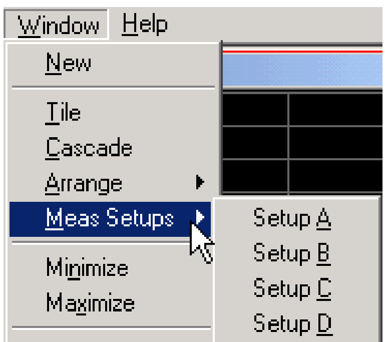
See other 'Setup Measurements' topics

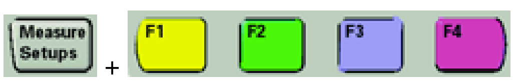
Pre-configured Setups for NEW Measurements

Each of the following setups **creates new traces**. Existing traces and their settings will be lost, unless you first save them.

How to select a pre-configured measurement setup

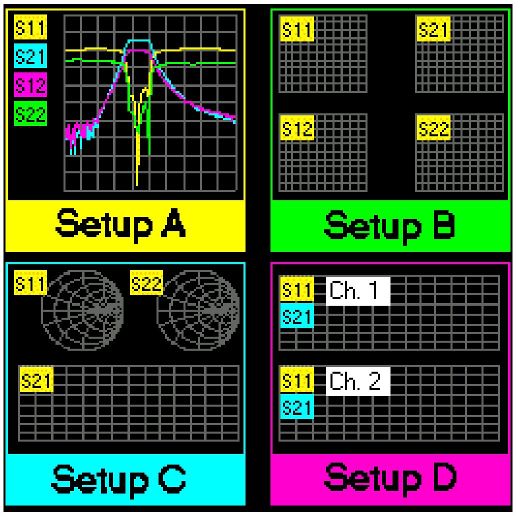
Use one of the following methods:

- 

- 

Learn more about using the front panel interface

The following are the four pre-configured measurement setups:

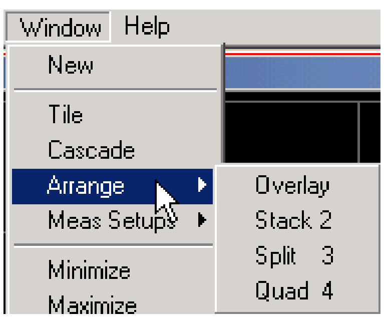


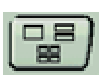
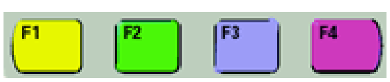
Arranging Existing Measurements

The following arrangements place EXISTING measurements into pre-configured Window arrangements using a sort algorithm.

How to select an Existing measurement arrangement

Use one of the following methods:

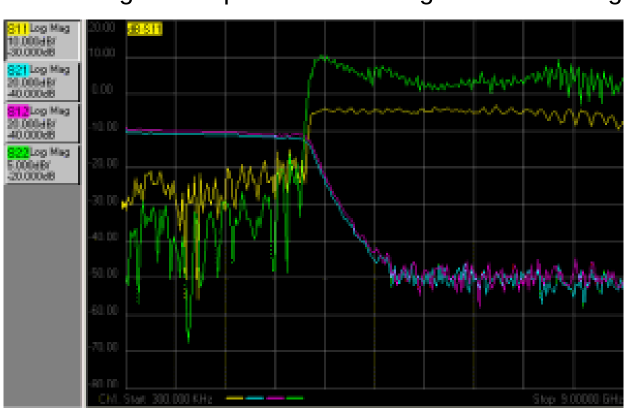
- 

-  + 

Learn more about using the front panel interface

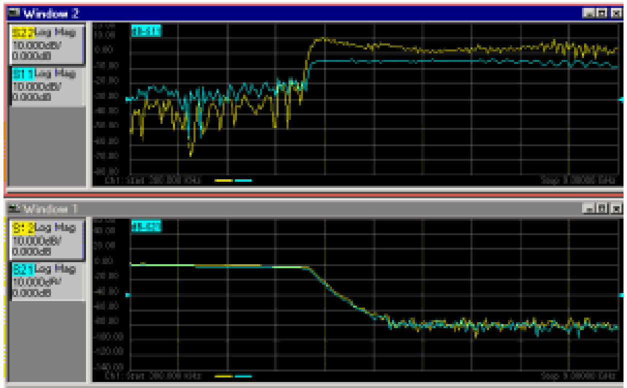
Overlay Arrangement

This configuration places all existing traces in a single window, all overlaid on each other.



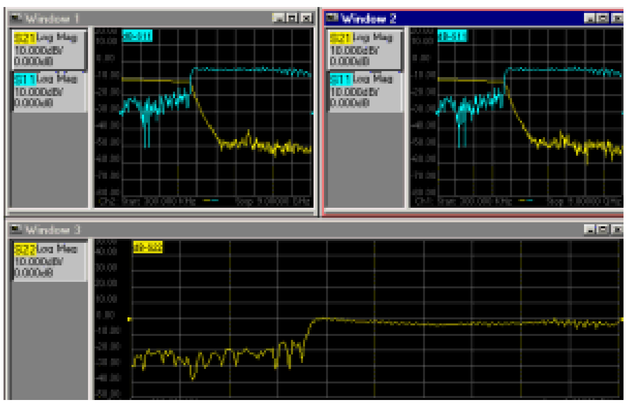
Stack 2 Arrangement

This configuration places all existing traces in two "stacked" windows.



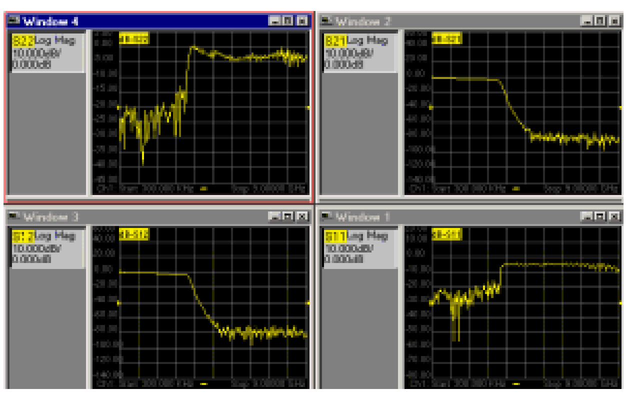
Split 3 Arrangement

This configuration places all existing traces in three windows, two on top and one below.



Quad 4 Arrangement

This configuration places all existing traces in four windows, one window in each screen quadrant.



Sort Algorithm

The sort algorithm for the Arrange Windows feature is designed to:

- Divide traces among windows based on their properties
- Group traces with common properties

The algorithm sorting is based on the following trace properties, in order of priority:

1. Format: circular (polar or Smith) versus rectilinear (log mag, lin mag, group delay, etc.)
2. Channel number
3. Transmission versus reflection

Note: The analyzer limitation of four traces per window overrides this algorithm. An error occurs if the arrange selection cannot be completed with the current number of traces on the screen.

Customize the PNA Screen

You can customize your PNA screen by showing or hiding the following display elements. All of these selections are made from the PNA **View** menu.

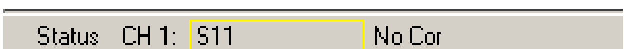
- **Status Bar**
- **Toolbars**
- **Tables**
- **Measurement Display**
- **Data and Memory Trace**
- **Title Bars**
- **Minimize Application**

Learn about using pre-configured measurements and windows arrangements

Learn about Traces, Channels, and Windows on the PNA

See other 'Setup Measurements' topics

Status Bar

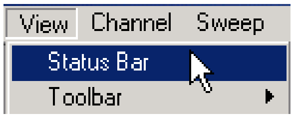


When enabled, the status bar is displayed along the bottom of the PNA screen. It shows the following:

- Active channel
- Measurement parameter for the active trace
- Trace Math
- Error correction for the active trace
- Averaging Factor for the active channel
- Smoothing Percentage
- Transform (On)
- Gating (On)
- IF Gating Enabled for Pulsed App: (G)
- Manual IF Filtering for Pulsed App: (F)
- Delay if invoked using Phase Offset, Electrical Delay, or Port Extensions.
- Loss if invoked using Magnitude Offset.
- GPIB status: local (LCL) or remote (RMT)
- Error Status: (LVL, LCK, etc)

The status bar state (ON or OFF) will not change when the PNA is Preset.

How to display the Status bar

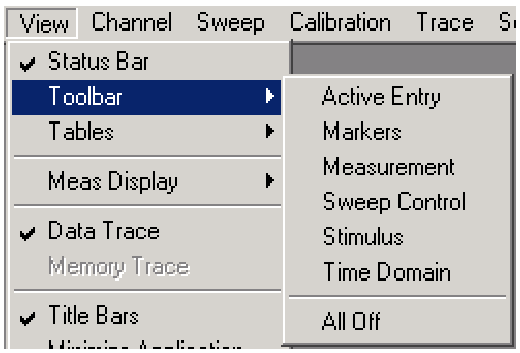


Learn more about using the front panel interface

Toolbars

You can display up to five different toolbars to allow you to easily set up and modify measurements.

How to display toolbars



Learn more about using the front panel interface

Learn about each toolbar:

- **Active Entry**
- **Markers**
- **Measurement**
- **Sweep Control**
- **Stimulus**
- **Time Domain**
- **All Off**

Note: There is also a Cal Set toolbar available for Monitoring Error Terms.

Active Entry Toolbar

The active entry toolbar is displayed at the top of the screen, below the menu bar. It allows you to make selections from the active function using the mouse or by pressing the front panel key with the corresponding color.

Learn more about using the front panel interface

Markers Toolbar

The markers toolbar allows you to set up and modify markers. It shows:

- Marker number
- Stimulation value
- Marker functions
    - Delta
    - Start/Stop
    - Center/Span

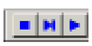
Tip: To use the Front Panel Knob to change marker position, first click the **Stimulus** field of the marker toolbar. Then turn the knob.

Learn more about Markers

Measurement Toolbar

The measurement toolbar allows you to **create a new trace** for a desired S-parameter measurement in a current window or new window.

Sweep Control Toolbar

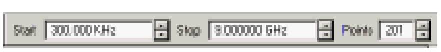


In left to right order, the buttons on this toolbar set the active channel to:

- **Hold** mode
- **Single** sweep, then Hold mode
- **Continuous** sweep

Learn more about Channel Trigger State.

Stimulus Toolbar



The stimulus toolbar allows you to view, set up, and modify the sweep stimulus. It shows the:

- **Start** value
- **Stop** value
- Number of **points**

Time Domain

The Time Domain toolbar allows you to do the following:

- Turn Transform and Gating ON / OFF
- Change the Start / Stop times for both Transform and Gating

- **More...**launches the Time Domain Transform dialog box
- **X** Closes the toolbar

The front panel Tab key steps through all of the settings on all of the toolbars on the display. If Tab does not work, press one of the Active Toolbar (color) keys.

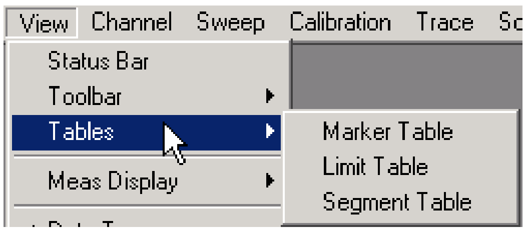
All Off

This allows you to **hide all toolbars** with a single selection.

Tables

Tables are displayed at the bottom of the selected window. Only one table may be displayed at a time for a window.

How to display Tables



Learn more about using the front panel interface

You can display tables for the following:

- **Marker Table**
- **Limit Line Table**
- **Segment Table**

Marker Table

You can display a table of marker settings. These settings include the:

- Marker number
- Marker reference (for delta measurements)
- Frequency
- Time and Distance (for Time Domain measurements)
- Response

Learn more about Markers

Limit Line Table

You can display, set up, and modify a table of limit test settings. These include:

- Type (MIN, MAX, or OFF)
- Beginning and ending stimulus values
- Beginning and ending response values

Learn more about Limit Lines

Segment Sweep Table

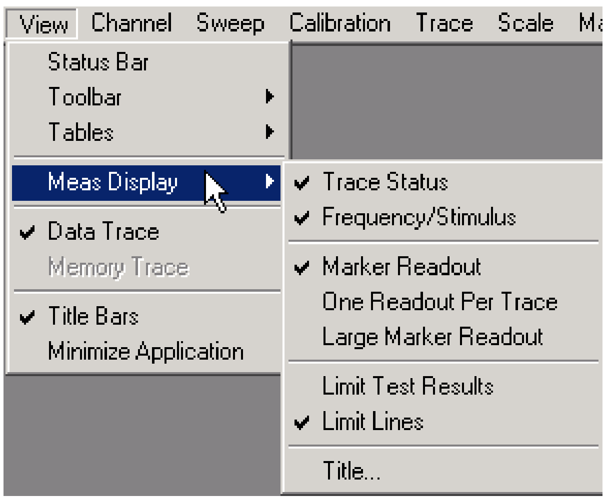
You can display, set up, and modify a table of segment sweep settings. These include:

- State (On/Off)
- Start and Stop frequencies
- Number of Points
- IF Bandwidth (if independent levels)
- Power Level (if independent levels)
- Sweep Time (if independent levels)

Learn more about Segment sweep

Measurement Display

How to display measurement information



Learn more about using the front panel interface

Trace Status

Frequency Stimulus

Marker Readout

Limit Test Results

Limit Lines

Title

Trace Status

Trace status is displayed to the left of each window on the screen. It shows the:

- Measurement parameter
- Format
- Scaling factor
- Reference level

Click a trace status button to make that measurement active.

How to display Trace Status buttons

Frequency/Stimulus

Frequency/stimulus information is displayed at the bottom of each window on the screen. It shows:

- Channel number
- Start value
- Stop value

How to display Frequency/Stimulus information

Marker Readout

- Checked - Shows the following Marker information in the top right corner of each window.
  - Marker number
  - Stimulus value
  - Response value in marker's selected format
- Cleared - Shows **no** Marker Readout display.

One Readout Per Trace

- Checked - Shows the Readout of the single **active marker** per trace.
- Cleared - Shows up to 5 markers on the **active trace**.

Large Marker Readout

- Checked - Shows the Marker Readout in **large font size for easy reading**.

- Cleared - Shows the Marker Readout in **normal font size**.

Learn more about Markers

How to change Marker readout settings

Limit Line Test Results

Limit line test results, **Pass** or **Fail,** are displayed on the right side of the designated window.

Limit Lines

Limit lines are displayed for the active trace in the designated window. Their position depends on:

- Limit levels
- Format
- Scaling
- Reference level

Learn more about Limit Lines

How to display Limit Lines and Results

Title

You can create and display a title for each measurement window using the keyboard. You can also use the following Title Entry dialog box.

The title is displayed in the upper-left corner of the selected window.

To clear a title, delete the title from the dialog box entry area and click OK.

How to display a Title

Data Trace and Memory Trace

You can view or hide the active data or memory trace.

- Make a trace active by clicking the trace status button
- To view a memory trace you must first store a trace in memory. Click **Trace**, then **Math / Memory**, then **Data => Memory.**

Learn more about Math operations

Title Bars

The Title bar shows the window number and Minimize / Maximize icons.

When checked, the title bars for all PNA windows are shown.

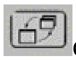
Clear Title Bars to hide the title bars of all the windows on the screen. This allows more room to display measurement results.

<u>How to change the Title Bar setting</u>

Minimize Application

On the PNA Menu, Click View / Minimize Application to minimize the PNA application to the Windows taskbar.

To restore the PNA application:

- Click the PNA application on the Windows taskbar
- Or press on the front panel.

Copy Channels

You can copy the channel settings from an existing channel to a new or another existing channel.

- **Why Copy Channels**
- **How to Copy Channels**
- **List of Channel Settings**

Other Setup Measurements Topics

Why Copy Channels

Copy channel settings if you need to create several channels that have slightly different settings.

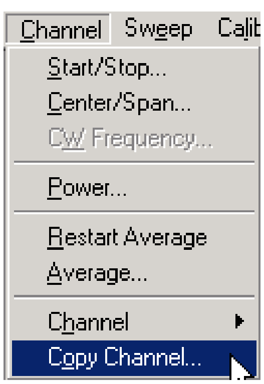
For example, you have an amplifier that you want to characterize over a frequency span with several different input power levels.

Follow these steps:

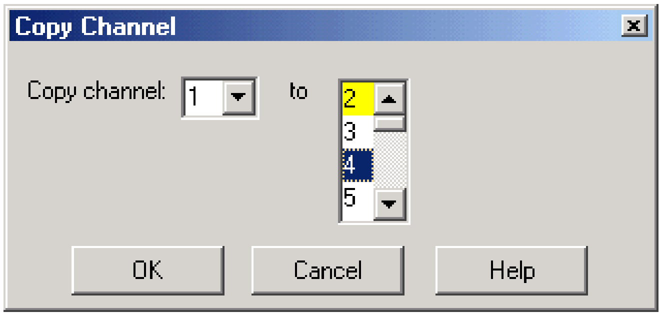
1. Create one measurement with your optimized channel settings.
2. Copy that channel to new channels.
3. Change the power level on the new channels.

The alternative to using Copy Channels is to create new default measurements on new channels. Then change every channel setting to your new requirement. This is very time consuming and thus shows the benefit of the Copy Channels feature.

How to copy channels



Learn more about using the front panel interface



Copy Channel dialog box help

Copies an existing channel's settings to another channel.

Copy channel: Select a channel to copy.

to: Scroll to select a channel to copy settings to. Channel numbers that are currently being used are highlighted. They can be selected and overwritten.

Notes:

- You can copy channel settings to ONLY one new or existing channel. Repeat this operation to copy to more than one channel.
- The new channel is ALWAYS copied to the Active window. If you want the new channel in its own window, first create a new measurement in a new window. Then make sure it is the Active window before you copy the channel into it.
- The measurement in the new channel becomes the active measurement.
- Only the channel settings are copied. The measurement trace is NOT copied to the new channel.
  - If measurements already exist on a channel being copied to, the measurements on that channel will not change, but they will assume the new channel settings.
  - If a NEW channel is copied TO, an S11 measurement is created in order to view the channel settings.

For example:

1. **Existing** channel 1: S21 measurement
2. **Copy** channel 1 to NEW channel 2
3. **Result**: channel 2: S11 measurement with channel 1 copied settings. Both measurements are in the same window. The S11 measurement is the active measurement.

For more information see Traces, Channels, and Windows on the PNA

List of Channel Settings

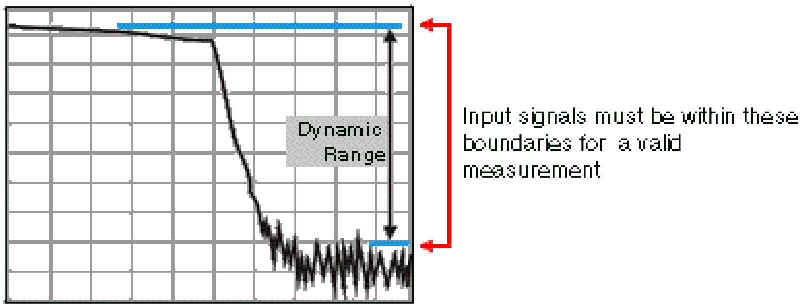
- Frequency Span
- Power

- Cal Set usage
- Source Power Cal data
- IF Bandwidth
- Number of Points
- Sweep Settings
- Average
- Trigger (some settings)

Dynamic Range

Dynamic range is the difference between the analyzer receiver's maximum input power and the minimum measurable power (noise floor). For a measurement to be valid, input signals must be within these boundaries.

Increasing dynamic range is important if you need to measure very large variations in signal amplitude, such as filter bandpass and rejection. The dynamic range is shown below for an example measurement.



To help reduce measurement uncertainty, the analyzer dynamic range should be greater than the response that the DUT exhibits. For example, measurement accuracy is increased when the DUT response is at least 10 dB above the noise floor. The following methods can help you increase the dynamic range.

- **Increase the Device Input Power**
- **Reduce the Receiver Noise Floor**
- **Use the Front-Panel Jumpers (if your PNA has a configurable test set)**

Other topics about Optimizing Measurements

Increase Device Input Power

Increase the DUT input power so that the analyzer can more accurately detect and measure the DUT output power. However, use caution - too much power can damage the analyzer receiver or cause compression distortion.

Caution! Receiver input damage level: +15 dBm.

increase input power to the device:

1. In the **Channel** menu, click **Power**
2. Select the **Power On** check box.
3. Under **Port Selection**, select the port that you are using as the source output.
4. In the **Test Port Power** box, type the value or press the arrow button to select the value you want.
5. Click **OK**.

Tip: You can further increase dynamic range by using an external booster amplifier to increase the input power to the DUT. See High Power Amplifier Measurements.

Reduce the Receiver Noise Floor

You can use the following techniques to lower the noise floor and increase the analyzer's dynamic range.

- Reduce **crosstalk** between the PNA receivers when measuring signals close to the noise floor. See Receiver Crosstalk.)
- Use **Sweep Averaging** - learn more about Sweep Average
- Reduce the **IF Bandwidth** - learn more about IF Bandwidth.
- In Segment sweep mode each segment can have its own IF bandwidth. For example, when measuring a filter:
  - In the passband, the IF bandwidth can be set wider for a fast sweep rate, as long as high-level trace noise is kept sufficiently small.
  - In the reject band, where noise floor contributes significantly to measurement error, the IF bandwidth can be set low enough to achieve the desired reduction in average noise level.

Use the Front-Panel Jumpers (if your PNA has the configurable test set)

If your PNA has FOUR front-panel jumpers, you can bypass the test-port couplers and apply signals directly into the receivers. See Dynamic Range - 4 Jumpers. Using this configuration, you can achieve up to 143 dB dynamic range with **Response Calibration** using segment sweep mode.

If your PNA has MORE THAN FOUR front-panel jumpers (Configurable Test Set), you can use the front-panel jumpers to reverse a test-port coupler. See Dynamic Range - Configurable Test Set Option. Using this configuration, you can achieve up to 143 dB dynamic range with **Full 2-port Calibration** using segment sweep mode.

Note: Bypassing a port's directional coupler increases the port mismatch by approximately 15 dB (the coupling factor of the directional coupler).

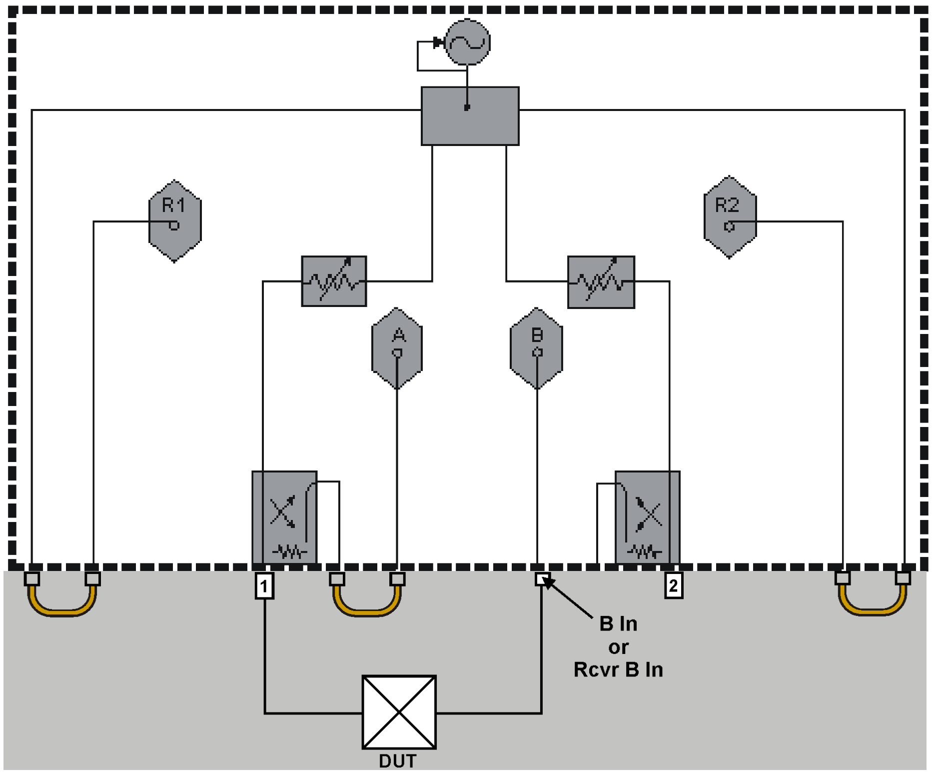
- For information about upgrading your PNA to include front-panel jumpers, see PNA Options.
- Discover the measurement possibilities using front-panel jumpers.

Improving Dynamic Range with FOUR front-panel jumpers

To improve dynamic range you can bypass the test-port coupler and apply the signal directly into the receiver. As shown in the following graphic, the signal is applied to the front-panel connector for the B In or Rcvr B In front-panel jumper rather than Port 2. Using this configuration, you can achieve up to 143 dB dynamic range with response calibration using segment sweep mode.

Explore the graphic with your mouse.

Note: Your PNA may not be equipped with front-panel jumpers or all of the components shown in this block diagram.

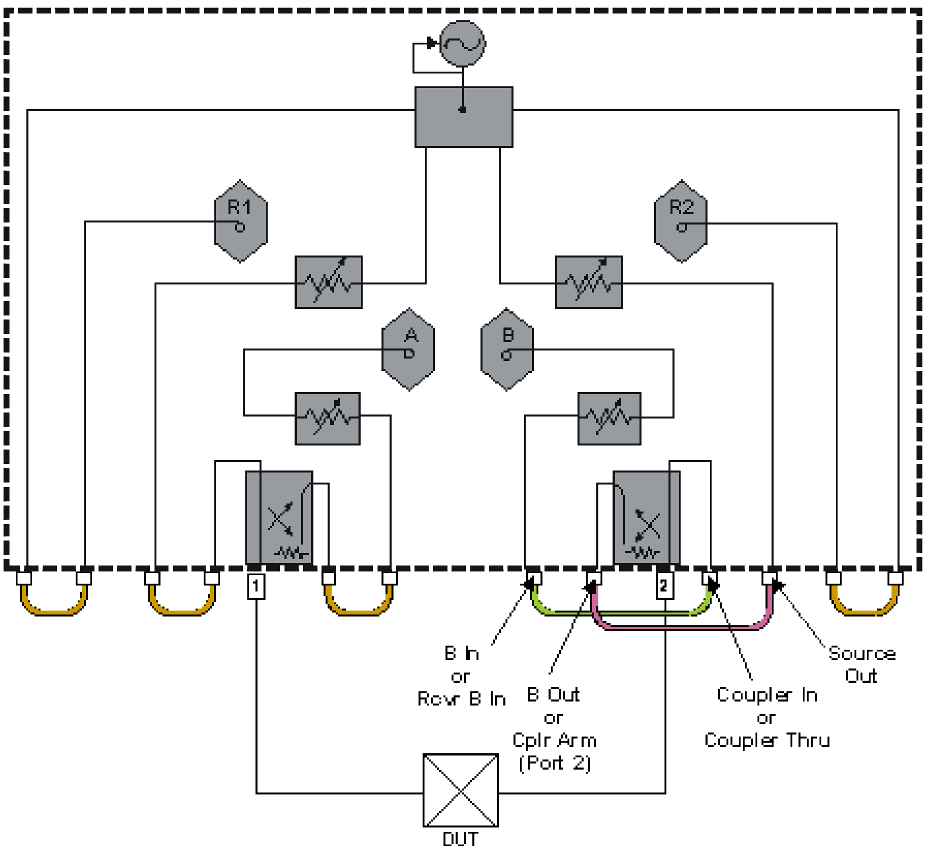


Improving Dynamic Range with Configurable Test Set Option

To improve dynamic range you can reverse the signal path in the test-port coupler and bypass the loss typically associated with the coupled arm. As shown in the following graphic, the signal is applied to Port 2. The signal bypasses the coupled arm via the jumper cable connected to the Coupler Thru (or Coupler In) and the Receiver B In (or B In) ports. Using this configuration, you can increase the forward measurement dynamic range up to 143 dB with full 2-port calibration using segment sweep mode. When making full 2-port error corrected measurements, the reverse measurement is degraded by 15 dB, with up to 113 dB of dynamic range available.

Explore the graphic with your mouse.

Note: Your analyzer's block diagram may contain different components than shown below.



Number of Points

A data point or "point" is a sample of data representing a measurement at a single stimulus value. You can specify the number of points that the analyzer measures across a sweep. (A "sweep" is a series of consecutive data point measurements, taken over a sequence of stimulus values.)

The PNA sweep time changes proportionally with the number of points. However, the overall measurement cycle time does not. See Technical Specifications for more information on how the number of points, and other settings, affect the sweep time.

Note: You may experience a significant decrease in computer processing speed with increased number of points, number of traces, and calibration error terms (full 2-port or 3-port). If this becomes a problem, you can increase the amount of RAM with PNA Option 022.

The default value is 201 points per sweep.

Other topics about Optimizing Measurements

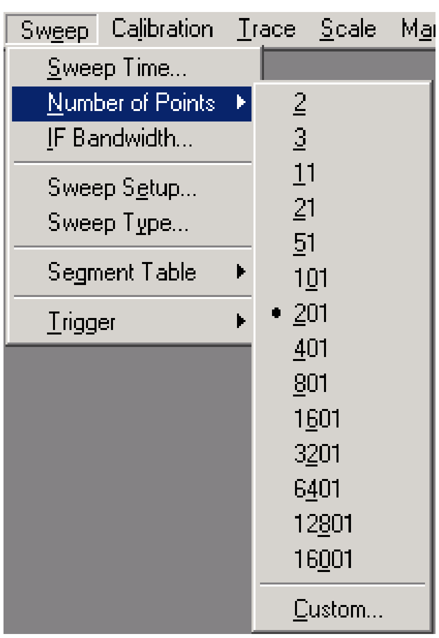
Tips:

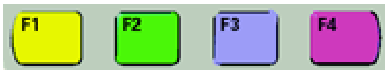
- To achieve the greatest trace resolution, use the maximum number of data points.
- For faster throughput use the smallest number of data points that will give you acceptable accuracy.
- To find an optimized number of points, look for a value where there is not a significant difference in the measurement when you increase the number of points.
- To ensure an accurate measurement calibration, perform the calibration with the same number of points that will be used for the measurement.

How to change the number of data points

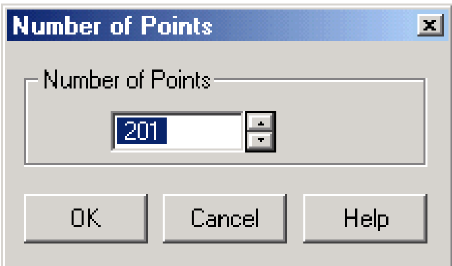
Use one of the following methods:

Select a number or click Custom to invoke a dialog box

- 

-  + 

Learn more about using the front panel interface



Number of Points dialog box help

Specifies the number of data points that the analyzer gathers during a measurement sweep.

You can specify any number from 2 to 16001.

The default value is 201.

Phase Measurement Accuracy

You can increase the accuracy of phase measurements by using the following PNA features.

- Electrical Delay
- Port Extensions
- Phase Offset
- Spacing Between Frequency Points (Aliasing)

Learn more about Phase measurements

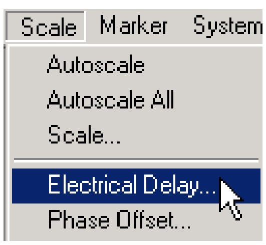
Electrical Delay

Electrical delay is a mathematical function that simulates a variable length of lossless transmission line.

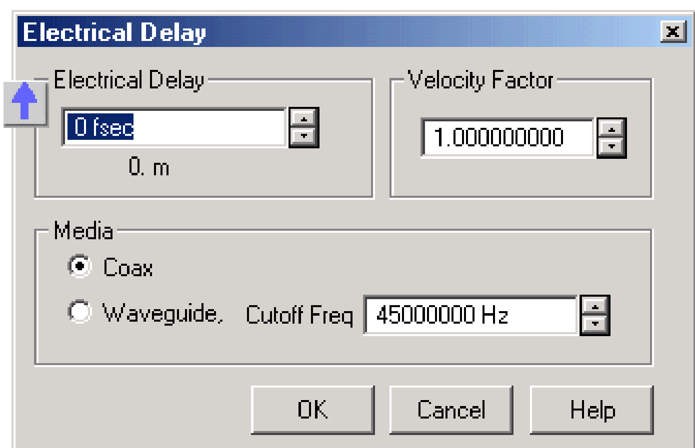
Use the electrical delay feature to compensate for the linear phase shift through a device. This feature allows you to look at only the deviation from linear phase of the device.

You can set the electrical delay independently for each measurement trace.

How to set Electrical Delay



Learn more about using the front panel interface



Electrical Delay dialog box help

Electrical Delay Specifies the value of delay added or removed, in units of time. This compensates for the linear phase shift through a device. You can set the electrical delay independently for each measurement trace.

Velocity Factor Specifies the relative velocity of the medium of the device inserted. The value for a polyethylene dielectric cable is 0.66 and 0.7 for Teflon dielectric.

Media

Coax select if the added length is coax. Also specify the velocity factor of the coax.

Waveguide Select if the added length is waveguide. Also specify the low frequency cutoff of the waveguide.

Cutoff Freq Low frequency cutoff of the waveguide.

Learn about Electrical Delay (scroll up)

Port Extensions

Use port extensions to electrically move the measurement reference plane after you have performed a calibration. This allows you to **avoid** performing another calibration. The following two scenarios show how port extensions can be useful.

1. You have already performed a calibration, and then decide that you need to add a length of cable in the measurement configuration. Use port extensions to "tell" the analyzer you have added the cable length to a specific port.
2. You are unable to perform a calibration directly at your device because it is in a test fixture. Use port extensions to compensate for the time delay (phase shift) caused by the fixture.

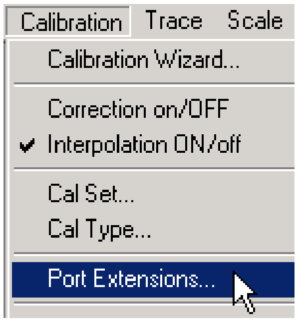
- Port extensions automatically apply electrical delay to **all** measurements associated with a particular port.
- You can add electrical length to the A and B receivers if you are using the receiver's direct access connectors on the front panel.
- Port extensions do NOT compensate for loss and mismatch errors of the cables, adapters, and fixtures; only the added electrical length.

Tip: To know when you have added enough delay:

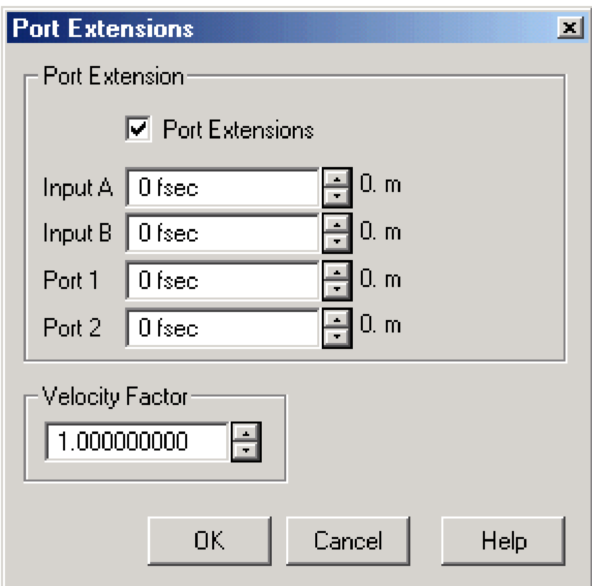
1. Connect a short in place of the device.
2. Adjust the port extension until the phase response is flat.

Note: Most short-circuit calibration standards have a non-zero delay. Therefore, adjusting delay with this method results in a delay error equal to two times the short delay. Determine the offset delay of the calibration short by examining the standard definition.

How to set Port Extensions



Learn more about using the front panel interface



Port Extensions dialog box help

Port Extensions When checked, applies the port extension function to the specified port or ports.

Input A Specifies the distance that the reference plane is extended for Input A.

Input B Specifies the distance that the reference plane is extended for Input B.

Port 1 Specifies the distance that the reference plane is extended for Port 1.

Port 2 Specifies the distance that the reference plane is extended for Port 2.

Velocity Factor Specifies the velocity factor that applies to the medium of the device that was inserted after the measurement calibration. The value for a polyethylene dielectric cable is 0.66 and 0.7 for Teflon dielectric. 1.0 corresponds to the speed of light in a vacuum.

Learn about Port Extensions (scroll up)

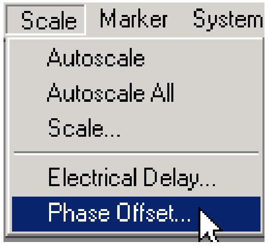
Phase Offset

Phase offset mathematically adjusts the phase measurement by a specified amount, up to 360°. Use this feature in the following ways:

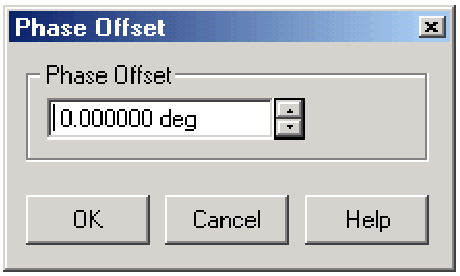
- **Improve the display of a phase measurement.** This is similar to the way you would change the reference level in an amplitude measurement. Change the phase response to center or align the response on the screen.

- **Emulate a projected phase shift in your measurement.** For example, if you know that you need to add a cable and that the length of that cable will add a certain phase shift to your measurement, you can use phase offset to add that amount and simulate the complete device measurement.

How to set Phase Offset



Learn more about using the front panel interface



Phase Offset dialog box help

Phase Offset Type a value or use the up and down arrows to select any value up to 360 degrees.

Learn about Phase Offset (scroll up)

Spacing Between Frequency Points (Aliasing)

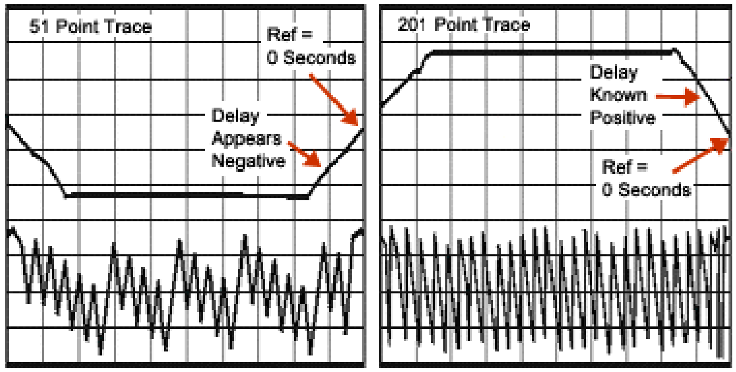
The analyzer samples data at discrete frequency points, then connects the points, creating a trace on the screen.

If the phase shift through a device is >180° between adjacent frequency points, the display can look like the phase slope is reversed. This is because the data is undersampled and aliasing is occurring.

If you are measuring group delay and the slope of the phase is reversed, then the group delay will change sign. For example, the following graphic shows a measurement of a SAW bandpass filter.

- The left measurement has 51 points and indicates the group delay is negative, which is a physical impossibility. That is, the response is below 0 seconds reference line.

- The right measurement shows an increase to 201 points which indicates the group delay is positive. That is, the response is above the 0 seconds reference line.



Tip: To check if aliasing might be occurring in a measurement, either increase the number of points or reduce the frequency span.

Electrically-Long Device Measurements

A signal coming out of a device under test may not be exactly the same frequency as the signal going in to a device at a given instant in time. This can sometimes lead to inaccurate measurement results. You can choose between two techniques to eliminate this situation and increase measurement accuracy.

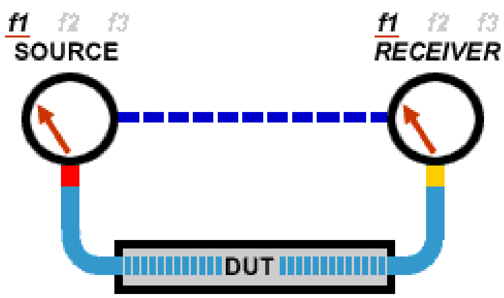
- Why Device Delay May Create Inaccurate Results
- Solutions to Increase Measurement Accuracy
  - Slow the Sweep Speed
  - Add Electrical Length to the R Channel

Other topics about Optimizing Measurements

Why Device Delay May Create Inaccurate Results

The following graphic shows an example of this situation:

- In the network analyzer, the source and receiver are phase locked together and sweep simultaneously through a span of frequencies.
- The signal flow through the Device Under Test (DUT) is shown as different colors for different frequencies.
- You can see as a stimulus frequency travels through the DUT, the analyzer tunes to a new frequency **just before** the signal arrives at the receiver. This causes inaccurate measurement results.



If the analyzer is measuring a long cable, the signal frequency at the end of the cable will lag behind the network analyzer source frequency. If the frequency shift is appreciable compared to the network analyzer's IF detection bandwidth (typically a few kHz), then the measured result will be in error by the rolloff of the IF filter.

Note: There is no fixed electrical length of a device where this becomes an issue. This is because there are many variables that lead to measurement speed. When high measurement accuracy is critical, lower the sweep speed until measurement results no longer change.

Solutions to Increase Measurement Accuracy

Choose from the following methods to compensate for the time delay of an electrically long device.

Slow the Sweep Speed

- [Add Electrical Length to the R Channel]

Slow the Sweep Speed

The following methods will slow the sweep speed.

- [Increase the Sweep Time]
- [Increase the Number of Points]
- [Use Stepped Sweep]
- [Set Dwell Time]

Add Electrical Length to the R Channel

Note: This method applies to PNA models with front panel loops.

Instead of slowing the sweep, you can compensate for the electrical length of a cable or fixture.

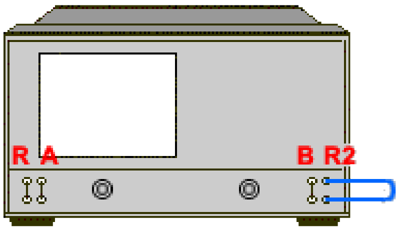
a. Remove the R-channel jumper on the front panel of the analyzer.

b. Replace the jumper with a cable of about the same length as the device under test.

   1. Add the cable on the R1 channel for S_{11} and S_{21} measurements.
   2. Add the cable on the R2 channels for S_{22} and S_{12} measurements.

c. Set the analyzer for a fast sweep.

Configuration for S_{22} and S_{12} Measurements



This method balances the delays in the reference and test paths, so that the network analyzer's ratioed transmission measurement does not have a frequency-shift error.

Note: This method works well if the delay is in a cable or fixture. For devices with long delays, this method is only suitable for uncalibrated measurements.

Reflection Accuracy on Low-Loss 2-Port Devices

To make accurate reflection measurements that have a 1-port calibration, you should terminate the unmeasured port.

- Why Terminate the Unmeasured Port
- How to Terminate the Unmeasured Port
- Resulting Measurement Uncertainty

Other topics about Optimizing Measurements

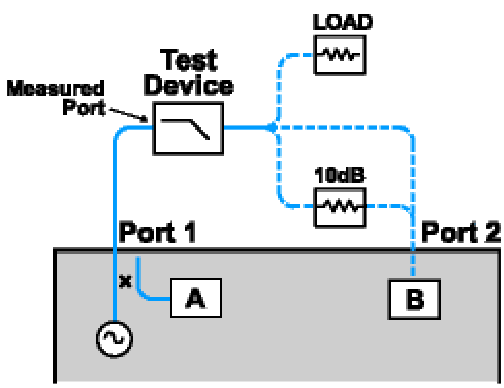
Why Terminate the Unmeasured Port

A 2-port calibration corrects for all 12 twelve error terms. A 1-port calibration corrects for directivity, source match and frequency response, but not load match. Therefore, for highest accuracy, you must make the load match error as small as possible. This especially applies for low-loss, bi-directional devices such as filter passbands and cables. You do not need to be concerned with load match when you are measuring a device with high reverse isolation, such as an amplifier.

How to Terminate the Unmeasured Port

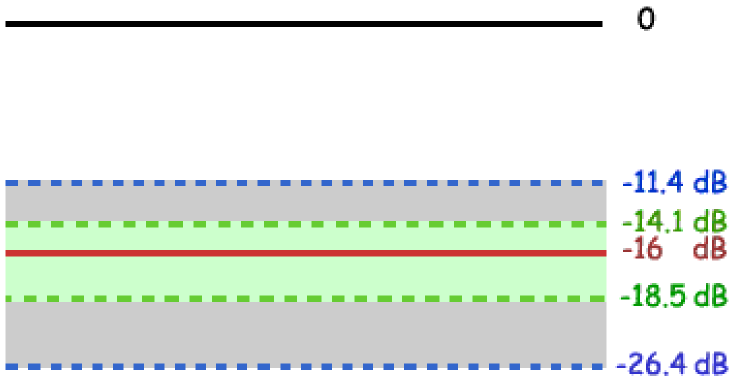
Use one of the following methods:

- Connect a high-quality termination load (from a calibration kit, for example) to the unmeasured port of your device. This technique yields measurement accuracy close to that of a Full SOLT 2-port calibration.
- Connect the unmeasured port of your device directly to the analyzer, inserting a 10 dB precision attenuator between the device output and the analyzer. This improves the effective load match of the analyzer by approximately twice the value of the attenuator, or 20 dB.



Resulting Measurement Uncertainty

The following graph illustrates the measurement uncertainty that results from terminating **with** and **without** a precision 10 dB attenuator on the output of the test device.



Legend

———	Filter Reflection
-------------	Uncertainty **with** attenuator
....	Uncertainty **without** attenuator

The calculations below show how adding a high-quality 10 dB attenuator improves the load match of the analyzer.

Note: The corresponding linear value is shown in parentheses.

Network Analyzer:

Load match (NA_{LM}) = 18 dB (.126)

Directivity (NA_{D}) = 40 db (.010)

Filter:

Insertion loss (F_{IL}) = 1dB (.891)

Return loss (F_{RL}) = 16 dB (.158)

Attenuator:

Insertion loss (A_{IL}) = 10 dB (.316)

SWR (A_{SWR}) = 1.05 (.024)
32.26 dB Return Loss

Calculations:

	Without Attenuator	With Attenuator
ρNA	$= (F_{IL})*(NA_{LM})*(F_{IL})$ = (.891)*(.126)*(.891) =.100	$= (F_{IL})*(A_{IL})*(NA_{LM})*(A_{IL})*(F_{IL})$ = (.891)*(.316)*(.126)*(.316)*(.891) = .010
ρAttenuator	NA	$= (F_{IL})*(A_{SWR})*(F_{IL})$ = (.891)*(.126)*(.891) = .019
Worst Case Error (E_{WC})	$= \rho_{NA}$ =.1	$= \rho_{NA} + \rho$Attn. =.01+.019 =.029
Uncertainty Adds	$= -20\log(F_{RL})+(E_{WC})+(NA_D)$ = -20log(.158)+(.100)+(.010) = **11.4 dB**	$= -20\log(F_{RL})+(E_{WC})+(NA_D)$ = -20log(.158)+(.029)+(.010) = **14.1 dB**
Uncertainty Subtracts	$= -20\log(F_{RL})-(E_{WC})-(NA_D)$ =-20log(.158)-(.100)-(.010) = **26.4 dB**	$= -20\log(F_{RL})-(E_{WC})-(NA_D)$ = -20log(.158)-(.029)-(.010) = **18.5 dB**

Measurement Stability

There are several situations that can cause unstable measurements. To ensure that you are making repeatable measurements, you can use various methods to create a stable measurement environment.

- **Frequency Drift**
- **Temperature Drift**
- **Inaccurate Measurement Calibrations**
- **Device Connections**

Other topics about Optimizing Measurements

Frequency Drift

The analyzer frequency accuracy is based on an internal 10 MHz frequency oscillator. See Technical Specifications for stability and aging specifications.

If your measurement application requires better frequency accuracy and stability, you can override the internal frequency standard and provide your own high-stability external frequency source through the 10 MHz Reference Input connector on the rear panel.

Temperature Drift

Thermal expansion and contraction changes the electrical characteristics of the following components:

- Devices within the analyzer
- Calibration kit standards
- Test devices
- Cables
- Adapters

To reduce the effects of temperature drift on your measurements, do the following.

- Switch on the analyzer 1/2 hour before performing a measurement calibration or making a device measurement.
- One hour before you perform a measurement calibration, open the case of the calibration kit and take the standards out of the protective foam.
- Use a temperature-controlled environment. All specifications and characteristics apply over a 25 °C ±5 °C range (unless otherwise stated).
- Ensure the temperature stability of the calibration kit devices.
- Avoid handling the calibration kit devices unnecessarily during the calibration procedure.
- Ensure the ambient temperature is ±1°C of the measurement calibration temperature.

Inaccurate Measurement Calibrations

If a measurement calibration is inaccurate, you will not measure the true response of a device under test. To ensure that your calibration is accurate, you should consider the following practices:

- Perform a measurement calibration at the points where you connect the device under test, that is, the reference plane.
- If you insert any additional accessory (cable, adapter, attenuator) to the test setup after you have performed a measurement calibration, use the port extensions function to compensate for the added electrical length and delay.
- Use calibration standards that match the definitions used in the calibration process.
- Inspect, clean, and gage connectors. See Connector Care.

See Accurate Measurement Calibrations for more detailed information.

Device Connections

Good connections are necessary for repeatable measurements. To help make good connections, do the following:

- Inspect and clean the connectors for all of the components in the measurement setup.
- Use proper connection techniques.
- Avoid moving the cables during a measurement.

Noise Reduction Techniques

Random electrical noise which shows up in the analyzer receiver chain can reduce measurement accuracy. The following PNA functions help reduce trace noise and the noise floor which can lead to better dynamic range and more accurate measurements.

Note: The trace noise in microwave PNAs becomes worse below 748 MHz and is especially obvious between 10 MHz and 45 MHz. See Reduce IFBW.

- **Sweep Average**
- **IF Bandwidth**
- **Trace Smoothing**

See Increase Dynamic Range to learn more about improving this critical parameter.

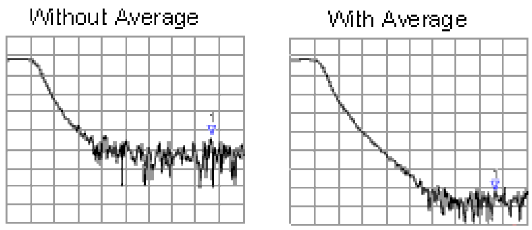
Other topics about Optimizing Measurements

Sweep Average

Sweep average is a feature that reduces the effects of random noise on a measurement. The PNA computes each data point based on the average of the same data point over several consecutive sweeps. You determine the number of consecutive sweeps by setting the Average factor. The higher the average factor, the greater the amount of noise reduction.

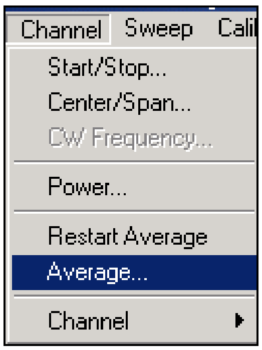
- When Averaging is ON, an **Average Counter** appears on the screen displaying the number of sweeps that has been averaged. The effect on the signal trace can be viewed as the Average Factor increases. This can assist in the selection of the optimum number of sweep averages.
- Averaging is applied to all measurements in a channel. The Average counter is displayed for each channel.
- Although you can average unratioed (single receiver) measurements, you may get unexpected results:
  - Phase results may tend toward 0. This is because phase measurements are relative by nature. Measuring absolute phase with a single receiver appears random. Averaging random positive and negative numbers will tend toward 0.
  - The noise floor does not drop when averaging unratioed measurements as on ratioed measurements.
- Use Average and IF bandwidth for the same benefit of general noise reduction. For minimizing very low noise, using Average is more effective than reducing system bandwidth. Generally, Averaging takes slightly longer than IF Bandwidth reduction to lower noise, especially if many averages are required.
- The PNA does NOT currently have a "point-averaging" feature like the **Agilent 8510 network analyzer**. That feature measures and averages each data point BEFORE moving to the next data point. Therefore, all data points are averaged in a single, slower sweep. To accomplish similar results with the PNA, try lowering the IFBW.

Effects of Sweep Average



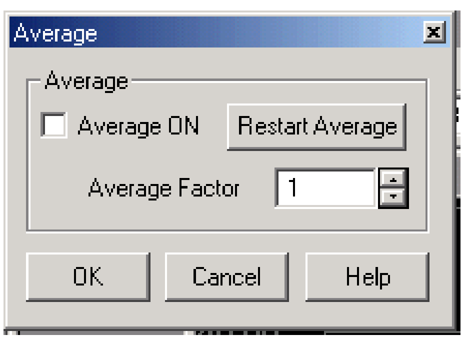
How to Set Averaging

Use one of the following methods:

- 

- 

Learn more about using the front panel interface



Average dialog box help

Average ON
Checked - Averaging is applied
Cleared - Averaging is NOT applied

Average Factor Specifies the number of sweeps that is averaged. Range of 1 to 1024.

Restart Begins a new set of measurements that are used for the average. This set of measurements is equal to the average factor.

Learn about Averaging (scroll up)

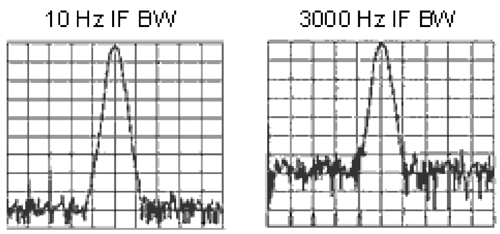
IF Bandwidth

The PNA converts the received signal from its source to a lower intermediate frequency (IF). The bandwidth of the IF bandpass filter is adjustable from 40 kHz (for most PNA models) down to a minimum of 1 Hz.

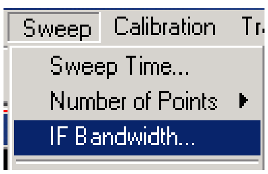
Reducing the IF receiver bandwidth reduces the effect of random noise on a measurement. Each tenfold reduction in IF bandwidth lowers the noise floor by 10 dB. However, narrower IF bandwidths cause longer sweep times.

You can set the IF bandwidth independently for each channel or each segment of segment sweep.

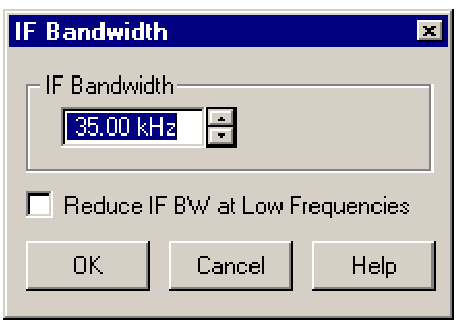
Effect of Reducing IF Bandwidth



How to set IF Bandwidth



Learn more about using the front panel interface



IF Bandwidth dialog box help

IF Bandwidth Specifies the IF (receiver) bandwidth. The value of IF bandwidth is selected by scrolling through the values available in the IF bandwidth text box. The IF BW is set independently for each channel.

The list of selectable IF Bandwidths is different depending on PNA model.

The following values are common to all models:

- **1 | 2 | 3 | 5 | 7 | 10 | 15 | 20 | 30 | 50 | 70 | 100 | 150 | 200 | 300 | 500 | 700 | 1000 | 1500 | 2000 | 3000 | 5000 | 7000 | 10000 | 15000 | 20000 | 30000**

In addition, the following values are PNA Model specific:

- N5230A: **50000 | 70000 | 100000 | 150000| 200000 | 250000**
- All other PNAs: **35000 | 40000 |**

Reduce IF BW at Low Frequencies

On PNA models with a maximum frequency of 20 GHz and higher, the trace noise becomes worse below 748 MHz. This is especially obvious between 10 MHz and 45 MHz and also when Time Domain is ON. See PNA models / maximum frequencies.

When this box is checked, the PNA uses a smaller IF Bandwidth than the selected value at frequencies below 748 MHz.

This setting:

- can be made for each channel.
- is ON (checked) by default.
- also applies to segment sweep.

Use the following calculations to determine the actual IF Bandwidth value that is used below 748 MHz.

If the result is NOT a selectable IF BW value, the next higher selectable value is used.

	10 MHz to 44.999999 MHz	**45 MHz to 748 MHz**
Less than 20 GHz models:	Not applicable	Not applicable
ALL 2-port 20 GHz PNA models*:	Actual IF BW = (selected IF BW) x **(.05)**	Actual IF BW = selected IF BW **(No reduction)**
ALL 40 GHz and higher models:	Actual IF BW = (selected IF BW) x **(.025)**	Actual IF BW = (selected IF BW) x **(.5)**

*NOT available on 4-port PNA (model N5230A Opt 240 and 245).

Example:

On a 67 GHz PNA, the selected IF BW is 30 KHz.

With **Reduce IF BW at Low Frequencies** checked, the actual IF Bandwidths used are:

- From **10 MHz to 44.999999 MHz**: 30,000Hz * **.025** = 750 Hz (PNA uses next higher selectable value: **1000 Hz**.)
- From **45 MHz to 748 MHz**: 30,000Hz * **.5** = **15 KHz**
- From **748 MHz** to stop sweep: **30 KHz**

OK Selects the value of IF bandwidth shown in the text box.

Learn about IF Bandwidth (scroll up)

Trace Smoothing

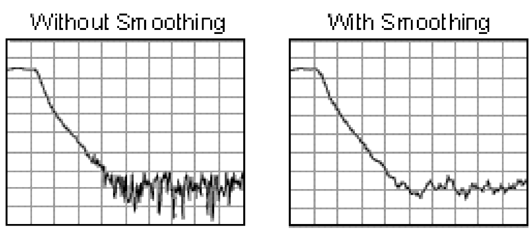
Trace smoothing averages a number of adjacent data points to smooth the displayed trace. The number of adjacent data points that get averaged together is also known as the smoothing aperture. You can specify aperture as either the number of data points or the percentage of the x-axis span.

Trace Smoothing reduces the peak-to-peak noise values on broadband measured data. It smooths trace noise and does not increase measurement time significantly.

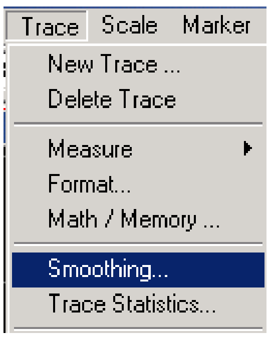
Tips:

- Start with a high number of display points and reduce until you are confident that the trace is not giving misleading results.
- Do not use smoothing for high-resonance devices, or devices with wide trace variations. It may introduce measurement errors.
- You can set the smoothing function independently for each trace.

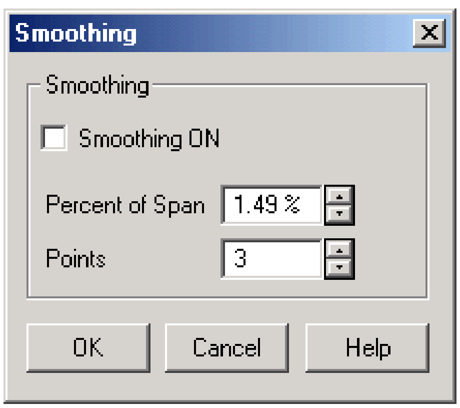
Effects of Smoothing on a Trace



How to set Trace Smoothing



Learn more about using the front panel interface



Smoothing dialog box help

Smoothing ON When checked, applies the smoothing function that averages data over a portion of the displayed trace.

Percent of Span Specifies the smoothing aperture as a percent of the swept stimulus span. For example, with a 401 points specified, the percent span is between 0.19% and 25%.

Points Specifies the smoothing aperture as a percent of the total number of measurement points in the swept stimulus span.

Learn about Trace Smoothing (scroll up)

Crosstalk

Crosstalk is energy leakage between analyzer signal paths. This can be a problem with high-loss transmission measurements. However, you can reduce the effects of crosstalk by doing the following:

- **Set the Sweep to Alternate**
- **Perform an Isolation Calibration**

Other topics about Optimizing Measurements

Set the Sweep to Alternate

Alternate sweep measures only one receiver per sweep. When one receiver is measured, the analyzer switches off the other receiver. This helps reduce receiver crosstalk.

Note: Alternate sweep mode can be set independently for each measurement channel. If multiple measurement channels are in use, you may want to set Alternate sweep for each channel.

1. In the **Sweep** menu, click **Sweep Setup.**
2. Check **Alternate Sweeps.**
3. Click **OK**.

Tips:

- The analyzer measures reflection (Receiver A) and transmission (Receiver B) on separate sweeps. Therefore, when the analyzer is making both reflection and transmission measurements, alternate sweep doubles the sweep cycle time.
- For transmission only or reflection only measurements, you can use alternate sweep to minimize the effects of crosstalk without affecting the sweep cycle time.
- The noise floor has to be lowered substantially before crosstalk is visible. You may need to use the average function or narrow the IF bandwidth.

Perform an Isolation Calibration

For transmission measurements, a response and isolation measurement calibration helps reduce crosstalk because the analyzer measures and then subtracts the leakage signal during the measurement calibration. The calibration improves isolation so that it is limited only by the noise floor.

See Isolation error to learn how crosstalk can be reduced in the calibration process.

Generally, the isolation error falls below the noise floor. So when you are performing an isolation calibration you should use a noise reduction technique such as sweep averages or reducing the IF bandwidth.

Effects of Accessories

Accessories in a configuration may affect the results of a device measurement. You can choose between two analyzer features that reduce the effects of accessories.

- Power Slope to Compensate for Cable Loss
- Gating to Selectively Remove Responses

Other topics about Optimizing Measurements

Power Slope to Compensate for Cable Loss

If you have a long cable or other accessory in a measurement configuration where a power loss occurs over frequency, apply the power slope function. This function increases the analyzer source power by a rate that you define (dB/GHz).

1. In the **Channel** menu, click **Power**.
2. If the slope function is not already switched on, click the **Slope** check box.
3. In the **dB/GHz** box, enter the rate that you want the source power to increase over the frequency sweep. Click **OK**.

Gating to Selectively Remove Responses

Gating is a feature in the time domain (option 010) that allows the analyzer to mathematically remove responses. You can set the gate for either a reflection or transmission response, but you will see different results.

- **Gating a reflection response** isolates a desired response (such as a filter's return loss), from unwanted responses (such as adapter reflections or connector mismatches).
- **Gating a transmission response** isolates a specific path in a multipath device that has long electrical lengths.

See Time Domain Gating for more information.

Fast Sweep Speed

Maintaining the fastest measurement sweep is one aspect of achieving the best measurement throughput. You can make the analyzer have the fastest sweep for your measurement application by adjusting the following:

- **Sweep Settings**
- **Noise Reduction Settings**
- **Measurement Calibration Choice**
- **Unnecessary Functions**

Other topics about Optimizing Measurements

Sweep Settings

Consider changing each of the following settings as suggested.

Frequency Span - Measure only the frequencies that are necessary for your device.

Segment Sweep - Use segments to focus test data only where you need it.

Switch Off Stepped Sweep - Use linear swept mode to minimize sweep time when possible.

Auto Sweep Time - Use this default to sweep as quickly as possible for the current settings.

Number of Points - Use the minimum number of points required for the measurement.

For more information on how number of points and other settings affect sweep cycle time, see Technical Specifications.

Noise Reduction Settings

Using a combination of these settings, you can decrease the sweep time while still achieving an acceptable measurement.

IF Bandwidth. Use the widest IF bandwidth that will produce acceptable trace noise and dynamic range.

erage. Reduce the average factor, or switch Average off.

Measurement Calibration Choice

Choose the fastest type of calibration for the required level of accuracy.

- Sweep speed is about the same for uncorrected measurements and measurements done using a response calibration, or one-port calibration.
- A full two-port calibration requires both forward and reverse sweeps to update all four S-parameters for error correction, even when only a single S-parameter is displayed.

For more information see Select a Calibration.

Unnecessary Functions

The analyzer must update information for all active functions. To achieve an additional increase in sweep speed, switch off all of the analyzer functions that are not necessary for your measurement application.

Delete Unwanted Traces

1. Click on the Trace Status bar to make an unnecessary trace active.
2. In the **Trace** menu, click **Delete Trace**.

Switch Off Unwanted Markers

1. In the **Marker** menu, click **Select Marker**, and click an unwanted marker on the list.
   - If you want to switch off all of the markers, click **All Off**.
2. Display the Marker toolbar: from the **View** menu, click **Toolbars**, and then click **Marker**.
3. Clear the **On** checkbox to switch off the marker.

Switch Off Smoothing

1. In the **Trace** menu, click **Smoothing**.
2. Click the **Smoothing On** box to clear the checkbox, switching off the smoothing function. Click **OK**.

Switch Off Limit Testing

1. In the **Trace** menu, click **Limit Test**.
2. Click the **Limit Test ON** box to clear the checkbox, switching off the limit function. Click **OK**.

Switch Off Math Functions

- In the **Trace** menu, point to the **Data Math** list and click **Data** to view only the current measurement trace.

Analyzer sweep speed is dependent on various measurement settings. Experiment with the settings to get the fastest sweep and the measurement results that you need.

Switch Between Multiple Measurements

If you need to make multiple measurements to characterize a device, you can use various methods to increase throughput. Experiment with these methods to find what is best for your measurement application needs.

- **Set Up Measurements for Increased Throughput**
  - **Arrange Measurements in Sets**
  - **Use Segment Sweep**
  - **Trigger Measurements Selectively**
- **Automate Changes Between Measurements**
- **Recall Measurements Quickly**

Other topics about Optimizing Measurements

Set Up Measurements for Increased Throughput

To achieve optimum throughput of devices that require multiple measurements, it is helpful to know the operation of the analyzer. This knowledge allows you to set up the measurement scenarios that are best for your applications.

Learn more about Traces, Channels, and Windows on the PNA

Arrange Measurements in Sets

If you arrange measurements to keep the complete set of device measurements in one instrument state, you can save them so that you can later recall a number of measurements with one recall function.

See Pre-configured Measurement Setups for more information.

Use Segment Sweep

Segment sweep is helpful if you need to change the following settings to characterize a device under test.

- Frequency Range
- Power Level
- IF Bandwidth
- Number of Points

The segment sweep allows you to define a set of frequency ranges that have independent attributes. This allows you to use one measurement sweep to measure a device that has varying characteristics.

See Segment Sweep for more information.

Trigger Measurements Selectively

You can use the measurement trigger to make measurements as follows:

- Continuously update only the measurements that have rapidly changing data.
- Occasionally update measurements that have infrequently changing data.

For example, if you had four channels set up as follows:

- Two channels measuring the data that is used to tune a filter
- Two channels measuring the data for the out-of-band responses of the filter

You would want to constantly monitor only the measurement data that you use for tuning the filter. If you continuously update all of the channels, this could slow the response of the analyzer so that you would not be able to tune the filter as effectively.

Note: You must either trigger the infrequent measurement manually or with remote interface commands.

To trigger measurements selectively:

This procedure shows you how to set up two different measurements with the following behavior:

- Channel 1 measurement will continuously update the data.
- Channel 2 measurement will occasionally update the data.

1. In the **Windows** menu, click **Meas Setups**, **Setup D**.

Set Up a Measurement Trigger for Continuous Updates

2. In the **Sweep** menu, click **Trigger**, **Trigger...**.
3. Under **Trigger Source**, click **Internal**.
4. Under **Channel Trigger State**, select **Channel 1**, and click **Continuous**.

Set Up a Measurement Trigger for Occasional Updates

5. Under **Channel Trigger State**, select **Channel 2**, and click **Single**, **OK**.
    - If you want the analyzer to trigger more than a single sweep, click the **Enable Groups** check box and enter the number of sweeps.
6. In the **System** menu, click **Keys, Trigger**.

Update the Measurement

7. Click on the lower window to make Channel 2 the active channel.
8. On the active entry toolbar, click the type of trigger you set up.
    - Click **Single** if you set up the analyzer for a single sweep per trigger.
    - Click **Groups** if you set up the multiple sweeps per trigger.

Note: A trace must be active for you to initiate a trigger for that measurement.

Automate Changes Between Measurements

If there are slight differences between the various measurements that you need to characterize a device, you may find that it is faster to change the measurement settings using programming. .

Recall Measurements Quickly

The most efficient way to recall measurements is to recall them as a set of measurements (instrument state).

- It only takes a short time longer to recall an instrument state that includes multiple measurements, than it does to recall an instrument state with only one measurement.
- Each recall function has time associated with it. You can eliminate that time by setting up the measurements as a set so you can recall them as a set.

See Save and Recall Files for more information.

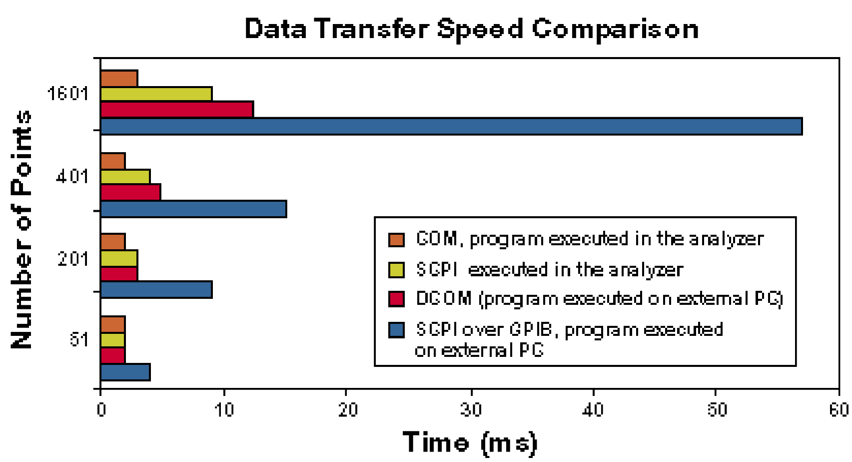
Data Transfer Speed

The fastest data transfer helps you achieve the best measurement throughput. Try these methods for improving data transfer speed.

- **Use single sweep mode** to ensure that a measurement is complete before starting a data transfer.

1. In the **Sweep** menu, click **Trigger**, **Trigger...**.
2. Under **Trigger Source**, click **Manual**.
3. Under **Channel Trigger State**, select the channel for the measurement that you want to transfer, and click **Continuous**, **OK**.
4. Update the Measurement
    1. In the **System** menu, click **Keys, Trigger**
    2. In the active entry toolbar, click **Single**

- **Transfer the minimum amount of data** needed. For example, a trace with a few points, using segment sweep rather than a full trace with many linearly spaced points. Also, use markers instead of trace transfers.
- **Choose the REAL data format** to provide the fastest transfer speed when using SCPI programs for automated applications.
- **Use SCPI over LAN** for applications that are automated with SCPI programs.
- **Use COM programs** to provide the fastest transfer speed when using an automated application. See Data Transfer Time.



Other topics about Optimizing Measurements

Using Macros

Macros are executable programs that you write, load into the analyzer, and then run from the analyzer. You can have up to 12 macros set up to run on the analyzer.

[How to Setup Macros]

[How to Run Macros]

[Macro Example]

How to Setup Macros

Use one of the following methods to access the Macro Setup dialog box:

1. 

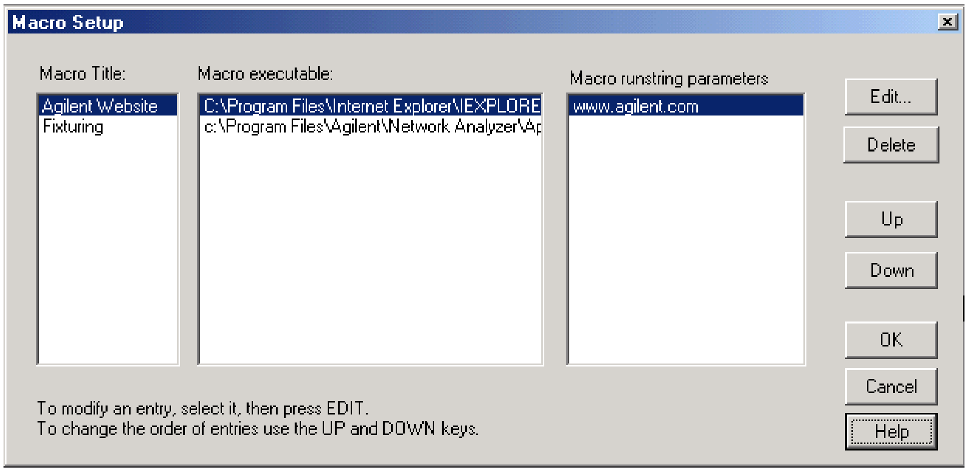
2. 

+ 

In the Macro Setup dialog box:

1. Click on a blank line below the last entry. (There may be NO entry.)
2. Click **Edit**
3. In the **Macro Title** box**,** type a descriptive title for your macro.
4. Click **Browse.**
5. Change **Files of Type**
6. Find and click your macro file
7. Click **OK**
8. Click **OK** on the Macro Setup dialog box.

Learn more about using the front panel interface



Macro Setup dialog box help

Allows you to create a set of 12 macros so that you can launch other programs from within the PNA application.

To add a Macro, select a blank line then click **Edit**

Macro Title Shows the titles that appear in the active entry toolbar when you press the Macro key. These titles are associated with the executable files and should be descriptive so you can easily identify them. For example, if you wanted to launch the Agilent Home Page, you could title the executable "Agilent Home."

Macro Executable Lists the complete path to the executable file. To follow the example of launching the Agilent PNA Series Home Page, the path to the executable could be "C:\Program Files\Internet Explorer\iexplore.exe.

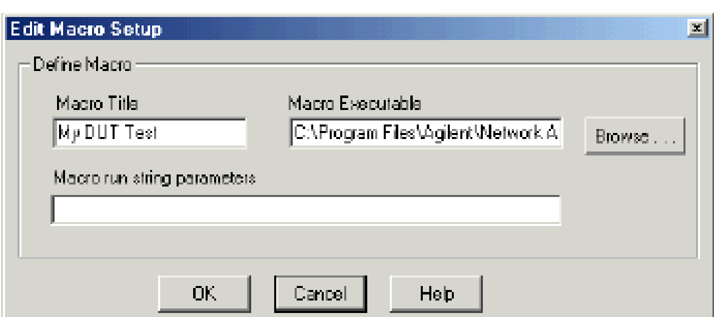
Macro Runstring Parameters Lists the parameters that get passed to the program that is referenced in the executable file. Again following the example of launching the PNA Series Home Page, you could assign the runstring parameters "http://www.agilent.com/find/pna".

Edit Invokes the Macro Edit dialog box.

Delete Deletes the selected macro.

Up Allows you to reorder the macros, moving the selected macro up one line. For the 12 possible macros there are 12 lines, indicating the order that they appear in the active entry toolbar when you press the Macro key. Since there are four titles that can be shown at one time in the toolbar, when you repeatedly press the Macro key, the toolbar changes the macro titles to the next set of four macro titles.

Down Moves the selection down one line in the list of macros.



Macro Edit dialog box help

Macro Title Allows you to modify the title that appears in the active entry toolbar.

Macro Executable Allows you to modify the complete path to the macro executable file.

Browse Allows you to look through drives and directories, to locate the macro executable file and establish the complete path to the file.

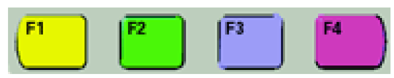
Macro run string parameters Allows you to modify the parameters that are passed to the program referenced in the executable file.

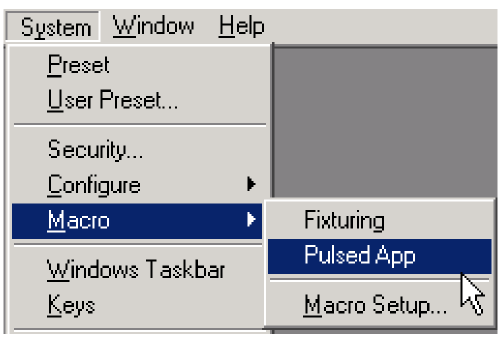
See Macro Setup dialog box

How to Run Macros

Use one of the following methods to run a Macro:

1. 

Press until your macro is visible, then 

2. 

Select your Macro

Learn more about using the front panel interface

Macro Example

The following is an example Visual Basic Scripting (vbs) program that you can copy, install, and run on your PNA

Note: Print these instructions if viewing in the analyzer. This topic will be covered by the Macro Setup dialog box.

1. Copy the following code into a Notepad file.
2. Save the file on the analyzer hard drive in the **C:\Documents** folder. Name the file **FilterTest.vbs**
3. Close Notepad
4. Setup the macro in the PNA
5. Run the macro

```
'Start copying here
'This program creates a S21 measurement, with Bandwidth
'markers for testing a 175MHz Bandpass filter
'It is written in VBscript using COM commands

Set PNA = CreateObject("AgilentPNA835x.Application")
PNA.Preset
Set chan=PNA.activechannel
Set meas=PNA.activemeasurement
Set limts = meas.LimitTest
Set trce = PNA.ActiveNAWindow.ActiveTrace

meas.ChangeParameter "S21",1
chan.StartFrequency = 45e6
chan.StopFrequency = 500e6
trce.ReferencePosition = 8
PNA.TriggerSignal = 3

'Do Test
for t=1 to 5
call measure
call compare
next
msgbox("Done Testing")

sub measure
msgbox("Connect Device " & t & " and press OK")
PNA.ManualTrigger True
meas.SearchFilterBandwidth
end sub

sub compare
BW = meas.FilterBW
if bw>6.5e7 then msgbox("Failed BW: " & BW)
Loss = meas.FilterLoss
if loss>5 then msgbox("Failed Loss: " & Loss)
end sub
'End copying here
```

Calibration Overview

The following is discussed in this topic:

What Is Measurement Calibration?

Why Is Calibration Necessary?

Conditions Where Calibration Is Suggested

What Is ECal?

See other Calibration Topics

What Is Measurement Calibration?

Calibration removes one or more of the systematic errors using an equation called an error model. Measurement of high quality standards (for example, a short, open, load, and thru) allows the analyzer to solve for the error terms in the error model. See Measurement Errors.

You can choose from different calibration types, depending on the measurement you are making and the level of accuracy you need for the measurement. See Select a Calibration Type.

The accuracy of the calibrated measurements is dependent on the quality of the standards in the calibration kit and how accurately the standards are modeled (defined) in the calibration kit definition file. The calibration-kit definition file is stored in the analyzer. In order to make accurate measurements, the calibration kit definition must match the actual calibration kit used. To learn more, see Accurate Calibrations.

The Calibration Wizard provides the different calibration methods used in the PNA. See Calibration Wizard.

There are quick checks you can do to ensure your measurement calibration is accurate. To learn more see Validity of a Measurement Calibration

If you make your own custom-built calibration standards (for example, during in-fixture measurements), then you must characterize the calibration standards and enter the definitions into a user modified calibration-kit file. For more information on modifying calibration kit files, see Calibration Standards.

Note: Instrument Calibration is ensuring the analyzer hardware is performing as specified. This is not the same as measurement calibration.

Why Is Calibration Necessary?

It is impossible to make perfect hardware that would not need any form of error correction.

In addition, the accuracy of network analysis is greatly influenced by factors external to the network analyzer. Components of the measurement setup, such as interconnecting cables and adapters, introduce variations in magnitude and phase that can mask the actual response of the device under test.

The best balance is to make the hardware as good as practically possible, balancing performance and cost. Calibration is then a very useful tool to improve measurement accuracy.

Conditions Where Calibration Is Suggested

Generally, you should calibrate for making a measurement under the following circumstances:

- You want the best accuracy possible.
- You are adapting to a different connector type or impedance.
- You are connecting a cable between the test device and an analyzer test port.
- You are measuring across a wide frequency span or an electrically long device.
- You are connecting an attenuator or other such device on the input or output of the test device.

If your test setup meets any of the conditions above, the following system characteristics may be affected:

- Amplitude at device input
- Frequency response accuracy
- Directivity
- Crosstalk (isolation)
- Source match
- Load match

What Is ECAL

ECal is a complete solid-state calibration solution. It makes one port (Reflection), full two and three-port calibrations fast and easy. See Using ECal.

- It is less prone to operator error.
- The various standards (located inside the calibration module) never wear out because they are switched with PIN-diode or FET switches.
- The calibration modules are characterized using a TRL-calibrated network analyzer.
- ECal is not as accurate as a good TRL calibration.

For information about ordering ECal modules, see Analyzer Accessories or contact your Agilent Support Representative

Select a Calibration Type

This section provides information about the calibration types available in the PNA.

Calibration Types	
OPEN Response	1-PORT Reflection
SHORT Response	Full SOLT 2-port
THRU Response	Full TRL 2-port
THRU Resp + Isol	Full SOLT 3-port

See other Calibration Topics

Type: **Open and Short Response**

Calibration Method: Unguided Calibration

General Accuracy: Low to Medium

Parameters: S11 or S22 (S33 on a 3-port PNA)

Standards Required: OPEN or SHORT

Systematic Errors Corrected:
- Frequency response reflection tracking

Application:
- Reflection measurements on any one port.

Type: **Thru Response**

Calibration Method: Unguided Calibration

General Accuracy: Medium

Parameters: S12 or S21 (S23, S32, S31, S13 on a 3-port PNA)

Standards Required: THRU

Systematic Errors Corrected:
- Frequency response transmission tracking

Application:
- Transmission measurements in any one direction.

Note: The THRU definition for a predefined calibration kit assumes zero-length and zero-loss. If you use an adapter as a THRU, characterize the adapter in the calibration kit file for better accuracy. See Accurate Calibrations

Type: **Thru Response and Isolation**

Calibration Method: Unguided Calibration

General Accuracy: Medium

Parameters: S11 or S22 (S23, S32, S31, S13 on a 3-port PNA)

Standards Required: THRU, several identical LOADS (one for each test port of the PNA)

Systematic Errors Corrected:

- Frequency response transmission tracking
- Crosstalk

Application:

- Use only for transmission measurements in any one direction.
- Isolation portion of calibration improves high dynamic range measurements.

Note: If you do not have several identical loads (one for each test port of the PNA), you cannot perform this calibration.

Type: **1-Port (Reflection)**

Calibration Method: Unguided Calibration, Guided Calibration, ECal

General Accuracy: High

Parameters: S11 or S22 (S33 on a 3-port PNA)

Standards Required: (SHORT, OPEN, LOAD) or ECal module

Systematic Errors Corrected:

- Directivity
- Source match
- Frequency response reflection tracking

Application:

- Reflection measurements on any one port.

PNA Models: All

Type: **Full SOLT 2-Port**

General Accuracy: High

Calibration Method: Unguided Calibration, Guided Calibration, ECal

Parameters: All

Standards Required: (SHORT, OPEN, LOAD, THRU) or ECal module

Systematic Errors Corrected:

- Directivity
- Source match
- Isolation
- Load match
- Frequency response transmission tracking
- Frequency response reflection tracking

Application:

- Use for all S-parameter measurements.
- 12-term error-correction

Type: **Full TRL 2-Port**

Only on PNA Models E8356A, E8357A, E8358A, E8362A, E8363A, E8364A

Calibration Method: Unguided Calibration

General Accuracy: Very High

Parameters: All

Standards Required: THRU, REFLECT, LINE

Systematic Errors Corrected:

- Directivity
- Source match
- Isolation
- Load match
- Frequency response transmission tracking
- Frequency response reflection tracking

Application:

- Use for all S-parameter measurements.
- 12-term error-correction
- Highest accuracy of all calibrations

For more information, see TRL Cal

To perform a TRL calibration with a connector type other than 3.5 mm or 7 mm, you must first modify or create TRL standards and definitions. For more information on modifying standards, see Calibration Standards.

Type: **Full SOLT 3-Port**

Only on PNA Models N3381A, N3382A, N3383A

General Accuracy: High
Calibration Method: Guided, ECal
Parameters: All
Standards Required: (SHORT, OPEN, LOAD, THRU) or ECal module
Systematic Errors Corrected: • Directivity • Source match • Isolation • Load match • Frequency response transmission tracking • Frequency response reflection tracking
Application: • Use for all S-parameter measurements. • 27-term error-correction

Calibration Wizard

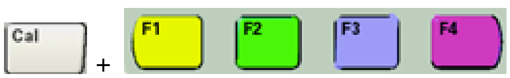
The Calibration Wizard allows you to choose a Calibration method and then perform the calibration.

- **How to Start Calibration Wizard**
- **Guided Calibration: Mechanical Standards**
- **Unguided Calibration**
- **Saving a Calibration**
- **Calibration Preferences**

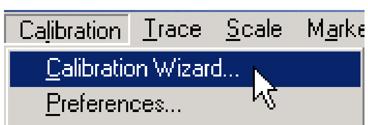
Other Cal Topics...

How to start Calibration Wizard

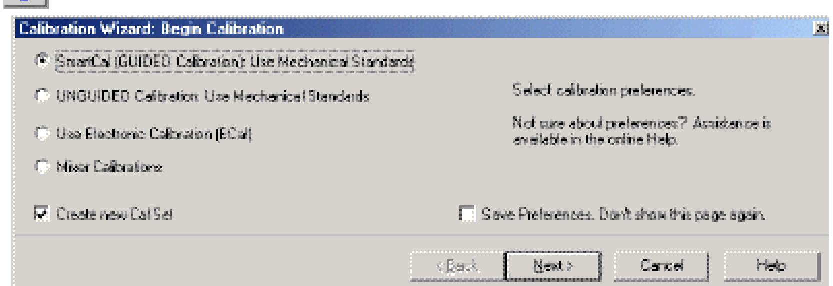
Use one of the following methods:

- 

- 

- 

Learn more about using the front panel interface



Calibration Wizard dialog box help

Allows you to select the calibration method:

SmartCal (Guided Calibration Use Mechanical Standards)

This method provides a step-by-step "wizard" interface. You describe the connectors on your DUT and the cal kits you will use; it walks you through the most accurate calibration possible.

- Supports all cals EXCEPT simple response cals (open, short, thru).

Learn more about SmartCal

See Also TRL Calibration

Unguided Calibration Use Mechanical Standards

This method provides a familiar calibration interface. You choose the type of cal to perform; it allows you the flexibility to measure the standards in any order.

- Supports all cals EXCEPT full 3-port or 4-port cal type.

Learn more about Unguided Calibration

See Also TRL Calibration

Use Electronic Calibration

This method provides fast, software controlled, 1-port, 2-port, and 3-port calibrations.

See Using ECal

Mixer Calibration

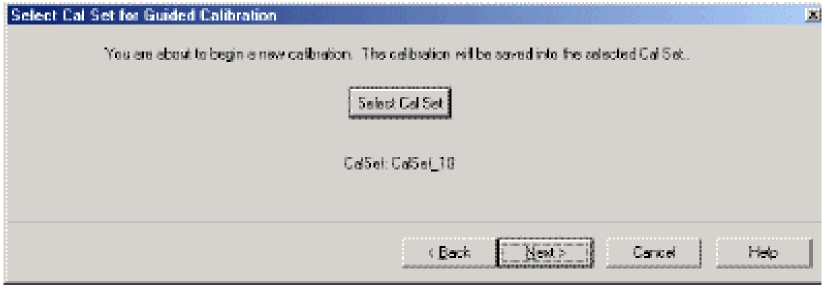
This method is only available if the PNA has the Frequency Converter Application (option 083).

Create new Cal Set

Check to create a new Cal Set for the calibration to be performed.
Clear to overwrite an existing Cal Set with the new calibration data. When cleared, click **Next** to invoke the Select Cal Set dialog box.

Save Preferences

When checked, saves your calibration method choice. You will not see this page again until you need to change the calibration method. To change, on the **Calibration** menu, click **Preferences**.
When cleared, you will continue to see this page on subsequent calibrations.

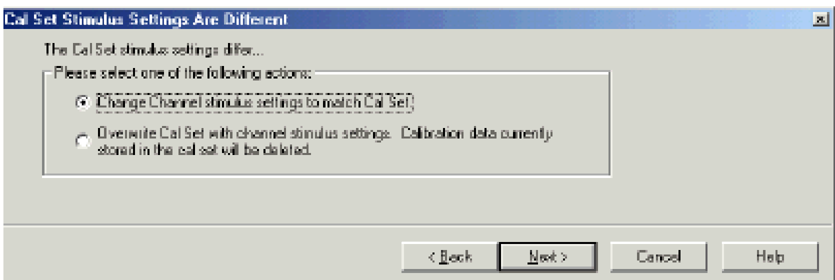


Select Cal Set dialog box help

Allows you to select a stored Cal Set to overwrite.

Select Cal Set Click to select a Cal Set to overwrite. Invokes the **Select A Cal Set** dialog box.

Note: If you do NOT click **Select Cal Set**, the current active Cal Set (which is displayed below the button) will be overwritten. If there is no active Cal Set, a new Cal Set will be created and displayed.



Cal Set Stimulus Settings Are Different dialog box help

The cal set that you selected has stimulus settings that differ from the active channel stimulus settings.

Change channel stimulus settings to match Cal Set This does NOT apply the Cal Set data, only change the stimulus settings. You will finish and apply the calibration that you are performing.

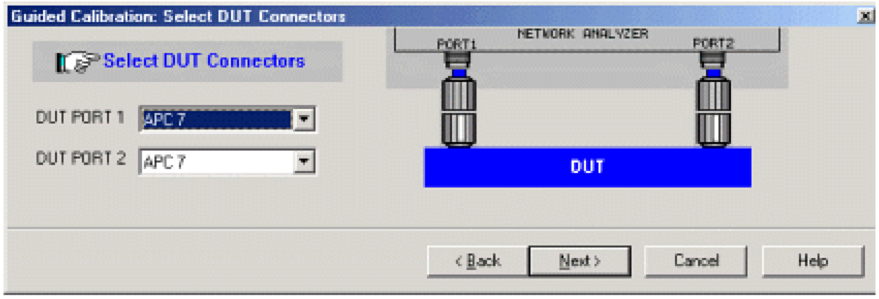
Overwrite cal set with channel stimulus settings. Calibration data currently stored in the cal set will be deleted. Overwrites the stimulus settings of the cal set with the channel stimulus settings. Deletes the calibration data currently stored in the cal set.

Guided Calibration   Mechanical Standards

Depending on the DUT connector types you specify, the "wizard" automatically determines the calibration type and suggests a calibration kit.

The PNA displays the following dialog boxes when performing a Guided calibration:

- **Select DUT Connectors**
- **Select Cal Kits**
- **Preview/Modify Settings**
- **Guided Calibration Steps**



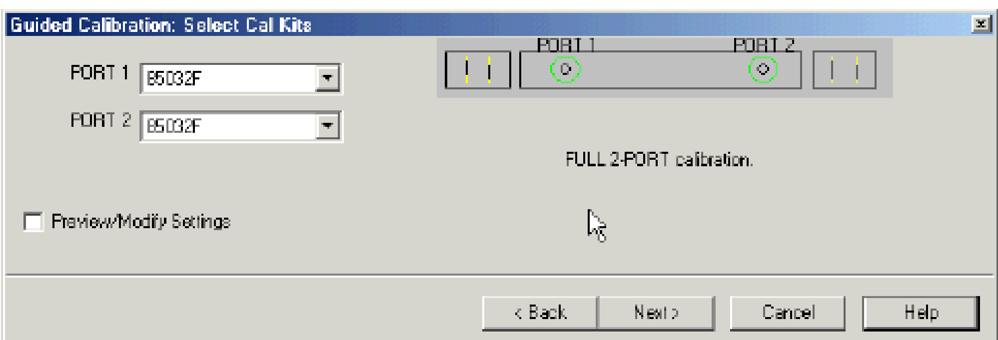
Select DUT Connectors dialog box help

Allows you to select the DUT connector types.

Note: To perform a Full 1-Port cal, select **'Not Used'** for the unused DUT connector.

If your DUT connectors are:

- **Waveguide** Change the system impedance to 1 ohm before performing a calibration. See Setting System Impedance.
- **Not listed** (male and female) Select **Type A** as the connector type. Type A requires a calibration kit file containing the electrical properties of the standards used for calibration (see Calibration kits).
- **Unspecified** (like a packaged device) Select **Type B** as the connector type. Type B requires a calibration kit file containing the electrical properties of the standards used for calibration (see Calibration kits).
- **Non-insertable** Guided Cal will perform an Adapter Removal calibration. This requires that you have an adapter that has the same connectors as the DUT. To override this behavior, check Preview/Modify Settings in the following dialog box.



Select Cal Kits dialog box help

Allows you to select the calibration kit used with each measurement port. The list for each PORT displays kits having the same connector type as the DUT.

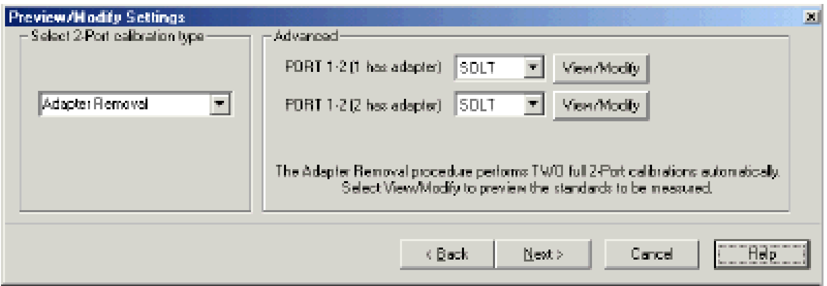
PORT 1 Lists the calibration kits available with port 1

PORT 2 Lists the calibration kits available with port 2

PORT 3 (on a three-port PNA) Lists the calibration kits available with port 3

Preview/Modify Settings Invokes a dialog box that shows the calibrations to be performed based on your selections in the previous dialog boxes.

Learn how to perform a TRL calibration.



Preview/Modify Settings dialog box help

Note: This dialog box looks slightly different on a 3 port PNA but functionally works the same.

Select calibration type

Shows the calibrations to be performed based on your selections in the previous dialog boxes. Click to select a different calibration type. Depending on your situation, there may be NO choices available.

The potential choices include:

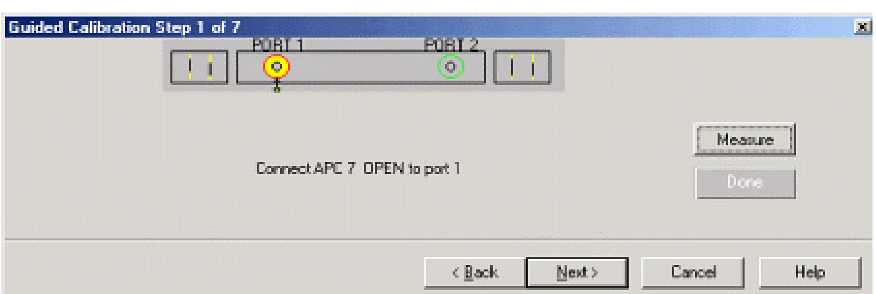
- Adapter Removal
- Unknown Thru
- Flush Thru - (use zero-length thru)
- Cal Kit Thru (Defined Thru)

Advanced

Shows calibrations to be performed on individual ports. Click to select a different calibration type. Depending on your situation, there may be NO choices available. The potential choices include:

- SOLT
- TRL Learn how to perform a TRL calibration.

View/Modify Invokes the Modify Calibration Class Assignments dialog box. Use this to view or change the standards associated with a calibration measurement.

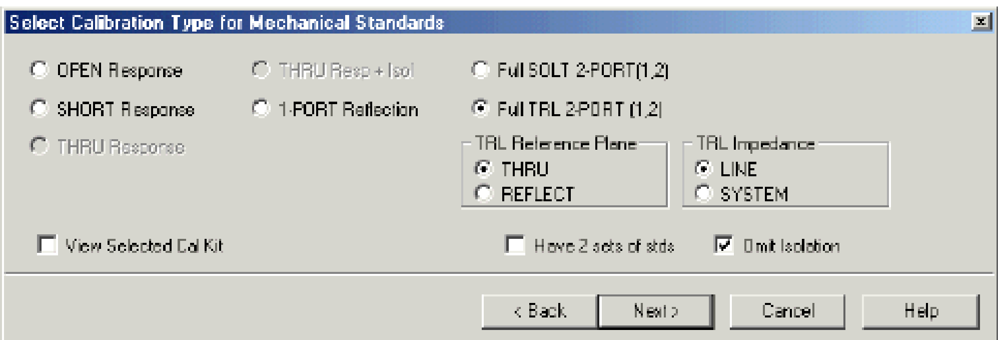


Guided Calibration Steps dialog box help
Prompts for standards to be measured. **Measure** Click to measure the standard. **Done** Click **after** a standard is re-measured and all measurements for the calibration are complete. **Next** Click to continue to the next calibration step. Does **NOT** measure the standard.

Unguided Calibration

Unguided calibration offers all calibration types except Full 3-Port and ECal. The PNA displays the following dialog boxes when performing an Unguided calibration:

- **Select Calibration Type for Mechanical Standards**
- **Select Cal Kit**
- **Measure Mechanical Standards**
- **Multiple Standards**
- **Sliding Load Measurement**



Select Calibration Type for Mechanical Standards dialog box help

Allows you to select the calibration type and settings. The Calibration types that are available depend on the parameter that is selected for the active channel and the PNA model. For example, if an S11 measurement is selected, then Thru Response cal type is not available.

OPEN Response (reflection)

SHORT Response (reflection)

THRU Response (transmission)

THRU Resp + Isol (transmission)

Full SOLT 2-PORT (1,2) The numbers in parenthesis represent the port pair used in the calibration.

Full TRL 2-PORT (1,2) TRL calibration is available with E835xA and E836xA series network analyzers.

Full SOLT 3-PORT 3-port PNAs only

TRL Reference Plane (when TRL is selected)

THRU – The THRU standard is used to establish the measurement reference plane. Select if the THRU standard is zero-length or very short.

REFLECT – The REFLECT standard is used to establish the position of the measurement reference plane. Select if the THRU standard is not appropriate AND the delay of the REFLECT standard is well defined.

TRL Impedance (when TRL is selected)

LINE – The impedance of the line standard is used as the reference impedance, or center of the Smith Chart. Any reflection from the line standard is assumed to be part of the directivity error.

SYSTEM – The system impedance is used as the reference impedance. Choose when the desired test port impedance differs from the impedance of the LINE standard.

View Selected Cal Kit Check to view the selected cal kit or select a different cal kit. Invokes the Select Cal Kit dialog box.

Have 2 sets of standards Select this checkbox to use two sets of calibration standards, reducing the number of measurements. You may also need to modify the Calibration Class Assignment Order.

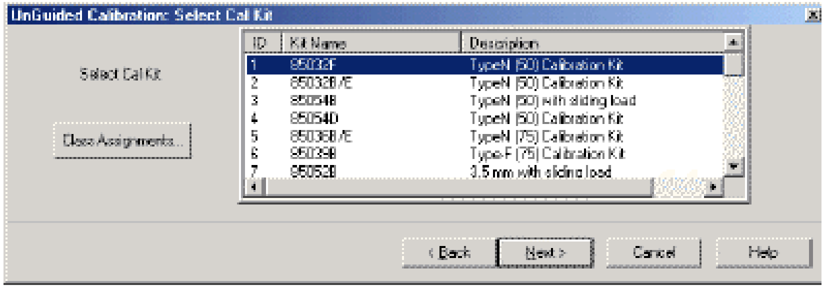
Clear to use one set of standards.

Omit Isolation Check to NOT correct for crosstalk (isolation). Use when your measurement does not require high dynamic range. Clear to calibrate and correct for isolation . Use when your measurement requires maximum dynamic range (> 90 dB). See also Isolation Portion of 2-Port Calibration.

Next Click to continue to Measure Mechanical Standards dialog box.

Notes:

- If the DUT connector type has an impedance other than 50 ohms (waveguide = 1 ohm), change the system impedance before performing a calibration. See Setting System Impedance.
- The Full SOLT 3-PORT calibration-type links to Guided calibration (on a three-port PNA).



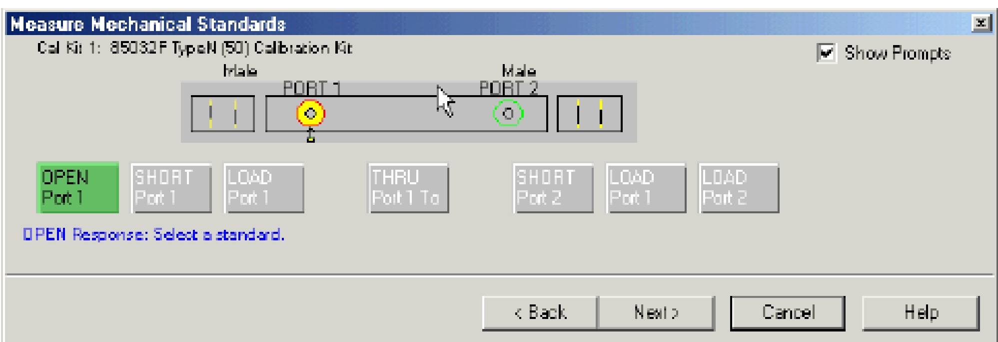
Select Cal Kit dialog box help

Displays the calibration kit files available for Unguided calibration.

Select the desired calibration kit file and click **Next**

Class Assignments Allows modification of the selected Cal Kit class assignments.

To learn to substitute other calibration kits, see Advanced Modify Cal Kits

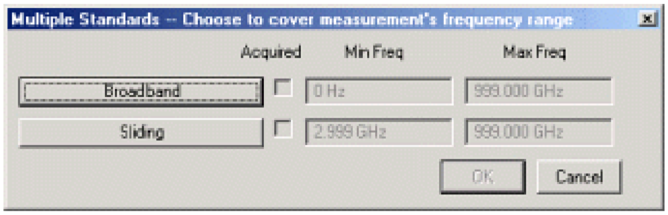


Measure Mechanical Standards dialog box help

Displays the calibration kit file and standards required for the calibration.

- Standards may be connected and measured in any order.
- Connect the standard to the measurement port and click its associated green button. A check mark indicates the standard has been measured.
- If a standard type contains multiple standards, a dialog box opens to display the multiple standards included in the calibration kit file.
- If a sliding load is included in the calibration kit file, a dialog box opens to perform the measurement with the standard.
- The THRU Resp + Isol calibration requires several identical loads (one for each test port of the PNA).

Show Prompts Check to provide a reminder for the required connection when you click on the standard.



Multiple Standards dialog box help

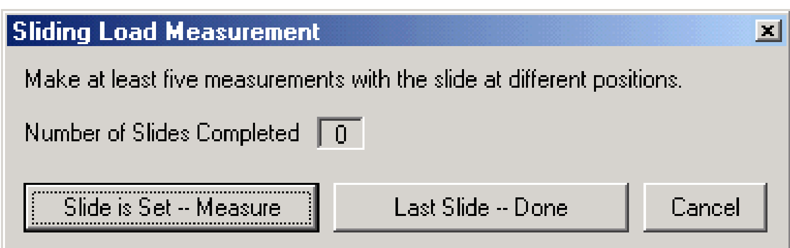
Select the standards to be measured.

Note: You may see both male and female standards. The Unguided cal has no knowledge of the gender of your connector types. **Choose the gender of your DUT connector;** NOT the test port. Then click OK.

To modify this calibration class to show only one standard, on the Calibration menu, click **Advanced Modify Cal Kits**. Select the Cal kit and click **Edit Kit.** In **Class Assignment**, click **Edit**. Learn more about Modify Calibration Class Assignments.

- Connect the standard to the measurement port and click its associated button. A check mark in the **Acquired** box indicates the standard has been measured.
- To cover the entire frequency range, you may need to measure more than one standard. The order in which the standards are measured is important. The last standard that is measured will override the others in respect to the frequency range of the standard definition. **Example:** In the case of measuring both a broadband load and a sliding load, you would measure the sliding load last. This is because the frequency range of the sliding load is a subset of the broadband load.

Learn more about Modify Calibration Class Assignments



Sliding Load Measurement dialog box help

Allows you to measure the sliding load standard.

To Measure a Sliding Load:

1. Connect the sliding load to the measurement port.
2. Position the sliding element, then click **Slide is Set -- Measure**. Do not move the sliding element until measurement is complete.
3. Measure the sliding load for at least **five** and up to **seven** positions for best accuracy.

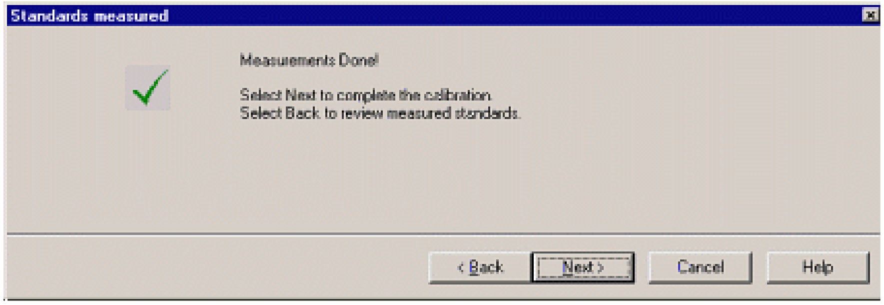
Note: The positions of the sliding element should be unequally spaced to reduce the possibility of overlapping data points.

5. Click **Last Slide -- Done** for final measurement.
6. Remove sliding load from the measurement port.
7. Measure the remaining standards.

Saving a Calibration

The PNA displays the following dialog boxes at the completion of a calibration. You can save the Cal Set data only or the Cal Set data and all instrument settings for the active channel.

- **Standards Measured**
- **Calibration Completed**
- **Save Instrument State**



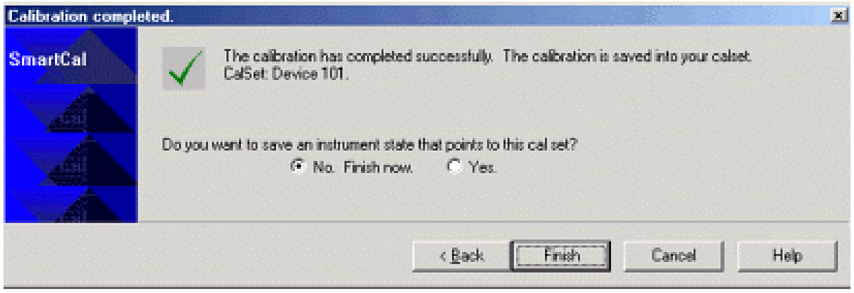
Standards Measured dialog box help

All standards required for Guided calibration have been measured.

Back Click to re-measure a calibration standard.

Cancel Exits Calibration Wizard.

Next Click to continue.



Calibration Completed dialog box help

Allows you to exit the calibration wizard or continue to **Save** options.

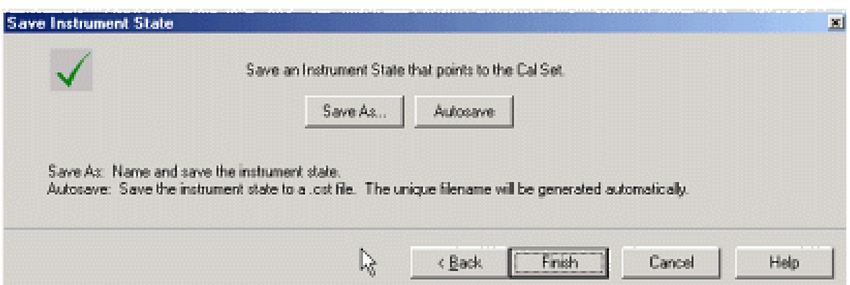
No. Finish now. Exits calibration wizard

Yes Allows selection of Save options.

Finish Performs the following:

- Saves Cal Set to memory
- Turns correction **ON**
- Exits calibration wizard

Note: You can apply Cal Set data to other channels. See Using Cal Sets.



Save Instrument State dialog box help

Allows you to save calibration data and instrument state settings to the following locations:

- Internal Hard Disk
- Floppy Disk
- External CD-RW Drive, PC or Server-Using a Mapped Network Drive

Save As Saves calibration and instrument settings (using **.cst** file type). You determine the name of the file and location to store it.

Autosave Automatically saves instrument settings and Cal Set to the current location and assigns a new file name in the **.cst** file type. A message temporarily appears on the display showing the path and new filename. The error terms are automatically stored as Cal Sets in the analyzer's memory.

Back Click to view previous dialog box.

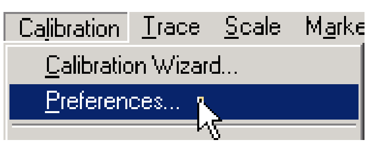
Cancel Exits calibration wizard.

Notes:

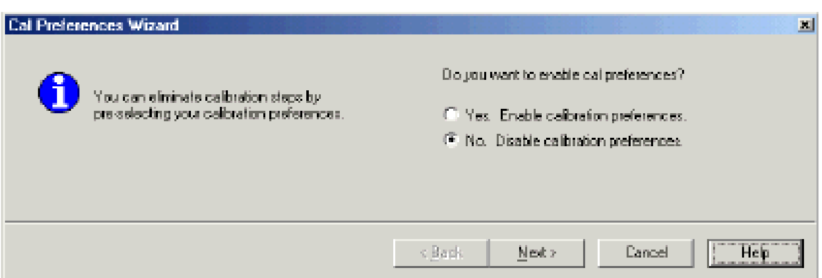
- You can recall the instrument settings and calibration data from memory or recall just the Cal Set. See Using Cal Sets.
- For more information about saving and recalling data, see Save and Recall.

Calibration Preferences

How to change Cal Preferences



Learn more about using the front panel interface



Cal Preferences dialog box help

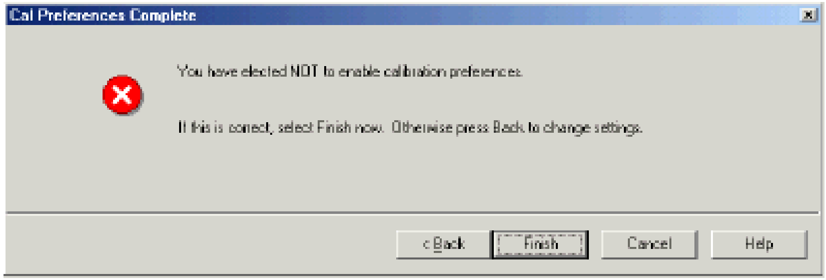
Cal preferences allows you to save the following selections from the Begin Calibration page of the Cal Wizard:

- Cal method (Guided, UnGuided, ECAL)
- Create New Cal Set (checked or cleared)

Yes. Enable calibration preferences The selections will always be made and you will not see the Begin Calibration page again until you disable calibration preferences. Click **Next** to make your selections on the Begin Calibration page.

Note: you can also Enable calibration preferences directly from the Begin Calibration page.

No. Disable calibration preferences See the Begin Calibration page every time you start Cal Wizard. Click **Next** to see the following dialog box confirming your choice to disable preferences.



Cal Preferences Complete dialog box help

Click **Finish** to complete cal preferences.

Using Calibration Sets

Calibration Sets (Cal Sets) enable you to use a single calibration on multiple channels.

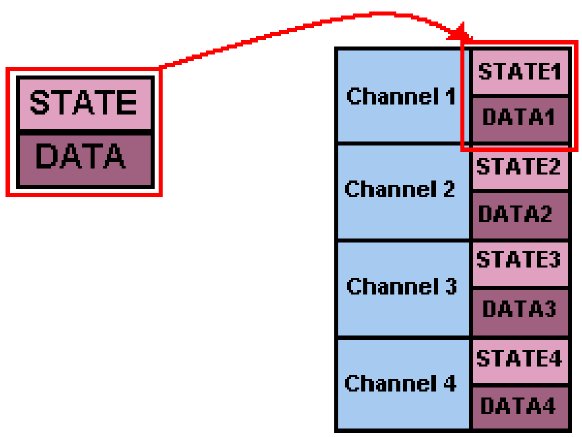
- PNA Calibration Data, past and present
- Creating a Cal Set
- Managing and Applying Cal Sets
- Examples of Cal Set Usage
- Archiving Cal Sets using .cal files

See other Calibration Topics

PNA Calibration Data, past and present

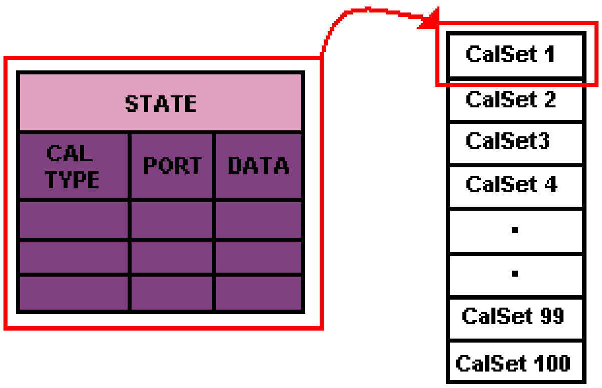
Past (before-PNA release 2.0)

One .cst File	Analyzer set up



- Calibration data was ONLY stored in either a .cal file or a .cst file.
- Calibration data could not be shared between channels.

Present with Cal Sets

<table>
<tr><td>One Cal Set (this image shows only some of the data saved in a cal set).</td><td>Cal Set file</td></tr>
<tr><td colspan="2">
</td></tr>
<tr><td colspan="2">

- Cal data is always saved to the cal set file (PNACalSets.dat).
- A cal set can be applied to any channel that has the same stimulus settings as the cal set.
- .cst (Instrument state plus cal data) files are NOW saved with a "pointer" to the cal set. See .cst file types
- Cal sets allow storage of multiple Cal Types.

</td></tr>
</table>

Each Cal Set contains the following items:

1. GUID Global Unique IDentifier
2. Cal Set description
3. Cal Set Attributes - stimulus settings, cal type, port association
4. Standards data
5. Error term data
6. Interpolated Error term data

Creating a Cal Set

After a guided or unguided calibration, a cal set is automatically created. At that time you can choose to also save the instrument state. When recalled, the instrument state will also recall the cal set.

Cal Sets can also be created programmatically:

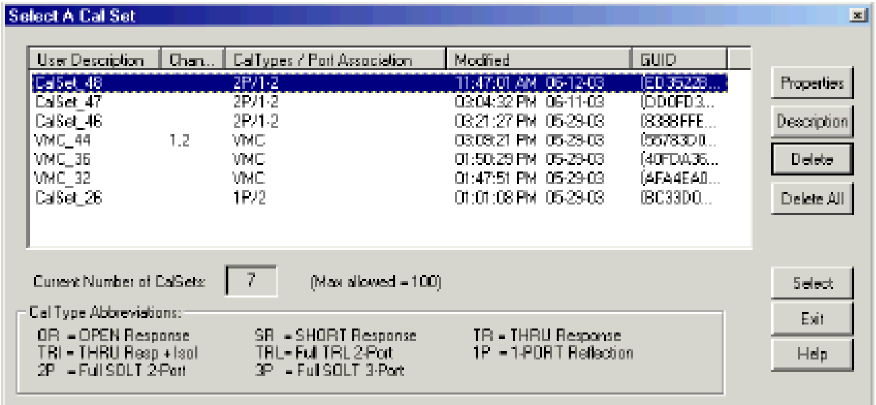
1. From COM programming using the Calibrator::SetCalInfo method.
2. From COM programming using the CalManager::CreateCalSet method.
3. From SCPI programming using the SYST:CORR:WIZARD to start the Cal Wizard.

Once the Cal Set has been created, it can be applied to a measurement and then managed.

Managing and Applying Cal Sets

Cal Sets are managed through the Select A Cal Set dialog box.

To access the dialog box, click **Calibration** then **Cal Set...**



Select a Cal Set dialog box help

Allows you to select (apply) a Cal Set to a measurement and view available Cal Sets, their properties, and associated Cal Types.

Although the number of Cal Sets you can have is limited to 100, it is considered unusual to have more than about 10 existing Cal Sets, or one current Cal Set for every unique channel setup. Old Cal Sets (with 'stale' data) should be deleted or overwritten.

Columns click a heading to sort by that column

User Description User name to identify the Cal Set. To change, click **Description.**

Channels Channel numbers that are currently using this Cal Set.

CalTypes / Port Association Type of Cal and port number of the Cal Set.

Modified Date and time the Cal Set was last modified.

GUID Unique identifier for each Cal Set.

Current Number of CalSets Number of listed Cal Sets. If the number reaches 100, the oldest Cal Set will be written over by the newest Cal Set.

Cal Type Abbreviations Description of the Calibration Type.

Buttons

Properties Invokes the Calibration Properties window.

Description Edit the Cal Set User Description.

Delete Permanently deletes the Cal Set after choosing OK to a warning prompt.

Delete All Permanently deletes ALL listed Cal Sets after choosing OK to a warning prompt.

Select Applies a Cal Set to the Active measurement. If the stimulus settings of the Cal Set and Measurement are different, a choice must be made. If more than one valid Cal Type is available, the Select Calibration Type dialog box allows you to choose.

Exit Exit the dialog box without selecting a Cal Set.

Help Invokes this section of Help

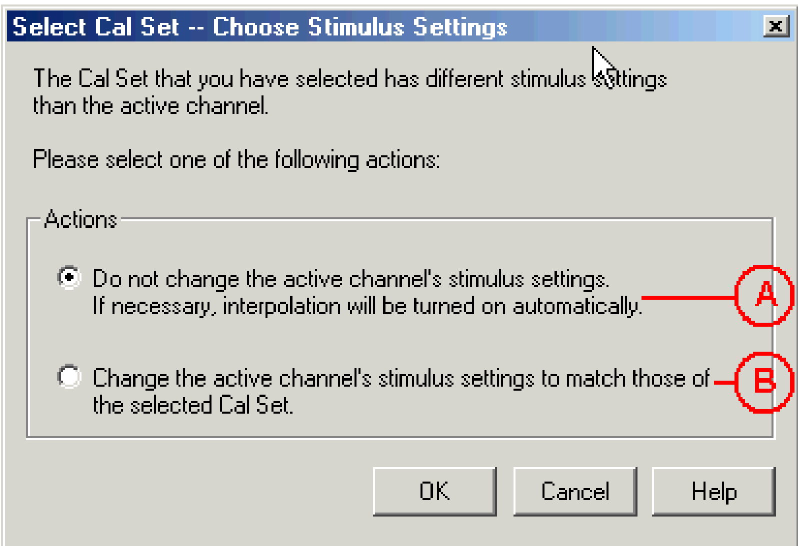
Stimulus Setting Different between Cal Set and Measurement

The Cal Set contains the measurement settings that were used at the time of the calibration. These settings may be different than those of the new measurement to which the Cal Set is being applied. If there is a difference, you can choose between the following options.

If the frequency span of the **Cal Set is equal to or greater than the measurement.**

	Frequency Range
Cal Set	|--------------|
Measurement	|--------|

Then choose



either:

A The calibration will be interpolated Learn more about Interpolation Accuracy

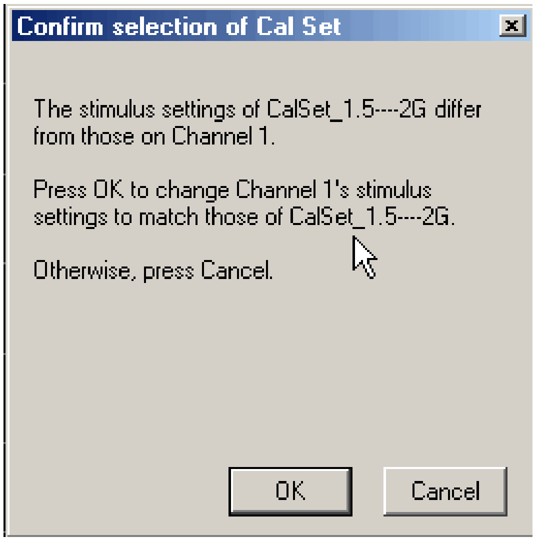
B The measurement will change to equal the Cal Set

For Cal Sets to be applied correctly, Interpolation must be turned on.

If the frequency span of the **Cal Set is less than the measurement.**

	Frequency Range
Cal Set	|--------|
Measurement	|-------------

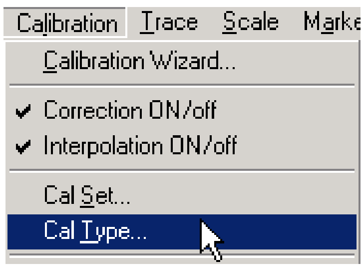
Then choose either:



OK Reduce the measurement frequency span

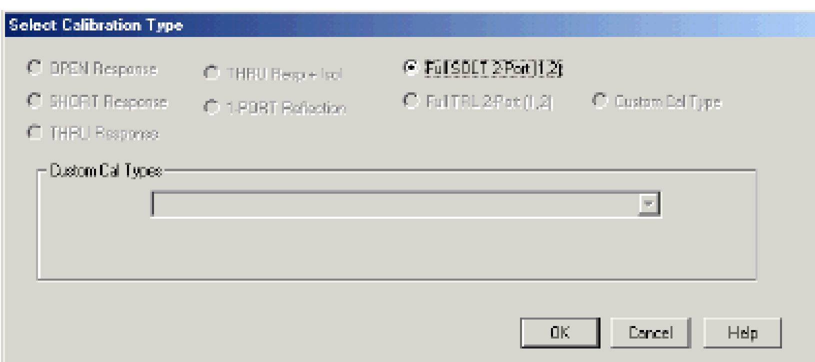
Cancel Cal Set will NOT be applied

How to Select a Calibration Type



Learn more about using the front panel interface

The Select Calibration Type dialog box opens.



Select Calibration Type dialog box help

It is possible for a Cal Set to have more than one Cal Type. When applying a Cal Set, if there is more than one Cal Type, this dialog box presents the choices.

1. Click an available Cal Type.
2. Click **OK** to select or **Cancel** to close the dialog box.

Learn about Custom Cal Type choices

Examples of Cal Set Usage

The following examples show how Cal Sets increase flexibility and speed in making analyzer measurements.

- **Using one Cal Set with many Measurements**
- **Using one Measurement with many Cal Sets**

Using one Cal Set with many Measurements

It is common to do one calibration, then to use it to set up several measurements, all using the same calibration data.

An example:

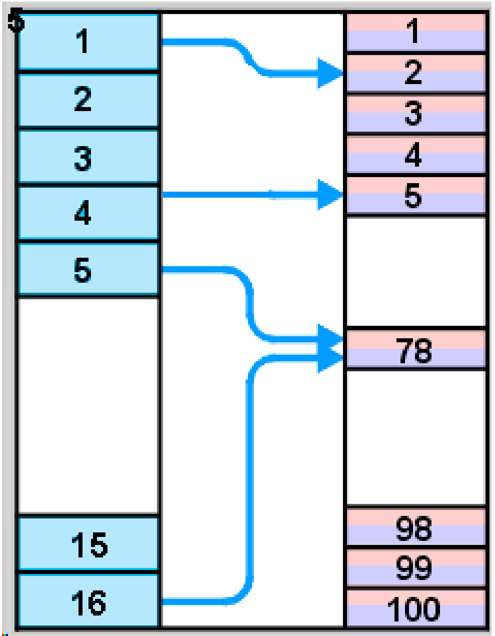
During a manufacturing process, you may have many calibrated S-parameter Measurements. You may wish to continuously cycle through the Measurements and examine them individually. Occasionally, you may wish to refresh the calibration without having to recreate all the Measurement state files.

Here is how: First do some planning. After examining the stimulus settings for the group of S-parameter Measurements, the Cal Set stimulus range is designed to be a super-set of the whole group. Each measurement can then use the same Cal Set.

Note: Make sure that interpolation is turned on. The individual S-parameter Measurements will have stimulus settings that will be a sub set of the Cal Set. Interpolation Error Corrected values will be calculated for these measurements.

Notice in the following image, Cal Set 78 is used on more than one Channel, in this case Channels 5 and 16 .

Channel **Cal Set**



Using one Measurement with many Cal Sets

The drawback with having one very large Cal Set associated with many instrument states could be a loss of measurement speed. In such cases, consider using one measurement with many Cal Sets. The stimulus conditions can then be changed for a measurement by applying different Cal Sets. Other settings (window setups, measurement definitions, scaling, limits, markers) will not change. This may result in faster state changes than if you saved and recalled *.cst files for each set of stimulus conditions

Example #1: An amplifier needs to be measured at several input power levels. Calibrate at several power levels and save each calibration in a separate Cal Set. Then, apply the Cal Sets to the single measurement consecutively.

Example #2: Making an S21 Measurement, you need to measure both wide span and narrow span characteristics of the device. One Cal Set covers the wide span setup, another the narrow span setup.

Example #3, Multi-port measurements: A 6-port device is being measured. It has one input port and 5 output ports. Measure S21 from the input to each output under identical measurement conditions (stimulus, markers, limits, etc). Create a Cal Set for each Port selection and recall them one at a time.

Archiving Cal Sets using .cal files

Because Cal Sets can easily be applied to measurements, they can also be easily deleted. To provide extra backup, you can still save your calibration as a .cal file (see saving a .cal file). All calibrations currently in use by the active channels on the analyzer as saved in a single .cal file.

Example:

One person performs a calibration, names and saves it as a Cal Set. This Cal Set is available for any other person to use. A second user could accidentally delete or modify the Cal Set requiring the originator to repeat the calibration.

Security can be provided for calibration data by saving the Cal Set to a **.cal** file. At a later time, the .cal file could be recalled and the original calibration restored.

Using ECal

This topic discusses all aspects of ECAL:

- **ECal Overview**
- **Connect ECal Module to the PNA**
- **Perform a Calibration Using ECAL**
- **ECAL User-Characterization**
- **Restore ECAL Module Memory**

See other Calibration Topics

ECal Overview

ECal is a complete solid-state calibration solution. Every ECal module contains electronic standards that are automatically switched into position during a PNA measurement calibration. These electronic standards have been measured at the factory and the data stored within the memory of the ECal module. The PNA uses this stored data, along with the PNA-measured data, to calculate the error terms for a measurement calibration.

ECal modules are available in 2-port and 4-port models and a variety of connector types, covering many frequency ranges. See Analyzer Accessories for more about available ECal modules and ordering information.

You can perform the following calibrations with ECal:

- 1-Port Reflection calibration
- Full 2-Port calibration
- Full 3-Port calibration

You can validate the accuracy of a mechanical or ECAL calibration with ECAL confidence check.

re and Handling of ECal Modules

You can improve accuracy, repeatability, and avoid costly repair of equipment in the following ways.

- Practice proper connector care. See Connector Care.
- Protect equipment against ESD damage. Read Electrostatic Discharge Protection.
- Do not apply excess power to ports. Refer to specifications provided with your ECal module.

Connect ECal Module to the PNA

ECal modules are controlled and powered through a USB connection to the PNA. When you connect the module, the PNA automatically recognizes the type of module, frequency range, and connector type.

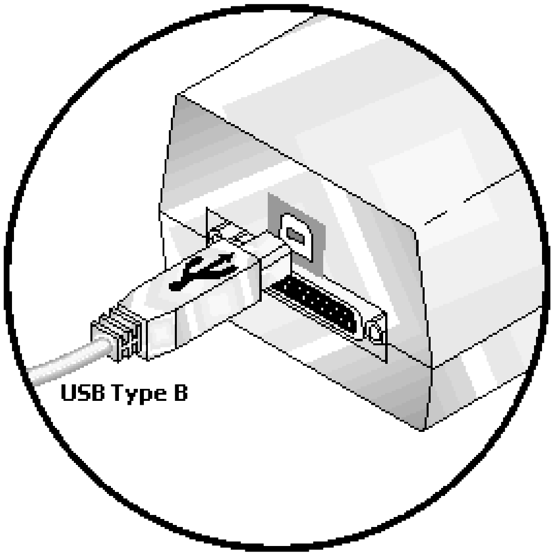
Note: Certain USB devices (such as ECAL modules) require you be logged on with Administrator privileges the first time you plug them into the PNA. This must be done for each serial number.

ECal modules connect to the USB port on the front or rear panel of the PNA.

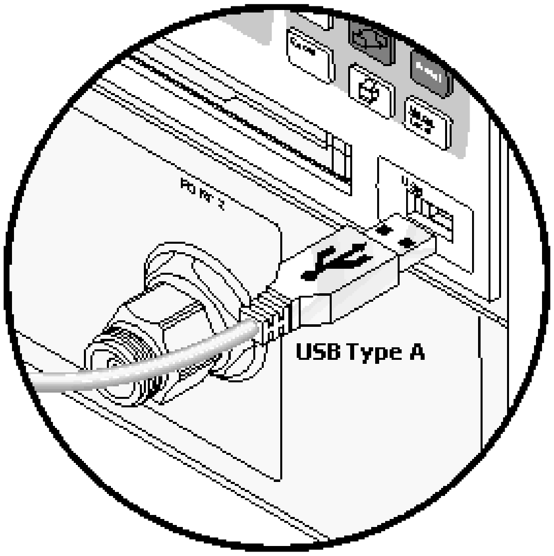
1. Wear a grounded wrist strap when making connections.

2. Connect the USB cable **Type B** connector to the ECal module and the USB cable **Type A** connector to the front or rear panel USB connector of the analyzer, as shown in the following graphics.

ECal Module USB Port



Analyzer Front Panel USB Port



Notes:

- Agilent 8509x and N4431 ECAL modules, when first connected, draw significantly more current than other modules. This could cause the USB to stop working. See USB limitations.
- Unused ECal modules that have completed a calibration may remain connected to the USB port.
- You can connect and disconnect the ECal module while the analyzer is operating. However, DO NOT connect or disconnect the module while data transfer is in progress. This can result in damage or at least corrupted data.
- A USB hub may be used to connect more than one USB device to the analyzer. See Analyzer Accessories for more information about USB peripheral equipment.

ECal module not found dialog box help

Displays an error message indicating the ECal module is not connected or has not been recognized by the network analyzer.

Retry Check the USB connections and click to continue.

Notes:

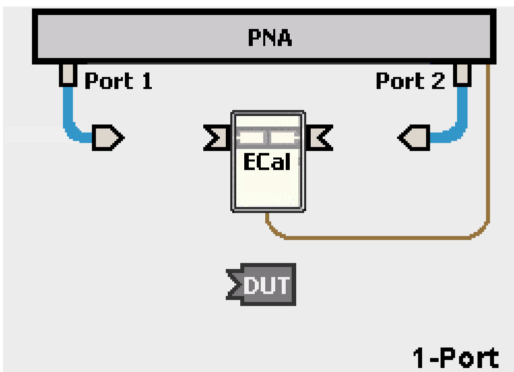
- When the ECal module is connected to the network analyzer for the first time, it may take approximately 30 seconds for the analyzer to recognize the module and make it available for calibration.
- For best accuracy, allow the ECal module to warm-up until it indicates READY.
- See Connect ECal Module USB to PNA USB.

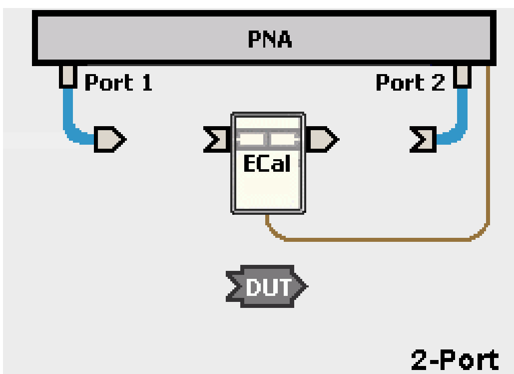
Perform a Calibration Using ECal

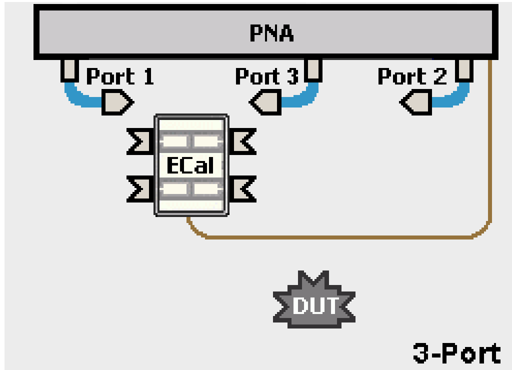
The following graphics show the typical connections for 1-port, 2-port, and 3-port calibrations and DUT measurements.

Note: Terminate any unused ECAL ports with a 50 ohm load.

Select an ECal module that has connectors of the same type and gender as the DUT. If such an ECal module is not available, a module with connectors different from the DUT can be used by using Advanced Settings or User Characterization.

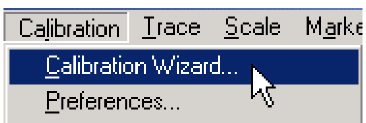




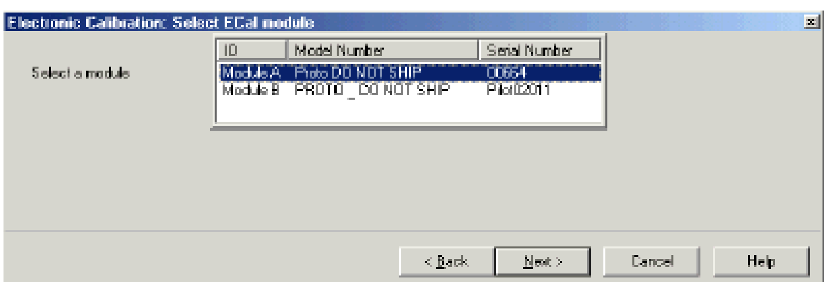


How to Perform a Calibration Using ECal:

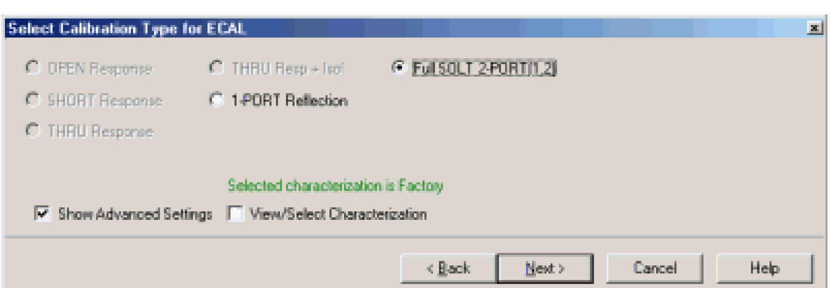
1. Connect the ECal module USB cable to the analyzer USB. See Connect ECal Module USB to PNA USB.
2. Allow the module to warm up until it indicates **READY**.
3. Enter the analyzer settings. See Set Up Measurements.
4. Start the Calibration Wizard as follows:



5. In the Calibration Wizard dialog box, click **Use Electronic Cal (ECal)**



Select ECAL Module dialog box help
If the PNA detects more than one ECAL module, select one of the modules, then click **Next.**



Select Calibration Type for ECal dialog box help

Allows you to select calibration type and settings.

1-PORT Reflection Click to select a 1-Port reflection calibration. Advanced Settings and Isolation are not available with 1-Port Reflection.

Full SOLT 2-PORT Click to select a Full 2-Port calibration.

Full SOLT 3-PORT Click to select a Full 3-Port calibration (3-port PNA only).

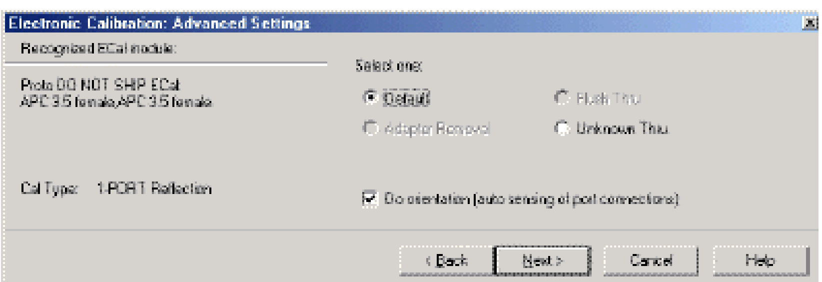
Show Advanced Settings Check to display the Advanced Settings for the calibration type selected.

View/Select Characterization Check to invoke Select Characterization dialog box. Learn more about User Characterization.

If the frequency range of the active channel is greater than that of the selected characterization, a **Caution: Additional Standards Required** message is displayed. To correct the problem, perform one of the following solutions:

- Reduce the frequency range of the active channel.
- Select a different characterization that covers the required frequency range.
- Re-characterize the module with an increased frequency range.

Note: The PNA no longer allows ECAL isolation to be performed. The inherent isolation of the PNA is better than that attained with correction using an ECAL module.



Advanced Settings dialog box help

Advanced Settings are enhancements that may be applied to a Full 2-Port or Full 3-Port calibration. Some settings are limited by the PNA model or ECal module connectors.

Recognized ECal module Displays the model number and connector type of the modules connected to the PNA.

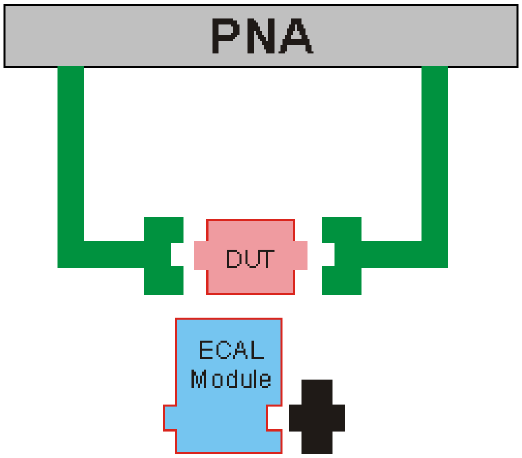
Select One

The following requires that you understand the terms "insertable and non-insertable" devices. See Calibration Accuracy to learn about these terms.

Default Measures the THRU state of the ECAL module. The easiest selection, and the most accurate if the PNA model being used has one reference receiver per port.

Adapter Removal Can be used with ECAL when your DUT is NON-insertable. However, the ECAL module MUST be insertable, and the adapter connectors must exactly match the connectors of the DUT as in the following diagram.

Note: With PNA release 4.8, adapter removal now performs 2-port measurements on both sides of the adapter. It previously performed 2-port measurements on one side and 1-port measurements on the other. This improves the accuracy of the adapter removal calibration.



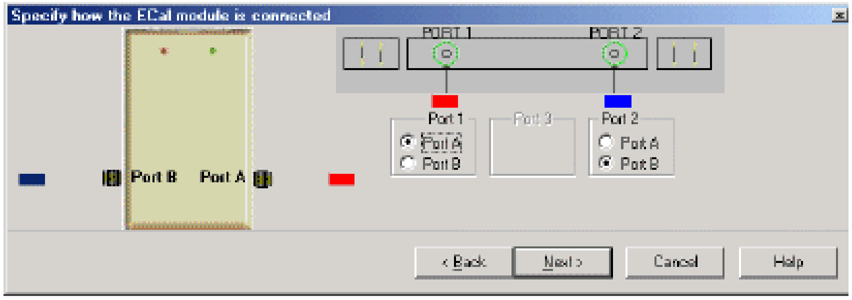
In cases when adapter removal cannot be performed, ECAL User Characterization is ALWAYS possible if you have the right adapters. A User Characterization is performed once and stored in the ECAL module. However, accuracy is compromised every time you remove, then reconnect, the adapter with the ECAL module.

Flush Thru (zero-length Thru) The THRU state of the ECal module has more loss than a zero-length thru, resulting in degraded transmission tracking error terms. This setting requires an insertable ECal module. When this setting is checked, you will be prompted to remove the ECal module and connect the two reference planes directly together for a zero-length thru.

Unknown Thru If your ECAL module pair of connectors are non-insertable, you cannot use a zero-length (or Flush) thru. However, you can use the Unknown Thru setting to achieve higher accuracy than using the Thru standard in the ECAL module. The ECal module is removed and a mechanical thru adapter is connected between the measurement ports. Learn more about Unknown Thru.

Do orientation When this box is checked (default) the PNA senses the model and direction in which an ECAL module port is connected to the PNA ports. If power to the ECAL module is too low, it will appear as if there is no ECAL module connected. If you use low power and are having this problem, clear this check box to provide the orientation manually.

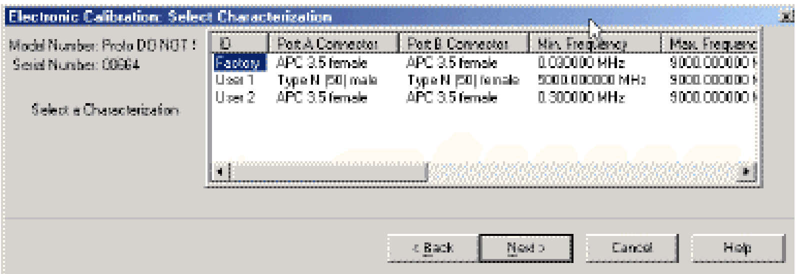
Orientation occurs first at the middle of the frequency range that you are calibrating. If a signal is not detected, it tries again at the lowest frequency in the range. If you have an **E8361A** or **E836xB** PNA and do an ECAL completely within 10 - 20 MHz OR 60 - 67 GHz, you may need to do orientation manually. There may not be sufficient power to orient the ECAL module at those frequencies.



Specify how the ECAL module is connected dialog box help

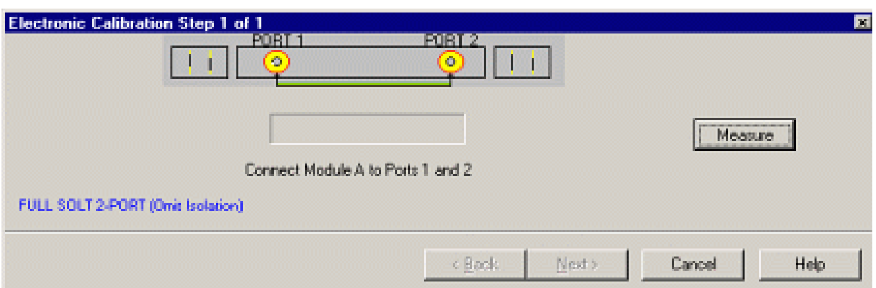
This dialog box appears when the **Do orientation** checkbox in the previous dialog box is cleared.

Click Port A and B to reflect the orientation of the ECAL module.



Select Characterization dialog box help

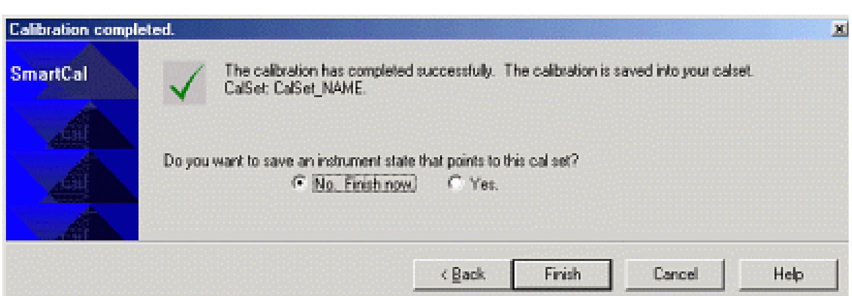
Select either the Factory Characterization of your ECAL module or a User Characterization. Once selected, that characterization becomes the default selection until the PNA is turned OFF and restarted. When restarted, **Factory** again becomes the default selection.



Electronic Calibration Steps dialog box help

Displays the instructions for each measurement required for calibration.

Measure Measures the ECal standards.



Calibration Completed dialog box help

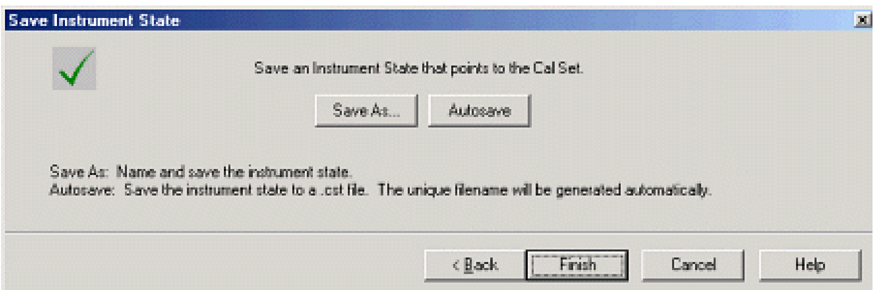
Allows you to finish the calibration or continue to **Save** options.

No. Finish now. Completes calibration.

Yes Allows selection of Save options.

Finish Performs the following:

- Saves Cal Set to memory
- Turns correction **ON**
- Exits Calibration Wizard



Save Instrument State dialog box help

Save As Saves calibration and instrument settings (using **.cst** file type). You determine the name of the file and location to store it.

Autosave Saves instrument settings and Cal Set data to the current location and automatically assigns a new file name of the .cst file type. A message temporarily appears on the display showing the path and new filename.

- Learn more about Using Cal Sets.
- Learn more about saving and recalling files.

ECAL User Characterization

A user characterized ECal allows you to add adapters to the ECal module, re-measure the ECal standards INCLUDING the adapters, then add that data to ECal memory. This extends the reference plane from the module's test ports to the adapters.

There are several reasons you might want to perform a user characterization:

- If you need to use adapters with your ECAL module, you could characterize your ECAL module with the adapters attached and perform subsequent ECALs with a single step.
- If you have a 4-port ECAL module, you could configure the module with adapters of different connector types. Then perform a user characterization of the module. When you need to test a DUT with a pair of the connector types on your module, calibrate the PNA with a 1-step ECAL using the two connectors of the User characterized module.
- If you test devices in a fixture, you could embed the characterization of the fixture in the characterization of

the module. To do this, during the mechanical calibration portion of the user characterization, calibrate at the reference plane of the device as you would normally calibrate. Then remove the fixturing to be embedded and insert the ECAL module to be characterized. When measuring the ECAL module, the PNA removes the effects of the fixturing and stores the measurement results in the user characterized ECAL module. Subsequent calibrations with that user characterized module will also remove the fixture effects.

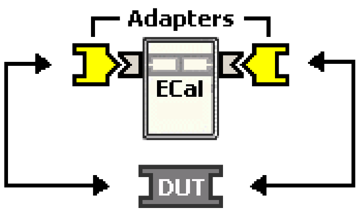
Notes:

- User Characterization does not delete the factory characterization data. The factory data is saved in the ECal module in addition to the user characterization data.
- You can save up to five different user characterizations in a single ECAL module. There are memory limitations; the PNA will determine if the contents of a user characterization will fit inside the module before it is performed. **Note:** This is a new feature with PNA Rev. 3.0. Previous versions of PNA will NOT recognize more than one user characterization.
- Both 2-port and 4-port ECal modules support user characterization.

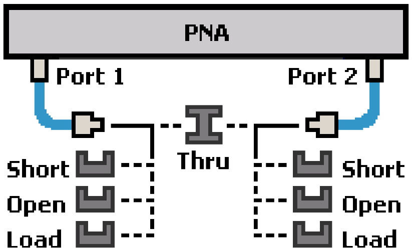
How to Perform a User Characterization

SUMMARY (A detailed procedure follows.)

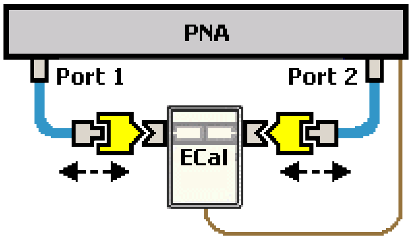
1. Select adapters for the module to match the connector configuration of the DUT.



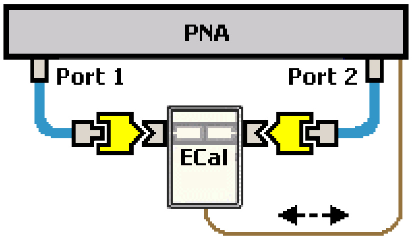
2. Either perform a Guided calibration using mechanical standards or recall a Cal Set. This is required to make accurate measurements of the ECAL module and adapters.



3. Measure the ECal module, including adapters, as though it were a DUT.

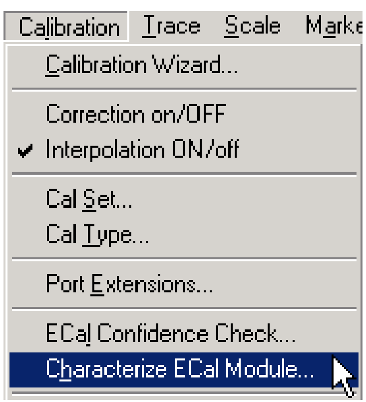


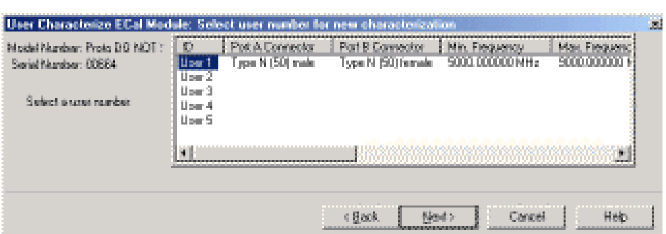
4. The measurement result is the characterization data that then gets stored inside the module.



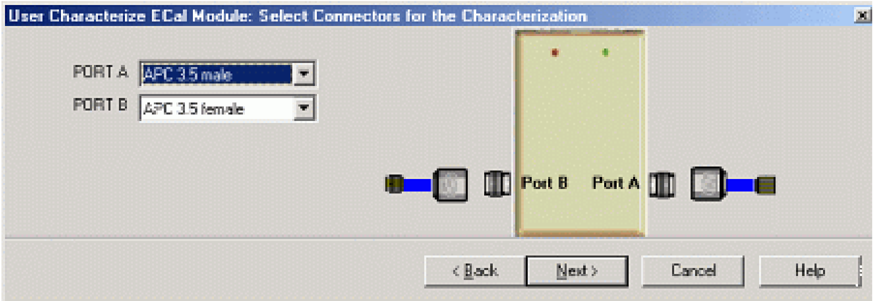
Detailed steps to Perform a User Characterization:

1. Connect the ECal module to the network analyzer with the USB cable. See <u>Connect ECal Module USB to PNA USB</u>.
2. Allow the module to warm up until it indicates **READY**.
3. **Preset** the analyzer.
4. <u>Set up the measurement</u>. For best accuracy, the **IF bandwidth** should be set to **1 kHz** or less.
5. Start the **Characterize ECAL Module** Wizard:





Select user number for new characterization dialog box help
Shows the characterizations already in the module. Scroll to view all of the parameters of the stored characterizations. Select an empty location or overwrite an existing characterization. **Next** Click to continue to the <u>Select Connectors for the Characterization</u> dialog box.



Select Connectors for the Characterization dialog box help

Allows you to select the adapters for the ECal module test ports. Select **No adapter** if no adapter is used on a port.

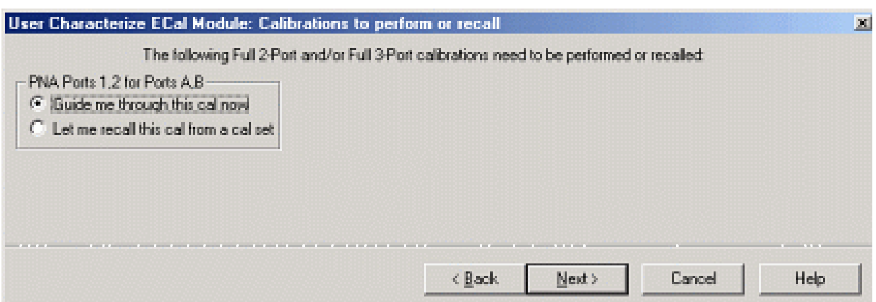
PORT A Lists the connector types available for Port A.

PORT B Lists the connector types available for Port B.

PORT C Lists the connector types available for Port C (available with a 4-port ECal module).

PORT D Lists the connector types available for Port D (available with a 4-port ECal module).

Next Click to continue to the Calibrations to perform or recall dialog box.



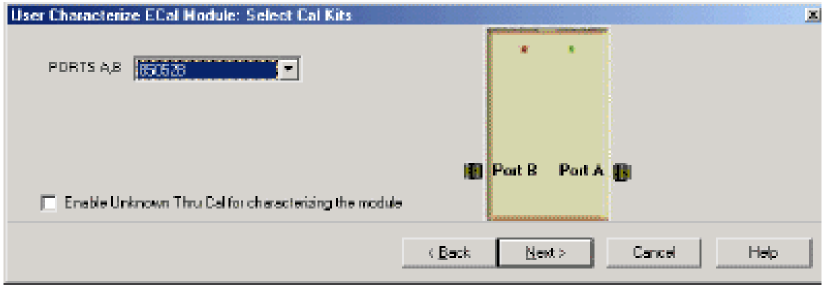
Calibrations to perform or recall dialog box help

The PNA must be calibrated before measuring the ECAL module and necessary adapters. This dialog box displays the number and types of mechanical calibrations required for the characterization.

Guide me through this cal now Click to perform a Guided calibration. A calibration kit is required for each connector type.

Let me recall this cal from a cal set Click to select a Cal Set from a previous calibration for the connector types. Learn more about Using Cal Sets.

Next Click to continue to either the Select Cal Kits or the Select Cal Set dialog box.

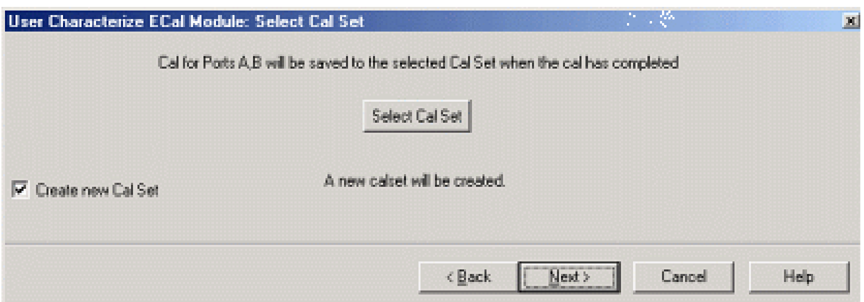


Select Cal Kits dialog box help

Provides a list of calibration kits to perform the calibration. Select the cal kit you will use for each port.

Enable Unknown Thru for characterizing the module Check to enable. This reduces the number of steps required to characterize the THRU standard. Learn more about Unknown Thru Cal This setting is available only on E835xA and E836xA series network analyzers.

Next Click to continue to the Select Cal Set dialog box.



Select Cal Set dialog box help

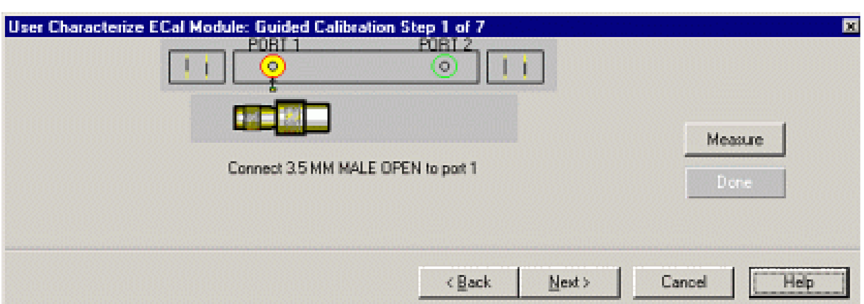
The calibration that you perform will be written to a Cal Set. This dialog box allows you to select a Cal Set to overwrite, or to write to a new Cal Set. The current choice is visible below the **Select Cal Set** button.

Select Cal Set Click to open the **Select A Cal Set** dialog box.

Create new Cal Set Check to create a new Cal Set to store the calibration. Clear to select and overwrite a stored Cal Set.

Next Click to continue to the Guided Calibration Steps dialog box.

Note: Make a note of the Cal Set name for future reference.



Guided Calibration Steps dialog box help

Instructs you to connect each calibration standard to the measurement port.

Measure Click to measure the standard.

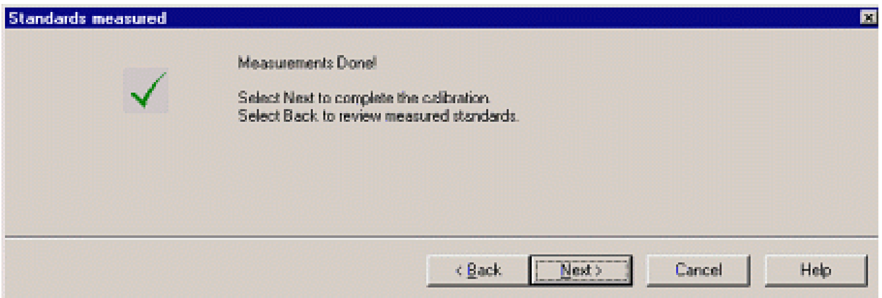
Back Click to repeat one or more calibration steps.

Done Click **after** a standard is re-measured and all measurements for the calibration are complete.

Next Click to continue to the next calibration step. (Does **not** measure the standard.)

Cancel Exits Calibration Wizard.

The Standards measured dialog box opens when the steps are completed.

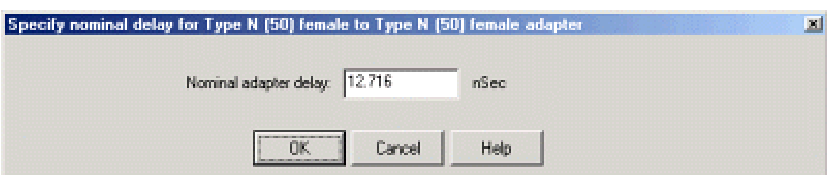


Standards measured dialog box help

Indicates that all standards required for the calibration have been measured.

Back Click to re-measure a calibration standard.

Next Click to continue to either the Specify nominal delay or the Guided Calibration Completed dialog box.



Specify nominal delay dialog box help

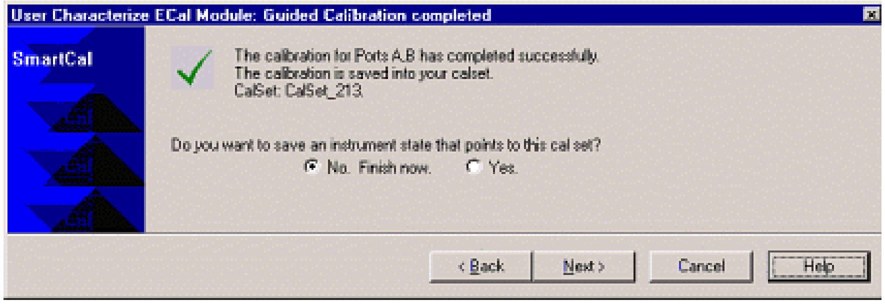
An estimate of the nominal delay is required.

OK Click to accept adapter delay.

For CW sweep or frequency sweeps with large frequency steps, overwrite the estimate and then click OK.

Compute delay by calculating the physical length of the adapter divided by the propagation velocity.

Delay = Length / Propagation Velocity



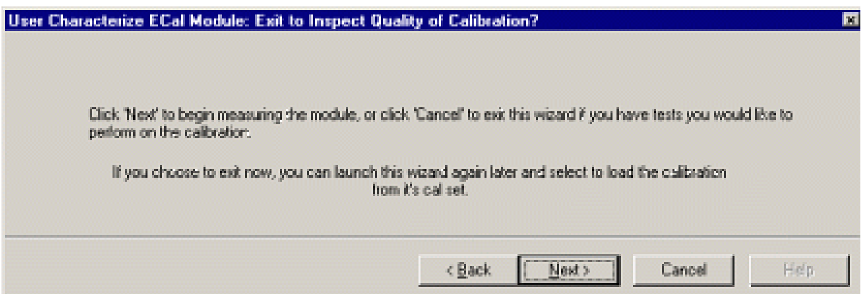
Guided Calibration completed dialog box help

Allows you to finish the calibration and continue to the next characterization steps.

No. Finish now Select to save Cal Set data.

Yes Allows selection of Save options.

Next Click to continue to the Exit to Inspect Quality of Calibration dialog box.



Exit to Inspect Quality of Calibration dialog box help

Allows you to exit User Characterization to validate the calibration before proceeding with the characterization.

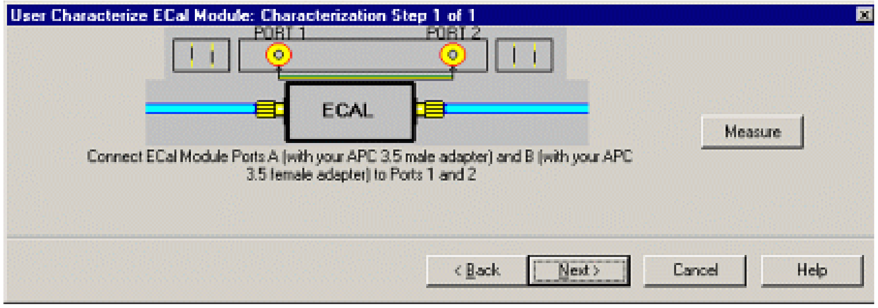
Back Allows you to repeat calibration.

Next Click to continue to the Characterization Steps dialog box.

Cancel Exits the Calibration.

To return to the current step:

1. Start User Characterization.
2. In the **Select user number for new characterization** dialog box, click **Next**.
3. In the **Select Connectors for Characterization** dialog box, click **Next**. (Previous entry is stored in memory.)
4. In the **Calibrations to perform or recall** dialog box, recall the Cal Set that you just performed.

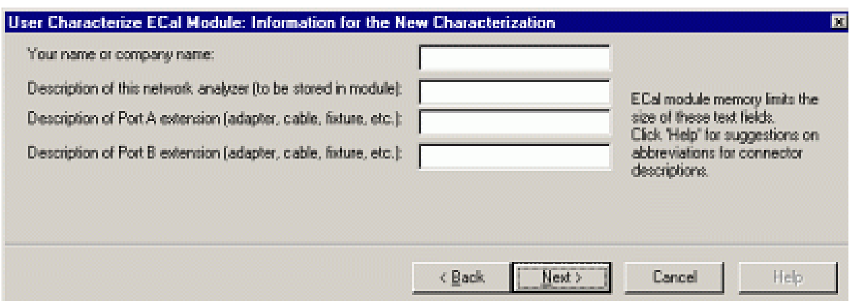


Characterization Steps dialog box help

Describes the instructions for each measurement required for characterization.

Measure Measures the ECal module.

Next Click to continue to the Information for the New Characterization dialog box when measurements are complete.



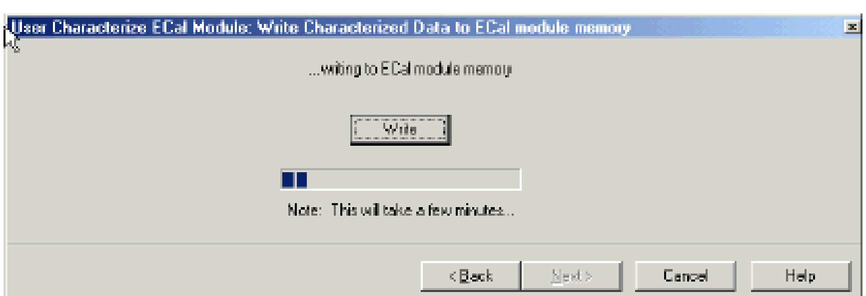
Information for the New Characterization dialog box help

Allows you to describe the properties of the User Characterization.

Next Click to continue to the Write Characterized Data to the ECal module dialog box.

To minimize the number of characters, use the following 3-character codes to describe the connectors listed.

Connector Type	3-Character Code
2.4 mm female	24F
2.4 mm male	24M
3.5 mm female	35F
3.5 mm male	35M
7-16 female	16F
7-16 male	16M
Type F female	F7F
Type F male	F7M
N50 female	N5F
N50 male	N5M
N75 female	N7F
N75 male	N7M
APC 7	7MM
K-band waveguide	KBW
P-band waveguide	PBW
Q-band waveguide	QBW
R-band waveguide	RBW
U-band waveguide	UBW
V-band waveguide	VBW
W-band waveguide	WBW
X-band waveguide	XBW

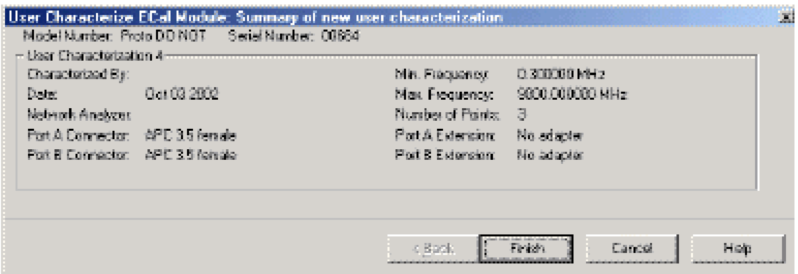


Write Characterized Data to the ECal module memory dialog box help

The PNA writes User Characterization and factory characterization data to the ECal module memory. For more information, see Restore ECAL module memory.

Write Click to write data into the ECal module.

The Updated Summary of ECal modules detected on USB dialog box opens after data is saved to module.



Updated Summary of ECal modules detected on USB dialog box help

Verify the status of ECal modules connected to the PNA.

Module 1, Module 2 Displays the following:

- ECal module model number
- summary from user characterization

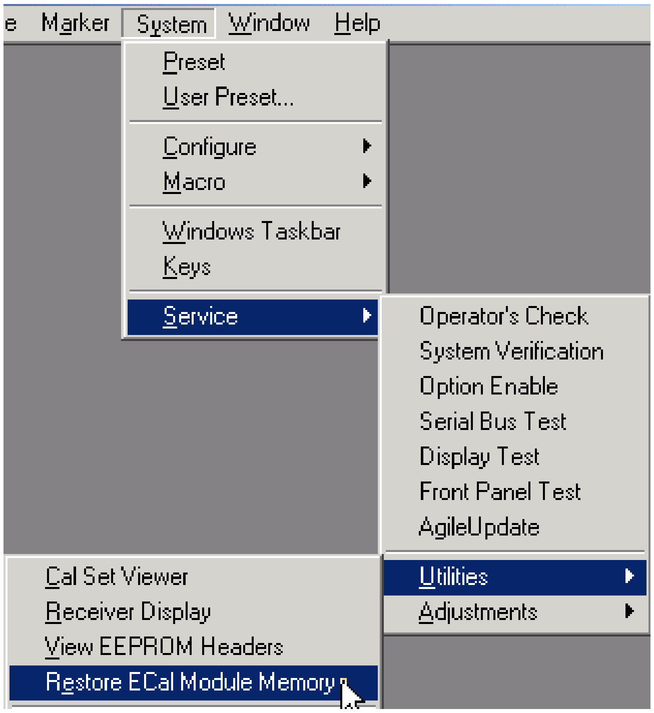
Cancel Click to exit (characterization complete).

Finish Click to exit (characterization complete).

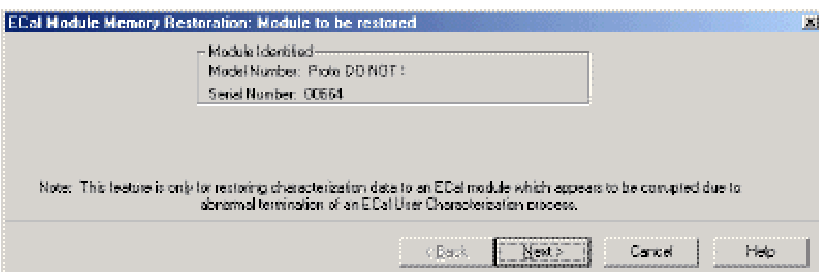
Restore ECAL Module Memory

When user-characterized data is written to the ECAL module, the entire contents of ECAL memory is also written to the PNA hard drive. In the unlikely event that your ECAL module memory is lost, you can restore the user-characterized data to ECAL memory.

How to Restore ECAL Module Memory



Learn more about using the front panel interface



Module to be restored dialog box help

Verify the Serial number of the module to be restored. If two modules are connected to the PNA , choose the one to have data restored.

Next Click to write data to the module.

Accurate Measurement Calibrations

Calibration accuracy is affected by the type of calibration, quality of the calibration standards, and the care with which the calibration is performed. This section provides additional information about how to make accurate calibrations.

- Measurement Reference Plane
- Effects of Using Wrong Calibration Standards
- Data-based versus Polynomial Calibration Kits
- Accuracy Level of Interpolated Measurement
- Effects of Power Level
- Setting System Impedance
- Using Port Extensions
- Isolation Portion of 2-Port Calibration
- Choosing a Thru Method

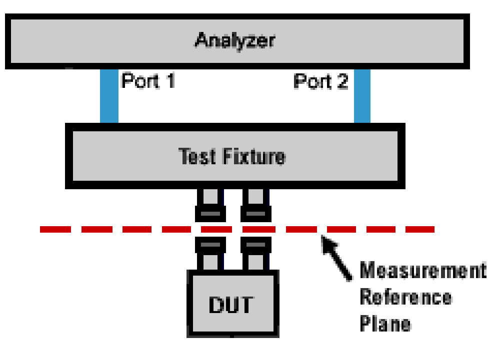
Learn how to determine the validity of your calibration.

See other Calibration Topics

Measurement Reference Plane

Most measurement setups will NOT allow you to connect a device under test directly to the analyzer's front panel test ports. More likely, you would connect your device to test fixtures or cables that are connected to the analyzer.

For the highest measurement accuracy, you should calibrate at the points where you connect your device. This is called the measurement reference plane (see graphic). If you calibrate at these points, the errors associated with the test setup (cables, test fixtures, and adapters used between the analyzer ports and the reference plane) are measured and removed in the calibration process.



Effects of Using Wrong Calibration Standards

Normally, a calibration is performed using a calibration kit that contains standards with connectors of the same type and sex as your device under test.

However, your calibration kit may not have the same connector type as your device. For example, suppose your device has 3.5mm connectors, but you have a Type-N calibration kit. So, you use an adapter to connect the Type-N standards to the 3.5mm test port. Because the adapter is part of the calibration and NOT part of the test setup, this will result in significant errors in your reflection measurements.

Data-based versus Polynomial Calibration Kits

The Select Cal Kit dialog box offers a data-based model and a polynomial model for the newest high-frequency cal kits. See PNA Accessories. The data-based models provide higher accuracy for describing calibration standards than the polynomial models. It is RECOMMENDED that the data-based model be used if the most accurate results are desired.

	Data-Based Model	**Polynomial Model**
How accurate is the model?	Provides highest calibration accuracy. Eliminates the errors that can be the result of polynomial model approximations.	Provides high calibration accuracy.
How does the model define calibration standards?	Uses S-Parameter measurements.	Uses traditional four-term polynomial calibration standard modeling parameters.
How do I manually edit the definitions of the calibration standards when using the model?	Use the Advanced Modify Cal Kit function.	Use the Advanced Modify Cal Kit function.
How do I use the Calibration Wizard with the model?	Use only the SmartCal (Guided) Calibration method.	Use the SmartCal (Guided) or the Unguided Mechanical Calibration methods.

Learn about the "Expanded Math" feature.

Accuracy Level of Interpolated Measurement

When your current instrument settings do not exactly match those under which the calibration was performed, the PNA may automatically interpolate your calibration. Therefore, the accuracy of the measurements cannot be predicted. The measurement accuracy may be affected significantly or not at all. Identifying measurement errors in these cases must be determined on a case-by-case basis.

Significant measurement inaccuracy WILL occur when the phase shift between measurement points increases more that 180 degrees. The PNA will incorrectly interpolate the new phase data. For more information, see phase accuracy.

In general, the chances of significant inaccuracy increases when interpolating measurements under the following conditions:

- when increasing, rather than decreasing, the frequency span between measurement points.
- when frequency span between measurement points becomes much greater.
- when measurement frequencies are very high, especially above 10 GHz.

See Validity of a Calibration to learn more about interpolation correction levels and determining the accuracy of your calibration.

Effects of Power Level

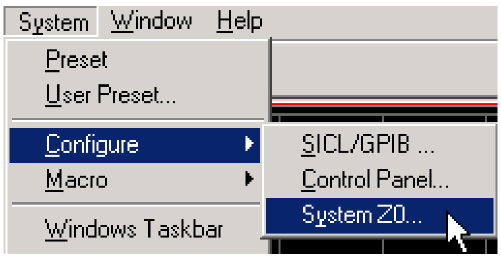
To attain the most accurate error correction, do NOT change the power level after a calibration is performed. However, when changing power within the same attenuator range at which the measurement calibration was performed, S-parameter measurements can be made with only a small degradation of accuracy. If a different attenuator range is selected, the accuracy of error correction is further degraded.

To check the accuracy of a calibration, see Validity of a Calibration.

Setting System Impedance

The system impedance can be changed for measuring devices with an impedance other than 50 ohms, such as waveguide devices.

How to change the System Impedance



Learn more about using the front panel interface

System Z0 dialog box help

Allows you to change the system impedance (default setting is 50 ohms).

Z0 Displays the current system impedance.

For 75 ohm devices:

1. Change the system Z0 to 75 ohms.
2. Connect minimum loss pads (75 ohm impedance) between the analyzer and the DUT.
3. Perform a calibration with 75 ohm calibration standards.

For waveguide devices:

1. Change the system Z0 to 1 ohm.
2. Perform a calibration with the appropriate waveguide standards.

Using Port Extensions

Use the port extensions feature after calibration to compensate for phase shift of an extended measurement reference plane due to additions such as cables, adapters, or fixtures.

Port extensions is the simplest method to compensate for phase shift between the calibration reference plane and the DUT. Applying port extensions does not, however, compensate for the mismatch and loss of the path between the reference plane and the DUT. For this reason, you should minimize the loss and mismatch caused by a test fixture to achieve best measurement accuracy.

To learn how to apply port extensions, see Port Extensions.

To learn about characterizing a test fixture, see Fixturing.

Isolation Portion of 2-Port Calibration

The isolation portion of a calibration corrects for crosstalk, the signal leakage between test ports when no device is present. When performing an UNGUIDED 2-port calibration, you have the option of omitting the isolation portion of the calibration.

Note: Isolation is never performed on a Smart (Guided) Calibration.

The uncorrected isolation between the test ports of the PNA is exceptional (typically >100dB). Therefore, you should only perform the Isolation portion of a 2-port calibration when you require isolation that is better than 100dB. Perform an isolation calibration when you are testing a device with high insertion loss, such as some filter stopbands or a switch in the open position.

The isolation calibration can add noise to the error model when the measurement is very close to the noise floor of the analyzer. To improve measurement accuracy, set a narrow IF Bandwidth.

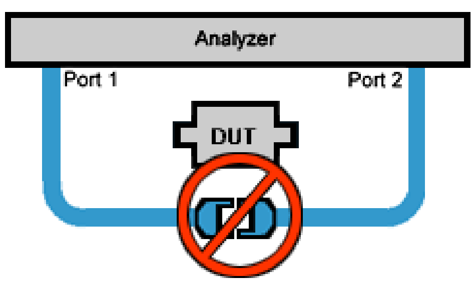
How to perform an isolation calibration

Isolation is measured when the Load standards are connected to the PNA test ports. For best accuracy, connect Load standards to BOTH test ports each time you are prompted to connect a load standard. If two Loads are not available, connect to the untested PNA port any device that will present a good match.

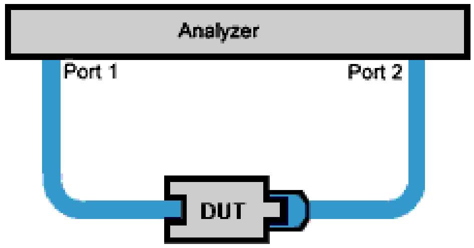
Choosing a Thru Method

To understand the Thru method choices, you must first understand the following terms. These definitions also apply to ECal modules. Substitute "ECal module" for "device".

A **non-insertable device** is one whose connectors could NOT mate together. They either do not have the same type of connector or they have the same gender. This also means that the test port cables would not mate together, as in the following diagram.



An **insertable** device is one whose connectors could mate together. They have the same type of connector and opposite,or no, gender. This also means that the test port cables would mate together, as in the following diagram.



Choices for Thru Method of Calibration

The Thru method is selected from the Cal Wizard. Select the **Preview/ Modify** checkbox in the Select Cal Kits dialog box.

Note: For ECal, the following choices have different meanings. See THRU methods for ECal.

Choice for Insertable Devices: FLUSH Thru (also known as **Zero-length Thru**)

When calibrating for an insertable device, the test ports at your measurement reference plane connect directly together. This is called a zero-length THRU, or Flush THRU meaning that the electrical description the analyzer uses in the calibration process has zero-length: no delay, no loss, no capacitance, and no inductance.

Although there is a zero-length thru definition, your calibration kit may not have a THRU standard because it is assumed you have an insertable device and will be using a zero-length THRU.

Choices for Non-Insertable Devices

The following methods calibrate for a non-insertable device:

- Adapter Removal Accurate, but least convenient.
- Defined Thru
- Unknown Thru Cal **Preferred method.**

(Swap-Equal-Adapters Method is NOT recommended in the PNA.)

Adapter Removal Calibration

This method is potentially very accurate. However, it requires many connections which increases the chances of inaccurate data.

Two full 2-port calibrations are performed: one with the adapter connected at port 1, and the other with the adapter connected to port 2. The result of the two calibrations is a single full 2-port calibration that includes accurate characterization and removal of the mismatch caused by the adapter.

Performing an Adapter Removal Cal requires:

- a THRU adapter with connectors that match those on the DUT.
- calibration standards for both DUT connectors.

Defined Thru (also known as **Cal Kit Thru** and **ECal Thru**)

Note: If performing an ECal calibration, this is the THRU standard in the ECal Module.

Defined Thru uses the THRU definition that is stored in the cal kit file or ECal module. The THRU standard may have worn over time, making it not as accurate as when it was new.

Defined Thru is usually more accurate than Adapter Removal, but not as accurate as Unknown Thru method.

To Define a THRU standard in a cal kit (not ECal module):

1. From the PNA Menu, click Calibration, Advanced Modify Cal Kits.
2. Select the Cal Kit
3. Click Edit Kit
4. Click Add
5. Select THRU
6. Complete the dialog box.

The next time you perform a Guided Cal, this Defined THRU standard will be available if the DUT connector types match the THRU standard.

Unknown Thru Cal

Unknown Thru Cal is the **preferred** method of calibrating the PNA to measure a non-insertable device. The Unknown Thru Cal setting is ONLY available on PNAs that have a reference receiver for each test port. See PNA models.

To perform an Unknown Thru, the PNA measures the THRU standard. The resulting data is used to improve the transmission tracking error term.

Note: Beginning with PNA code release 4.86, the measurement of calibration standards is performed in a dedicated PNA window which becomes visible during the calibration process. Previously, the 16th window was used to perform unknown thru standard measurements. This is no longer necessary.

At the end of the calibration, the PNA estimates the electrical length of the adapter. This estimate may be wrong if there are too few frequency points over the given frequency span.

You can compute and change the delay by calculating the physical length of the adapter divided by the propagation velocity.

Delay = Length / Propagation velocity

See Using ECAL for ECAL Thru calibration.

Validity of a Calibration

This section helps you determine if your calibration is valid and how the analyzer displays correction level information for your measurement.

- Viewing Correction Level
- Validating a Calibration
    - Quick Check
    - ECal Confidence Check
    - Verification Kit

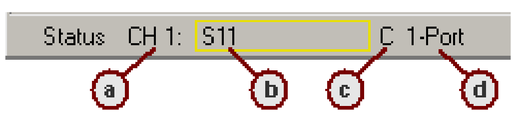
See other Calibration Topics

Viewing Correction Level

The correction level provides information about the accuracy of the active measurement. Correction level notation is displayed on the status bar for different calibration types like response, full 2-port, TRL or power calibration.

To View Correction Levels:

In the **View** menu, click **Status Bar**. The status bar appears and displays the following items:



a. Active Channel

b. Measurement parameter

c. Correction Level (see description below)

d. Calibration type

Correction Level	Status Bar Notation	Accuracy
Full	C	Highest
Interpolated	C*	Uncertain
No Correction	No Cor	Lowest

Full Correction (C)

Full Correction level is displayed immediately after a calibration is performed or when a valid calibration is recalled from memory.

If you require optimum accuracy, avoid adjusting analyzer settings after calibration so your measurement remains at this level.

Interpolated Correction (C*)

An interpolated measurement is indicated by (C*) in the status bar. Interpolation occurs when you change any of the following settings:

- Sweep time
- Frequency (subset of Start/Stop)
- IF Bandwidth
- Port power
- Stepped sweep enabled/disabled
- Number of points

To learn how measurement accuracy is affected, see Accuracy Level of Interpolated Measurements

Note: Changing frequency outside of Start/Stop will turn correction **OFF**.

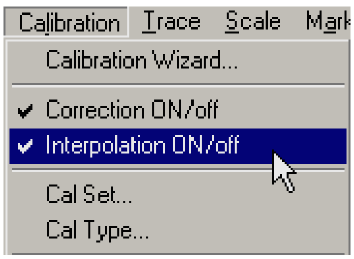
No Correction

If the frequency span is increased outside the original start/stop frequency, or sweep type is changed, the current calibration becomes invalid and error correction is turned **OFF**.

The analyzer indicates a no-correction status with **No Cor** on the status bar.

Interpolation ON/OFF

Interpolation **ON** in the calibration menu makes interpolation **available** for the active measurement. The opposite is true for interpolation **OFF**.



Validating a Calibration

At the completion of a calibration or selection of a stored Cal Set, validation can accomplish the following:

Improve Measurement Accuracy – Once a measurement calibration has been performed, its performance should be checked before making device measurements. There are several sources of error that can invalidate a calibration: bad cables, dirty or worn calibration standards that no longer behave like the modeled standards, and operator error.

Verify Accuracy of Interpolation – You should validate the calibration if you are testing a device and the measurements are uncertain because of interpolation**.** For more information see Accuracy Level of Interpolated Measurements.

Verify Accuracy of Cal Standards – To check accuracy, a device with a known magnitude and phase response should be measured.

Quick Check

For this test, all you need are a few calibration standards. The device used should not be one of the calibration standards; a measurement of one of these standards is merely a measure of repeatability. The reflection and transmission tests can be applied to all test ports.

To verify reflection measurements, perform the following steps:

1. Connect either an OPEN or SHORT standard to port 1. The magnitude of S11 should be close to 0 dB (within a few tenths of a dB).
2. Connect a load calibration standard to port 1. The magnitude of S11 should be less than the specified calibrated directivity of the analyzer (typically less than -30 dB).

To verify transmission measurements:

1. Connect a THRU cable (or known device representative of your measurement) from port 1 to port 2. Verify the loss characteristics are equivalent to the known performance of the cable or device.
2. To verify S21 isolation, connect two loads: one on port 1 and one on port 2. Measure the magnitude of S21 and verify that it is less than the specified isolation (typically less than -80 dB).

Note: To get a more accurate range of expected values for these measurements, consult the analyzer's specifications.

ECal Confidence Check

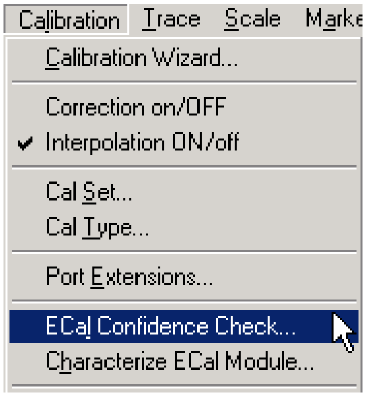
ECal Confidence Check is a method to check the accuracy of a calibration performed with mechanical standards or an ECal module. The confidence check allows you to measure an impedance state in the ECal module (called the confidence state), and compare it with factory measured data stored in the module.

In order for this test to be valid, the test ports of the ECal module must connect directly to the calibration reference plane (without adapters).

To Perform ECal Confidence Check:

1. Connect ECal module to the analyzer with the USB cable. See Connect ECal Module to the PNA. **Note:** Terminate any unused ECAL ports with a 50 ohm load.
2. Allow the module to warm up for 15 minutes (20 minutes for a four-port module) or until the module indicates **READY**.
3. To start ECal Confidence Check, press **Menu/Dialog**, tab to the **Calibration** menu, and then click **ECal Confidence Check**.

>

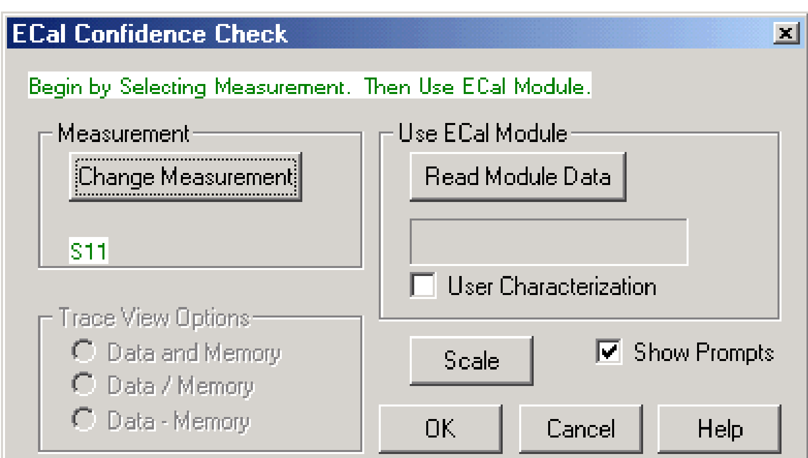



The ECal Confidence Check dialog box opens.

4. Click **Read Module Data**. The following occurs:
   - ECal module is set to "confidence state".
   - Analyzer reads and displays stored data.
   - Analyzer measures and displays "confidence state".
5. If you want to view a different parameter, select **Change Measurement** and select the check box for the desired parameter. (The default is the active channel parameter).
6. Select the viewing option in the Trace View Options block.
7. Compare the stored and measured data for each measurement parameter.

Notes:

- If the two traces show excessive difference, there may be a loose or dirty connection at the test ports or damage to the test cables. Carefully inspect the cables and connections. Then clean and gage each connector, and re-calibrate if needed.
- The User Characterization setting selects the user-characterization data instead of the factory characterization data (available when a User-Characterization is stored in the ECal module).



ECal Confidence Check dialog box help

Compares the accuracy of corrected (calibrated) data with stored data in the ECal module. For the check to be valid, the module test ports must connect directly to the calibration reference plane (without an adapter).

Measurement

Change Measurement Opens the Measure dialog box.

Use ECal Module

Read Module Data

- Copies stored data from the ECal module to Memory.
- Changes state of ECal module to confidence state.
- Measures and displays confidence state and Memory trace.

User Characterization Selects the user-characterization data (stored in the module) instead of the factory characterization data (available when a User-Characterization is stored in the ECal module).

Scale Opens the Scale dialog box.

Show Prompts Check to show a reminder for the connection (default).

Trace View Options

Data and Memory Trace Displays current measurement data and Memory trace.

Data / Memory Performs an operation where the current measurement data is divided by the data in memory.

Data + Memory Performs an operation where the current measurement data is added to the data in memory.

Verification Kit

Measuring known devices, other than calibration standards, is a straightforward way of verifying that the network analyzer system is operating properly. Verification kits use accurately known verification standards with well-defined magnitude and phase response. These kits include precision airlines, mismatch airlines, and precision fixed attenuators. Traceable measurement data is shipped with each kit on disk and verification kits may be re-certified by Agilent.

See Analyzer Accessories for a list of Agilent verification kits.

Measurement Errors

You can improve accuracy by knowing how errors occur and how to correct for them. This topic discusses the sources of measurement error and how to monitor error terms.

- **Drift Errors**
- **Random Errors**
- **Systematic Errors**
- **Monitoring Error Terms**

See other Calibration Topics

Drift Errors

- Drift errors are due to the instrument or test-system performance changing after a calibration has been done.
- Drift errors are primarily caused by thermal expansion characteristics of interconnecting cables within the test set and conversion stability of the microwave frequency converter and can be removed by re-calibrating.
- The time frame over which a calibration remains accurate is dependent on the rate of drift that the test system undergoes in your test environment.

Providing a stable ambient temperature usually minimizes drift. For more information, see Measurement Stability.

Random Errors

Random errors are not predictable and cannot be removed through error correction. However, there are things that can be done to minimize their impact on measurement accuracy. The following explains the three main sources of random errors.

Instrument Noise Errors

Noise is unwanted electrical disturbances generated in the components of the analyzer. These disturbances include:

- Low level noise due to the broadband noise floor of the receiver.
- High level noise or jitter of the trace data due to the noise floor and the phase noise of the LO source inside the test set.

You can reduce noise errors by doing one or more of the following:

- Increase the source power to the device being measured.
- Narrow the IF bandwidth.
- Apply several measurement sweep averages.

Switch Repeatability Errors

- Mechanical RF switches are used in the analyzer to switch the source attenuator settings.
- Sometimes when mechanical RF switches are activated, the contacts close differently from when they were previously activated. When this occurs inside of the analyzer, it can adversely affect the accuracy of a measurement.
- You can reduce the effects of switch repeatability errors by avoiding switching attenuator settings during a critical measurement.

Connector Repeatability Errors

Connector wear causes changes in electrical performance. You can reduce connector repeatability errors by practicing good connector care methods. See Connector Care.

Systematic Errors

Systematic errors are caused by imperfections in the analyzer and test setup.

- They are repeatable (and therefore predictable), and are assumed to be time invariant.
- They can be characterized during the calibration process and mathematically reduced during measurements.
- They are never completely removed. There are always some residual errors due to limitations in the calibration process. The residual (after measurement calibration) systematic errors result from:
    - imperfections in the calibration standards
    - connector interface
    - interconnecting cables
    - instrumentation

Reflection measurements generate the following three systematic errors:

- **Directivity**
- **Source Match**
- **Frequency Response Reflection Tracking**

Transmission measurements generate the following three systematic errors:

- **Isolation**
- **Load Match**
- **Frequency Response Transmission Tracking**

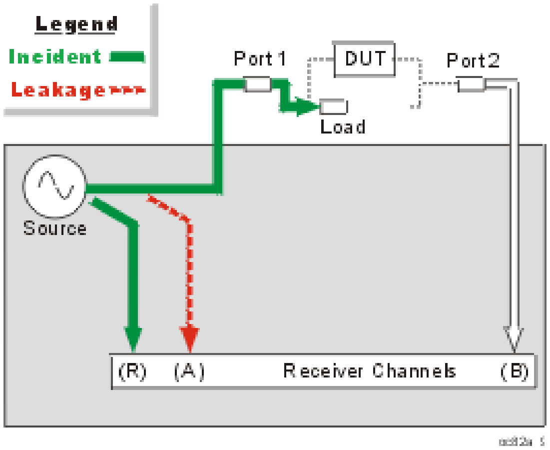
Note: The figures for the following six systematic errors show the relevant hardware configured for a forward measurement. For reverse measurements, internal switching in the analyzer makes Port 2 the source and Port 1 the receiver. Channel 'A' becomes the transmitted receiver, channel 'B' becomes the reflected receiver, and channel 'R2' becomes the reference receiver. These six systematic errors, times two directions, results in 12 systematic errors for a two port device.

Directivity Error

All network analyzers make reflection measurements using directional couplers or bridges.

With an ideal coupler, only the reflected signal from the DUT appears at channel 'A'. In reality, a small amount of incident signal leaks through the forward path of the coupler and into channel 'A'.

This measurement error is called directivity error. It can be measured and reduced by the analyzer.



How the Analyzer Measures and Reduces Directivity Error.

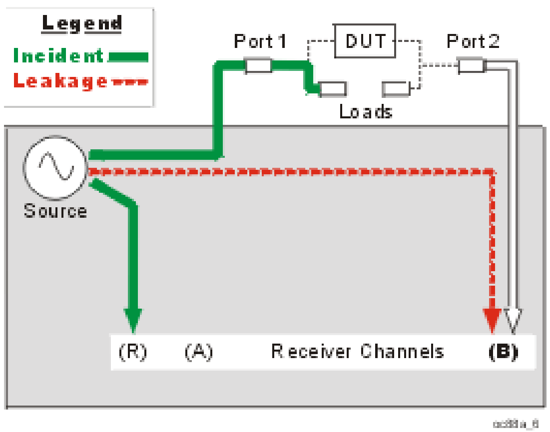
1. During calibration, a load standard is connected to Port 1. We assume no reflections from the load.
2. The signal measured at channel 'A' results from the incident signal leakage through the coupler.
3. Directivity error is subtracted from subsequent reflection measurements.

Isolation Error

Ideally, only signal transmitted through the DUT is measured at channel 'B'.

In reality, a small amount of signal leaks into the channel 'B' receiver through various paths in the analyzer.

The signal leakage, also known as crosstalk, is isolation error which can be measured and reduced by the analyzer.



How the Analyzer Measures and Reduces Isolation Error

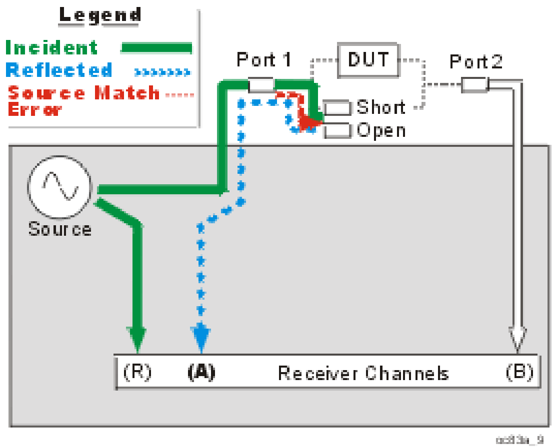
1. During calibration, load standards are connected to both Port 1 and Port 2.
2. The signal measured at channel 'B' is leakage through various paths in the analyzer.
3. This isolation error is subtracted from subsequent transmission measurements.

Source Match Error

Ideally in reflection measurements, all of the signal reflected off of the DUT is measured at channel 'A'.

In reality, some of the signal that is reflected off of the DUT is reflected again off of Port 1 and is not measured at channel 'A'.

This measurement error is called source match error which can be measured and reduced by the analyzer.



How the Analyzer Measures and Reduces Source Match Error

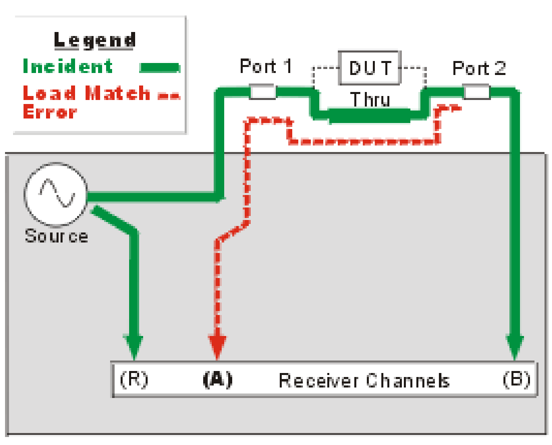
1. During calibration, a short standard is connected to Port 1. Known reflection from the short is measured at channel 'A' and stored in the analyzer.
2. An open standard is connected to Port 1. Known reflection from the open is measured at channel 'A' and stored in the analyzer.
3. The analyzer compares the two stored signals. The difference between the two signals is due to source match error.
4. Source match error is subtracted from subsequent reflection and transmission measurements.

Load Match Error

Ideally in transmission measurements, an incident signal is transmitted through the DUT and is measured at channel 'B'.

In reality, some of the signal is reflected off of Port 2 and is not measured at channel 'B'.

This measurement error is called load match error which can be measured and reduced by the analyzer.



How the Analyzer Measures and Reduces Load Match Error

1. The Port 1 and Port 2 test connectors are mated together for a perfect zero-length thru connection. (If this is not possible, a thru-line adapter is inserted.) This allows a known amount of incident signal at Port 2.
2. The signal measured at channel 'A' is reflection signal off of Port 2
3. The resulting load match error is subtracted from subsequent transmission and reflection measurements.

Frequency Response Reflection Tracking Error

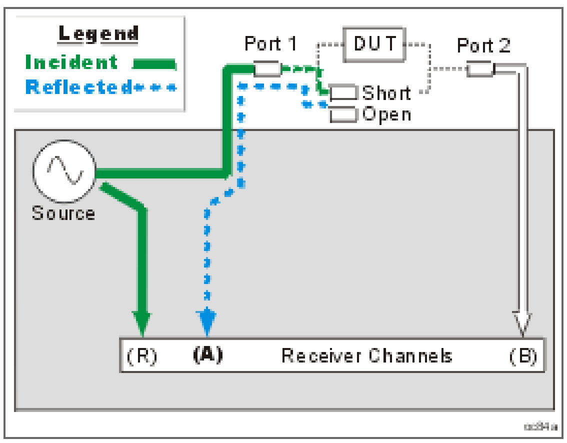
Reflection measurements are made by comparing signal at channel 'A' (the reflection channel) to signal at channel 'R1' (the reference channel). This is called a ratio measurement or "A over R1" (A/R1).

For ideal reflection measurements, the frequency response of the channel 'A' and channel 'R1' receivers would be identical.

In reality, they are not, causing a frequency response reflection tracking error. This is the vector sum of all test variations in which magnitude and phase change as a function of frequency. This includes variations contributed by:

- signal-separation devices
- test cables
- adapters
- variations between the reference and test signal paths

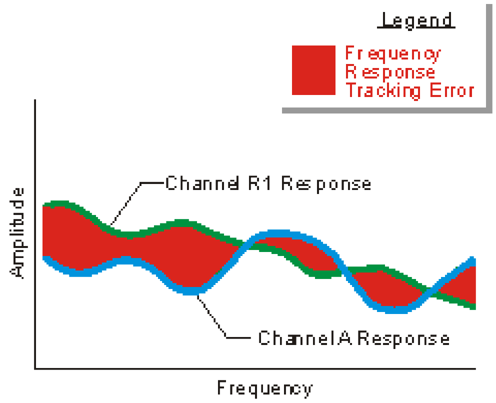
Frequency response reflection tracking error can be measured and reduced by the analyzer.



How the Analyzer Measures and Reduces Frequency Response Reflection Tracking Error.

1. During calibration, a short standard is connected to Port 1. Known reflection from the short is measured at channel 'A' and stored in the analyzer.
2. An open standard is connected to Port 1. Known reflection from the open is measured at channel 'A' and stored in the analyzer.
3. The analyzer averages the two stored channel 'A' signals.
4. The average channel 'A' response is compared with the reference signal measured at 'R1'.
5. The result is the difference in frequency response of the channel 'A' and channel 'R1' receivers (see the following diagram). This frequency response reflection tracking error is subtracted from subsequent DUT

measurements.



Note: In reflection response calibrations, only a single calibration standard is measured (open or short) and thus only its contribution to the error correction is used.

Frequency Response Transmission Tracking Error

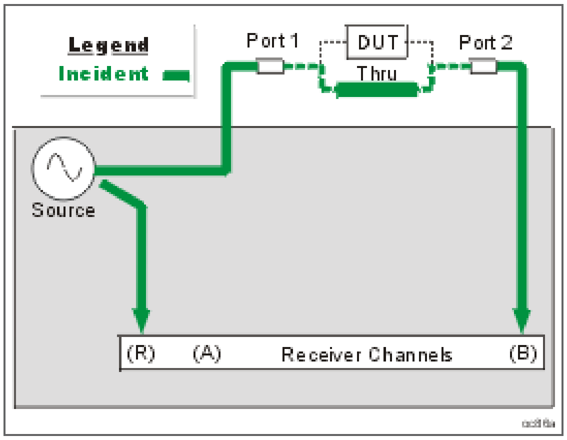
Transmission measurements are made by comparing signal at channel 'B' (the transmission channel) to signal at channel 'R1' (the reference channel). This is called a ratio measurement or "B over R1" (B/R1).

For ideal transmission measurements, the frequency response of the channel 'B' and channel 'R1' receivers would be identical.

In reality, they are not, causing a frequency response transmission tracking error. This is the vector sum of all test variations in which magnitude and phase change as a function of frequency. This includes variations contributed by:

- signal-separation devices
- test cables
- adapters
- variations between the reference and test signal paths

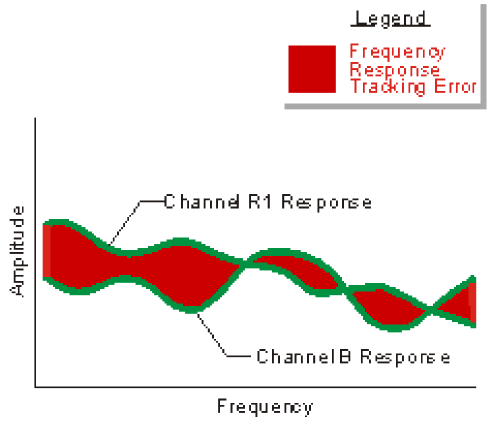
Frequency response transmission tracking error can be measured and reduced by the analyzer.



How the Analyzer Measures and Reduces Frequency Response Transmission Tracking Error.

1. During calibration, the Port 1 and Port 2 test connectors are mated together for a perfect zero-length thru connection. (If this is not possible, a thru-line adapter is inserted.) This allows a known amount of incident signal to reach Port 2.

2. Measurements are made at channel 'B' and channel 'R1' and stored.
3. The analyzer compares the channel 'R1' and channel 'B' signals.
4. The result is the difference in frequency response of the channel 'B' and channel 'R1' receivers (see the following diagram). This frequency response transmission tracking error is subtracted from subsequent device measurements.



Monitoring Error Terms using Cal Set Viewer

The PNA calculates error terms by measuring well-defined calibration devices over the frequency range of interest. It then compares the measured data with an ideal model for each device. The differences of the compared data represent systematic (repeatable) errors of the analyzer system. These differences are stored as error terms and removed from subsequent measurements when error correction is on. Learn about these error terms.

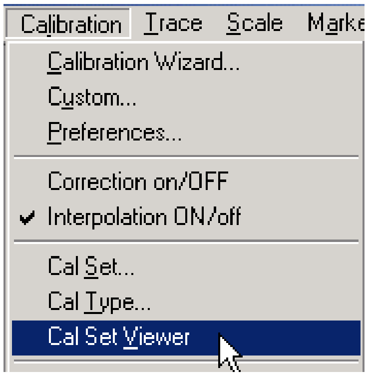
You can use **Cal Set Viewer** to monitor the measured data and the calculated error term. This will help to determine the health of your PNA and the accuracy of your measurements.

By printing or saving the error terms, you can periodically compare current error terms with previously recorded error terms that have been generated by the same PNA, measurement setup, and calibration kit. If previously generated values are not available, refer to Typical Error Term Data in Appendix A, "Error Terms", of the Service Guide.

Note: The service guide for your PNA is available at http://www.agilent.com/find/pna
It is also on the CDROM that was shipped with your PNA.

- A stable system should generate repeatable error terms over about six months.
- A sudden shift in error terms over the same frequency range, power, and receiver settings, may indicate the need for troubleshooting system components. For information on troubleshooting error terms, see Appendix A , "Error Terms", of the Service Guide.
- A subtle, long-term shift in error terms often reflects drift or connector and cable wear. The cure is often as simple as cleaning and gauging connectors or inspecting cables.

How to monitor Error Terms



Use the down arrow to select a Cal Set

Click either

- **Error Terms -** calculated data
- **Standards** - the raw measurement data of the Standard

Use the down arrow to select the desired error term or standard of interest

Select the **Enable** check box to view the data on the PNA screen.

Learn more about using the front panel interface

Viewing Cal Set Data

- Existing measurement traces are unaffected by the Cal Set Viewer.
- The Cal Set data trace is presented in the highest unused channel number (usually 32) in the active window.
- The Cal Set data trace is labeled as S11 in the status bar regardless of the type of error term or standard.
- Only one Cal Set error term or standard data can be viewed at a time. However, you can store a data trace into memory and compare it with the same error term from another Cal Set.
- ECAL data does not include standard data, only error terms.

Automated Retrieval of Error Terms

Refer to the Cal Tab of Command Finder to see the SCPI and COM commands that retrieve and store error terms.

Modify Calibration Kits

You can create or modify calibration kit files using Advanced Modify Cal Kits.

About Modifying Calibration Kits

Creating a New Cal Kit from an Existing Cal Kit

Creating Custom Calibration Kits using a New Connector Family

How to Modify Cal Kits

See other Calibration Topics

About Modifying Calibration Kits

You can modify calibration kit files or create a custom one.

Note: You CAN NOW modify Data-based Cal Kits. Learn more.

For most applications, the default calibration kit models provide sufficient accuracy for your calibration. However, several situations exist that may require you to create a custom calibration kit:

- Using a connector interface different from those used in the predefined calibration kit models.
- Using standards (or combinations of standards) that are different from the predefined calibration kits. For example, using three offset SHORTs instead of an OPEN, SHORT, and LOAD to perform a 1-port calibration.
- Improving the accuracy of the models for predefined kits. When the model describes the actual performance of the standard, the calibration is more accurate. (Example: A 7 mm LOAD is determined to be 50.4O instead of 50.0O.)
- Modifying the THRU definition when performing a calibration for a non-insertable device.
- Performing a TRL calibration.

Creating a New Cal Kit from an Existing Cal Kit

You can create a new custom Cal Kit using a copy of an existing Cal Kit as a starting point. Here is how:

1. From the Edit PNA Cal Kits dialog, click **Import Kit** to load the Cal Kit you want to use as a starting point. A "Duplicate Name..." message appears. Click **OK** to load a duplicate copy of the Cal kit into the last position of the Edit PNA Cal Kits dialog.
2. Select the imported kit.
3. Click **Edit Kit**, then change the Cal Kit Name and Description.
4. Click **Installed Kits - Save As** to save the new Cal Kit to a .ckt file.
5. Recommended: Also click **Edit PNA Cal Kits - Save As** to save the entire collection of Cal Kits to a .wks file.
6. If using a new or modified connector, click Change Family to change the connector family.

7. Click **Add or Edit** to change connector descriptions and parameters.
8. Make modifications to your new custom Cal Kit as required. Save your work by clicking **Installed Kits - Save As**

Creating Custom Calibration Kits using a New Connector Family

To create a custom calibration kit that uses a new connector type, you must first define the connector family. The connector family is the name of the connector-type of the calibration kit, such as:

- APC7
- 2.4 mm
- Type-N (50O)

Although more than one connector family is allowed, it is best to limit each calibration kit to only one connector family.

If you are using a connector family that has male and female connectors, include definitions of both genders. If you are using a family with no gender, such as APC7, only one connector definition is required.

Use the following steps to create a custom calibration kit:

1. In the Edit PNA Cal Kits dialog box, click **Insert New** to add the new connector family.
2. In the Edit Kit dialog box:
    - Type the Kit Description for the custom cal kit.
    - Click **Add** in the Connectors section of the dialog box.
3. In the Add Connector dialog box:
    - Type a Connector Family name.
    - Type a Description of the connector.
    - Select the Gender of one of the connectors.
    - Type the minimum and maximum Frequency Range.
    - Type the Impedance.
    - Click the down-arrow to select the Media.
    - Type the cut-off frequency.
    - Click **Apply**.
    - Click **OK**.
4. If you need to add another connector gender, in the Edit Kit dialog box :
    - Click **Add** in the Connectors section again for the next connector gender.
5. If you are adding another connector gender, repeat step 3.

Note: If you have male and female versions of the connector family, you probably do NOT also have a NO GENDER version.

6. Now that the connector family is added to the custom cal kit, you are ready to add new calibration standards. In the Edit Kit dialog box:

    - Under the list of standards, click **Add**.

7. In the Add Standard dialog box:

    - Select the type of standard (OPEN, SHORT, LOAD, or THRU).
    - Click **OK**.

8. In the Edit/Add Standards dialog box:

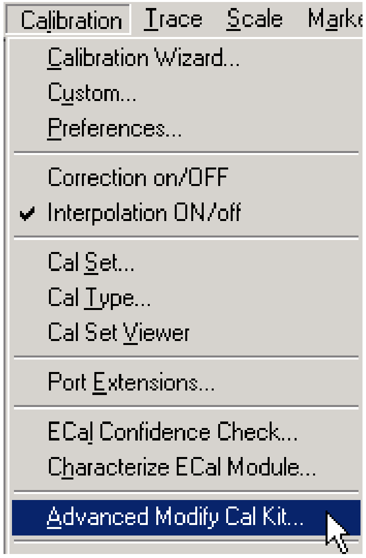
    - Complete the information in the dialog box for the standard you selected. Note that for banded standards, the start and stop frequency may be different than the frequency range of the specified connector. Edit the start and stop frequencies as needed.
    - Click **OK** when all the settings are correct.

9. Repeat steps 6 - 8, as necessary, to add all standards and definitions to the new custom cal kit.
10. Assign each of the standards to a calibration class. This is done through the Modify Calibration Class Assignment dialog box.
11. Click **File**, **PrintToFile**. PrintToFile will generate a .prn file (ascii file with comma delimiters) that can be imported into a spreadsheet.
12. Import the .prn file into an application such as Microsoft Excel, and print the results.
13. Use the spreadsheet to verify that each standard in the kit belongs to the same connector family and the gender of each standard is properly specified. It is important that the connectors and genders for your standards are correctly defined and verified in order for your SmartCal (guided calibrations) to work properly.

How to Modify Cal Kits

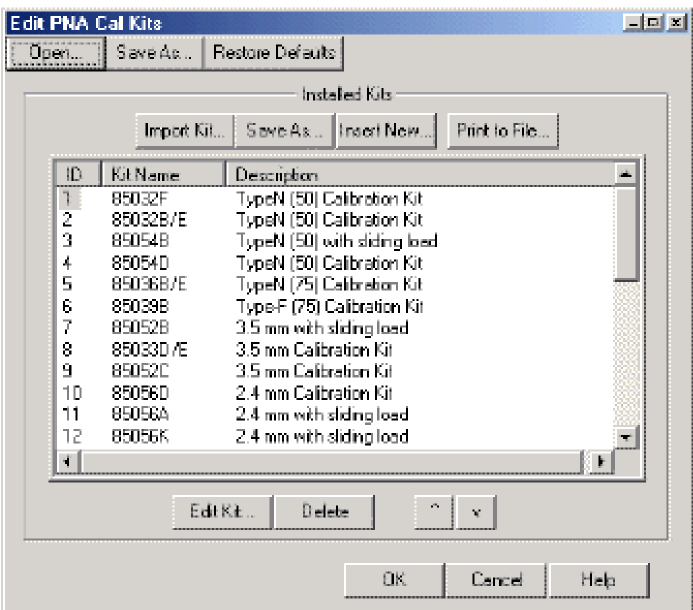
The series of dialog boxes that follow allow you to modify the standard definitions or class assignments of calibration kit files.

How to start Advanced Modify Cal Kit



Learn more about using the front panel interface

Edit PNA Cal Kits



Edit PNA Cal Kits dialog box help

Provides access to all Agilent cal kits and allows modification of their standard definitions.

PNA Cal Kits and Firmware Upgrades

- The default "factory" cal kits are overwritten when new firmware is installed. Your custom cal kits (files with custom filenames) are NOT overwritten. However, the custom cal kits must be imported (click **Import Kit**) into the new firmware.
- All PNA cal kits can only be imported by the current firmware revision and later. They can NOT be imported by PAST firmware revisions. Once a Cal Kit has been imported by a later firmware revision, it cannot be imported by the previous version of firmware from which it originated.
- When a firmware upgrade takes place, ALL cal kits, both factory and custom, that are present on the PNA are saved to a single *.wks file using a unique filename. These files are NOT Excel spreadsheet files. They are opened using the **Open** button (see below). They can be used as archives of cal kits from previous firmware versions.

Open Opens an archive of cal kits from past firmware upgrades and 'Save As' operations.

Save As Saves ALL cal kits in the PNA to a *.wks file.

Restore Defaults Re-installs the default factory contents of all Agilent cal kits from the PNA hard drive. The factory Agilent cal kits are stored on the PNA hard drive at C:\Program Files\Agilent\Network Analyzer\PnaCalKits\factory.

Installed Kits

Import Kit Invokes the Import Kit dialog box.

Save As Saves the selected calibration kit and definitions (using **.ckt** file type).

Insert New Invokes a blank Edit Kit dialog box to create new calibration kit definitions.

Print to File Prints the contents of the selected cal kit to a .prn file

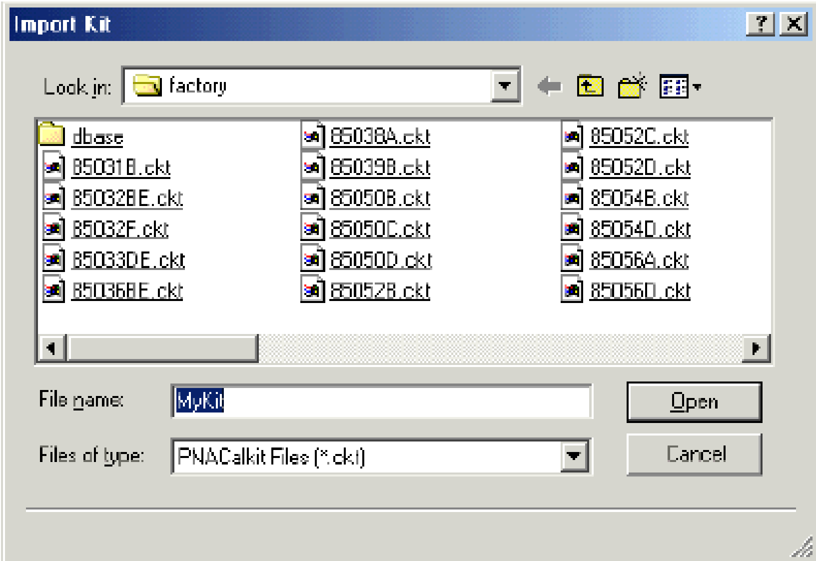
Edit Kit... Invokes the Edit Kit dialog box to modify selected calibration kit definitions.

Note: You CAN NOW modify Data-based Cal Kits. Learn more.

Delete Deletes selected calibration kit file.

^ Selects previous / next calibration kit in list.

For more information see Creating Custom Calibration Kits using a New Connector Family.



Import Kit dialog box help

Imports calibration kit definitions from hard disk or other drive that are saved in the various formats. With PNA version 4.0 or later, four kit types can be imported.

Note: See PNA Cal Kits and Firmware Upgrades

Files of type Select the file type of your Cal Kit

Cal Kit Format	File Type
Current PNA Series Cal Kit	*.ckt
Old PNA Series Cal Kit (Version 1)	*.ck1
8510 Cal Kit	CK_*
8753, 8752, 8719, 8720, or 8722 Cal Kit	*.ck

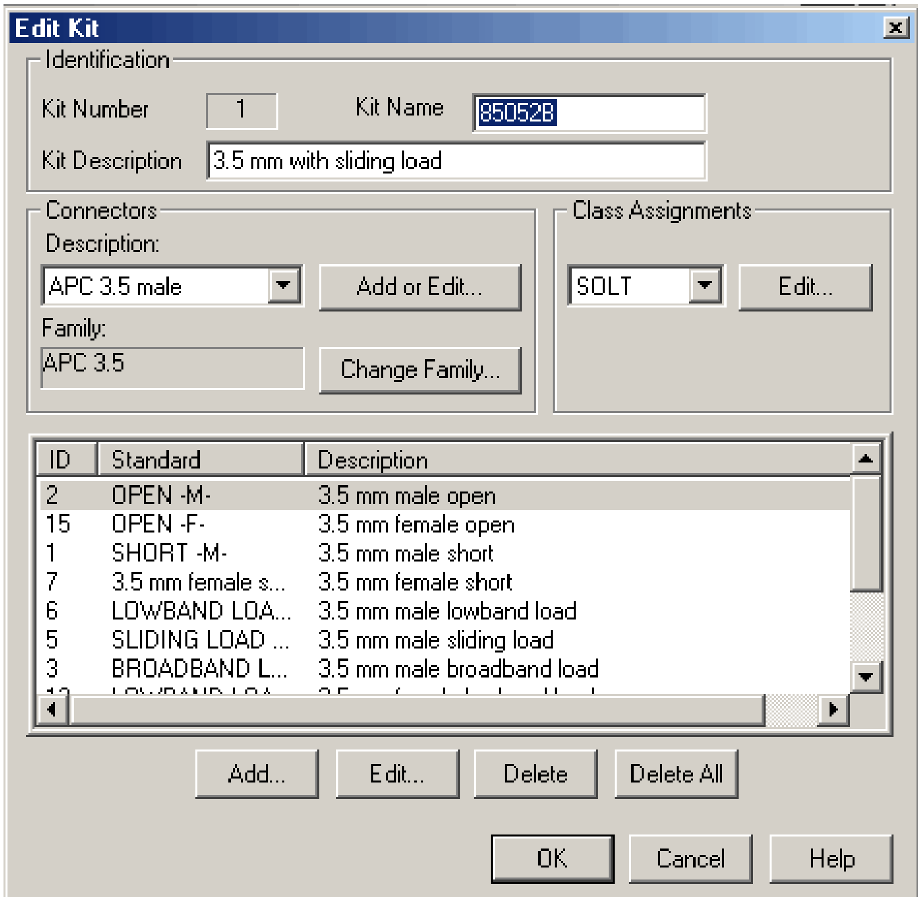
File name Navigate and select your cal kit file.

Open Imports the selected file. The kit is added at the end of the list of cal kits.

Importing Kits other than Current PNA Series Kits

Cal kit files from Agilent "legacy" network analyzers (listed above) may not contain information that the PNA requires. Therefore, the PNA may modify the cal kit name and description, the cal standards, and the cal class assignments in a “best effort” manner. You may need to correct these modifications after importing your “legacy” cal kit to meet your specific requirements.

- "Legacy" cal kit files are based on the analyzer test port sex; PNA cal kits are based on the Device Under Test (DUT) connector sex. Therefore, when the kit is imported the standard's label and description are reversed and are noted as –F- (female) and –M- (male) .
- When a Coaxial standard is detected in the kit file, a pair of male/female connectors is typically created.
- Waveguide standards that are created as “connector” have no gender.



Edit Kit dialog box help

Identification

Kit Number Number of the selected calibration kit.

Kit Name Allows you to change the Name of the selected calibration kit.

Kit Description Allows you to change the description of the selected calibration kit.

Connectors

Note: You can NOT use a connector with a new or modified name to perform an ECal User Characterization.

Click the down arrow to change the connector type.

Add or Edit Invokes the Add or Edit Connector dialog box which allows you to add new connector type to the calibration kit or edit the connector properties.

Change Family Invokes the Change Connector Family dialog box which allows you to rename the entire connector family name.

Class Assignments

Click the down arrow to change the Class Assignment.

Edit Invokes the Modify Calibration Class Assignments dialog box.

Standards in Kit

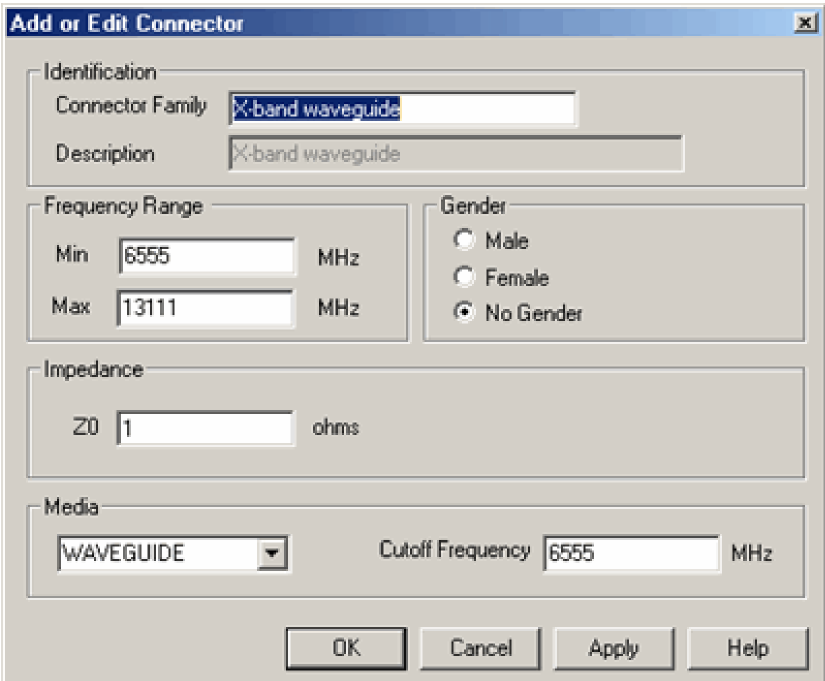
Lists the current standards and descriptions in the cal kit.

Add... Invokes the Add Standard dialog box that allows you to add definitions for a standard.

Edit... Invokes the Edit dialog box that allows you to modify standard definitions for the selected standard:

either Open, Short, Load, or Thru.

Delete Deletes selected standard from calibration kit.



Add or Edit Connector dialog box help

Identification

Note: You can NOT use a connector with a new or modified name to perform an ECal User Characterization.

Connector Family Allows you to Add or Edit a specific connector name. If you change Connector Family to a unique name, the name and selected Gender is ADDED to the list of connectors in that kit.

Note: To change the Connector Family Name of all connectors in the Kit, click Change Family on the previous dialog box.

Description Displays connector type and gender.

Frequency Range

Min Allows you to define the lowest frequency at which the standard is used for calibration.

Max Allows you to define the highest frequency at which the standard is used for calibration.

Gender

Allows you to define the connector gender.

Impedance

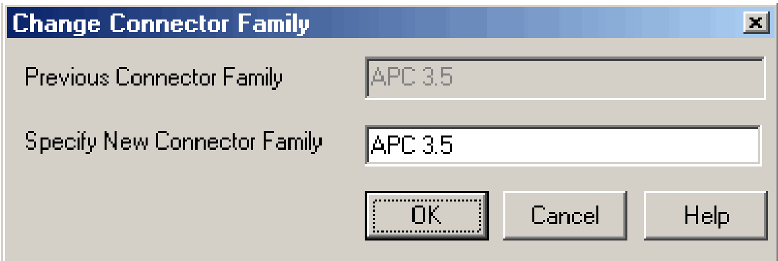
Allows you to define the impedance of the standard.

Media

Allows you to define the medium (or 'geometry') of the connector (COAX or WAVEGUIDE).

Cutoff Frequency If Media is Waveguide, type the low-end cutoff frequency.

See Also: Creating a New Cal Kit from an existing Cal Kit



Change Connector Family dialog box help

Note: You can NOT use a connector with a new or modified name to perform an ECal User Characterization.

Performs a text "Search and Replace" function. Within the description field of each of the standards of the current Cal Kit, it searches for the Previous Connector Name and replaces it with the New Connector Name.

Specify New Connector Name Allows you to replace the primary connector-family name from the selected kit with the new connector-family name. The PNA allows multiple connector-families per kit.

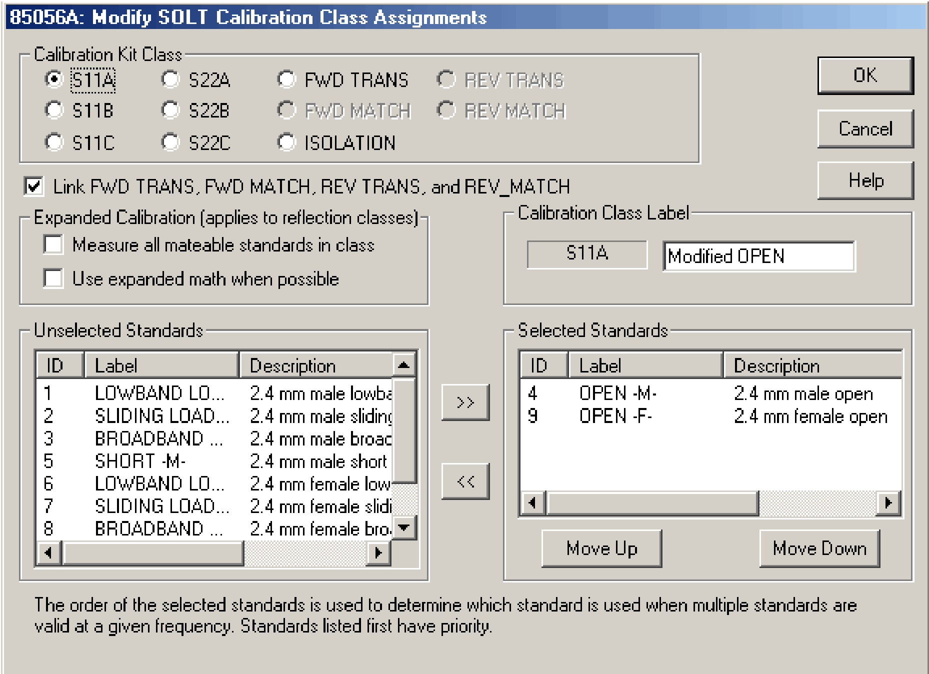
Previous Connector Name Displays the primary connector-family name. All occurrences of the previous connector name will be replaced throughout calibration dialog boxes. This includes calibration kit labels and description fields.

Notes:

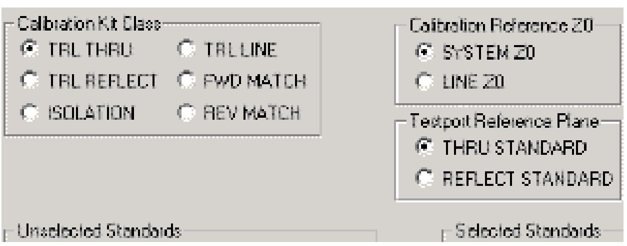
- String replacement requires an exact match and is case sensitive. For example, "Type N" does not match "type N", and "apc 7" does not match "APC 7".
- Some calibration kits may include connector names that do not match strings within labels or description fields. You may reuse the Change Connector Name dialog to standardize the name within the kit, and then to replace the standard name with the new name.
  Example:
  Select the 85056A calibration kit. The default connector-family name is "APC 2.4". However, many standard description files are labeled "2.4 mm". You may want to replace the connector family name with a new name and update the standard descriptions to match the new name. For this kit, use a two step procedure.

  1. Use the Change Connector Name dialog to replace "APC 2.4" with "2.4 mm".
  2. Use the Change Connector Name dialog to replace "2.4 mm" with the new name, "PSC 2.4 mm".

See Also Creating a New Cal Kit from an existing Cal Kit



If TRL is selected as Class Assignment in the Edit Kit dialog box, the following changes appear:



Modify Calibration Class Assignments dialog box help

Allows you to assign single or multiple standards to classes. To Add or Edit standards, click Calibration then, click Advanced Modify Cal Kit.

To assign a standard to a calibration class:

1. Select the **Calibration Class**
2. Select the standard from the **Unselected Standards** field
3. Click the right arrow to move the standard to the **Selected Standards** field.

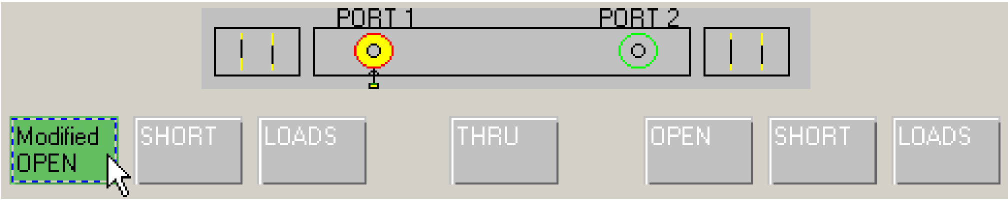
Notes:

- During an Unguided Cal all of the **Selected Standards** are presented. You then choose which of these standards to measure.
- The MATCH standards must be assigned to the FWD MATCH, REV MATCH, and LINE classes. See TRL calibrations to learn more about TRL standards.
- Use MOVE UP and MOVE DOWN to change the **ORDER** of the standard. The order is used during a Guided Cal to determine overlap priorities when:
  - **Multiple standards are valid for a frequency** - standards are presented in the order in which they appear.

- **Using two sets of standards** - modify the order in which standards appear to reflect the configuration of your DUT. For example, for a DUT with a male connector on port 1 and a female connector on port 2, order the devices within the S11 classes (A, B, and C) such that the MALE standards are first in the list. Then order the S22 classes specifying the FEMALE standards as the first in the list.

Calibration Class Label

The label that appears on the Unguided Cal - Measure Mechanical Standards dialog box. For example, the Calibration Class Label "**Modified OPEN"** would yield the following prompt:



The following selections in this dialog box depend on your Class Assignment selection (**SOLT** or **TRL**) in the Edit Kit dialog box.

SOLT ONLY

Link FWD TRANS, FWD MATCH, REV TRANS, and REV MATCH Check to automatically assign the standard definition for FWD TRANS to FWD MATCH, REV MATCH, and REV TRANS. Clear to separately assign FWD MATCH, REV MATCH and REV TRANS classes (SOLT calibrations only).

Expanded Calibration

The following two check boxes **apply ONLY during Guided Calibrations**. For Unguided Calibration, these check boxes are ignored, including the case where the multiple standards dialog box is presented.

Note: You can make these selections during the calibration process with Preview/Modify settings selected from within the Cal Wizard or by modifying the cal kit from Advanced Modify Cal Kit.

Measure all mateable standards in class Check this box to attain the very highest accuracy possible. For example, if a cal kit contains several load standards, during the calibration process you will be prompted to measure each of the standards. This could require a significant amount of calibration time. When checked, the "Use expanded math when possible" box is also checked automatically.

Use expanded math when possible Some kits contain multiple calibration standards of the same type that together cover a very wide frequency range. (For example: multiple shorts, or a lowband load and a sliding load.) If a calibration requires more than one standard to cover the calibration frequency range, there can be regions of overlapping measurements. When this checkbox is selected, the PNA automatically computes the most accurate measurement in the overlap regions using a "weighted least squares fit" algorithm. This function improves accuracy without slowing the calibration speed.

- Manually select this checkbox only when using a cal kit that contains multiple standards of the same type. (For example: multiple shorts, or a lowband load and a sliding load.)
- The checkbox is cleared by default when a polynomial model is selected from the cal kit menu.
- The checkbox is selected by default when the 85058B or 85058E data-based model is selected from the cal kit menu.

TRL ONLY

The MATCH standards must be assigned to the FWD MATCH, REV MATCH, and TRL LINE classes. See TRL calibrations to learn more about TRL standards.

LRL line auto characterization

Note: This setting ONLY applies if an LRL Cal Kit is being modified **AND** Testport Reference Plane is set to Thru Standard **AND** the TRL Thru class standard and the TRL Line/Match class standard both have the same values for Offset Z0 and Loss. Otherwise, this setting is ignored.

- Check the box to allow the PNA to automatically correct for line loss and dispersion characteristics.
- Clear the box if anomalies appear during a calibrated measurement which may indicate different loss and impedance values for the Line standards.

Calibration Reference Z0 (TRL only)

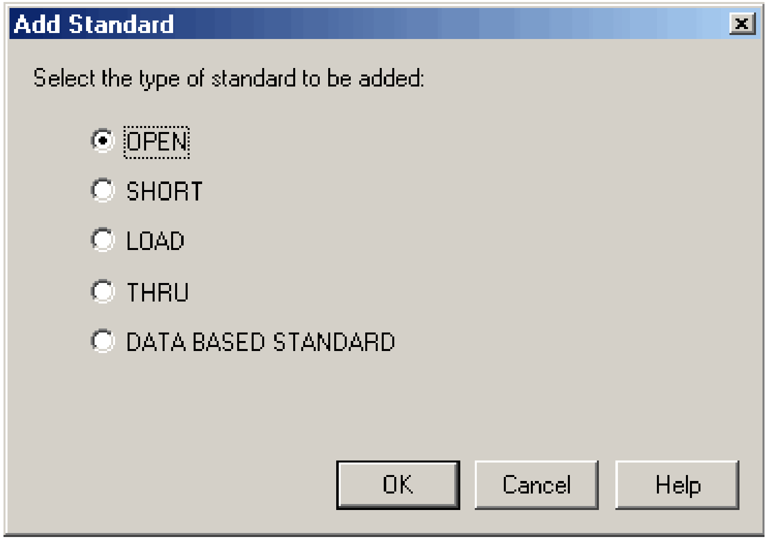
System Z0 The system impedance is used as the reference impedance. Choose when the desired test port impedance differs from the impedance of the LINE standard. Also, choose when skin effect impedance correction is desired for coax lines.

Line Z0 The impedance of the line standard is used as the reference impedance, or center of the Smith Chart. Any reflection from the line standard is assumed to be part of the directivity error.

Testport Reference Plane (TRL only)

Thru Standard The THRU standard definition is used to establish the measurement reference plane. Select if the THRU standard is zero-length or very short.

Reflect Standard The REFLECT standard definition is used to establish the position of the measurement reference plane. Select if the THRU standard is not appropriate AND the delay of the REFLECT standard is well defined.



Add Standard dialog box help

Allows you to add standards to the calibration kit file.

OPEN Adds an open to the calibration kit file.

SHORT Adds a short to the calibration kit file.

LOAD Adds a load to the calibration kit file.

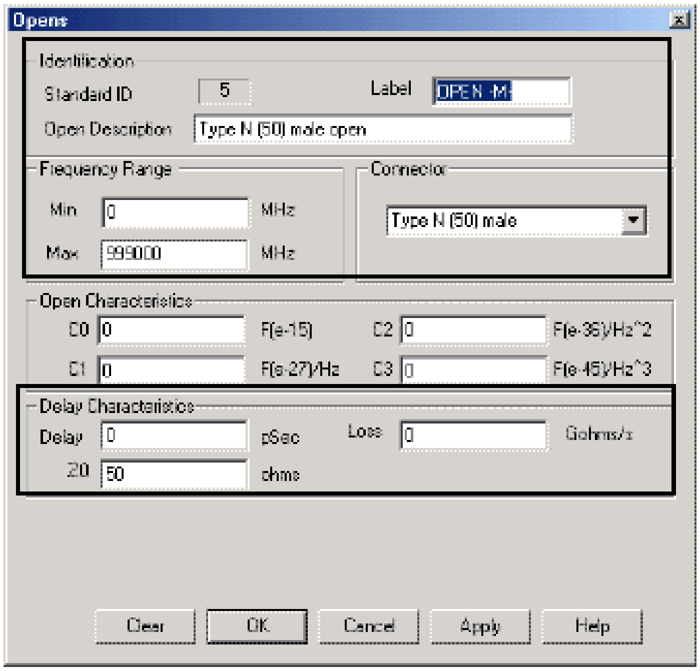
THRU Adds a thru to the calibration kit file.

DATA BASED STANDARD Adds a data-based standard to the calibration kit file.

OK Invokes a blank Edit Standards: Open, Short, Load, or Thru dialog box.

For more information see Creating Custom Calibration Kits using a New Connector Family.

Edit / Add Standards (Open, Short, Load, or Thru)



Edit / Add Standards dialog box help

This dialog box changes depending on the type of standard selected.

The boxed areas of the previous graphic applies to all standard types:

Identification

Standard ID Number in list of standards

Label Type of standard.

Description Description of standard.

Frequency Range

Min Defines the lowest frequency at which the standard is used for calibration.

Max Defines the highest frequency at which the standard is used for calibration.

Connector

Indicates the type and gender (Male, Female, None) of the standard.

Delay Characteristics

Delay Defines the one-way travel time from the calibration plane to the standard in seconds.

Z0 Defines the impedance of the standard.

Loss Defines energy loss, due to skin effect, along a one-way length of coaxial cable.

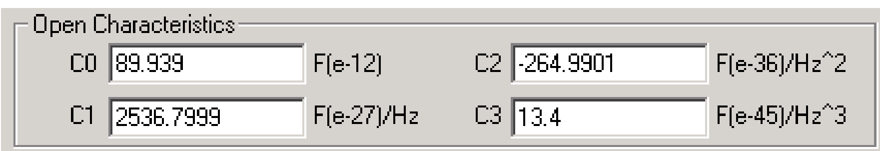
• The value of loss is entered as ohms/second at 1 GHz.

• To compute the loss of the standard, measure the delay in seconds and the loss in dB at 1 GHz. Then use the following formula:

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

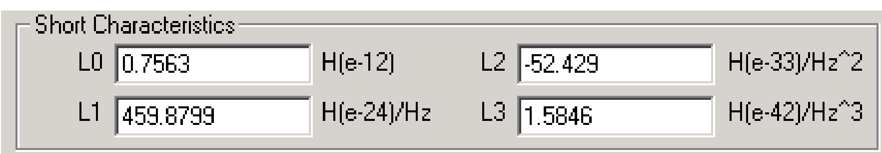
The following applies to standard types <u>Open</u>, <u>Short</u>, <u>Load</u>, <u>Thru</u>, and <u>Data-based</u>

Open Standard



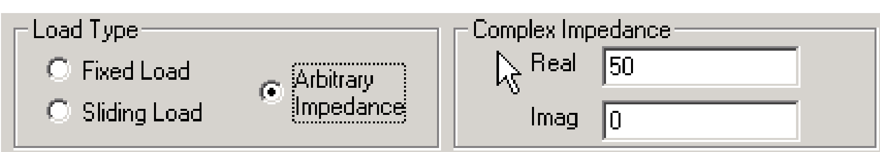
C0, C1, C2, C3 Specifies the fringing capacitance.

Short Standard



L0, L1, L2, L3 Specifies the residual inductance.

Load Standard



Allows you to select the type of load.

Load Type

Fixed Load Specifies the load type as Fixed

Sliding Load Specifies the load type as Sliding

Arbitrary Impedance Specifies the load type to be have an impedance value different from system Z0. Only when Arbitrary Impedance is selected are the following settings available:

- **Real** The real portion of the impedance value.

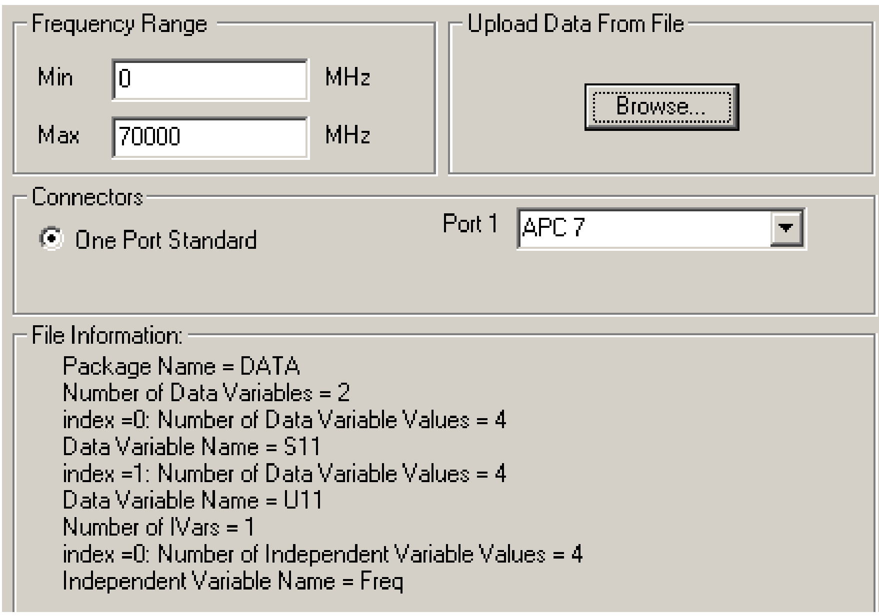
- **Imaginary** The imaginary portion of the impedance value.

Thru Standard

Connectors

Defines connector type at both ports.

Data-Based Standard



Note: To learn how to modify data-based standard files, visit http://na.tm.agilent.com/pna/dbcal.html

The modified file can then be uploaded into the PNA.

Upload Data From File

Click Browse to load data from an existing Data-Based Standard

Connectors

One Port Standard Currently only 1-port standards can be modified.

Port 1 Select the type of connector.

File Information Information about the standard that is read from the uploaded file.

3-Port Measurements

PNA Series network analyzers, N3381A, N3382A, and N3383A, provide true vector error-corrected measurements for three port devices. This topic provides information about making 3-Port measurements.

- **Overview**
- **3-Port Calibrations**
- **Selecting Parameters**

See other Calibration Topics

Overview

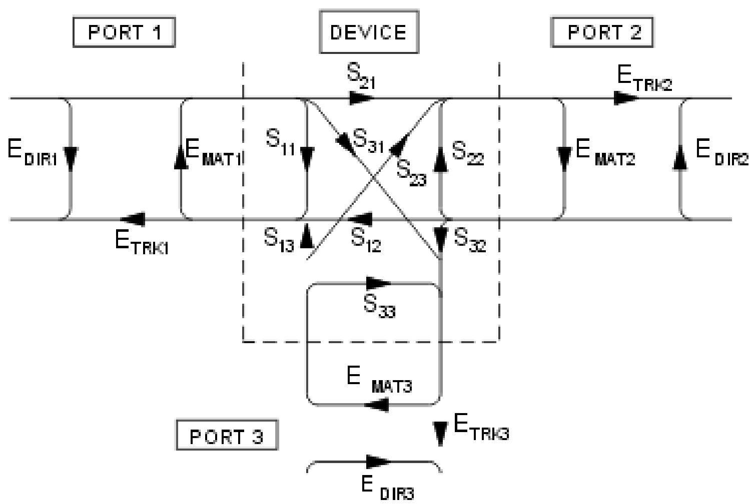
3-port PNA Series network analyzers offer complete characterization of three-port devices quickly, compared to two-port analyzers. You can measure devices like directional couplers, splitters, and antenna diplexers without setup changes. 3-port measurements allow viewing up to nine different S-parameters in a variety of format and scale settings.

3-Port Calibrations

To perform a Full 3-port calibration, use Guided calibration in the Calibration Wizard. In Guided Calibration, you can perform the calibration with either mechanical standards or an ECal module. When you use mechanical standards, Guided calibration automatically performs adapter removal calibrations for noninsertable devices. See Using Calibration Wizard.

Error Terms

The following flowgraph displays the error terms in the 3-port model:



where:

E = error term

DIR = Directivity

MAT = Forward Source Match and Reverse Load Match

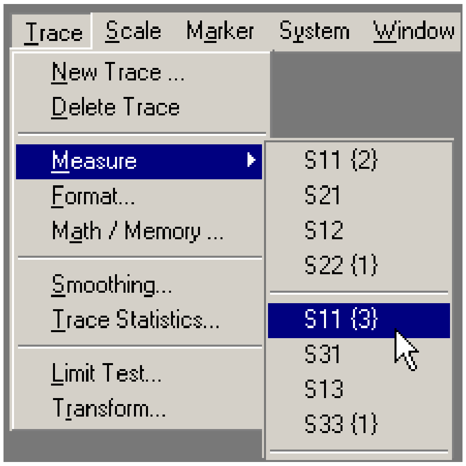
TRK = Forward Reflection Tracking and Reverse Transmission Tracking

Viewing Measurements

Because there are nine available parameters to view, it is useful to know how to configure the measurement in order to view data. You can configure multiple measurements in a variety of ways using up to sixteen Windows and four traces per Window. Once your measurement is configured, you can resize and arrange the placement of Windows as well as print data.

Selecting Parameters

You can select from nine S-parameters; three reflection and six transmission. As shown in the following graphic, the reflection parameters are followed by a number in braces{}. This number identifies the port to be used for the reverse sweep when performing a full 2-port calibration.



Select these and other parameters as you would on a 2-port PNA. See How to select a measurement

Power Calibration

Source and Receiver Power Calibrations work together to provide very accurate power levels from the source and power measurements from the receiver.

- Source Power Calibration
- Reducing Time to Complete a Source Power Calibration
- Receiver Power Calibration
- Saving Source and Receiver Cals

See programming examples in SCPI and COM

See other Calibration Topics

Source Power Calibration

Perform Source Power Calibration when you need accurate power levels at some point in the measurement path between the PNA test ports. For example, you need to characterize the gain of an amplifier across a frequency range at a specified input power. You would perform a source power cal at the input of the amplifier to ensure the **exact** power level into the amplifier across the frequency range.

Using a Source Power Cal, you can reasonably expect the power at the point of calibration to be within the range of the uncertainty of the power meter and sensor that is used.

> **Note:** You may not be allowed to perform a Source Power Cal unless you are logged on to the PNA with an Administrator user account. To correct this, see http://na.tm.agilent.com/pna.

Source Power Calibration:

- Is independent of measurement type. It corrects the PNA source regardless of which receivers are being used in a measurement. Therefore, it can be used with both ratio or non-ratio measurements.
- Applies to all measurements on the channel that is active when the calibration is performed.
- ONLY applies to those measurements that use the test port that was specified as the Source for the calibration. For example, if you specify Port 1 as the source to be calibrated, only those measurements on the active channel that use port 1 as the source will be corrected.
- Can be used in conjunction with other measurement calibrations, such as a full 2-port calibration. For highest accuracy, perform the measurement calibration AFTER the source calibration.
- Can be used with Power Sweep type. Source Power Cal will correct the power at all power levels across the power sweep.

Overview of How it works:

Click to see the detailed procedure

1. Specify the measurement settings (frequency range, IFBW and so forth).
2. Start Source Power Calibration.

3. Connect a power meter sensor to the point at which you want a known power level. This may be at the input or output of your device, or some other point between the test ports.
4. The program steps the PNA source through the specified frequency range, at the specified power level, and measures the power using a power meter.
5. The program calculates and stores the difference between the desired corrected power (Cal power) and what was measured by the power meter, at each data point.
6. When complete, the source power calibration can be saved to disk as part of the instrument state.

- The power meter is removed and the measurement path reconnected.
- The calibration is automatically applied to the measurement so that during subsequent sweeps, the source power is adjusted by the difference values to provide the corrected power level.

Test Equipment Supported

Power Meters

Agilent Model Number	Critical Specification
E4416A	Single Channel
E4417A	Dual Channel
EPM-441A, E4418A/B	Single Channel
EPM442A, E4419A/B	Dual Channel
437B	Single Channel
438A	Dual Channel
See Create a Custom Power Meter Driver.	

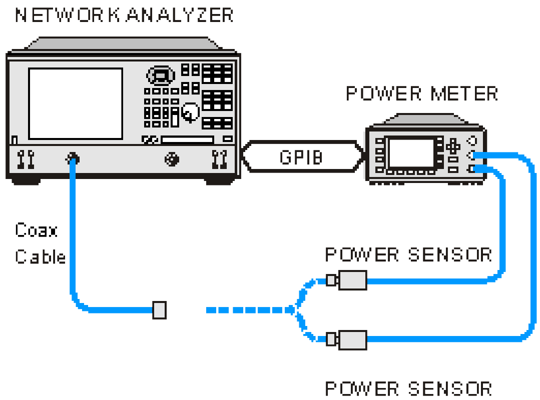
Power Sensors

You can perform a Source Power Calibration with ALL power sensors that are supported by the above list of power meters. However, Source Power Calibration and Scalar Mixer Calibration, operates slowly with the Agilent E930x and E932x power sensors, as the two calibrations are not optimized for use with those sensors.

You can use up to two sensors to cover the frequency span of the measurement.

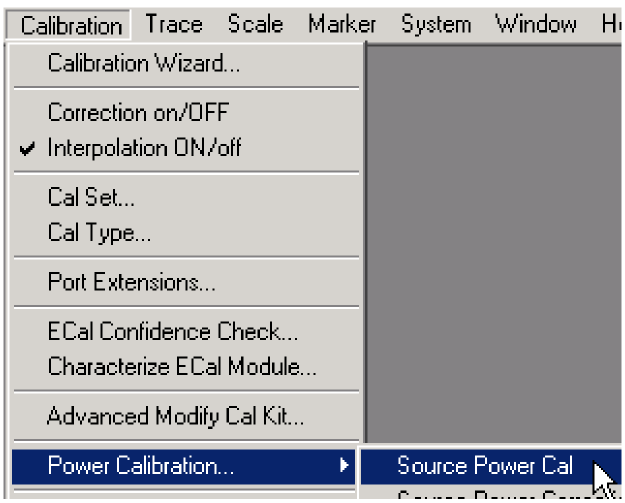
Detailed Procedure: How to perform Source Power Calibration

1. Setup your measurement (sweep type, frequency range, IFBW, and so forth).
2. Connect coax cable, GPIB cable, and power sensors to the PNA as shown in graphic below.



Note: You can use the 82357A USB/GPIB Interface to control the power meter.

3. Apply power to the power meter and allow 30 minutes warm-up time before beginning calibration.
4. Select **Source Power Cal** as follows:

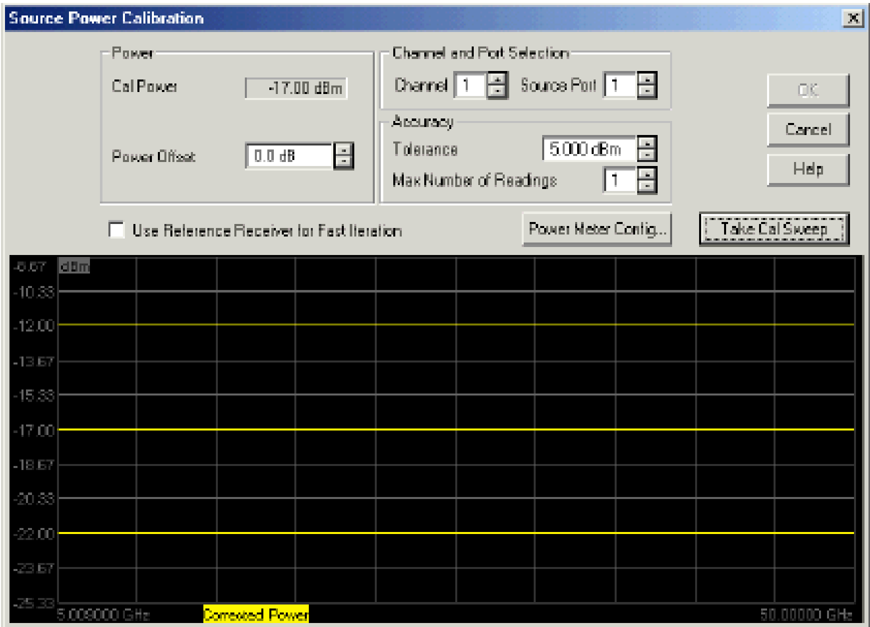


5. Complete the Source Power Cal dialog box (below), including Loss Compensation and Power Sensor Settings, as needed.
6. When complete, click **Take a Cal Sweep** in the Source Power Cal dialog box.
7. Follow the prompts to connect the sensors as required.
8. When calibration is finished, click **OK**.
9. Remove sensor.

- **Src Pwr Cal** is displayed in the status bar and correction data is applied to subsequent sweeps.
- To turn correction **OFF**, on the **Calibration** menu, point to **Power Calibration**, then set **Source Power Correction** to **OFF**.

Interpolation

If the original stimulus settings are changed, Interpolation or EXTRAPOLATION is applied and **Src Pwr Cal*** is displayed in the status bar. This is different from measurement calibration interpolation. For example, if the frequency span is increased, the PNA will extrapolate new correction values rather than turn correction off. This is to protect your test device from being overpowered by the source. If the original settings are restored, then source power calibration returns to full correction.



Source Power Cal dialog box help

Note: Be sure that the frequency range of your power sensor covers the frequency range of your measurement. This does NOT occur automatically.

Power

Cal Power The calculated power (in dBm) at the calibration point. This value is the specified PNA source power plus the Power Offset value.

Power Offset Allows you to specify a gain or loss (in dB) to account for components you connect between the source and the reference plane of your measurement. For example, specify 10 dB to account for a 10 dB amplifier in the path to your DUT. Following the calibration, the PNA power readouts are adjusted to this value.

Channel and Port Selection

Channel Specifies the channel on which to perform the calibration. This setting defaults to the active channel.

Source Port Specifies the selected source port. This setting defaults to the source port for the active channel.

Accuracy

At each data point, power is measured (using the specified settling) and adjusted, until the reading is within the specified **Tolerance** or the **Max Number of Readings** has been met. The last power reading is plotted on the screen (below the dialog box) against the Tolerance limit lines.

Tolerance Sets the maximum desired deviation from the specified **Cal Power** level.

Max Number of Readings Sets the maximum number of readings to take at each data point for iterating the source power.

Use Reference Receiver for Fast Iteration When checked, the first reading at each data point is used to calibrate the reference receiver. Subsequent readings, if necessary to meet your accuracy requirement, are measured using the reference receiver. This technique is much faster than using the power meter with almost no degradation in accuracy.

NOTE:Do NOT use the Reference Receiver for Iteration feature if there is a component before the power sensor that exhibits non-linear behavior, such as a power amplifier in compression.

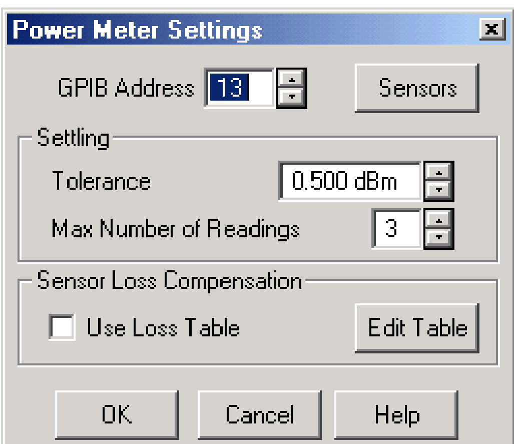
Power Meter Config Invokes the Power Meter Settings dialog box

Take Cal Sweep Begins source power calibration measurement.

OK Applies calibration. This button is disabled until the **Take Cal Sweep** has been pressed.

Cancel Cancels calibration.

Learn more about Source Power Cal



Power Meter Settings dialog box help

GPIB Address GPIB address for the power meter . Default is 13. When performing a source power cal, the PNA will search VISA interfaces that are configured in the Agilent IO LIbraries on the PNA.

Sensors Invokes the power sensor settings dialog box.

Settling

Each power reading is "settled" when either consecutive meter readings are within the **Tolerance** value or when the **Max Number of Readings** has been met. The readings that were taken are averaged together to become the "settled" reading. The settled reading is then compared to the Accuracy requirements (tolerance and max readings) specified on the Source Power Cal dialog box.

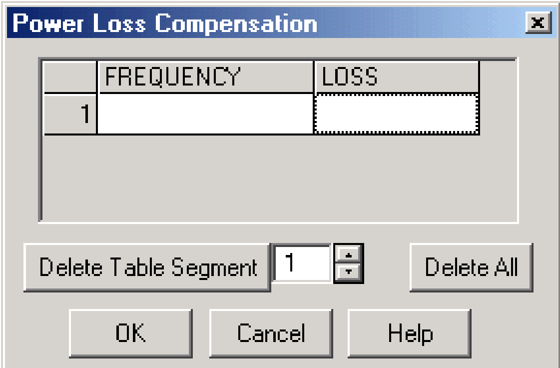
Tolerance When consecutive power meter readings are within this value of each other, then the reading is considered settled.

Max Number of Readings Sets the maximum number of readings the power meter will take to achieve settling.

Sensor Loss Compensation

Use Loss Table Select this checkbox to apply loss data to Source Power calibration correction (such as for an adapter on the power sensor).

Edit Table Invokes the Power Loss Compensation dialog box.



Power Loss Compensation dialog box help

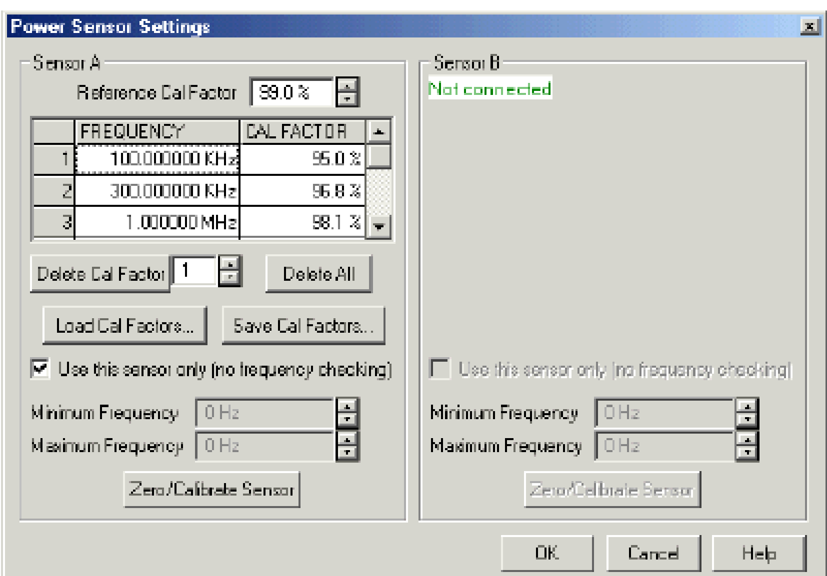
Allows you to compensate for losses that occur when you use an adapter or coupler to connect the power sensor to the measurement port.

Delete Table Segment Deletes row indicated in the field.

Delete All Deletes all data in the table.

Note: To Add a Row to the table, click on a row in the table and press the down arrow on either the PNA front panel or keyboard.

- If you enter a single frequency/loss segment, the analyzer applies that value to the entire frequency range.
- You can enter up to 100 segments to achieve greater accuracy.



Power Sensor dialog box help

Note: Be sure that the frequency range of your power sensor covers the frequency range of your measurement. This does NOT occur automatically.

Sensor A (B) Displays one of the following messages depending on type of sensor.

- **Not connected** The PNA is not detecting a power sensor.
- **Cal factors are contained within this sensor** The PNA detects an Agilent E-Series power sensor. Reference Cal Factor and Cal Factor data are loaded automatically.
- **Sensor Data** Allows entry for power sensor data:

Reference Cal Factor Specifies the sensor's Reference Cal Factor.

Cal Factor Table Specifies the frequency and corresponding Cal Factor for the sensor.

Delete Cal Factor Deletes the indicated row in the table.

Delete All Deletes all data in the table.

Use this sensor only Select this checkbox to use this sensor over the entire frequency span of the measurement, even if two sensors are connected to power meter. Clear to allow entry of minimum and maximum frequencies for the sensor.

Minimum Frequency Specifies the minimum frequency range for the sensor when using dual sensors.

Maximum Frequency Specifies the maximum frequency range for the sensor when using dual sensors.

Perform Sensor Zeroing and Calibration Zero and calibrate the power sensor before taking data.

Note: To Add a Row to the table, click on a row in the table and press the down arrow on either the PNA front panel or keyboard. A row is added to the bottom of the table. The table is automatically sorted by frequency when OK is pressed.

Reducing Time to Complete a Source Power Calibration

The time required to perform a Source Power Calibration depends on source power, number of points, and number of readings taken. You can reduce this measurement time with the following methods:

- **Reduce number of points before calibration.** You can reduce the number of points before the measurement, then return the number of points to its original value after calibration is complete and correction is ON. The analyzer will perform a linear interpolation, although with some loss in accuracy.
- **Use an Agilent E-Series sensor.** You can obtain 200+ readings per second over GPIB with this type of sensor.
- **Increase power to the sensor.** Lower power may have longer settling time with some sensors.
- **Check Use Reference Receiver for Iteration.**

Receiver Power Calibration

Receiver power calibration mathematically removes frequency response errors in the specified PNA receiver, and adjusts readings to the same, or a value offset from, the source power calibration level. It is the same as doing **Data / Memory,** but with the data shifted to the Cal Power value.

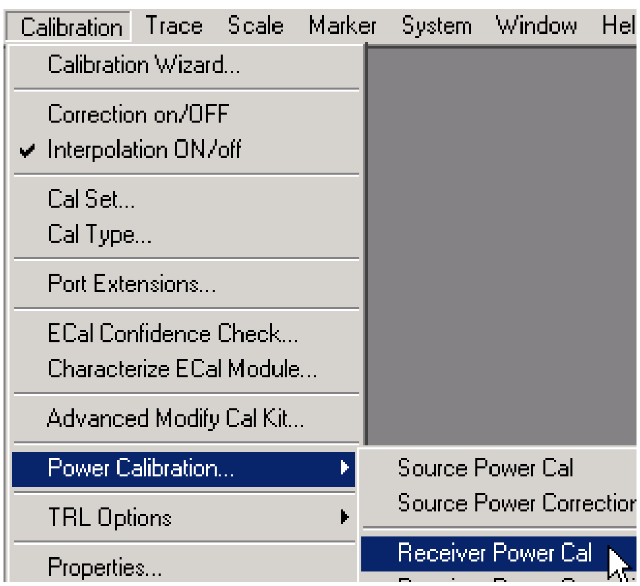
Use Receiver Power Calibration to make very accurate absolute power (amplitude) measurements.

Receiver Power Calibration:

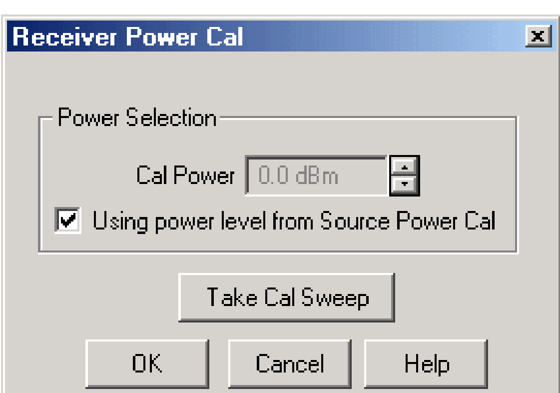
- Expects that a source power calibration was performed first.
- Is ONLY allowed when making absolute power (unratioed) measurements.
- ONLY applies to the active measurement.

How to perform a Receiver Power Calibration

1. Perform a Source Power Calibration.
2. Set the active measurement to unratioed. Learn How.
3. Connect a thru line from the source port to the receiver port. If you are performing a receiver power cal on a reference receiver, no connection is necessary as the receiver is internally connected to the source.
4. Ensure correction for Source Power Calibration is ON as indicated by **Src Pwr Cal** or **Src Pwr Cal*** in the status bar.
5. Select **Receiver Power Cal** as follows:



6. Complete the following Receiver Power Cal dialog box



Receiver Power Cal dialog box help

Cal Power Specifies the power to be measured by the receiver.

Using power level from Source Power Cal. Select this checkbox to use the power level from the Source Power Calibration. Clear to enter a different Cal Power value.

Take a Cal Sweep Click to trigger a Receiver Power Calibration sweep.

OK Click when Receiver Power Calibration sweep is complete.

Notes:

- When Receiver Power Cal is finished, **C Rcvr Pwr** is displayed in the status bar and correction data is applied to subsequent sweeps.
- To turn correction **OFF**, click **Calibration,** point to **Power Calibration**, then set **Receiver Power Correction** to **OFF**.

Interpolation

Like a measurement calibration, if the original stimulus settings are narrowed, interpolation is applied and **C* Rcvr Pwr** is displayed in the status bar. If the original stimulus settings are made wider, the PNA will turn Receiver Power Correction **OFF.**

If the original settings are restored, then receiver power calibration returns to full correction.

Saving Source and Receiver Power Calibration

Both Source and Receiver Cals are saved as part of the Instrument State; NOT as Calibration Data. Therefore, you can save them with either a .sta file, or a .cst file. If you would like to save measurement calibration data along with the Source or Receiver Cals, you must use a .cst file. Learn more about Saving PNA files types.

Calibration Standards

This following section explains the general principles and terms regarding calibration kit files. To learn **how** to modify calibration kit files, See Modify Calibration Kits.

- **About Calibration Kits**
- **Calibration Standards**
- **Standard Type**
- **Standard Definitions**
- **Class Assignments**

See other Calibration Topics

About Calibration Kits

A calibration kit is a set of physical devices called standards. Each standard has a precisely known or predictable magnitude and phase response as a function of frequency.

In order for the analyzer to use the standards of a calibration kit, the response of each standard must be mathematically defined and then organized into standard classes that correspond to the error models used by the analyzer.

To be able to use a particular calibration kit, the known characteristics from each standard in the kit must be entered into analyzer memory or recalled from a default list of calibration kits stored in the analyzer.

For a list of calibration kits, see Analyzer Accessories.

Calibration Standards

Calibration standards provide the reference for error-corrected measurements in the network analyzer. Each standard has a precisely known definition that includes electrical delay, impedance, and loss. The analyzer stores these definitions and uses them to calculate error correction terms.

During measurement calibration, the analyzer measures standards and mathematically compares the results with "ideal models" of those standards. The differences are separated into error terms that are later removed from device measurements during error correction.

Standard Type

A standard type is one of four basic types that define the form or structure of the model to be used with that standard. The four standard types are shown below:

Standard	Terminal Impedance
SHORT	zero ohms
OPEN	infinite ohms
LOAD	system impedance, Z0
THRU/LINE	no terminal impedance
ARBITRARY	user-defined

Standard Definitions

Standard definitions describe the electrical characteristics of the standards and the frequencies they will be used. Refer to the "standards definition table" included with the calibration kit. Standard definitions include:

- **Minimum Frequency** Specifies the minimum frequency the standard is used for calibration.
- **Maximum Frequency** Specifies the maximum frequency the standard is used for calibration.
- **Z0** Specifies the characteristic impedance of the standard (not the system characteristic impedance or the terminal impedance of the standard).
- **Delay** Specifies a uniform length of transmission line between the standard being defined and the actual calibration plane.
- **Type** Specifies type of standard (SHORT, OPEN, THRU/LINE, LOAD, ARBITRARY).
- **Loss** Specifies energy loss, due to skin effect, along a one-way length of coaxial cable.

Loss model equation:

- The value of loss is entered as ohms/second at 1 GHz.
- To compute the loss of the standard, measure the delay in seconds and the loss in dB at 1 GHz. Then use the following formula:

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

Capacitance model equation:

C0, C1, C2, C3. Specifies the fringing capacitance for the open standard.

- C = (C0) + (C1 x F) + (C2 x F^2) + (C3 x F^3)
- (F is the measurement frequency).
- The terms in the equation are defined when specifying the open as follows:
  - C0 term is the constant term of the third-order polynomial and is expressed in Farads.
  - C1 term is expressed in F/Hz (Farads/Hz).
  - C2 term is expressed in F/Hz^2.

- C3 term is expressed in F/Hz^3.

Inductance model equation:

L0, L1, L2, L3. Specifies the residual inductance for the short standard.

- $L = (L0) + (L1 \times F) + (L2 \times F^2) + (L3 \times F^3)$
- (F is the measurement frequency).
- The terms in the equation are defined when specifying the short as follows:
    - L0 term is the constant term of the third-order polynomial and is expressed in Henries.
    - L1 term is expressed in H/Hz (Henries/Hz)
    - L2 term is expressed in H/Hz^2.
    - L3 term is expressed in H/Hz^3.

Class Assignments

Once a standard is characterized, it must be assigned to a standard "class". A standard class is a group of standards that are organized according to the calibration of the PNA error model.

The number of classes needed for a particular calibration type is equal to the number of error terms being corrected.

A class often consists of a single standard, but may be composed of multiple standards, such as loads or delay lines. Refer to the calibration kit "class assignment" table.

Example: A response calibration requires only one class, and the standards for that class may include an OPEN, or SHORT, or THRU. A 1-port calibration requires three classes. A full 2-port requires 10 classes, not including two for isolation.

The number of standards assigned to a given class may vary from one to seven for unguided calibrations. Guided calibrations allow as many standards as needed.

The different classes used in the PNA:

S11A, S11B, S11C (S22A, S22B, S22C and so forth)

These are the three classes for port 1-reflection calibrations (three classes also for S22 and S33). They are used in the one-port calibrations and the full two-port calibration. They are required in removing the directivity, source match, and reflection tracking errors. Typically, these classes might consist of an open, a short and a load standard for each port.

Transmission and Match (forward and reverse)

These classes are used to perform a full two-port calibration. The transmission class relates primarily to the transmission tracking, while the match class refers to load match. For both of these classes, the typical standard is a thru or delay.

Isolation

The isolation classes are used to perform a full two-port and the TRL two-port calibrations. The isolation classes apply to the forward and reverse crosstalk terms in the PNA error model.

TRL thru

These are used to perform a TRL two-port calibration. The TRL thru class should contain a thru standard or a short line. If it contains a non-zero length thru standard, then the calibration type is called LRL or LRM.

TRL reflect

This class is used to perform a TRL two-port calibration. The TRL reflect class should contain a standard with a high reflection coefficient, typically an open or short. The actual reflection coefficient need not be known, but its phase angle should be specified approximately correctly (± 90 deg). The exact same reflection standard must be used on both ports in the TRL calibration process.

TRL line (or match)

These are used to perform a TRL two-port calibration. The TRL line or match class should contain line standards, load standards, or both. If a line standard is used, its phase shift must differ from that of the TRL thru standard by 20° to 160°. This limits the useable frequency range to about 8 to 1. Two or more line standards of different lengths may be specified to get broader frequency coverage. It is also common to include a load standard for covering low frequencies, where the line's length would be impractically long. When a load is used, the calibration type is called TRM or LRM.

Note: For more information, read application note 8510-5A, "*Specifying Calibration Standards for the Agilent 8510 Network Analyzer*". Although the application note is written for the Agilent 8510 series of network analyzers, it applies to the PNA as well. The part number for the publication is 5956-4352.

TRL Calibration

TRL (Thru - Reflect - Line) represents a **family** of calibration techniques that measure two transmission standards and one reflection standard to determine the 2-port 12-term error coefficients. For example, TRM (Thru - Reflect - Match) is included in this family. The traditional SOLT calibration measures one transmission standard (T) and three reflection standards (SOL) to determine the same error coefficients.

- **Why Perform a TRL Cal?**
- **The TRL Calibration Process**
- **TRL Cal Kits**
- **Cal Standards Used in TRL**
- **TRL Options**

Note: You can perform a TRL cal only on 2-port PNA models with a dedicated reference receiver for every test port. See PNA Standard Configurations for a list of model numbers. 4-port PNA models (N5230A Opts 240/245) do NOT support TRL or TRL* calibrations because there is only one reference receiver.

See other Calibration Topics

Why Perform a TRL Cal?

TRL calibration is extremely accurate, in some cases more accurate than an SOLT cal. However, very few calibration kits contain TRL standards. TRL Cal is most often performed when you require a high level of accuracy and do not have calibration standards in the same connector type as your DUT. This is usually the case when using test fixtures, or making on-wafer measurements with probes. Therefore, in some cases you must construct and characterize standards in the same media type as your DUT configuration. It is easier to manufacture and characterize three TRL standards than the four SOLT standards.

A limitation for TRL cal with broad frequency coverage is the requirement for multiple LINE standards. For example, a span from 2 GHz to 26 GHz requires two line standards. Also, for lower frequencies, the LINE standard can be too long for practical use.

The TRL Cal Process

Although TRL can be performed using the Cal Wizard Unguided Cal selection, the following process uses the easier SmartCal selection. Both selections require that you already have TRL calibration standards defined and included in a PNA cal kit.

1. Preset the PNA
2. Connect the DUT to the PNA
3. Set up a S-parameter measurement and the desired stimulus settings.
4. Click **Calibration / Calibration Wizard**
5. Click **SmartCal Guided Cal / Use Mechanical Stds**
6. Select the DUT connectors and Cal Kit for each port. The selected Cal Kits MUST include TRL standards. TRL should appear as the Cal Method.

7. Check **Modify Cal, Next**, then **View/Modify** to change default TRL options if necessary.
8. Follow the prompts to complete the calibration.
9. Check the accuracy of the calibration

TRL Cal Kits

Agilent Technologies offers two cal kits that include the required standards to perform a TRL calibration: 85050C (APC 7mm) and 85052C (3.5mm). Both kits include the traditional Short, Open, and Load standards. (The Thru standard, not actually supplied, assumes a zero-length Thru). In addition, the kits include an airline which is used as the LINE standard. To use the airline, the kits include an airline body, center conductor, and insertion / extraction tools. The APC 7 kit includes an adapter to connect the airline to the APC connector.

Cal Standards Used in TRL

These standards must be defined in your TRL cal kit:

THRU

- The THRU standard can be either a zero-length or non-zero length. However, a zero-length THRU is more accurate because it has zero loss and no characteristic impedance.
- The THRU standard cannot be the same electrical length as the LINE standard.
- If the insertion phase and electrical length are well-defined, the THRU standard may be used to set the reference plane.

REFLECT

- The REFLECT standard can be anything with a high reflection, as long as it is the same when connected to both PNA ports.
- The actual magnitude of the reflection need not be known.
- The phase of the reflection standard must be known within 1/4 wavelength.
- If the magnitude and phase of the reflection standard are well-defined, the standard may be used to set the reference plane.

LINE

The LINE standard establishes the reference impedance for the measurement after the calibration is completed. TRL calibration is limited by the following restrictions of the LINE standard:

- Must be of the same impedance as the THRU standard
- The electrical length need only be specified within 1/4 wavelength.
- Cannot be the same length as the THRU standard.
- Must be an appropriate electrical length for the frequency range: at each frequency, the phase difference between the THRU and the LINE should be greater than 20 degrees and less than 160 degrees. This means in practice that a single LINE standard is only usable over an 8:1 frequency range (Frequency Span / Start Frequency). Therefore, for broad frequency coverage, multiple lines are required.

- At low frequencies, the LINE standard can become too long for practical use. The optimal length of the LINE standard is 1/4 wavelength at the geometric mean of the frequency span (square root of f1 x f2).

MATCH

If the LINE standard of appropriate length or loss cannot be fabricated, a MATCH standard may be used instead of the LINE.

- The MATCH standard is a low-reflection termination connected to both Port 1 and Port 2.
- In the TRL computation, this standard is treated as a high-loss, infinite length transmission line.
- The impedance of the MATCH standard becomes the reference impedance for the measurement.

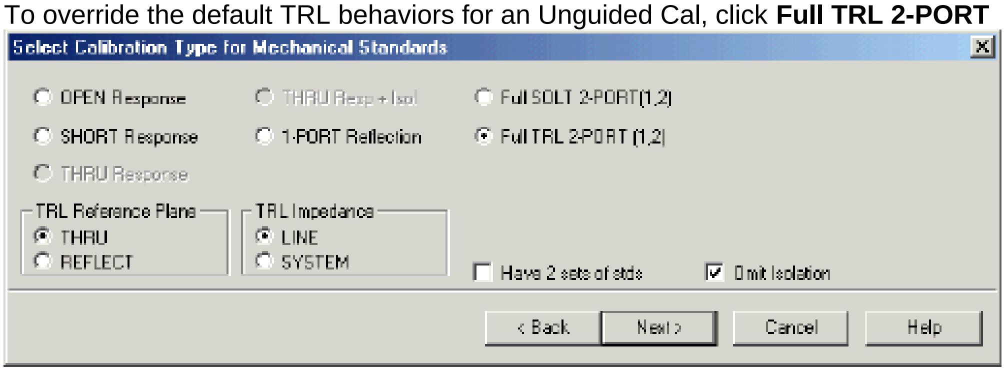
Note: The MATCH standards must be assigned to the FWD MATCH, REV MATCH, and TRL LINE classes. See Modify Calibration Kits for detailed information about creating and modifying Calibration kit definitions.

Find more information about TRL standards at http://www.tm.agilent.com. Click "Technical Support". Use "Application Notes" to search for App Note 8510-5A (part number 5956-4352). Although the application note is written for the Agilent 8510 Series Network Analyzers, it also applies to the PNA.

How to set TRL Options

Use one of the following methods:

- In Guided Cal, Select the DUT connectors and Cal Kit dialog, check Modify Cal. In the Modify Thru Method of Cal dialog box, click View/Modify to invoke the Modify Calibration Class Assignments dialog box. You can also access this dialog box directly by clicking Calibration, Advanced Modify Cal Kit. These settings determine the default TRL options behavior.
- To override the default TRL behaviors for an Unguided Cal, click **Full TRL 2-PORT**



Learn more about using the front panel interface

Select Calibration Type dialog box help

Note: You can perform a TRL cal only on 2-port PNA models with a dedicated reference receiver for every test port. See PNA Standard Configurations for a list of model numbers. 4-port PNA models (N5230A Opts 240/245) do NOT support TRL or TRL* calibrations because there is only one reference receiver.

TRL Reference Plane

Thru The THRU standard is used to establish the position of the measurement reference plane. Select if the THRU standard is zero-length or very short.

Reflect The REFLECT standard is used to establish the position of the measurement reference plane. Select if the THRU standard is not appropriate AND the delay of the REFLECT standard is well defined.

TRL Impedance

Line Z0 Specifies that the characteristic impedance of the LINE standard should be used as the system impedance. This ignores any difference between Offset Z0, Offset Loss and system Z0.

System Z0 Transforms the LINE standard impedance and loss to that of the system impedance for use with the calibration error terms. The TRL calibration will first compute the error terms assuming the LINE standard impedance is the system's characteristic impedance (same as LINE option), then modify the error terms to include the impedance transformation. This should only be used with coax since the skin effect model used is a coaxial model.

Learn how to change System Z0.

1.

Fixturing Macro

This macro allows you to compensate your PNA measurements for the effects of test fixtures or other external components. Using this macro you can do the following:

- Add Port Extensions with or without loss
- Modify the Reference Impedance of the measurement port
- Embed or De-embed a two-port device

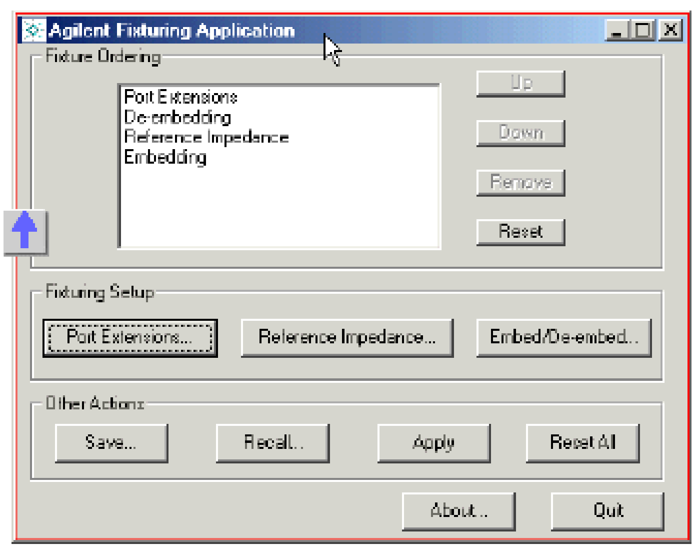
The compensation which results from these operations is applied to specific ports and channels.

For example, you can measure a DUT in a 50 ohm, 75 ohm, and 100 ohm environment by creating three measurement channels, each channel at a different reference impedances. No physical connection changes are required.

How to start the Fixturing Macro

1. Press the Macro key repeatedly until Fixturing appears on the Active Entry screen area.
2. Press the corresponding Active Entry key.

Learn more about using the front panel interface



Agilent Fixturing Macro dialog box help

Fixture Ordering

Different results will occur depending on the order in which these operations are performed. For example, applying a port extension and then embedding a fixture will not give the same results as embedding followed by applying a port extension.

To reorder the operations:

1. Select one of the fixturing operations.
2. Use **Up** and **Down** buttons to reorder the operations as necessary
3. **Remove** any operation that will not be used.

Reset Restores all operations to the default order.

Fixturing Setup

Port Extension Invokes the Port Extensions dialog box.

Reference Impedance Invokes the Reference Impedance dialog box.

Embed/De-Embed Invokes the Embed/De-Embed dialog box.

Other Actions

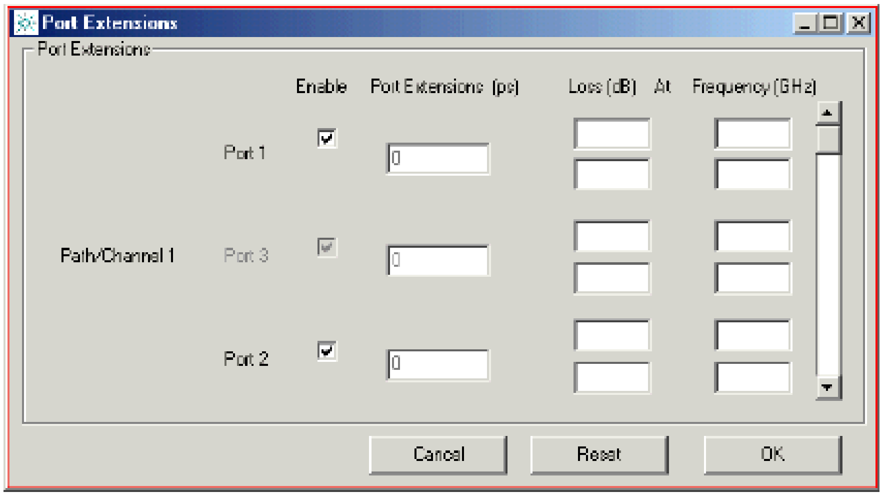
Save Saves the fixturing information for later recall, eliminating the need to re-enter parameters.

Recall Recalls a previously saved fixturing file. Click Apply to apply the recalled fixturing information.

Apply Modifies the PNA error coefficients to include the fixturing information.

Note: Click **Apply** after a file is recalled to implement the compensation on the current measurements.

Reset All Restores the default order of the fixturing operations.
Clears the data entered on the individual operational dialogs.
Restores the original calibration.



Port Extensions dialog box help

Note: Port Extensions apply to individual ports on EACH CHANNEL. Use the scroll bar (far right of dialog box) to scroll to the appropriate channel.

Enable Check to enable port extension on the associated PNA port.

Port Extensions (ps) The amount of extension entered in picoseconds. This value will be reflected in the corrected measurement as a corresponding phase rotation.

Loss (dB) At Frequency One or two loss values may be associated with a Port Extension value.

If one loss value is specified with no frequency value, that loss is taken as a constant across all frequencies. If loss is specified at one frequency, then the loss is extrapolated such that:

L(f) = Loss (1 GHz) * f ^.5

Where:
L(f) is the loss in dB at the frequency of measurement in GHz
Loss(1 GHz) is the loss in dB at 1 GHz.

If loss is specified at two frequencies, then the loss is extrapolated for all frequencies such that:

L(f) = L * f ^n
n = (log (L1/L2)) / (log (f1/f2))
L = L1 / f1^n

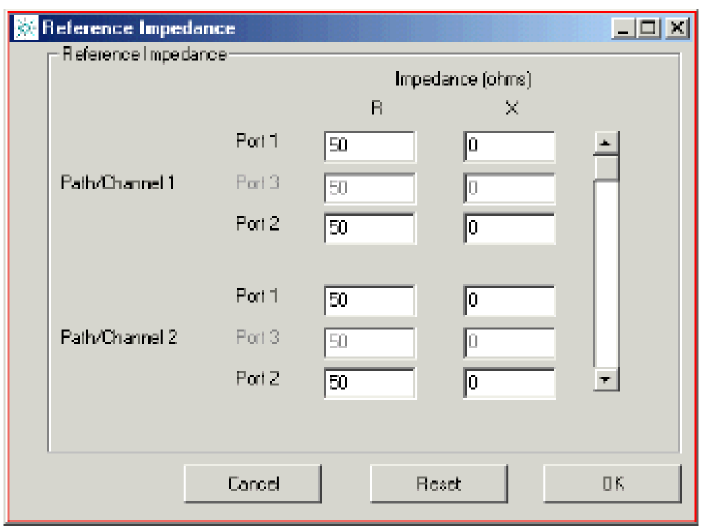
Where:
L1 and L2 are the loss in dB at f1 and f2 (in GHz) respectively.

OK Saves the entries

Reset Restores the form back to default values

Cancel Exits without saving the entries.

Learn about Port Extensions



Reference Impedance dialog box help

The reference impedance algorithm corrects the measurement and displays the results as if the measurement had been made into a reference impedance of the specified value. The physical port termination is still approximately 50 ohms.

Note: Reference Impedance applies to individual ports on EACH CHANNEL. Use the scroll bar (far right of dialog box) to scroll to the appropriate channel.

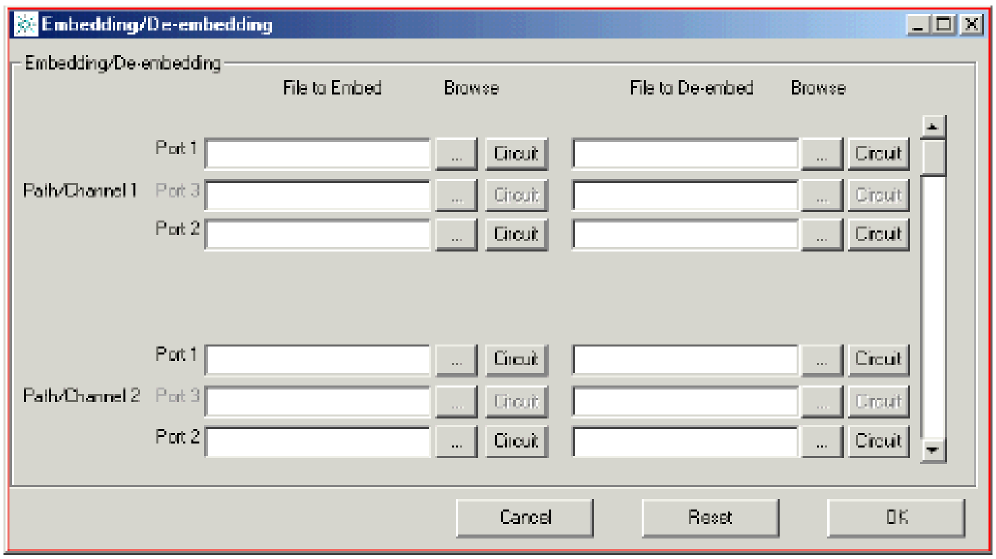
R Resistance part of the desired reference impedance for the specified port and channel.

X Reactance part of the desired reference impedance for the specified port and channel.

Cancel Exits without saving the entries.

Reset Restores the form to default values

OK Saves the entries



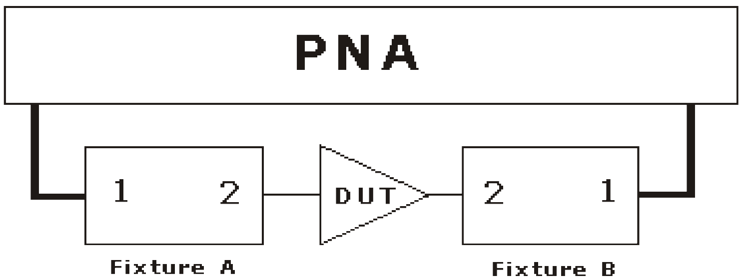
Embed/De-Embed dialog box help

Embedding is used to **add** the effect of a device to the measurement results. For example, you could add a reactive load that is not currently in place.

De-embedding is used to **remove** the unwanted effects of a device from the measurement results. For example, you could remove the effects of a test fixture.

The embedding/de-embedding operation creates and recalls an s2p file (Touchstone format) for a 2-port device. The file includes the electrical characteristics of a supplemental fixture or device. The file can be in any standard format (real-imaginary, magnitude-angle, dB-angle) and can represent any 2-port device.

Note: In all cases, Port 1 of the supplemental device is assumed to be connected to the PNA and Port 2 of the device is assumed to be connected to the DUT.

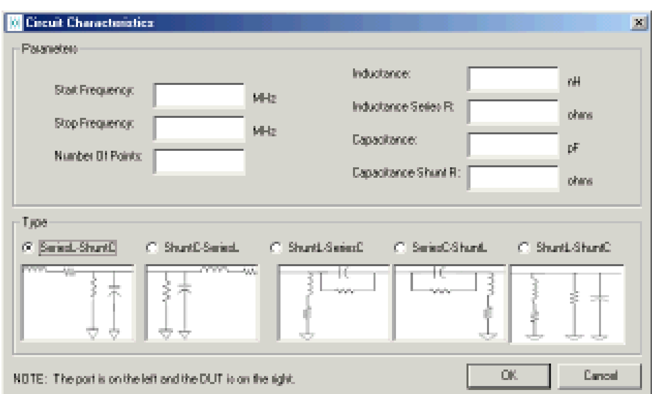


Do one of the following for each port on each channel:

Type the full path name of the file to be embedded or de-embedded in the appropriate box.

Browse Finds a file that already exists.

Circuit Creates a file by starting the Circuit Characteristics dialog box



Circuit Characteristics dialog box help

Allows the design of one of several simple matching circuits. This circuit simulates the device that you are embedding or de-embedding.

Note: The PNA is always on the left and the DUT is always on the right of the circuit.

Parameters

Start Frequency Start frequency of the measurement that the compensation will be applied to.

Stop Frequency Stop frequency of the measurement that the compensation will be applied to.

Number of Points Number of data points of the measurement that the compensation will be applied to.

Inductance, Inductance Series R, Capacitance, Capacitance Shunt R Values for the specific components of the circuit type that models the supplemental device. Capacitance and Capacitance Shunt R are not allowed to be zero.

Type Type of circuit to be created that matches the fixture

OK Creates a Touchstone (.s2p) data file for the specified circuit. Provide a file name when prompted. The Circuit Characteristics dialog box then closes.

Cancel Exits without saving the entries.

Markers

The Markers function provides a numerical readout of measured data, a search capability for specific values, and can change stimulus settings. There are 9 regular markers and one Reference marker available per trace. This topic discusses all aspects of markers.

Note: Marker Readout can be turned ON / OFF and customized from the **View** menu. See Marker Readout

- **Creating and Moving Markers**
- **Delta Markers**
- **Searching with Markers**
- **Marker Functions (Change Instrument Settings)**
- **Advanced Marker Settings**
- **Marker Table**

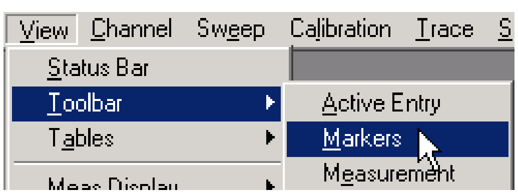
Other Analyze Data topics

How to Create and Move Markers

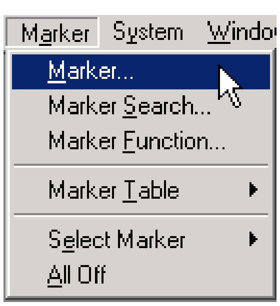
Use one of the following methods:

- Active Entry toolbar

- Marker toolbar



- Marker dialog box



Learn more about using the front panel interface

Create a Marker

- From the Active Entry toolbar, press a marker number. To access markers 4 through 9, press 

repeatedly.

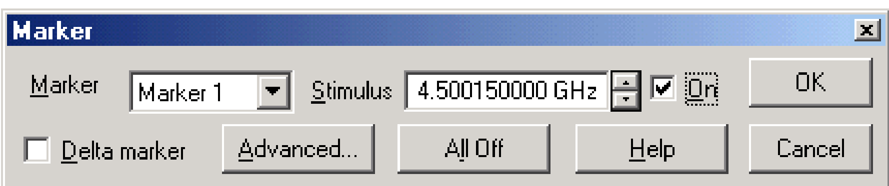
- From the Marker Toolbar or Marker Dialog Box, select a marker number and click **ON**

Moving a Marker

To move a marker, make the marker active by selecting its number (in any of the previous 3 methods. Then change the stimulus value using any of the following methods:

- Type a value
- Scroll to a stimulus value using the up / down arrows
- Click the stimulus box, then use the front-panel knob.

Note: To change marker properties, the marker must be active. The **active marker** appears on the analyzer display as Ñ. All of the other markers are inactive and are represented on the analyzer display as D.



Marker dialog box help

Marker Specifies the marker number that you are defining.

Stimulus Specifies the X-axis value of the selected marker. To change stimulus value, type a value, use the up and down arrows, or click in the text box and use the front-panel knob.

On Check to display the marker and corresponding data on the screen.

Delta Marker Check to make the specified marker data relative to the reference marker. If not already on, the reference (R) marker will be displayed automatically. Learn more about Delta Markers

Advanced... Invokes the Advanced Markers dialog box.

All Off Switches OFF all markers on the active trace.

Delta Markers

Delta Markers allow you to view data that is relative to the **reference** marker.

A delta marker can be set from the Marker dialog box or the Marker Toolbar.

When a Delta marker is created, the reference marker will be activated automatically.

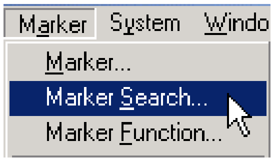
Searching with Markers

You can use markers to search measurement data for specific criteria.

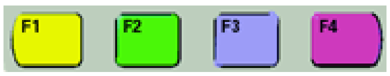
If there is no valid data match for any of the search types, the marker will not move from its current position.

How to Search with Markers

Use one of the following methods:

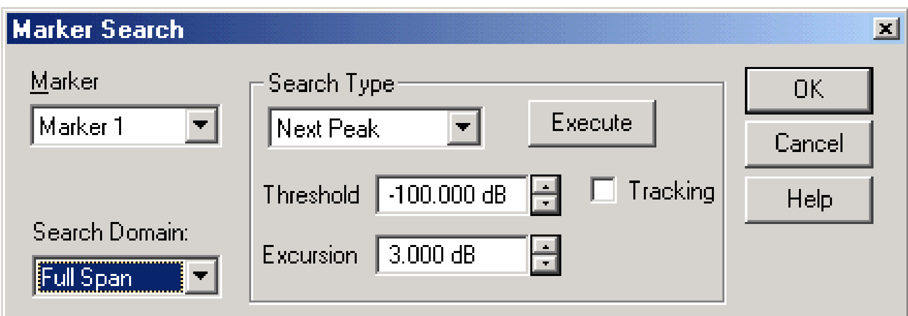
- 

- 

+ 

Only **Max, Min, Left Peak**, and **Right Peak** search types are available from Active Entry Keys

Learn more about using the front panel interface



Marker Search dialog box help

Marker Specifies the marker that you are defining.

Search Domain Defines the area where the marker can move or search. For full span, the marker searches for specified values within the full measurement span. For user span, the marker searches for specified values within a measurement span that you define. Learn more about Search Domain.

Search Type

Maximum Marker locates the maximum (highest) data value.

Minimum Marker locates the minimum (lowest) data value.

Next Peak Marker locates the peak with the next lower amplitude value relative to its starting position. See What is a Peak.

Peak Right The marker locates the **next valid peak to the right** of its starting position on the X-axis .See What is a Peak.

Peak Left The marker locates the **next valid peak to the left** of its starting position on the X-axis .See What is a Peak.

- **Threshold** - Minimum amplitude (dB). To be considered valid, the peak must be **above** the threshold level. The valley on either side can be below the threshold level.
- **Excursion** The vertical distance (dB) between the peak and the valleys on both sides. To be considered a peak, data values must "fall off" from the peak on both sides by the excursion value.

Target Enter the Target value. The marker moves to the first occurrence of the Target value **to the right of its current position**. Subsequent presses of the Execute button cause the marker to move to the next value to the right that meets the Target value. When the marker reaches the upper end of the stimulus range, it will "wrap around" and continue the search from the lower end of the stimulus range (left side of the window).

- If **Discrete Marker** is OFF, the marker locates the interpolated data point that equals the target value.
- If **Discrete Marker** is ON and there are two data points on either side of the target value, the marker locates the data point closest to the Target value

Bandwidth Marker1 locates the highest peak which falls off on both sides by the specified level. (When Bandwidth is selected, a Level box appears to specify the level in dB down from the peak where bandwidth is measured - default is -3dB.)

This search type always uses Markers 1-4.

- Marker 1: Peak
- Marker 2: Specified level down to the left of the peak
- Marker 3: Specified level down to the right of the peak
- Marker 4: Center frequency of the bandwidth

Display readout for Bandwidth Search:

- **BW**: (Marker 3 x-axis value) **-** (Marker 2 x-axis value) = width of the filter
- Center Mathematical midpoint between markers 2 and 3
- Q Ratio of Center Frequency to Bandwidth (Center Frequency / Bandwidth)
- Loss Y-axis value of Marker 4. This is the loss of the filter at its center frequency. The ideal filter has no loss (0 dB) in the passband.

Note: You must either press **Execute** or check **Tracking** to initiate all search types.

Execute Click to cause the marker to search for the specified criteria.

Tracking Check to cause the marker to search for the specified criteria with each new sweep. The searches begin with the first sweep after Tracking has been checked, based on the current search type and domain information. Therefore, make sure that the search criteria are in the desired state before using the data. You cannot manually change the stimulus setting for a marker if Tracking is selected for that marker.

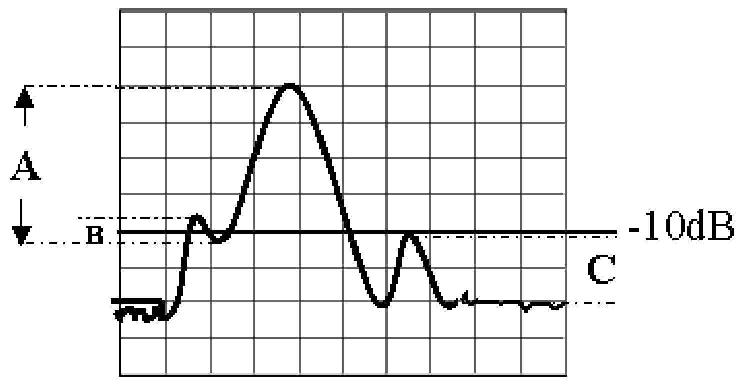
What Is a "Peak"?

You define what the analyzer considers a "peak" by selecting the following two peak criteria settings:

- **Threshold** - Minimum amplitude (dB). To be considered valid, the peak must be **above** the threshold level. The valley on either side can be below the threshold level.
- **Excursion** - The vertical distance (dB) between the peak and the valleys on both sides. To be considered a peak, data values must "fall off" from the peak on both sides by the excursion value.

Example:

Threshold Setting: -10dB
Excursion Setting: 1dB
Scale = 1 dB / Division
Mouse over the graphic to find a valid peak.



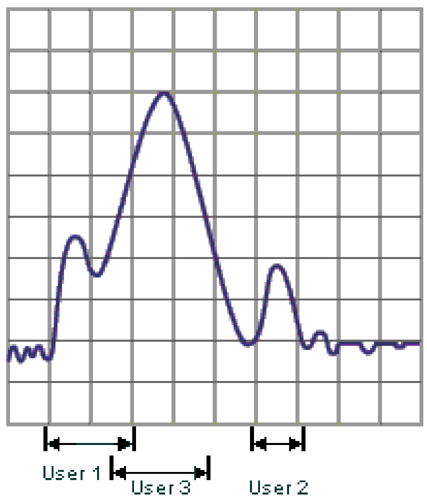
Peak A = Valid Peak (Above Threshold and Excursion Settings)
Peak B = Invalid Peak (Below Excursion Setting)
Peak C = Invalid Peak (Below Threshold Setting)

Search Domain

Search domain settings restrict the stimulus values (X-axis for rectangular format) to a specified span. Set the Start and Stop stimulus settings of these **User** spans. If Start is greater than Stop, the marker will not move.

- The default domain of each new marker is "full span".
- There are 9 user-defined domains for every channel.
- The user-defined domains can overlap.
- More than one marker can use a defined domain.

The graphic below shows examples of search domains.

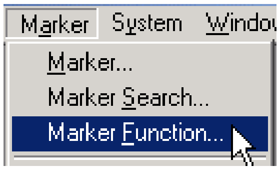


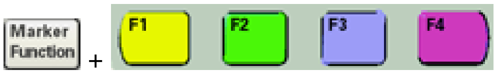
Marker Functions - Change Instrument Settings

The following settings change the relevant PNA settings to the position of the active maker.

How to change instrument setting using markers

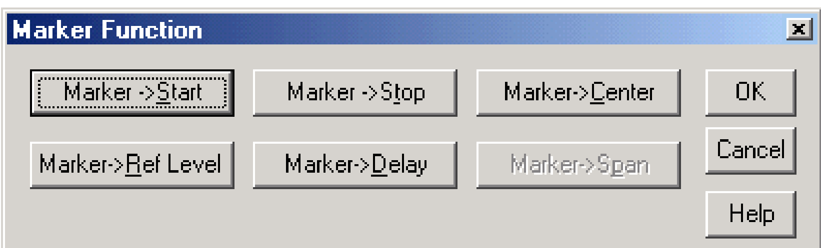
Use one of the following methods:

- 

- 

Only **Center, Ref Level**, and **Delay** are available
- From the Marker Toolbar only Start, Stop Center and Span Frequencies are available.

Learn more about using the front panel interface



Marker Function dialog box help

Note: Marker Functions do not work with channels that are in CW or Segment Sweep mode.

Marker =>Start Sets the start sweep setting to the value of the active marker.

Marker =>Stop Sets the stop sweep setting to the value of the active marker.

Marker =>Center Sets the center of the sweep to the value of the active marker.

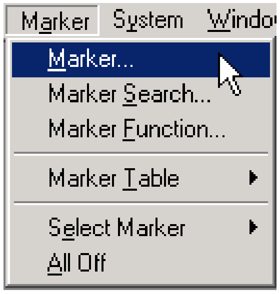
Marker =>Ref Level Sets the screen reference level to the value of the active marker.

Marker =>Delay The phase slope at the **active marker** stimulus position is used to adjust the line length to the receiver input. This effectively flattens the phase trace around the active marker. (Additional Electrical Delay adjustments are required on devices without constant group delay over the measured frequency span.) You can use this to measure the electrical length or deviation from linear phase.

This feature adds phase delay to a variation in phase versus frequency; therefore, it is only applicable for ratioed measurements. (See Measurement Parameters.)

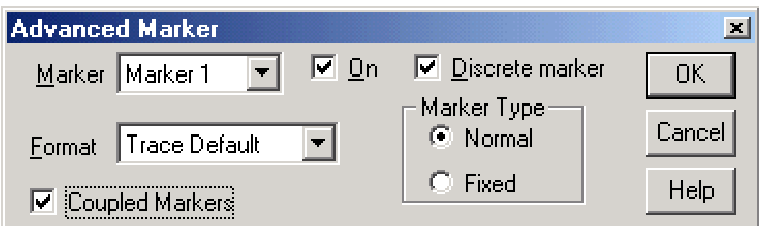
Marker =>Span Sets the sweep span to the span that is defined by the delta marker and the marker that it references. Unavailable if there is no delta marker.

How to select Advanced Marker settings



Then click **Advanced** on the Marker dialog box

Learn more about using the front panel interface



Advanced Marker dialog box help

Marker Specifies the marker number that you are defining.

On Check to display the marker and corresponding data on the screen.

Format Displays the marker data in a format that you choose. The marker format could be different from the grid format. In the default setting, the marker and grid formats are the same.

Discrete Marker Check to display values at only the discrete points where data is taken. Clear to display values that are interpolated from the data points.

Coupled Markers Check to couple markers by marker number, 1 to 1, 2 to 2 and so forth. The markers will remain coupled until this box in unchecked. Learn more about coupled markers.

Marker Type

Normal Has a fixed stimulus position (X-axis) and responds to changes in data amplitude (Y-axis). It can be scrolled left and right on the X-axis by changing the marker stimulus value. Use this marker type with one of the marker search types to locate the desired data.

Fixed Has a fixed X and Y-axis position based on its placement on the trace when it was set to fixed. It does NOT move with trace data amplitude. It can be scrolled left and right on the X-axis by changing the marker stimulus value.

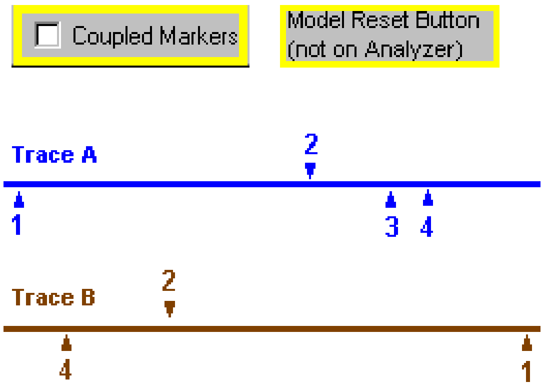
Use this marker type to quickly monitor "before and after" changes to your test device. For example, you could use fixed markers to record the difference of test results before and after tuning a filter.

Coupled Markers

The coupled markers feature causes markers on different traces to line up with the markers on the selected trace. Markers are coupled by marker number, 1 to 1, 2 to 2, 3 to 3, and so forth. If the x-axis domain is the same (such as frequency or time), coupling occurs across all channels, windows, and traces. Trace markers in a different x-axis domain will not be coupled. If a trace marker has no marker to couple with on the selected trace, the marker remains independent.

Coupled Markers Model

This model simulates the use of coupled markers in the PNA

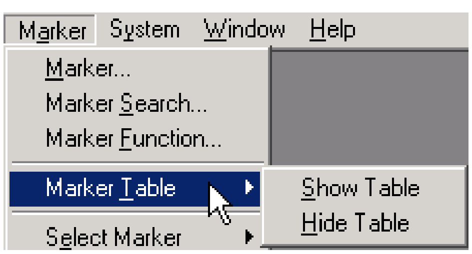


1. Click **Trace A** or **Trace B**
2. Click **Coupled Markers**
3. Notice the following:
   * Markers on the unselected trace move to the x-axis position of the selected trace.
   * If a marker number on the unselected trace has no corresponding marker on the selected trace, no movement occurs for that marker.
4. Click **Reset** to run the model again. (There is no Reset for coupled markers on the PNA.)

Set Coupled Markers from the Advanced Markers dialog box.

Marker Table

You can display a table that provides a summary of marker data for the active trace. The marker data is displayed in the specified format for each marker.

How to view the Marker Table
Use one of the following methods: •  •  Learn more about using the front panel interface

Note: The Marker Table and Marker Readout can also be turned on from the **View** menu. (See Customize Your Analyzer Screen.)

Using Math Operations

You can perform four types of math on the active trace versus a memory trace. In addition three statistics (Mean, Standard Deviation and Peak to Peak) can be calculated and displayed for the active data trace.

- [Trace Math](#)
- [Trace Statistics](#)

Other Analyze Data topics

Trace Math

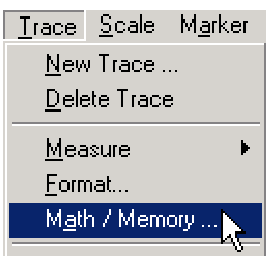
To perform any of the math operations, you must first store a trace to memory. You can display the memory trace using the View options.

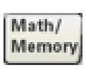
Trace math is performed on the complex data before it is formatted for display. See the PNA data processing map.

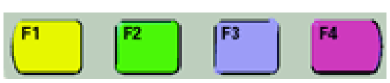
Markers can be used while viewing any trace, including the memory trace.

How to select Trace Math

Use either of the following methods:

- 

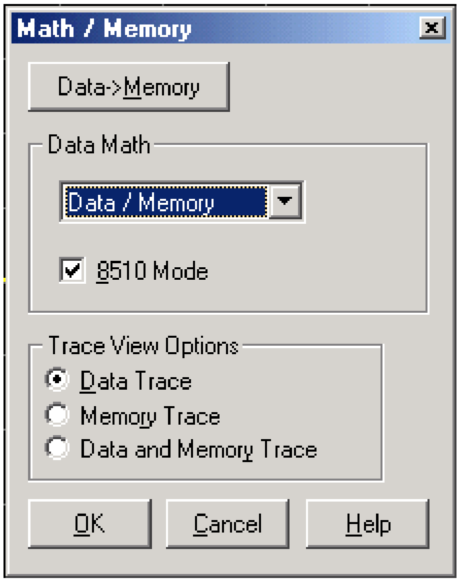
- 

 + 

Only the following are available from Active Entry

- **Data>>Mem** (stores Data trace into Memory)
- **Data/Mem** (performs math operation: Data divided by memory)
- **Data** (displays data trace with no math operation applied)
- **Mem on/OFF** (turns Memory trace on or off)

Learn more about using the front panel interface



Math / Memory dialog box help

Data=>Memory Puts the active data trace into memory.

Data Math

All math operations are performed on linear (real and imaginary) data before being formatted. See the PNA Data flow (below).

Data Does no mathematical operation.

Data / Memory - Current measurement data is divided by the data in memory. Use for ratio comparison of two traces, such as measurements of gain or attenuation. Learn more.

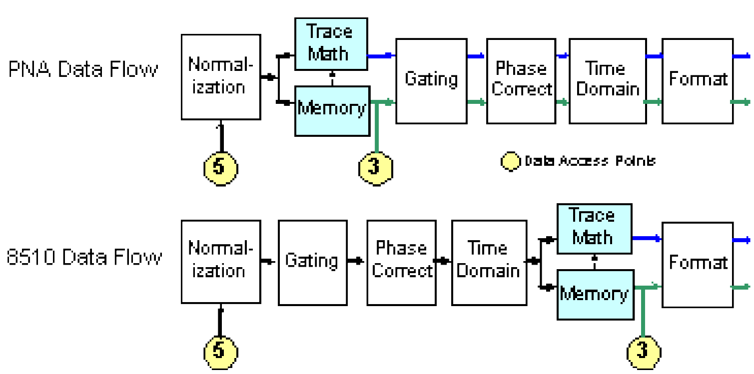
Data – Memory - Data in memory is subtracted from the current measurement data. For example, you can use this feature for storing a measured vector error, then subtracting this error from the DUT measurement. Learn more.

Data + Memory - Current measurement data is added to the data in memory. Learn more.

Data * Memory - Current measurement data is multiplied by the data in memory. Learn more.

8510 Mode Check to simulate the Agilent 8510 data processing chain as it pertains to Trace Math and Memory. This setting applies to all channels. When the box is checked or cleared, the PNA performs an Instrument Preset and retains its setting through subsequent Instrument Presets.

This setting is saved as part of an instrument state. However, when recalled, this setting is assumed only temporarily. When a subsequent PNA Preset is performed, the PNA reverts to the setting that was in effect before the state was recalled.



This represents the relevant portion of the data flow. See the entire PNA data processing chain.

Trace View Options

Data Trace Displays ONLY the Data trace (with selected math operation applied).

Memory Trace Displays ONLY the trace that was put in memory.

Data and Memory Trace Displays BOTH the Data trace (with selected math operation applied). and the trace that was put in memory.

Learn more about Trace Math (scroll up)

(Data / Memory) and (Data - Memory)

(Data / Memory) and (Data - Memory) math operations are performed on linear data before it is formatted. Because data is often viewed in log format, it is not always clear which of the two math operations should be used. Remember: dividing linear data is the same as subtracting logarithmic data. The following illustrates, in general, when to use each operation.

Use **Data / Memory** for normalization purposes, such as when comparing S21 traces "before" and "after" a change is made or measurement of trace noise. In the following table, the Data/Mem values intuitively show the differences between traces. It is not obvious what Data-Mem is displaying.

S21 values to compare	Data/Mem	Data-Mem
0.5 dB and 0.6 dB	0.1 dB	-39 dB
0.5 dB and 0.7 dB	0.2 dB	-33 dB

Use **Data - Memory** to show the relative differences between two signals. Use for comparison of very small signals, such as the S11 match of two connectors.

In the following table, Data/Mem shows both pairs of connectors to have the same 2 dB difference. However, the second pair of connectors have much better S11 performance (-50 and -52) and the relative significance is shown in the Data-Mem values.

S11 values to compare	Data/Mem	Data-Mem
-10 dB and -12 dB	2 dB	-24 dB
-50 dB and -52 dB	2 dB	-64 dB

Data * Memory and Data + Memory

Use **Data * Memory and Data + Memory** to perform math on an active data trace using data from your own formulas or algorithms rather than data from a measurement. For example, if you want to simulate the gain of a theoretical amplifier placed in series before the DUT, you could:

1. Create an algorithm that would characterize the frequency response of the theoretical amplifier.
2. Enter complex data pairs that correspond to the number of data points for your data trace.
3. Load the data pairs into memory with SCPI or COM commands. The analyzer maps the complex pairs to correspond to the stimulus values at the actual measurement points.

4. Use the **data + memory** or **data * memory** function to add or multiply the frequency response data to the measured data from the active data trace.

Note: The data trace must be configured before you attempt to load the memory.

Trace Statistics

You can calculate and display statistics for the active data trace. These statistics are:

- Mean
- Standard deviation
- Peak-to-peak values

You can calculate statistics for the full stimulus span or for part of it with user ranges.

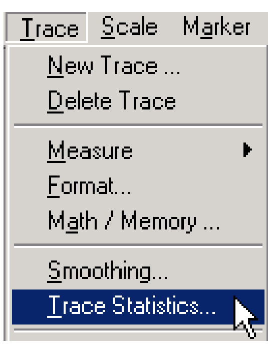
There are nine user ranges per channel. These user ranges are the same as the search domains specified for a marker search in that same channel; they use the same memory registers and thus share the same stimulus spans. If you specified search domains with marker search for a channel, you can recall these same spans by selecting the corresponding user ranges. The user ranges for a channel can overlap each other.

A convenient use for trace statistics is to find the peak-to-peak value of passband ripple without searching separately for the minimum and maximum values.

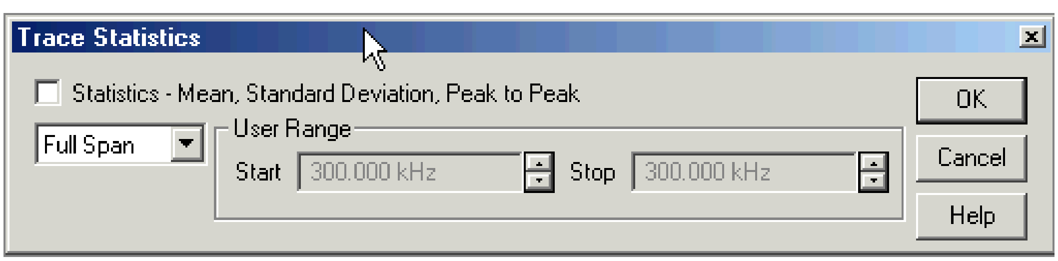
The trace statistics are calculated based on the format used to display the data.

- Rectangular data formats are calculated from the scalar data represented in the display
- Polar or Smith Chart formats are calculated from the data as it would be displayed in Log Mag format

How to activate Trace Statistics



Learn more about using the front panel interface



Trace Statistics dialog box help

Statistics Check to display mean, standard deviation, and peak to peak values for the active trace.

Span Specifies the span of the active trace where data is collected for a math operation. You can define up to 9 user spans per channel with Start and Stop. You can also define the user spans from the Marker Search dialog box.

Start Defines the start of a user span.

Stop Defines the stop of a user span.

Learn more about Trace Statistics (scroll up)

Use Limits to Test Devices

Limit lines allow you to compare measurement data to performance constraints that you define.

- Overview
- Create and Edit Limit Lines
- Display and Test with Limit Lines
- Testing with Sufficient Data Points

Other Analyze Data topics

Overview

Limit lines are visual representations on the PNA screen of the specified limits for a measurement. You can use limit lines to do the following:

- Give the operator **visual guides** when tuning devices.
- Provide **standard criteria** for meeting device specification.
- Show the **comparison** of data versus specifications.

Limit testing compares the measured data with defined limits, and provides optional **Pass or Fail** information for each measured data point.

You can have up to **100** discrete lines for each measurement allowing you to test all aspects of your DUT response.

Limit lines and limit testing are NOT available with **Smith Chart** or **Polar** display format. If limit lines are ON and you change to Smith Chart or Polar format, the analyzer will automatically disable the limit lines and limit testing.

Note: Limit Lines are NOT currently supported on the Frequency Converter Application. They CAN be used on basic Frequency Offset measurements.

Create and Edit Limit Lines

You can create limit lines for all measurement traces. The limit lines are the same color as the measurement trace.

Limit lines are made up of discrete lines with four coordinates:

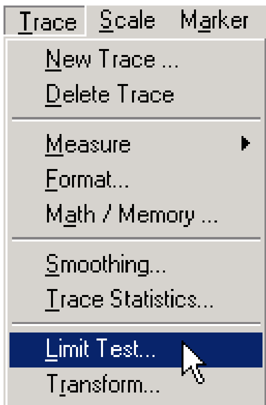
- BEGIN and END stimulus - X-axis values.
- BEGIN and END response - Y-axis values.

How to create and edit Limit Lines

Limits are created, edited, and activated, with the limit table.

Use one of the following methods to show the limit table:

- 

- 

Learn more about using the front panel interface

Limit Table

	TYPE	BEGIN STIMULUS	END STIMULUS	BEGIN RESPONSE	END RESPONSE
1	MIN	1.930000 GHz	1.990000 GHz	-5.000000 dB	-5.000000 dB
2	MAX	1.000000 GHz	1.900000 GHz	-60.000000 dB	50.000000 dB
3	MAX	2.050000 GHz	3.000000 GHz	-50.000000 dB	-60.000000 dB
4	OFF	0 Hz	0.000000 Hz	0.000000 dB	0.000000 dB

Note: To ADD a limit line to the table, change the last limit line to either MAX or MIN

1. In the **Type** area of the Limit Table, select **MIN** or **MAX** for Limit Line 1.
   The MIN value will fail measurements BELOW this limit.
   The MAX value will fail measurements ABOVE this limit.
2. Click **BEGIN STIMULUS** for Limit Segment 1. Enter the desired value.
3. Click **END STIMULUS** for Limit Segment 1. Enter the desired value.
4. Click **BEGIN RESPONSE** for Limit Segment 1. Enter the desired value.
5. Click **END RESPONSE** for Limit Segment 1. Enter the desired value.
6. Repeat Steps 1-5 for each desired limit line.

Displaying and Testing with Limit Lines

After creating limit lines, you can then choose to **display** or **hide** them for each trace. The specified limits remain valid even if limit lines are not displayed.

Limit testing cannot be performed on memory traces.

You can choose to provide a visual and / or audible PASS / FAIL indication.

With limit testing turned ON:

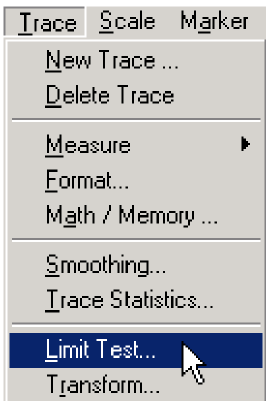
- Any portion of the measurement trace that **fails** is **displayed in red**.
- Any portion of the measurement trace that does **NOT fail** remains unchanged and silent.

PASS is the default mode of Pass / Fail testing. A data point will FAIL only if a measured point falls outside of the limits.

- If the limit line is set to OFF, the entire trace will PASS.
- If there is no measured data point at a limit line stimulus setting, that point will PASS.

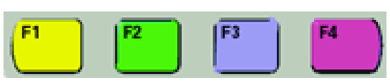
How to display and test with Limit Lines

The following method Invokes a dialog box

- 

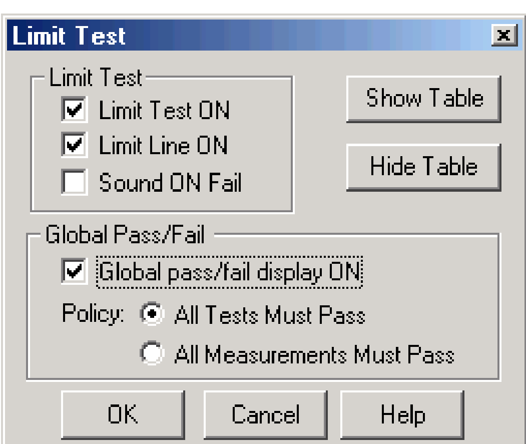
The following method shows the Limit Table and provides Test ON/OFF and Line ON/OFF capability.

- 

+ 

  Sound ON Fail selection is not available from the Active Entry

Learn more about using the front panel interface



Limit Test dialog box help

Limit Test

Limit Test ON Check the box to compare the data trace to the limits and display PASS or FAIL.

Limit Line ON Check the box to make the limits visible on the screen. (Testing still occurs if the limits are not visible.)

Sound ON Fail Check the box to make the PNA beep when a point on the data trace fails the limit test.

Global Pass/Fail

The Pass/Fail indicator provides an easy way to monitor the status of all measurements.

Global pass/fail display ON Check to display the Global Pass/Fail status.

Policy: Choose which of the following must occur for the Global Pass/Fail status to display PASS:

- All Tests (with **Limit Test** ON) Must Pass - This setting reads the results from the Limit Tests. If all tests (with **Limit Test** ON) PASS, then the Global Pass/Fail status will PASS.
- All Measurements Must Pass - This more critical setting shows FAIL unless all measured data points fall within established test limits and Limit Test is ON. **Note:** With this selection, all Limit Tests could PASS, yet the Global Pass/Fail status show FAIL. This is because Limit Tests default to Pass and Global Pass/Fail with 'All Measurements Must Pass' selected defaults to FAIL.

Show Table Shows the table that allows you to create and edit limits.

Hide Table Makes the limits table disappear from the screen.

Note: To ADD a limit line to the table, change the last limit line to either MAX or MIN

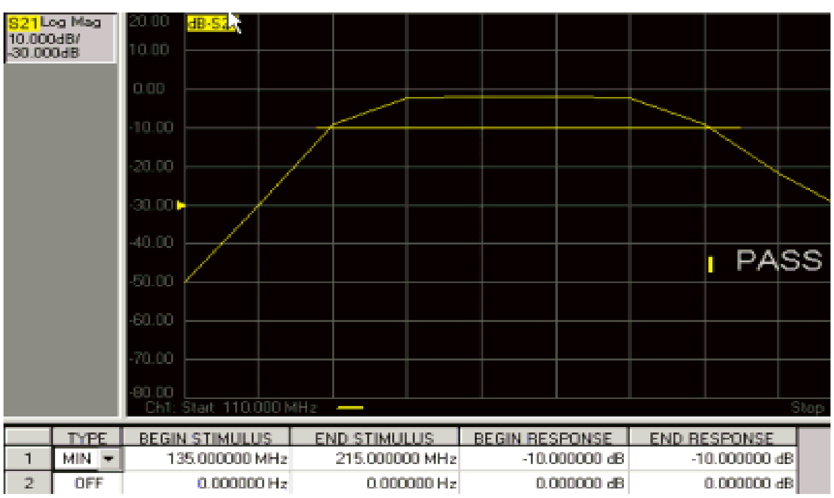
Learn more about displaying and testing with Limits (scroll up)

Testing with Sufficient Data Points

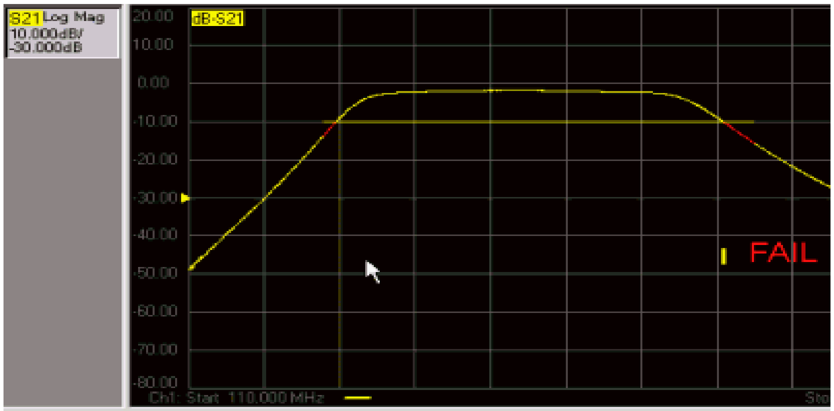
Limits are checked only at the actual measured data points. Therefore, It is possible for a device to be out of specification without a limit test failure indication if the data point density is insufficient.

The following image is a data trace of an actual filter using 11 data points (approximately one every vertical graticule). The filter is being tested with a minimum limit line (any data point under the limit line fails).

Although the data trace is clearly below the limit line on both sides of the filter skirts, there is a PASS indication because there is no data point being measured at these frequencies.



The following image shows the exact same conditions, except the number of data points is increased to 1601. The filter now fails the minimum limit test indicated by the red data trace.



	TYPE	BEGIN STIMULUS	END STIMULUS	BEGIN RESPONSE	END RESPONSE
1	MIN	135.000000 MHz	215.000000 MHz	-10.000000 dB	-10.000000 dB
2	OFF	0.000000 Hz	0.000000 Hz	0.000000 dB	0.000000 dB

Save and Recall a File

The PNA allows you to save and recall files to and from an internal or external storage device in a variety of file formats.

- **How to Save a File**
- **How to Recall a File**
- **Instrument / Calibration State Files (.cst, .sta, .cal)**
- **Measurement Data Files (.prn, .SnP, .cti)**
- **Define Data Saves**
- **Managing Files without a Mouse**

Other Data Outputting topics

How to Save a File

Use one of the following methods:

- 

- 

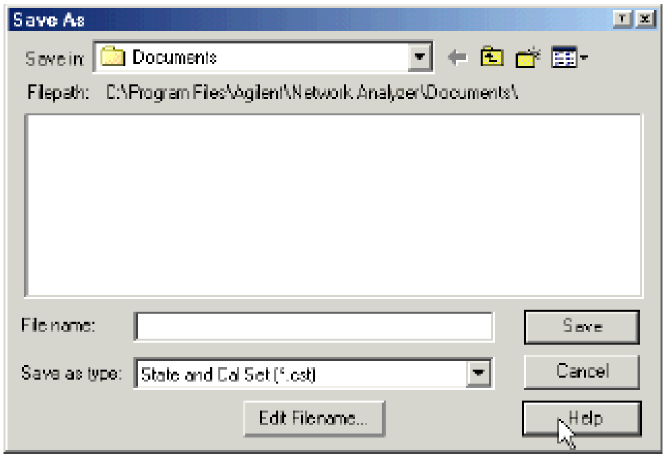
Save Immediately saves the PNA state and possibly calibration data to the filename and extension you used when you last performed a Save. Only .cst and .sta files are remembered when Save is performed. This file will be overwritten the next time you click Save. To prevent this, use one of the following methods.

Save As Invokes the Save As dialog box.

Auto Save (Only available from the Active Entry keys) Saves state and calibration data to the internal hard disk in the C:\Program Files\Agilent\Network Analyzer\Documents folder. A filename is generated automatically using the syntax "atxxx.cst"; where xxx is a number that is incremented by one when a new file is Auto Saved.

Note: You can NOT save Frequency Converter Application .S2P files using this method. To learn how, see Using FCA, Save Data.

Learn more about using the front panel interface



Save As dialog box help

Save in Allows you to navigate to the directory where you want to save the file.

File name Displays the filename that you either typed in or clicked on in the directory contents box.

Save as type

The following file types save **Instrument states and Calibration data**. These file types are only recognized by Agilent PNA Series analyzers. Learn more about these file types.

- *.cst - save both Instrument state and a reference to the Cal Set data
- *.sta - save instrument state only (**no** calibration data)
- *.cal - save Calibration data only (no Cal Set)

The following file types save **Measurement data** for use in spreadsheet or CAE programs. Click to learn more about these file types.

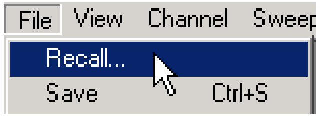
- *.prn
- *.s1p , *.s2p, *.s3p
- *cti (citifile)

Note: To save the PNA screen as .bmp, .jpg, or .png graphics file types, click **File / Print to File.** Learn more.

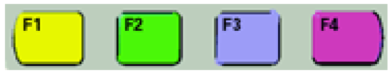
Save Saves the file to the specified file name and directory.

How to Recall (open) a file

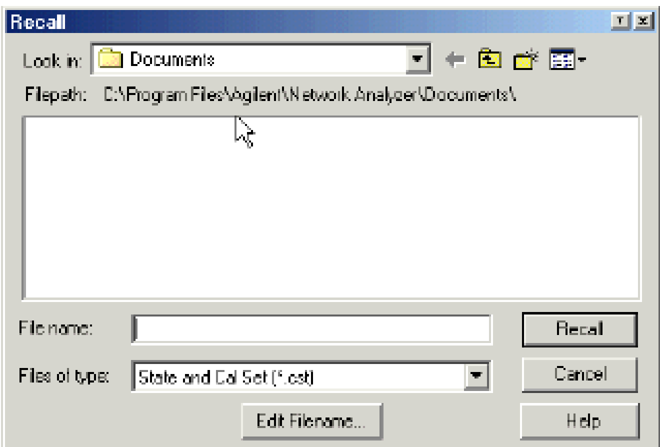
Use one of the following methods:

- 

- Press 

repeatedly, then 

Learn more about using the front panel interface



Recall dialog box help

Look in Allows you to select the directory that contains the file that you want to recall.

File name Displays the filename that you either typed in or clicked on in the directory contents box.

Files of type Allows you view and select files that are listed in categories of a file type.

Recall Recalls the file displayed in the file name box.

Note: *s1p, *s2p, *s3p files cannot be recalled by the PNA.

Instrument State / Calibration Files

You can save, and later recall, instrument setups and calibration data. Almost every PNA setting is saved and recalled with the Instrument State. The following PNA settings are NOT saved and recalled with Instrument State:

1. GPIB address
2. RF power ON/OFF (Learn more.)
3. Test set I/O settings

There are three file types that save and recall these files:

- ***.sta** files contain **Instrument State** but no cal data or reference to cal data.

- ***.cal** files contain **Calibration** data; NOT as a reference to the Cal Set.
- ***.cst** files contain both **Instrument State AND** a reference to the Cal Set being used. Learn more about Cal Sets. Recalling a *.cst file saves time recalling both the instrument state and the calibration data for each measurement.

Measurement Data Files

Measurement data is saved as ASCII file types for use in a spreadsheet or CAE programs. The following three file types are used by the PNA. You can select the content and the format of *.SnP files and *.cti files through the Define Data Saves dialog box.

***.prn files**

***.SnP (Touchstone)**

***.cti (Citifile)**

*.prn Files

- Read directly into rows and columns by spreadsheet software, such as Microsoft® Excel.
- Contain formatted and corrected stimulus and response data.
- Contain comma-separated list for the current active trace ONLY.
- Are Output only - it cannot be read by the analyzer.

Example:

"S_{11} Log Mag"

"**Frequency (Hz)**",	"**dB**"
3.000000e+005 ,	**-3.528682e+001 ,**
4.529850e+007 ,	**-2.817913e+001 ,**
9.029700e+007 ,	**-3.216808e+001 ,**
1.352955e+008 ,	**-3.101017e+001 ,**

SnP Format (*.s1p, *.s2p, *.s3p)

This file format is used by CAE programs such as Agilent's Microwave Design System (MDS) and Advanced Design System (ADS).

Note: Frequency Converter Application .S2P files are saved using a different method. See Using FCA, Save Data.

- SnP data is Output only; it cannot be read by the analyzer.
- SnP data can be saved in various formats. See Define Data Saves
- The amount of data that is saved depends on the file type that you specify:

File Type	# of Ports	# of S-parameters saved
***.s1p**	1	1 S-parameter
***.s2p**	2	4 S-parameters
***.s3p**	3	9 S-parameters

SnP data is generally used to gather all 4 S-parameters for a fully corrected measurement. The PNA fulfills this request using the data that is available on the channel of the active measurement.

- If full 2-port correction is applied, then valid data is returned for all 4 corrected s-parameters.
- If correction is NOT applied, the PNA returns as much applicable raw data as possible using S-parameter measurements on the selected channel.
- Data that is not available is zero-filled.

For example, if correction is NOT applied and the active measurement is S11, and an S21 measurement also exists on the channel, then data is returned for the S11 and S21 measurements. Data for S12 and S22 is not available and therefore returned as zeros.

All valid data is saved using the same format and settings as the active measurement. For example, if the active measurement settings include port extensions, time domain, and smoothing, then all valid data is processed and saved using these same settings.

SnP Data Output

SnP files contain header information, stimulus data, a response data pair for EACH S-parameter measurement. The only difference between S1P, S2P, and S3P files is the number of S-parameters that are saved.

The following is a sample of **Header information:**

```
!Agilent Technologies,E8362B,US42340026,Q.03.54
!Agilent E8362B: Q.03.54
!Date: Friday, April 25, 2003 13:46:41
!Correction: S11(Full 2 Port SOLT,1,2) S21(Full 2 Port SOLT,1,2) S12(Full 2 Port
SOLT,1,2) S22(Full 2 Port SOLT,1,2)
!S2P File: Measurements:S11,S21,S12,S22:
# Hz  S RI R 50
```

Note: Although the following shows Real / Imag pairs, the format could also be LogMag / Phase or LinMag / Phase

***.s1p Files**

Each record contains 1 stimulus value and 1 S-parameter (total of 2 values)

Stim Real (Sxx) Imag(Sxx)

***.s2p Files**

Each record contains 1 stimulus value and 4 S-parameters (total of 8 values)

Stim Real (S11) Imag(S11) Real(S21) Imag(S21) Real(S12) Imag(S12) Real(S22) Imag(S22)

***.s3p Files**

Each record contains 1 stimulus value and 9 S-parameters (total of 18 values)

Stim   Real (S11) Imag(S11) Real(S12) Imag(S12) Real(S13) Imag(S13)
Real (S21) Imag(S21) Real(S22) Imag(S22) Real(S23) Imag(S23)

Real (S31) Imag(S31) Real(S32) Imag(S32) Real(S33) Imag(S33)

Note: *.s3p files require 3 lines per stimulus.

*cti CitiFiles

Citifile format is compatible with the Agilent 8510 Network Analyzer and Agilent's Microwave Design System (MDS). You can do the following using citifiles :

- save the active trace, or all traces. (see Define Data Saves
- save formatted or unformatted citifile data

Save Formatted data

1. Set the format using Define Data Saves.
2. Click **File** then **Save As**
3. Select **Citifile Formatted Data (*cti)**

- See Note under Recalling Citifiles Using the PNA
- On the data access map, Formatted data is taken from location 2 or 4.

Save Unformatted data

1. Click File Save As
2. Select **Citifile Data Data (*cti)**

On the data access map, Unformatted data is taken from the block just before Format.

Recalling Citifiles Using the PNA

To read citifiles using the PNA, click **File** then **Recall.** Specify **(*.cti)**

The recalled files are displayed on the PNA using new windows and channels.

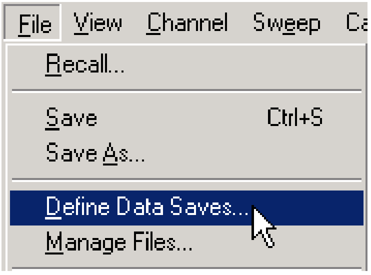
- The new window numbers begin with the next available window number.
- New channel numbers begin with channel 9.

When recalling citifiles, an error is displayed if you exceed the PNA windows and channels limitations.

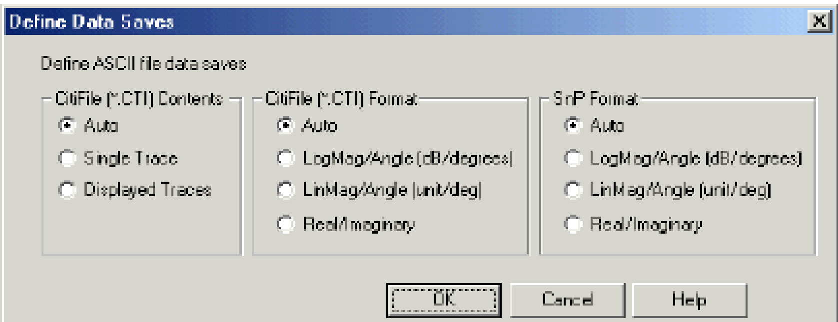
Note: Recalled citifile data is ALWAYS displayed on the PNA using LogMag format, regardless of how the file was stored.

Define Data Saves

How to select Define Data Saves



Learn more about using the front panel interface



Define Data Saves dialog box help

CitiFile Contents Determines what is saved to a .cti file.

Auto - Saves the active trace. Additional traces are saved if correction is ON and Full 2-port or Full 3-port calibration is performed. For Full 2-port calibration, 4 traces are saved. For Full 3-port calibration, 9 traces are saved.

Single Trace - Saves the active trace in the currently selected window.

Displayed Traces - Saves all displayed data traces

Citifile Formatted Data

Auto - Output is in the currently selected trace format. If other than LogMag, LinMag, or Real/Imag, then output is in Real/Imag.

LogMag, LinMag, Real/Imag - Select output format. All **LogMag** and **LinMag** output is in degrees.

SnP Formatted Data (.s1p, .s2p, .s3p)

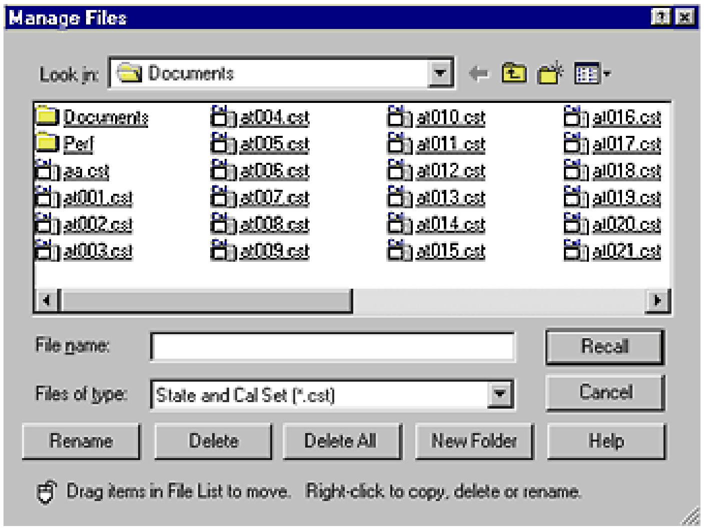
Auto - Output is in the currently selected trace format. If other than LogMag, LinMag, or Real/Imag, then output is in Real/Imag.

LogMag, LinMag, Real/Imag - Select output format. All **LogMag** and **LinMag** output is in degrees.

Manage Files without a Mouse

The Manage Files dialog box is designed to be used from the front panel. It performs the same function as Windows Explorer, but can be used without the use of a mouse or keyboard.

Learn more about using the Front-panel interface.



Manage Files dialog box help

Recall Opens a Network Analyzer State already stored in memory.

Rename Renames a file that is selected in the open folder.

Delete Removes a selected file from the open folder.

Delete All Removes all files of the file type selected that appear in the open folder.

New folder Create a new folder and give it a name

If using a mouse:

Files can be moved by dragging them in the file contents.

Right click in the topic page to carry out file management features.

Drive Mapping

Drive mapping allows you to share disk drives between the PNA and an external computer. You can either map from the PNA, or from your PC, to the other.

- From the PNA, map to a drive on an External PC
- From an External PC, map to a drive on the PNA

To prepare for Drive Mapping:

1. Both the PC and PNA must be connected to a shared computer network
2. You must know the full computer name of the PC (or analyzer) you are mapping **TO**. Tell me how
3. Your logon and password on the analyzer must be the same as that on the external PC. You can add your PC logon to the analyzer. Tell me how

---

Note: These procedures require a mouse and keyboard. Also, the external PC must have Windows NT 4.0 (or later).

From the Analyzer, map to a drive on the External PC

1. On the external computer desktop, go to **Windows Explorer**. In the listing of drives, right click on the drive you want to share. Click **Sharing**.
2. In the dialog box, select **Shared As**. In the **Share Name** box, use the arrow key or type in a share name for the drive. For example: **C$**. Click **OK**.
3. On the analyzer desktop, click **Windows Explorer**. From the **Tools** menu, click **Map Network Drive**. (To get to the analyzer desktop, click **View**, then click **Title Bars)**
4. If you would like to connect to your external PC using a different logon, click **Connect using a different Logon**. This logon must be registered on the analyzer and you must be currently logged on the external PC using this logon.
    1. In the **Connect as** box, type your logon name.
       The logon name and password must be exactly the same on both the external PC and the analyzer.
    2. In the **Password** box, type the logon password that you use on the external computer. Click **OK**. The logon name and password must be exactly the same on both the external PC and the analyzer.
5. In the **Folder** box, type \(full computer name of analyzer)\share name (from step 2). (For example: **\SLT1234\C$**)
6. Click **Finish**.

From an External PC, map to a drive on the Analyzer

1. On the analyzer desktop, click **Windows Explorer**. Right click on the drive you want to share. Click on **Sharing...**

2. In the dialog box, select **Shared this folder**. In the **Share Name** box, type in a share name for the drive. For example: **C$**. Click **OK**.

3. On the external PC desktop, click **Windows Explorer**. From the **Tools** menu, click **Map Network Drive**.

4. If the current logon on your PC is different from the current logon on the analyzer, click **Connect using a different Logon** to connect to using the current analyzer logon, .This logon must be registered on the external PC. To see the current logon on either the PC or analyzer, hold **Ctrl - Alt**, and press **Delete**.

    1. In the **Connect as** box, type the logon currently being used by the analyzer.

    2. In the **Password** box, type the logon password that you use on the external computer. Click **OK**

5. In the **Folder** box, type *computername (prep1)**share name* (from step 2). (For example: **\\SLT1234\C$**)

6. Click **Finish**.

Print a Displayed Measurement

The analyzer allows you to print a displayed measurement to a printer or to a file. The printer can be either a networked or local. Click on a button to learn how to connect to a printer or print a displayed measurement.

Connecting a Printer

Printing

Other Outputting Data topics

Connecting a Printer

You can connect your printer to the PNA using three different connector types:

- Parallel connector
- Serial connector
- USB

Note: Early PNAs have a Centronics connector for connecting a printer. An adapter (36-pin male - 1284-C - to 25-pin female) was shipped with those PNAs to allow connection with a standard parallel printer cable.

CAUTION: Do NOT connect your printer to the 25-pin female port labeled **Ext. Test Set Interface.** Voltage levels of signal lines may damage the printer's I/O.

To Add a Printer

Note: If you try to print from the PNA application and the **Add Printer Wizard** appears, click **Cancel** and add the printer using the following procedure.

1. From the PNA application, click **View** then click **Minimize Application**
2. On the Windows taskbar, click **Start**, point to **Settings**, then click **Printers**.
3. Double-click **Add Printer**.
4. Follow the instructions in the **Add Printer** Wizard.

For more information, refer to Microsoft Windows Help or your printer documentation.

Printing

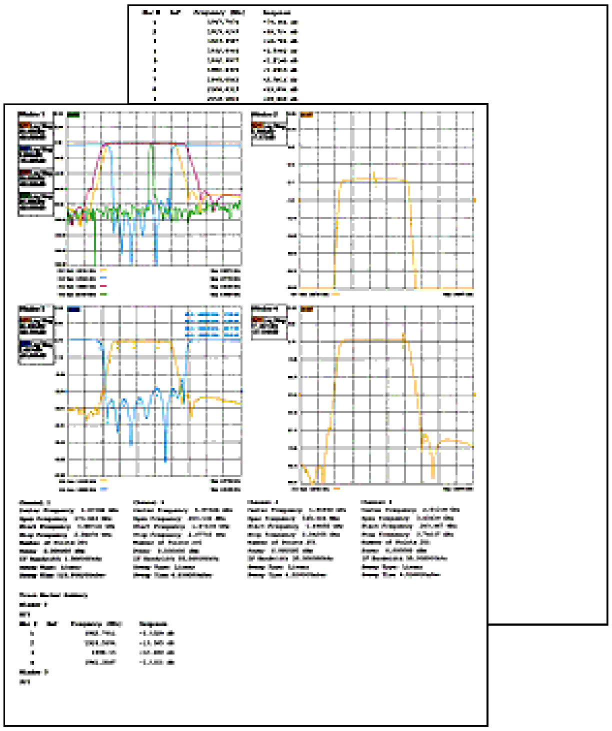
Print a Hardcopy

Print Options

Print to File

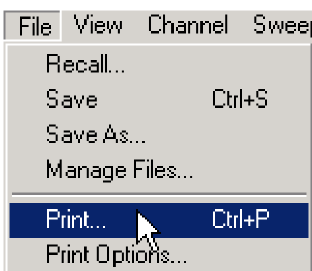
The measurement information on the screen can be printed to any local or networked printer that is connected to the analyzer. The amount of measurement information printed is selected by the print options settings.

The graphic below shows an example of how a screen-capture image appears when printed. The print options settings allows you to customize the printed form of the measurement information.



How to Print a Hardcopy

Use one of the following methods

- 

- 

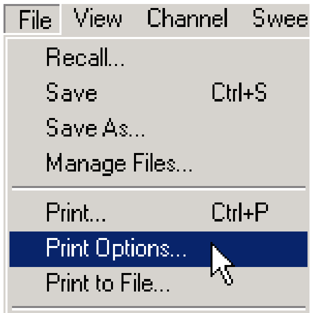
Learn more about using the front panel interface

Note: For information on the choices in the Print dialog box, see Windows 2000 Help.

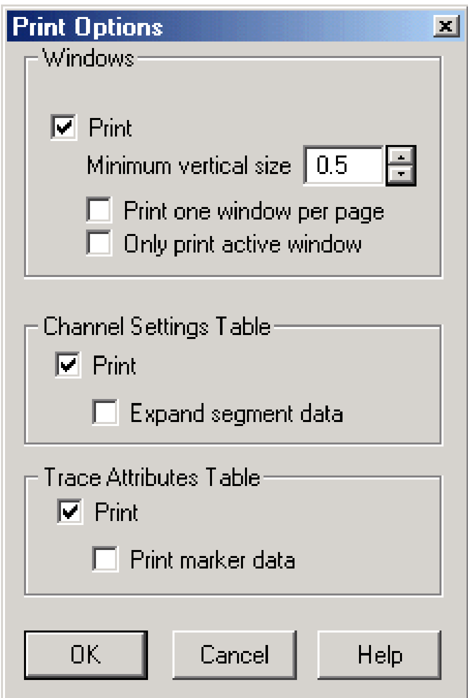
Print Options

The Print Options Dialog Box allows flexibility in the appearance that measurement data is printed. After selecting the options, click **Print...** to obtain a hard-copy.

How to select Print Options



Learn more about using the front panel interface



Print Options dialog box help

Windows

Print Check to print measurement windows.

Minimum vertical size Adjust to change the amount of a page that the measurement window fills. The adjustment range is from .4 to 1.0 of a page.

Print one window per page Check to print one window per page. Clear to print all selected windows without a forced page break.

Only print active window Check to print only the active window. Clear to print all windows.

Channel Settings Table

Print Check to print the channel settings table.

Expand segment data Check to print segment data. The amount of data depends on how much the sweep is segmented.

Trace Attributes Table

Print Check to print the Trace Attributes Table. The Trace Attributes are measurement type, correction factors ON or OFF, smoothing, options, and marker details. The Trace Attributes are listed by Trace ID# (1 to 4), for each window.

Each Trace ID# can have multiple entries depending on the number of markers associated with the trace. The marker details are marker number, position and response. If there are multiple markers on a trace, the trace attributes are only shown for the first marker. However, the trace attributes for the first marker apply to all other markers on that trace.

The options column can have one or more options. **D** for Delay, **M** for Marker, **G** for Gating. Multiple options selected would appear as follows: DMG.

Print marker data Check to print all marker data. The amount of data depends on how many markers are created.

Print to a File

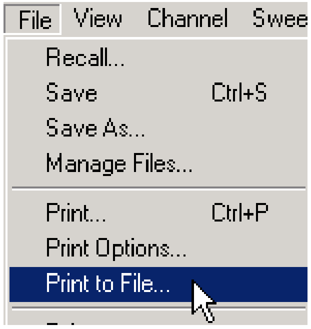
The analyzer can save a screen-capture image in any of the following formats:

- **.bmp** (bitmap) format
- **.jpg** format
- **.png** format

The analyzer automatically saves the file to the current path. If not previously defined, the analyzer automatically selects the default path C:\Program Files\Agilent\Network Analyzer\Documents\

A .bmp file, like a .prn file, can be imported into software applications such as Microsoft Excel, Word, or Paint to display a screen-capture image.

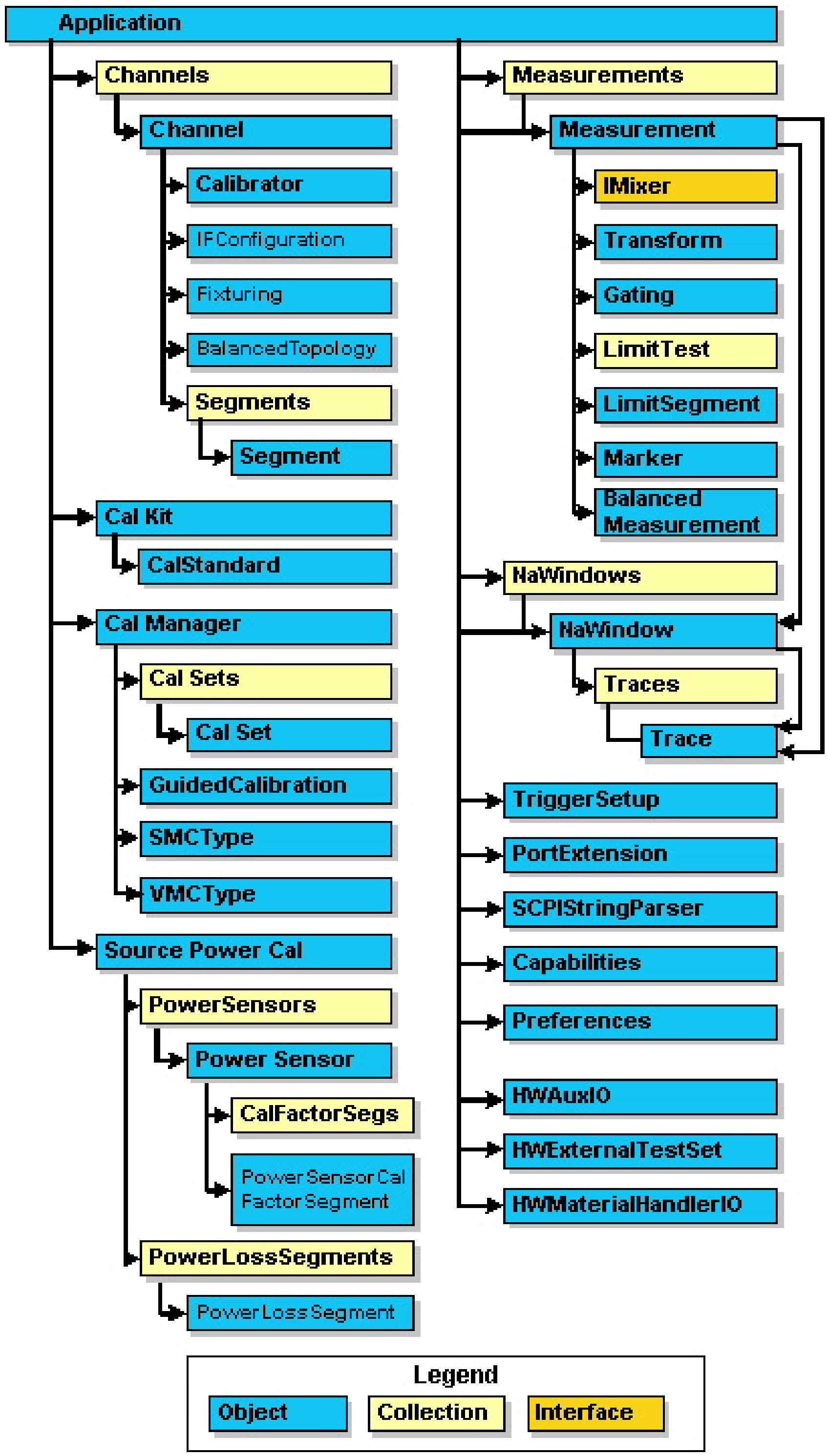
How to Print to a File



Learn more about using the front panel interface

See Save and Recall files for more information.

PNA Object Model



See a list of obsolete commands.

Application Object

Description

The Application object is the highest object in the PNA object model. This object presents methods and properties that affect the entire analyzer, rather than a specific channel or measurement. For example, the application object provides the GetIDString method. There's only one ID string for the instrument, unrelated to the channel or parameter being measured. Likewise, the TriggerSignal Property is global to the instrument. You can elect to use an internally generated (free run) trigger or a manual trigger. Either way, that type of trigger generation will be used on all measurements, on all channels. Therefore, it is under the Application object.

Accessing the Application object

This object is unique in that you must **create** this object rather than just get a handle to it.

```
Dim app As AgilentPNA835x.Application
Set app = CreateObject("AgilentPNA835x.Application", <analyzerName>)
```

Replace <analyzerName> with the full computer name of your PNA. For example, "My PNA". See Change Computer Name.

See Also:

PNA Automation Interfaces

The PNA Object Model

Getting a Handle to an Object.

(**Bold** Methods or Properties provide access to a child object)

Methods	Interface See History	Description
ActivateWindow	IApplication	Makes a window object the Active Window.
AllowAllEvents	IApplication	Monitors all events
owEventCategory	IApplication	Monitors an event category
AllowEventMessage	IApplication	Monitors an event
AllowEventSeverity	IApplication	Monitors an event severity level
BuildHybridKit	IApplication	Defines the user kit as port1kit + port2kit.
Channel	IApplication	Stimulus values like frequency, power, IF bandwidth, and number of points.
CreateCustomMeasurementEx	IApplication3	Creates a new custom measurement with initialization.
CreateCustomMeasurement	IApplication	**OBSOLETE** - CreateCustomMeasurementEx Method
CreateMeasurement	IApplication	Creates a new measurement.
CreateSParameter	IApplication	**OBSOLETE** - Use Create SParameterEX Method

CreateSParameterEx	IApplication	Creates a new S-Parameter measurement with a 3-port load.
DeleteShortCut	IApplication	Removes a macro (shortcut) from the list of macros
DisallowAllEvents	IApplication	Monitors NO events
DoPrint	IApplication	Prints the screen to the active Printer.
ExecuteShortcut	IApplication	Executes a macro (shortcut) stored in the analyzer.
GetAuxIO	IApplication	Returns a handle to the AuxIO interface
GetCalManager	IApplication	Returns a handle to the CalManager interface
GetExternalTestSetIO	IApplication	Returns a handle to the ExternalTestSet IO interface
GetMaterialHandlerIO	IApplication	Returns a handle to the MaterialHandlerIO interface
GetShortcut	IApplication	Returns the title and path of the specified macro (shortcut).
LaunchCalWizard	IApplication	Launches the Cal Wizard
ManualTrigger	IApplication	Triggers the analyzer when TriggerSignal = naTriggerManual.
Preset	IApplication	Resets the analyzer to factory defined default settings.
PrintToFile	IApplication	Saves the screen data to bitmap (.bmp) file of the screen.
PutShortcut	IApplication	Puts a Macro (shortcut) file into the analyzer.
Quit	IApplication	Ends the Network Analyzer application.
Recall	IApplication	Recalls a measurement state, calibration state, or both from the hard drive into the analyzer.
RecallKits	IApplication	Recalls the calibration kits definitions that were stored with the SaveKits command.
Reset	IApplication	Removes all existing windows and measurements.
RestoreCalKitDefaults	IApplication	Restores the factory defaults for the specified kit.
RestoreCalKitDefaultsAll	IApplication	Restores the factory defaults for all kits.
Save	IApplication	Saves files to disk
SaveCitiDataData	IApplication4	Saves UNFORMATTED trace data to .cti file.
SaveCitiFormattedData	IApplication4	Saves FORMATTED trace data to .cti file.
SaveKits	IApplication	Saves all cal kits to disk.
SetFailOnOverRange	IApplication	Causes over range values to return an error code
ShowStatusBar	IApplication	Shows and Hides the Status Bar.

ShowStimulus	IApplication	Shows and Hides Stimulus information.
ShowTitleBars	IApplication	Shows and Hides the Title Bars.
ShowToolbar	IApplication	Shows and Hides the specified Toolbar.
UserPreset	IApplication7	Performs a User Preset.
UserPresetLoadFile	IApplication7	Loads an existing instrument state file (.sta or .cst) to be used for User Preset.
UserPresetSaveState	IApplication7	Saves the current instrument settings as UserPreset.sta.
Properties		**Description**
ActiveCalKit	IApplication	Returns a pointer to the kit identified by kitNumber.
ActiveChannel	IApplication	Returns a handle to the Active Channel object.
ActiveMeasurement	IApplication	Returns a handle to the Active Measurement object.
ActiveNAWindow	IApplication	Returns a handle to the Active Window object.
ArrangeWindows	IApplication	Sets or returns the arrangement of all the windows.
CalKitType	IApplication	Sets or returns the calibration kit type for to be used for calibration or for kit modification. Shared with the CalKit object.
Capabilities	IApplication4	Return capabilities of the remote PNA.
Channels	IApplication	Collection for iterating through the channels
CoupledMarkers	IApplication	Sets (or reads) coupled markers ON and OFF
DisplayAutomationErrors	IApplication2	Enables or disables automation error messages from being displayed on the screen. U
DisplayGlobalPassFail	IApplication6	Shows or hides the dialog which displays global pass/fail results.
ExternalALC	IApplication	Sets or returns the source of the analyzer leveling control.
GPIBAddress	IApplication8	Sets and returns the PNA GPIB address.
GPIBMode	IApplication	Makes the analyzer the system controller or a talker/listener.
IDString	IApplication	Returns the model, serial number and software revision of the analyzer
LocalLockoutState	IApplication4	Prevents use of the mouse, keyboard, and front panel while your program is running.
Measurements	IApplication	Collection for iterating through the Application measurements.
MessageText	IApplication	Returns text for the specified eventID

NaWindows	IApplication	Collection for iterating through the Application windows.
NumberOfPorts	IApplication	Returns the number of hardware source ports on the PNA
Options	IApplication	Returns the options on the analyzer
Port Extensions	IApplication	
Preferences	IApplication5	Preferences for saving citifiles.
ScpiStringParser	IApplication	Provides the ability to send a SCPI command from within the COM command.
SecurityLevel	IApplication4	Turns ON or OFF the display of frequency information.
SICL	IApplication5	Allows control of the PNA via SICL
SICLAddress	IApplication8	Sets and returns the PNA SICL address
SourcePowerCalibrator	IApplication2	Allows capability for performing source power calibrations.
SourcePowerState	IApplication	Turns Source Power ON and OFF.
SystemImpedanceZ0	IApplication	Sets the analyzer impedance value.
SystemName	IApplication	Returns the full computer name of the PNA.
TriggerDelay	IApplication	Sets or returns the delay time for a trigger.
TriggerSetup	IApplication4	Controls triggering for the entire PNA application.
TriggerSignal	IApplication	**Obsolete** - use Source Property
TriggerType	IApplication	**Obsolete** - use Scope Property
UserPresetEnable	IApplication7	'Checks' and 'clears' the enable box on the User Preset dialog box.
VelocityFactor	IApplication	Sets the velocity factor to be used with Electrical Delay, Port Extensions, and Time Domain marker distance calculations.
Visible	IApplication	Makes the Network Analyzer application visible or not visible.
WindowState	IApplication	Sets or returns the window setting of Maximized, Minimized, or Normal. Shared with the NAWindow Object

Events	**Interface**	**Description**
OnCalEvent	IApplication	Triggered by a calibration event.
OnChannelEvent	IApplication	Triggered by a channel event.
OnDisplayEvent	IApplication	Triggered by a display event.

OnHardwareEvent	IApplication	Triggered by a hardware event.
OnMeasurementEvent	IApplication	Triggered by a measurement event.
OnSCPIEvent	IApplication	Triggered by a SCPI event.
OnSystemEvent	IApplication	Triggered by a system event.
OnUserEvent	IApplication	For future use

IApplication History

Interface	Introduced with PNA Rev:
IApplication	1.0
IApplication2	3.0
IApplication3	3.2
IApplication4	3.5
IApplication5	4.0
IApplication6	5.0
IApplication7	5.0
IApplication8	5.2

CalFactorSegments Collection

Description

A collection object that provides a mechanism for iterating through the segments of a power sensor cal factor table.

Accessing the CalFactorSegments collection

```
Dim app As AgilentPNA835x.Application
Set app = CreateObject("AgilentPNA835x.Application", <analyzerName>)

Dim calFact As CalFactorSegments
Set calFact = app.SourcePowerCalibrator.PowerSensors(1).CalFactorSegments
```

See Also:

Collections in the Analyzer

The PNA Object Model

Methods	Description
Add	Adds a PowerSensorCalFactorSegment object to the collection
Item	Use to get a handle to a PowerSensorCalFactorSegment object in the collection.
Remove	Removes an object from the collection.
Properties	**Description**
Count	Returns the number of objects in the collection.
Parent	Returns a handle to the Parent object (PowerSensor) of this collection.

Calibrator Object

See Calibrator Methods and Properties

See ICalData Interface for putting and getting typed Calibration data.

Description

The Calibrator object, a child of the channel, is used to perform an Unguided calibration.

Note: You can NOT perform a full 3 or 4-port using the Calibrator object; you must use the GuidedCalibration object.

There must be a measurement present for the calibrator to use or you will receive a "no measurement found" error. Therefore, to perform a 2-port cal, you must have any S-parameter measurement on the channel. For a 1-port measurement, you must have the measurement (S11 or S22) on the channel. The same is true for a response measurement.

There are a number of approaches to calibration with the calibrator object:

- You can collect data yourself and download it to the ACQUISITION buffer. The acquisition buffer holds the actual measured data for each standard. See the PNA data map.

1. Calibrator.SetCalInfo
2. Connect a standard
3. Trigger a sweep
4. Retrieve the data for the standard
5. Download the data - calibrator.putStandard
6. Repeat for each standard
7. Calibrator.CalculateErrorCoefficients

- You can tell the calibrator to acquire a standard. In this case, the calibrator collects the data and places it in the ACQUISITION buffer.

1. Calibrator.SetCalInfo
2. Connect a standard
3. Calibrator.AcquireCalStandard2
4. Repeat for each standard
5. Calibrator.CalcuateErrorCoefficients

- You can put previously-retrieved error terms in the error correction buffer.

1. PutErrorTerm
2. Repeat for each term
3. Measurement.Caltype = pick one

- You can also "piece together" a 2-port cal from two 1-port cals (S11 and S22) and four response (thru) cals. The system will detect that all the standards needed for a 2-port cal have been acquired even though they may not have gathered at the same time.

Accessing the Calibrator object

```
Dim app As AgilentPNA835x.Application
Set app = CreateObject("AgilentPNA835x.Application", <analyzerName>)

Dim cal As ICalibrator
Set cal = app.ActiveChannel.Calibrator
```

See Also:

- PNA Automation Interfaces
- The PNA Object Model
- Learn about reading and writing Calibration data.

Methods	Interface See History	Description
AcquireCalConfidenceCheckECAL	ICalibrator	**Obsolete -** use AcquireCalConfidenceCheckECALEx
AcquireCalConfidenceCheckECALEx	**ICalibrator4**	Transfers ECAL confidence data into analyzer memory
AcquireCalStandard	ICalibrator	**Obsolete** - use AcquireCalStandard2
AcquireCalStandard2	ICalibrator	Causes the analyzer to measure a calibration standard. Also provides for sliding load.
CalculateErrorCoeffecients	ICalibrator	Generates Error Terms from standard and actual data in the error correction buffer.
DoECAL1Port	ICalibrator	**Obsolete** - use DoECAL1PortEx
DoECAL1PortEx	**ICalibrator4**	Completes a 1 port ECAL
DoECAL2Port	ICalibrator	**Obsolete** - use DoECAL2PortEx
DoECAL2PortEx	**ICalibrator4**	Completes a 2 port ECAL
DoneCalConfidenceCheckECAL	ICalibrator	Concludes an ECAL confidence check
GetECALModuleInfo	ICalibrator	**Obsolete** - use Get ECALModuleInfoEx
Get ECALModuleInfoEx	**ICalibrator4**	Returns information about the attached module
getErrorTerm	ICalibrator	**Obsolete**- replaced by GetErrorTermByString
getStandard	ICalibrator	**Obsolete** - replaced by GetStandardByString
putErrorTerm	ICalibrator	**Obsolete** - replaced by PutErrorTermByString

putStandard	ICalibrator	**Obsolete** - replaced by PutStandardByString
SaveCalSets	ICalibrator	Writes new or changed Cal Sets to disk. Shared with the CalManager Object
setCalInfo	ICalibrator	Specifies the type of calibration and prepares the internal state for the rest of the calibration.
Properties	**Interface**	**Description**
AcquisitionDirection	ICalibrator	Specifies the direction in a 2-Port cal using one set of standards.
ECALCharacterization	**ICalibrator2**	**Obsolete** - use ECALCharacterizationEx
ECALCharacterizationEx	**ICalibrator4**	Specifies which set of characterization data within an ECal module will be used for ECal operations with that module.
ECALCharacterizationIndexList	**ICalibrator6**	Returns a list of characterizations stored in the specified ECal module.
ECAL Isolation	ICalibrator	**Obsolete** - No replacement
ECALModuleNumberList	**ICalibrator6**	Returns a list of index numbers to be used for referring to the ECal modules that are currently attached to the PNA.
ECALPortMap	**ICalibrator3**	**Obsolete** - use ECALPortMapEx
ECALPortMapEx	**ICalibrator4**	Specifies which ports of the ECal module are connected to which ports of the PNA.
IsECALModuleFound	ICalibrator	**Obsolete** - use IsECALModuleFoundEx
IsECALModuleFoundEx	**ICalibrator4**	**Obsolete** - use ECALModuleNumberList and ECALCharacterizationIndexList
OrientECALModule	**ICalibrator3**	Specifies if the PNA should perform orientation of the ECal module during calibration.
Simultaneous2PortAcquisition	ICalibrator	Allows the use of 2 sets of standards at the same time.

ICalibrator History

Interface	Introduced with PNA Rev:
ICalibrator	1.0
ICalibrator2	3.1
ICalibrator3	3.1
ICalibrator4	3.5
ICalibrator6	4.87

ICalData Interface

Description

Contains methods for putting Calibration data in and getting Calibration data out of the analyzer using typed data. This interface transfers data more efficiently than variant data.

There is also an ICalData Interface on the CalSet Object

Learn about reading and writing Calibration data.

Methods	Description
getErrorTermComplex	Retrieves error term data
getStandardComplex	Retrieves calibration data from the acquisition data buffer (before error-terms are applied).
putErrorTermComplex	Puts error term data
putStandardComplex	Puts calibration data into the acquisition data buffer (before error-terms are applied).
Properties	**Description**
None	

ICalData History

Interface	Introduced with PNA Rev:
ICalData	1.0

CalKit Object

Description

The calkit object provides the properties and methods to access and modify a calibration kit. The calkitType property can be set from two objects:

- Application object - app.calKitType
- CalKit object - calKit.calKitType

Both of these commands specify or read the calibration kit type. When specified, the cal kit also becomes the Active cal kit.

Accessing a CalKit object

To get a handle to a cal kit, use **app.ActiveCalKit**.

The calKit object behaves differently from other objects in the system in that you can only have a handle to **one** cal kit -- the active calkit. Therefore, when you change the calkitType from either the Application object or the CalKit object, you may also be changing the object to which you may have other references.

For example, the following example specifies two calKit type objects and in turn, assigns them to two different variables: ck1 and ck2.

```
Dim app As AgilentPNA835x.Application
Dim ck1 As calKit
Dim ck2 As calKit

Set app = CreateObject("AgilentPNA835x.Application", <analyzerName>)
app.CalKitType = naCalKit_User1
Set ck1 = app.ActiveCalKit
ck1.Name = "My CalKit1"

app.CalKitType = naCalKit_User2
Set ck2 = app.ActiveCalKit
ck2.Name = "My CalKit2"

Print "ck1: " & ck1.Name
Print "ck2: " & ck2.Name
```

When the pointer to each of these kits is read (printed), they each have a pointer to the last kit to be assigned to the Active cal kit:

```
ck1: My CalKit2
ck2: My CalKit2
```

See Also:

PNA Automation Interfaces

The PNA Object Model

(**Bold** Methods or Properties provide access to a child object)

Methods	Description
getCalStandard	Returns a handle to a calibration standard for modifying its definitions.
GetStandardsForClass	Returns the calibration standard numbers for a specified calibration class.
SetStandardsForClass	Sets the calibration standard numbers for a specified calibration class

Properties	Description
CalKitType	Sets or returns the calibration kit type for to be used for calibration or for kit modification. Shared with the Application object.
Name	Sets and returns the name of the cal kit
PortLabel	Labels the ports for the kit; only affects the cal wizard annotation.
StandardForClass	**Obsolete** Use GetStandardForClass and SetStandardForClass. Maps a standard device to a cal class.

ICalKit History

Interface	Introduced with PNA Rev:
ICalKit	1.0

CalManager Object

Description

Use this interface to list, save, and delete Cal Sets.

Accessing the CalManager object

Get a handle to a the CalManager with the app.GetCalManager Method.

```
Dim app As AgilentPNA835x.Application
Set app = CreateObject("AgilentPNA835x.Application", <analyzerName>)

Dim mgr as ICalManager
Set mgr = app.GetCalManager
```

See Also:

PNA Automation Interfaces

The PNA Object Model

(**Bold** Methods or Properties provide access to a child object)

Methods	Interface See History	Description
CreateCalSet	ICalManager	Creates a new Cal Set
CreateCustomCal	ICalManager2	Creates a custom cal object.
DeleteCalSet	ICalManager	Deletes a Cal Set
GetCalSetByGUID	ICalManager	Get a handle to a Cal Set
GetCalSetCatalog	ICalManager	Gets a list of Cal Sets
GetCalSetUsageInfo	ICalManager	Returns the Cal Set ID and Error Term ID currently in use
GetCalTypes	ICalManager2	Query for a list of available calibration types.
GetRequiredEtermNames	ICalManager2	Returns an array of strings specifying the error terms required by the Cal Type correction algorithm.
SaveCalSets	ICalManager	Writes new or changed Cal Sets to disk Shared with the Calibrator Object
Properties		
Cal Sets	ICalManager	Collection for iterating through all the Cal Sets in the analyzer.

GuidedCalibration	ICalManager4	Used to perform a Guided Calibration.

ICalManager History

Interface	Introduced with PNA Rev:
ICalManager	2.0
CalManager2	3.1
CalManager3	3.5
CalManager4	5.0

CalSet Object

See ICalData Interface for putting and getting typed Cal Set data.

Description

Use this interface to query and or change the contents of a Cal Set.

Accessing the CalSet object

Get a handle to a CalSet object by using the CalSets collection. This is done through the CalManager object with the app.GetCalManager Method.

```
Dim app As AgilentPNA835x.Application
Set app = CreateObject("AgilentPNA835x.Application", <analyzerName>)

Dim calst As ICalSet
Set calst = app.GetCalManager.CalSets.Item(1)
```

See Also:

PNA Automation Interfaces

The PNA Object Model

Reading and Writing Calibration data.

Methods	Interface See History	Description
CloseCalSet	ICalSet	**Obsolete** - No longer necessary.
ComputeErrorTerms	ICalSet	Computes error terms for the CalType specified by a preceding OpenCal Set call.
Copy	ICalSet	Creates a new Cal Set and copies the current Cal Set data into it.
getErrorTerm	ICalSet	**Obsolete** - Use getErrorTermByString
getErrorTermByString	ICalSet2	Returns variant error term data by specifying the string name of the error term.
getErrorTermList	ICalSet	**Obsolete** - Use getErrorTermList2
getErrorTermList2	ICalSet2	Returns a list of error term names found in a calset.
GetGUID	ICalSet	Returns the GUID identifying a Cal Set
getStandard	ICalSet	**Obsolete** - Use getStandardByString
getStandardByString	ICalSet2	Returns variant standard acquisition data by specifying the string name of the standard.
getStandardsList	ICalSet	**Obsolete** - Use getStandardList2

getStandardList2	ICalSet2	Returns a list of standard names found in a Cal Set.
HasCalType	ICalSet	Verifies that the Cal Set object contains the error terms required to apply the specified CalType to an appropriate measurement.
OpenCalSet	ICalSet	**Obsolete** - No longer necessary.
putErrorTerm	ICalSet	**Obsolete** - Use putErrorTermByString
putErrorTermByString	ICalSet2	Writes variant error term data by specifying the string name of the error term.
putStandard	ICalSet	**Obsolete** - Use putStandardByString
putStandardByString	ICalSet2	Writes variant standard acquisition data by specifying the string name of the standard.
Save	ICalSet	Saves the current Cal Set to disk.
StringToNACalClass	ICalSet	Converts string values from GetStandardsList into enumeration data
StringToNAErrorTerm2	ICalSet	Converts string values from GetErrorTermList into enumeration data
Properties		**Description**
AlternateSweep	ICalSet3	Reads sweep either alternate or chopped.
Attenuator	ICalSet3	Returns the value of the attenuator control for the specified port number.
AttenuatorMode	ICalSet3	Returns the mode of operation (auto or manual) of the attenuator control for the specified port number.
CouplePorts	ICalSet3	Returns state of couple ports (ON or OFF)
CWFrequency	ICalSet3	Returns CW Frequency
Description	ICalSet	Set or return the descriptive string assigned to the Cal Set
DwellTime	ICalSet3	Returns the dwell time for the channel.
FrequencyOffsetCWOverride	ICalSet3	Reads state of CW Override (ON or OFF)
FrequencyOffsetDivisor	ICalSet3	Reads Frequency Offset Divisor value
FrequencyOffsetFrequency	ICalSet3	Reads Offset Frequency
FrequencyOffsetMultiplier	ICalSet3	Reads Frequency Offset Multiplier value
FrequencyOffsetState	ICalSet3	Reads Frequency Offset state (ON or OFF)
IFBandwidth	ICalSet3	Reads IF Bandwidth of the channel

LastModified	ICalSet3	Reads the time stamp of when the file was last modified
NumberOfPoints	ICalSet3	Returns the Number of Points of the channel.
PowerSlope	ICalSet3	Returns the Power Slope value.
ReceiverAttenuator	ICalSet3	Returns the value of the specified receiver attenuator control.
StartFrequency	ICalSet3	Returns the start frequency of the channel.
StartPower	ICalSet3	Returns the start power of the PNA when sweep type is set to Power Sweep.
StimulusValues	ICalSet3	Returns x-axis values for stimulus or response frequencies
StopFrequency	ICalSet3	Returns the stop frequency of the channel.
StopPower	ICalSet3	Returns the stop power of the PNA when sweep type is set to Power Sweep.
SweepGenerationMode	ICalSet3	Returns the method being used to generate a sweep: analog or stepped.
SweepTime	ICalSet3	Returns the sweep time of the analyzer.
SweepType	ICalSet3	Returns the type of X-axis sweep that is performed on a channel.
TestPortPower	ICalSet3	Returns the RF power level for the channel.

ICalSet History

Interface	Introduced with PNA Rev:
ICalSet	2.0
ICalSet2	3.0
ICalSet3	3.2

ICalData Interface

Description

Use this interface as an alternative to the ICalSet Interface to avoid using variants when transmitting data to and from the Cal Set

Learn about reading and writing Calibration data.

Methods	Interface See History	Description
get ErrorTermComplex	ICalData2	**Obsolete** - Use getErrorTermComplexByString
getErrorTermComplexByString	ICalData3	Returns typed error term data by specifying the string name of the error term.
getStandardComplex	ICalData2	**Obsolete** - Use getStandardComplexByString
getStandardComplexByString	ICalData3	Returns typed standard acquisition data by specifying the string name of the standard.
putErrorTermComplex	ICalData2	**Obsolete** - Use putErrorTermComplexByString
putErrorTermComplexByString	ICalData3	Writes typed error term data by specifying the string name of the error term.
putStandardComplex	ICalData2	**Obsolete** - Use putStandardComplexByString
putStandardComplexByString	ICalData3	Writes typed standard acquisition data by specifying the string name of the standard.

Properties	Description
None	

History

Interface	Introduced with PNA Rev:
The original ICalData Interface was introduced with PNA 1.0 on the Calibrator Object.	
ICalData2	2.0
ICalData3	3.1

Cal Sets Collection

Description

A collection object that provides a mechanism for iterating through all the Cal Sets in the analyzer. There is no ordering to the items in the collection. Therefore make no assumptions about the formatting of the collection.

Accessing the CalSets collection

Get a handle to the CalSets collection through the CalManager object with the app.GetCalManager Method.

```
Dim app As AgilentPNA835x.Application
Set app = CreateObject("AgilentPNA835x.Application", <analyzerName>)

Dim calsts As CalSets
Set calsts = app.GetCalManager.CalSets
```

See Also:

Collections in the Analyzer

The PNA Object Model

Methods	Description
Item	Returns a handle to a Cal Set object in the collection.
Remove	Deletes the Cal Set residing at position index in the collection.

Properties	Description
Count	Returns the number of Cal Sets in the collection.

CalStandard Object

Description

Contains all of the settings that are required to modify a calibration kit.

Accessing the CalStandard object

Get a handle to a standard with the calkit.GetCalStandard Method.

```
Dim app As AgilentPNA835x.Application
Set app = CreateObject("AgilentPNA835x.Application", <analyzerName>)

Dim std As ICalStandard
Set std = app.ActiveCalKit.GetCalStandard(1)
std.Delay = 0.00000003
```

See Also:

PNA Automation Interfaces

The PNA Object Model

Reading and Writing Calibration data.

Method
None

Property	**Interface** See History	**Description**
C0	ICalStandard	Sets and Returns the C0 (C-zero) value (the first capacitance value) for the calibration standard, when the Type is set to "naOpen".
C1	ICalStandard	Sets and Returns the C1 value (the second capacitance value) for the calibration standard, when the Type is set to "naOpen".
C2	ICalStandard	Sets and Returns the C2 value (the third capacitance value) for the calibration standard, when the Type is set to "naOpen".
C3	ICalStandard	Sets and Returns the C3 value (the fourth capacitance value) for the calibration standard, when the Type is set to "naOpen".
Delay	ICalStandard	Sets and Returns the electrical delay value for the calibration standard.
L0	ICalStandard	Sets and Returns the L0 (L-zero) value (the first inductance value) for the calibration standard, when the Type is set to "naShort".
L1	ICalStandard	Sets and Returns the L1 value (the second inductance value) for the calibration standard, when the Type is set to "naShort"..
L2	ICalStandard	Sets and Returns the L2 value (the third inductance value) for the calibration standard, when the Type is set to "naShort"..

L3	ICalStandard	Sets and Returns the L3 value (the third inductance value) for the calibration standard, when the Type is set to "naShort"..
Label	ICalStandard	Sets and Returns the label for the calibration standard.
loss	ICalStandard	Sets and Returns the insertion loss for the calibration standard.
Maximum Frequency	ICalStandard	Sets and Returns the maximum frequency for the calibration standard.
Medium	ICalStandard	Sets and Returns the media type of the calibration standard.
Minimum Frequency	ICalStandard	Sets and Returns the minumum frequency for the calibration standard.
Type	ICalStandard	Sets and Returns the type of calibration standard. Selections are: **naOpen, naShort, naLoad, naThru, naArbitraryImpedance and naSliding.**
TZReal	**ICalStandard2**	Sets and Returns the TZReal value (the Real Terminal Impedance value) for the calibration standard, when the Type is set to "naArbitraryImpedance".
TZImag	**ICalStandard2**	Sets and Returns the TZImag value (the Imaginary Terminal Impedance value) for the calibration standard, when the Type is set to "naArbitraryImpedance".
Z0	ICalStandard	Sets and Returns the characteristic impedance for the calibration standard.

ICalStandard History

Interface	Introduced with PNA Rev:
CalStandard	1.0
CalStandard2	3.0

Capabilities Object

Description

These properties return capabilities of the remote PNA.

Accessing the Capabilities object

```
Dim app As AgilentPNA835x.Application
Set app = CreateObject("AgilentPNA835x.Application", <analyzerName>)

Dim cap As Capabilities
Set cap = app.Capabilities
```

See Also:

PNA Automation Interfaces

The PNA Object Model

ICapabilities History

Method
None

Property	Description
FirmwareMajorRevision	Returns integer portion of firmware revision number.
FirmwareMinorRevision	Return decimal portion of firmware revision number.
FirmwareSeries	Returns the Alpha portion of the firmware revision number.
InternalTestsetPortCount	Returns the number of test ports (2 or 3).
IsFrequencyOffsetPresent	Returns the presence of Frequency Offset Option 080 (True or False).
IsReceiverStepAttenuatorPresent	Returns the presence of receiver step attenuators (True or False).
IsReferenceBypassSwitchPresent	Returns the presence of the reference switch (True or False).
MaximumFrequency	Returns the maximum frequency of the PNA.
MaximumNumberOfPoints	Returns the maximum possible number of data points.
MaximumReceiverStepAttenuator	Returns the maximum amount of receiver attenuation.
MaximumSourceALCPower	Returns the maximum amount of source ALC power.
MaximumSourceStepAttenuator	Returns the maximum amount of source attenuation.
MinimumFrequency	Returns the minimum frequency of the PNA.

MinimumNumberOfPoints	Returns the minimum possible number of data points.
MinimumReceiverStepAttenuator	Returns the minimum amount of receiver attenuation.
MinimumSourceALCPower	Returns the minimum amount of source ALC power.
ReceiverCount	Returns the number of receivers in the PNA.
ReceiverStepAttenuatorStepSize	Returns the step size of the attenuator.
SourceCount	Returns the number of sources.

ICapabilities History

I Interface	Introduced with PNA Rev:
ICapabilities	3.5

Channel Object

See SourcePowerCalData Interface for putting and getting typed source power calibration data.

Description

The channel object is like the engine that produces data. Channel settings consist of stimulus values like frequency, power, IF bandwidth, and number of points.

Accessing the Channel object

You can get a handle to a channel in a number of ways. But first you have to make sure that the channel exists. When you first startup the analyzer, there is one S11 measurement on channel 1. Thus there is only one channel in existence. You can do the following:

```
Dim app As AgilentPNA835x.Application
Set app = CreateObject("AgilentPNA835x.Application", <analyzerName>)

Dim chan As IChannel
Set chan = app.ActiveChannel
```

or

```
Set chan = app.Channels(2)
```

The first method returns the channel object that is driving the active measurement. If there is no measurement, there may not be a channel. Once a channel is created, it does not go away. So if there once was a measurement (hence a channel), the channel will still be available.

If there is no channel you can create one in a couple ways. You can do the following:

```
Pna.CreateMeasurement( ch1, "S11", port1, window2)
```

or

```
Pna.Channels.Add(2)
```

The latter will have no visible effect on the analyzer. It will simply create channel 2 if it does not already exist.

See Also:

PNA Automation Interfaces

The PNA Object Model

ading and Writing Calibration data.

(**Bold** Methods or Properties provide access to a child object)

Methods	Interface See History	Description
Abort	IChannel	Aborts the current measurement sweep on the channel.
AveragingRestart	IChannel	Clears and restarts averaging of the measurement data.
Continuous	IChannel	The channel continuously responds to trigger signals.
CopyToChannel	IChannel	Sets up another channel as a copy of this object's channel.
GetNumberOfGroups	**IChannel3**	Returns the number of groups a channel has yet to acquire.

getSourcePowerCalData	IChannel	**OBSOLETE** - use Get SourcePowerCalDataEx Method
getSourcePowerCalDataEx	**IChannel4**	Returns requested source power calibration data, if it exists.
GetXAxisValues	IChannel	Returns the channel's X-axis values into a dimensioned Variant array.
GetXAxisValues2	IChannel	Returns the channel's X-axis values into a dimensioned NON-Variant array.
Hold	IChannel	Puts the Channel in Hold - not sweeping.
Next_IFBandwidth	IChannel	A function that returns the Next higher IF Bandwidth value.
NumberOfGroups	IChannel	Sets the Number of trigger signals the channel will receive.
Preset	IChannel	Resets the channel to factory defined settings.
PreviousIFBandwidth	IChannel	Returns the previous IF Bandwidth value.
putSourcePowerCalData	IChannel	**OBSOLETE** - use Put SourcePowerCalDataEx Method
putSourcePowerCalDataEx	**IChannel4**	Inputs source power calibration data to this channel for a specific source port.
SelectCal Set	IChannel	Specifies the Cal Set to use for the Channel
Single	IChannel	Channel responds to one trigger signal from any source (internal, external, or manual). Then channel switches to Hold.
Properties	**Interface**	**Description**
AlternateSweep	IChannel	Sets sweeps to either alternate or chopped.
Attenuator	IChannel	Sets or returns the value of the attenuator control for the specified port number.
AttenuatorMode	IChannel	Sets or returns the mode of operation of the attenuator control for the specified port number.
Averaging	IChannel	Turns trace averaging ON or OFF for all measurements on the channel.
AveragingCount	IChannel	Returns the number of sweeps that have been averaged into the measurements.
AveragingFactor	IChannel	Specifies the number of measurement sweeps to combine for an average.
BalancedTopology	IChannel6	Sets and returns the topology of a balanced DUT.
Calibrator	IChannel4	Used to perform an Unguided calibration.
CalSet	IChannel	

centerFrequency	IChannel	Sets or returns the center frequency of the channel. Shared with the Segment Object
channelNumber	IChannel	Returns the Channel number. Shared with the Measurement Object
CoupleChannelParams	IChannel5	Turns ON and OFF Time Domain Trace Coupling.
CouplePorts	IChannel	Turns ON and OFF port power coupling.
CWFrequency	IChannel	Set the Continuous Wave (CW) frequency.
DwellTime	IChannel	Sets or returns the dwell time for the channel. Shared with the Segment Object
ExternalTriggerDelay	IChannel6	Sets or returns the external trigger delay value for the channel.
Fixturing	IChannel6	Port Ext, Embedding, and De-embedding functions.
FrequencyOffsetDivisor	IChannel2	Part of formula used to determine offset frequency of receivers.
FrequencyOffsetFrequency	IChannel2	Part of formula used to determine offset frequency of receivers.
FrequencyOffsetMultiplier	IChannel2	Part of formula used to determine offset frequency of receivers.
FrequencyOffsetCWOverride	IChannel2	Establishes a fixed (CW) stimulus frequency while measuring swept response frequency range.
FrequencyOffsetState	IChannel2	Turns frequency Offset ON and OFF.
FrequencySpan	IChannel	Sets or returns the frequency span of the channel. Shared with the Segment Object.
IFBandwidth	IChannel	Sets or returns the IF Bandwidth of the channel. Shared with the Segment Object.
IFConfiguration	IChannel4	Control the IF gain and source path settings for the H11 Option.
IsContinuous	IChannel3	Returns whether or not a channel is in continuous mode.
IsHold	IChannel3	Returns whether or not a channel is in hold mode.
NumberOfPoints	IChannel	Sets or returns the Number of Points of the channel. Shared with the Segment Object.
Parent	IChannel	Returns a handle to the parent object of the channel.
PowerSlope	IChannel	Sets or returns the Power Slope value.
R1InputPath	IChannel2	Throws internal reference switch (option 081).
ReceiverAttenuator	IChannel	Sets or returns the value of the specified receiver attenuator control.

ReduceIFBandwidth	IChannel	Sets or returns the state of the Reduced IF Bandwidth at Low Frequencies setting.
Segments	IChannel	Collection for iterating through the sweep segments of a channel.
SourcePowerCalPowerOffset	IChannel4	Sets or returns a power level offset from the PNA test port power.
SourcePowerCorrection	IChannel	Turns source power correction ON or OFF for a specific source port.
StartFrequency	IChannel	Sets or returns the start frequency of the channel. Shared with the Segment Object
StartPower	IChannel	Sets the start power of the analyzer when sweep type is set to Power Sweep.
StopFrequency	IChannel	Sets or returns the stop frequency of the channel. Shared with the Segment Object
StopPower	IChannel	Sets the Stop Power of the analyzer when sweep type is set to Power Sweep.
SweepGenerationMode	IChannel	Sets the method used to generate a sweep: continuous ramp (analog) or discrete steps (stepped).
SweepTime	IChannel	Sets the Sweep time of the analyzer.
SweepType	IChannel	Sets the type of X-axis sweep that is performed on a channel.
TestPortPower	IChannel	Sets or returns the RF power level for the channel. Shared with the Segment Object
TriggerMode	IChannel	Determines the measurement that occurs when a trigger signal is sent to the channel.
UserRangeMax	IChannel	Sets the stimulus stop value for the specified User Range.
UserRangeMin	IChannel	Sets the stimulus start value for the specified User Range.
XAxisPointSpacing	IChannel	Sets X-Axis point spacing for the active channel.

IChannel History

Interface	Introduced with PNA Rev:
IChannel	1.0
IChannel2	3.0
IChannel3	4.0
IChannel4	4.0
IChannel5	4.2
IChannel6	5.0
IChannel7	5.2

ISourcePowerCalData Interface

Description

Contains methods for putting source power calibration data in and getting source power calibration data out of the analyzer using typed data. The methods in this interface transfer data more efficiently than methods that use variant data.

Note: Interface **ISourcePowerCalData** is abbreviated as **ISPCD** in the following table.

Methods	Interface See History	Description
getSourcePowerCalDataScalar	ISPCD	**OBSOLETE** - use PutSourcePowerCalDataScalarEx Method
getSourcePowerCalDataScalarEx	**ISPCD2**	Returns requested source power calibration data, if it exists.
putSourcePowerCalDataScalar	ISPCD	**OBSOLETE** - use PutSourcePowerCalDataEx Method
putSourcePowerCalDataScalarEx	**ISPCD2**	Inputs source power calibration data to a channel, for a specific source port.
Properties		**Description**
None		

ISourcePowerCalData History

Interface	Introduced with PNA Rev:
ISourcePowerCalData	2.0
ISourcePowerCalData2	4.0

Channels Collection

Description

A collection object that provides a mechanism for iterating through the channels

Collections are, by definition, unordered lists of like objects. You cannot assume that Channels.Item(1) is always Channel 1.

Accessing the Channels collection

```
Dim app As AgilentPNA835x.Application
Set app = CreateObject("AgilentPNA835x.Application", <analyzerName>)

Dim chans As Channels
Set chans = app.Channels
```

See Also:

Collections in the Analyzer

The PNA Object Model

Methods	Interface	Description
Add	IChannels	An alternate way to create a measurement.
Hold	IChannels	Places all channels in Hold trigger mode.
Item	**IChannels2**	Use to get a handle on a channel in the collection.
Resume	**IChannels2**	Resumes the trigger mode of all channels that was in effect before sending the channels.Hold method.
Properties		**Description**
Count	IChannels	Returns the number of channels in the analyzer.
Parent	IChannels	Returns a handle to the current Application.
UnusedChannelNumbers	**IChannels2**	Returns an array of channel numbers that are NOT in use.
UsedChannelNumbers	**IChannels2**	Returns an array of channel numbers that are in use.

Gating Object

Description

Contains the methods and properties that control Time Domain Gating.

Accessing the Gating Object

```
Dim app As AgilentPNA835x.Application
Set app = CreateObject("AgilentPNA835x.Application", <analyzerName>)

Dim gate As Gating
Set gate = app.ActiveMeasurement.Gating
```

See Also:

PNA Automation Interfaces

The PNA Object Model

Time Domain Topics

Methods
None

Property	Interface (see History)	Description
Center	IGating	Sets or returns the Center time. Shared with the Transform Object
CoupledParameters	**IGating2**	Select Gating parameters to couple
Shape	IGating	Specifies the shape of the gate filter.
Span	IGating	Sets or returns the Span time. Shared with the Transform Object
Start	IGating	Sets or returns the Start time. Shared with the Transform Object
State	IGating	Turns an Object ON and OFF.
Stop	IGating	Sets or returns the Stop time. Shared with the Transform Object
Type	IGating	Specifies the type of gate filter used.

History

Interface	Introduced with PNA Rev:
IGating	1.0
IGating2	4.2

HWAuxIO Object

Description

Contains the methods and properties that control the rear panel Auxiliary Input / Output connector.

Accessing the HWAuxIO object

```
Dim app As AgilentPNA835x.Application
Set app = CreateObject("AgilentPNA835x.Application", <analyzerName>)

Dim AuxIO As HWAuxIO
Set AuxIO = app.GetAuxIO
```

See Also:

Pinout of the Aux IO Connector

PNA Automation Interfaces

The PNA Object Model

Method	Interface See History	Description
get_InputVoltage	IHWAuxIO	Reads the ADC input voltage
get_OutputVoltage	IHWAuxIO	Reads voltages on the DAC/Analog Output 1 and Output 2
get_OutputVoltageMode	**IHWAuxIO2**	Reads mode setting for either DAC output.
get_PortCData	IHWAuxIO	Reads a 4-bit value from Port C
put_OutputVoltage	IHWAuxIO	Writes voltages to the DAC/Analog Output 1 and Output 2
put_OutputVoltageMode	**IHWAuxIO2**	Writes mode setting for either DAC output.
put_PortCData	IHWAuxIO	Writes a 4-bit value to Port C
Property		**Description**
FootSwitch	IHWAuxIO	Reads the Footswitch Input
FootswitchMode	**IHWAuxIO3**	Determines the action that occurs when the footswitch is pressed.
PassFailLogic	IHWAuxIO	Sets and reads the logic of the PassFail line Shared with the HWMaterialHandler Object

PassFailMode	IHWAuxIO	Sets and reads the mode of the PassFail line Shared with the HWMaterialHandler Object
PassFailPolicy	**IHWAuxIO4**	Sets the policy used to determine how global pass/fail is computed. Shared with the HWMaterialHandler Object
PassFailScope	IHWAuxIO	Sets and reads the scope of the PassFail line Shared with the HWMaterialHandler Object
PassFailStatus	**IHWAuxIO4**	Returns the most recent pass/fail status value. Shared with the HWMaterialHandler Object
PortCLogic	HWAuxIO	Sets and reads the logic mode of Port C
PortCMode	HWAuxIO	Sets and reads the mode of Port C
SweepEndMode	HWAuxIO	Sets and reads the event that causes the Sweep End line to go to a false state. Shared with the HWMaterialHandler Object

IHWAuxIO History

Interface	Introduced with PNA Rev:
IHWAuxIO	2.0
IHWAuxIO2	3.0
IHWAuxIO3	3.0
IHWAuxIO4	5.0

HWExternalTestSetIO Object

Description

Contains the methods and properties that control the rear panel External Test Set Input / Output connector

Accessing the HWExternalTestSetIO object

```
Dim app As AgilentPNA835x.Application
Set app = CreateObject("AgilentPNA835x.Application", <analyzerName>)

Dim ExtTS As HWExternalTestSetIO
Set ExtTS = app.GetExternalTestSetIO
```

See Also:

Pinout of the Aux IO Connector

Pinout for the External Test Set Connector

PNA Automation Interfaces

The PNA Object Model

Method	Description
ReadData	Reads data and generates the appropriate timing signals
ReadRaw	Reads data, but does NOT generate appropriate timing signals
WriteData	Writes data and generates the appropriate timing signals
WriteRaw	Writes data, but does NOT generate the appropriate timing signals
Property	**Description**
Interrupt	Returns the state of the Interrupt line
SweepHoldOff	Returns the state of the Sweep Holdoff line

IHWExternalTestSetIO History

Interface	Introduced with PNA Rev:
HWExternalTestSetIO	2.0

HWMaterialHandlerIO Object

Description

Contains the methods and properties that control the rear panel Material Handler Input / Output connector.

Accessing the HWMaterialHandlerIO object

```
Dim app As AgilentPNA835x.Application
Set app = CreateObject("AgilentPNA835x.Application", <analyzerName>)

Dim MatHdlr As HWMaterialHandlerIO
Set MatHdlr = app.GetMaterialHandlerIO
```

See Also:

Pinout for the Material HandlerIO Connector

PNA Automation Interfaces

The PNA Object Model

Method	Interface	Description
get_Input1	HWMaterialHandlerIO	Reads a hardware latch that captures low to high transition on Input1
get_Output	HWMaterialHandlerIO	Returns the last value written to the selected output pin.
get_Port	HWMaterialHandlerIO	Returns the value from the specified "readable" port.
put_Output	HWMaterialHandlerIO	Writes a TTL HI or TTL Low to output pins 3 or 4.
t_Port	HWMaterialHandlerIO	Writes a value to the specified port.
Property		**Description**
IndexState	**HWMaterialHandlerIO2**	Determines the control of Material Handler connector Pin 20
ReadyForTriggerState	**HWMaterialHandlerIO2**	Determines the control of Material Handler connector Pin 21
PassFailLogic	HWMaterialHandlerIO	Sets and reads the logic of the PassFail line Shared with the HWAuxIO Object

PassFailMode	HWMaterialHandlerIO	Sets and reads the mode for the PassFail line Shared with the HWAuxIO Object
PassFailPolicy	**HWMaterialHandlerIO2**	Sets the policy used to determine how global pass/fail is computed. Shared with the HWAuxIO Object
PassFailScope	HWMaterialHandlerIO	Sets and reads the scope for the PassFail line Shared with the HWAuxIO Object
PassFailStatus	**HWMaterialHandlerIO2**	Returns the most recent pass/fail status value. Shared with the HWAuxIO Object
PortLogic	HWMaterialHandlerIO	Sets and returns the logic mode of data ports A-H
PortMode	HWMaterialHandlerIO	Sets and returns whether Port C or Port D is used for writing or reading data
SweepEndMode	HWMaterialHandlerIO	Sets and reads the event that cause the Sweep End line to go to a low state. Shared with the HWAuxIO Object

HWMaterialHandlerIO History

Interface	Introduced with PNA Rev:
HWMaterialHandlerIO	2.0
HWMaterialHandlerIO2	5.0

IFConfiguration Object

Description

These properties control the IF gain and source path settings for the H11 Option.

Accessing the IFConfiguration object

```
Dim app as AgilentPNA835x.Application
Dim chan as Channel
Set chan = app.ActiveChannel
Dim cfg as IIFConfiguration
Set cfg = chan.IFConfiguration
```

See Also:

PNA Automation Interfaces

The PNA Object Model

About the H11 Option

Pulsed Application

ConfigNarrowBand2 Method

Pulsed Measurement Example

Method	Description
None	

Property	Interface See History	Description
IFFilterSampleCount	IFConfiguration2	Sets or returns the number of taps in the IF filter.
IFFilterSamplePeriod	IFConfiguration2	Sets or returns the IF filter sample period time.
IFFilterSamplePeriodList	IFConfiguration2	Returns the list of available IF filter sample periods for the instrument.
IFFilterSamplePeriodMode	IFConfiguration2	Sets or returns the IF filter sample period mode.(Auto or Manual).
IFFilterSource Property	IFConfiguration2	Sets or retrieves type of IF filter to be used.
IFGainLevel Property	IFConfiguration	Sets the gain level for the specified receiver.
IFGainMode Property	IFConfiguration	Sets the gain state for ALL receivers.

IFGateEnable Property	IFConfiguration2	Sets or retrieves the state of the IF Gate.
IFSourcePath Property	IFConfiguration	Sets the source path of the specified receiver to Internal or External.
MaximumIFFilterSampleCount	IFConfiguration2	Returns the maximum allowed value for the IFFilterSampleCount.
MinimumIFFilterSampleCount	IFConfiguration2	Returns the minimum allowed value for the IFFilterSampleCount.

IFConfiguration History

Interface	Introduced with PNA Rev:
IIFConfiguration	4.0
IIFConfiguration2	4.0

IMixer Interface (Option 083)

Description

Contains the methods and properties to setup Mixer measurements. For performing calibrations, use either the SMC Type Object or the VMC Type Object.

Accessing the IMixer Interface

Access the IMixer Interface through the Measurement Object. If the particular type of Measurement that was created supports IMixer, then the program determines this at run time and can access the functionality exposed by IMixer. Because the determination of IMixer support is not made until runtime, the program should handle the case where IMixer is not supported on the object.

```
Dim app As AgilentPNA835x.Application
Set app = CreateObject("AgilentPNA835x.Application", "analyzerName")
app.Preset

' FCA Measurements can't share the channel with standard measurements
' Because preset creates a single measurement in channel 1, we first delete the
standard measurement
Dim standardMeas As IMeasurement
Set standardMeas = app.ActiveMeasurement
standardMeas.Delete

' Create a Measurement object, in this case using the IMeasurement interface
Dim meas As IMeasurement
Set meas = app.CreateCustomMeasurementEx(1, "SMC_Forward.SMC_ForwardMeas", "SC21")

' See if this measurement object supports IMixer
Dim mixer As IMixer
```

See an example program that shows how to create and calibrate a standard SMC or VMC measurement or a fixed output SMC measurement.

See Also:

PNA Automation Interfaces

The PNA Object Model

Methods	Interface See History	Description
Apply	**IMixer3**	Applies mixer settings.
Calculate	IMixer	Automatically calculate Input and Output frequencies for mixer setup.
LoadFile	IMixer	Loads a previously-configured mixer attributes file (**.mxr**)
SaveFile	IMixer	Saves the settings for the mixer/converter test setup to a mixer attributes file.

Properties		Description
ActiveXAxisRange	**IMixer3**	Sets or returns the swept parameter to display on the X-axis.
AvoidSpurs	IMixer	Sets and returns the state of the avoid spurs feature.
IFDenominator	IMixer	Sets or returns the denominator value of the IF Fractional Multiplier.
IFNumerator	IMixer	Sets or returns the numerator value of the IF Fractional Multiplier.
IFSideband	IMixer	Sets or returns the value of the IF sideband.
IFStartFrequency	IMixer	Returns the start frequency of the mixer IF.
IFStopFrequency	IMixer	Returns the stop frequency of the mixer IF.
InputDenominator	IMixer	Sets or returns the denominator value of the Input Fractional Multiplier.
InputNumerator	IMixer	Sets or returns the numerator value of the Input Fractional Multiplier.
InputPower	IMixer	Sets or returns the value of the Input Power.
InputStartFrequency	IMixer	Sets or returns the start frequency of the mixer input.
InputStopFrequency	IMixer	Sets or returns the stop frequency of the mixer input.
IsInputGreaterThanLO	**IMixer2**	Specifies whether to use the Input frequency that is greater than the LO or less than the LO.
LODenominator	IMixer	Sets or returns the denominator value of the LO Fractional Multiplier.
LOFixedFrequency	IMixer	Sets or returns the fixed frequency of the specified LO.
LOName	IMixer	Sets or returns the LO name.
LONumerator	IMixer	Sets or returns the numerator value of the LO Fractional Multiplier.
LOPower	IMixer	Sets or returns the value of the LO Power.
LORangeMode	**IMixer3**	Sets or returns the LO sweep mode to fixed or swept.
LOStage	IMixer	Returns the number of stages.
LOStartFrequency	**IMixer3**	Sets or returns the start frequency of the specified LO.
LOStopFrequency	**IMixer3**	Sets or returns the start frequency of the specified LO.
NominalIncidentPowerState	**IMixer4**	Toggles Nominal Incident Power ON and OFF.
OutputFixedFrequency	**IMixer3**	Sets or returns the fixed frequency of the mixer output.
OutputSideband	IMixer	Sets or returns the value of the output sideband.
OutputStartFrequency	IMixer	Sets or returns the start frequency of the mixer output.

OutputStopFrequency	IMixer	Sets or returns the stop frequency of the mixer output.

IMixer History

Interface	Introduced with PNA Rev:
IMixer	1.0
IMixer2	3.5
IMixer3	4.0
IMixer4	4.8

Limit Test Collection

Description

Child of the **Measurement** Object. A collection that provides a mechanism for iterating through the Measurement's Limit Segment objects (Limit Lines). The collection has 100 limit lines by default.

Accessing the LimitTest collection

Get a handle to an individual limit segment by specifying an item of the LimitTest collection.

```
Dim app As AgilentPNA835x.Application
Set app = CreateObject("AgilentPNA835x.Application", <analyzerName>)

Dim limSegs As LimitTest
Set limSegs = app.ActiveMeasurement.LimitTest
limSegs.Item(1).BeginResponse = 1000000000#
```

See Also:

Collections in the Analyzer

The PNA Object Model

Limit Line Testing Example

Methods	Description
GetTestResult	Retrieves the Pass/Fail results of the Limit Test (State).
Item	Use to get a handle on a limit line in the collection.

Properties	Description
Count	Returns the number of limit lines used in the measurement.
LineDisplay	Displays the limit lines on the screen.
SoundOnFail	Enables a beep on Limit Test fails.
State	Turns ON and OFF limit testing.

LimitTest History

Interface	Introduced with PNA Rev:
ILimitTest	1.0

LimitSegment Object

Description

The LimitSegment object is an individual limit line.

Accessing the LimitSegment object

Get a handle to an individual limit line by using the LimitTest collection.

```
Dim app As AgilentPNA835x.Application
Set app = CreateObject("AgilentPNA835x.Application", <analyzerName>)

Dim limSegs As LimitTest
Set limSegs = app.ActiveMeasurement.LimitTest
limSegs(1).BeginResponse = 1000000000#
```

See Also:

PNA Automation Interfaces

The PNA Object Model

Methods	Description
None	

Properties	Description
BeginResponse	Specifies the Y-axis value that corresponds with Begin Stimulus (X-axis) value.
BeginStimulus	Specifies the beginning X-axis value of the Limit Line.
EndResponse	Specifies the Y-axis value that corresponds with End Stimulus (X-axis) value.
EndStimulus	Specifies the End X-axis value of the Limit Line.
Type	Specifies the Limit Line type.

LimitSegment History

Interface	Introduced with PNA Rev:
ILimitSegment	1.0

Marker Object

Description

Contains the methods and properties that control Markers. There are 10 markers available per measurement:

- 1 reference marker
- 9 markers for absolute data or data relative to the reference marker (delta markers).

There are two ways to control markers through COM.

1. The Measurement object has properties that apply to ALL of the markers for that measurement. For example, **meas.MarkerFormat = naLinMag** applies formatting to all markers.
2. Marker object properties override the Measurement object properties. For example, you can then override the format setting for an individual marker by specifying **mark.Format = naLogMag** on the marker object.

Note: SearchFilterBandwidth is available through the measurement object.

Accessing the Marker object

To turn ON a marker, get a handle to the marker through the measurement object. If not already activated, this command will turn ON marker 1

```
Dim app As AgilentPNA835x.Application
Set app = CreateObject("AgilentPNA835x.Application", <analyzerName>)

app.ActiveMeasurement.marker(1).Format = naLinMag
```

You can also set the marker object to an object variable:

```
Dim m1 As Marker
Set m1 = app.ActiveMeasurement.Marker(1)
m1.Format = naMarkerFormat_LinMag
```

See Also:

PNA Automation Interfaces

e PNA Object Model

Methods	Interface	Description
Activate	IMarker	Makes an object the Active Object. Shared with the Marker Object
SearchMax	IMarker	Searches the marker domain for the maximum value.
SearchMin	IMarker	Searches the marker domain for the minimum value.
SearchNextPeak	IMarker	Searches the marker's domain for the next largest peak value.

SearchPeakLeft	IMarker	Searches the marker's domain for the next VALID peak to the left of the marker.
SearchPeakRight	IMarker	Searches the marker's domain for the next VALID peak to the right of the marker.
SearchTarget	IMarker	Searches the marker's domain for the target value.
SearchTargetLeft	IMarker	Moving to the left of the marker position, searches the marker's domain for the target value.
SearchTargetRight	IMarker	Moving to the right of the marker position, searches the marker's domain for the target value.
SetCenter	IMarker	Changes the analyzer's center frequency to the X-axis position of the marker.
SetCW	IMarker	Changes the analyzer to sweep type CW mode and makes the CW frequency the marker's frequency.
SetElectricalDelay	IMarker	Changes the measurement's electrical delay to the marker's delay value.
SetReferenceLevel	IMarker	Changes the measurement's reference level to the marker's Y-axis value.
SetStart	IMarker	Changes the analyzer's start frequency to the X-axis position of the marker.
SetStop	IMarker	Changes the analyzer's stop frequency to the X-axis position of the marker.
Property		**Description**
Bucket Number	IMarker	Marker data point number
DeltaMarker	IMarker	Makes a marker relative to the reference marker
Distance	**IMarker2**	Sets or returns distance value for time domain trace.
Format	IMarker	Linear, SWR, and so forth
Interpolated	IMarker	Turn marker interpolation ON and OFF
Number	IMarker	Read the number of the active marker
PeakExcursion	IMarker	Sets and reads the peak excursion value for the specified marker.
PeakThreshold	IMarker	Sets peak threshold for the specified marker.
SearchFunction	IMarker	Emulates the Tracking function in the marker search dialog box.
Stimulus	IMarker	Sets and reads the X-Axis value of the marker.
Target Value	IMarker	Sets the target value for the marker when doing Target Searches.

Tracking	IMarker	The tracking function finds the selected search function every sweep.
Type	IMarker	Sets and reads the marker type.
UserRange	IMarker	Assigns the marker to the specified User Range.
UserRangeMax	IMarker	Sets the stimulus stop value for the specified User Range.
UserRangeMin	IMarker	Sets the stimulus start value for the specified User Range.
Value	IMarker	Reads the Y-Axis value of the marker.

Marker History

Interface	Introduced with PNA Rev:
IMarker	1.0
IMarker2	4.2

Measurement Object

See IArrayTransfer Interface for putting and getting typed data.

See IMixer Interface (used with Option 083)

Description

The Measurement object is probably the most used object in the PNA Object Model. A measurement object represents the chain of data processing algorithms that take raw data from the channel and make it ready for display, which then becomes the scope of the Trace object.

A Measurement object is defined by it's parameter (S11, S22, A/R1, B and so forth). The measurement object is associated with a channel which drives the hardware that produces the data that feeds the measurement. The root of a measurement is the raw data. This buffer of complex paired data then flows through a number of processing blocks: error-correction, trace math, phase correction, time domain, gating, formatting. All of these are controlled through the measurement object.

The ACTIVE measurement is the measurement that will be acted upon if you make a setting from the front panel. It is the measurement whose "button" is pressed in the window with the red "active window" frame. If you create a new measurement, that measurement becomes the active measurement.

Therefore, all automation methods with the word "Active" in them refer to the object associated with the Active measurement, whether that object is a Channel, Window, Trace or Limit line.

Learn about the IMeasurement2 Interface for reading stimulus properties.

Accessing the Measurement object

```
Dim app As AgilentPNA835x.Application
Set app = CreateObject("AgilentPNA835x.Application", <analyzerName>)

Dim meas As IMeasurement
Set meas = app.ActiveMeasurement
```

or

```
Set meas = app.Measurements(n)
```

You can access four other objects through the Measurement object: markers, limit test, transform, and gating. For example, because each measurement has its own set of markers, you can set a marker by doing this:

```
Dim meas as measurement
Set meas = app.ActiveMeasurement
meas.marker(1).Stimulus = 900e6
```

IMeasurement2 Interface

Some of the properties and methods for the IMeasurement2 Interface return stimulus values that are set using the channel object. The following is the reason these properties and methods are duplicated.

Every measurement carries with it a snapshot of the stimulus properties of the channel that were in effect when the measurement last acquired data. Therefore, it is the measurement that provides the most accurate stimulus description of its data. Any change made to the channel after the measurement was acquired renders the IChannel interface unreliable in terms of describing the measurement.

See Also:

PNA Automation Interfaces

The PNA Object Model

(**Bold** Methods or Properties provide access to a child object)

Methods	Interface	Description
Activate	IMeasurement	Makes an object the Active Object. Shared with the Marker Object
ActivateMarker	IMeasurement	Makes a marker the Active Marker.
ChangeParameter	IMeasurement	Changes the parameter of the measurement.
DataToDivisor	IMeasurement	Stores data for receiver power cal of unratioed measurements.
DataToMemory	IMeasurement	Stores the active measurement into memory.
Delete	IMeasurement	Deletes the measurement object.
DeleteAllMarkers	IMeasurement	Deletes all of the markers from the measurement.
DeleteMarker	IMeasurement	Deletes a marker from the active measurement.
getData	IMeasurement	Retrieves Complex data from analyzer memory
getDataByString	IMeasurement	Retrieves variant data from the specified location in your choice of formats.
GetFilterStatistics	IMeasurement	Returns all four Filter Statistics
GetReferenceMarker	IMeasurement	Returns a handle to the reference marker.
Get SnPData	IMeasurement3	Returns SnP data.
GetTraceStatistics	IMeasurement	Returns the Trace Statistics.
GetXAxisValues	IMeasurement2	Returns the stimulus values for the measurement.
InterpolateMarkers	IMeasurement	Turns All Marker Interpolation ON and OFF for the measurement.
putDataComplex	IMeasurement	Puts complex data into one of five data buffers.
putDataScalar	IMeasurement	Puts formatted variant data into the measurement results buffer.
SearchFilterBandwidth	IMeasurement	Searches the domain with the current BW target.

Properties	Interface	Description
ActiveMarker	IMeasurement	Returns a handle to the Active Marker object.
BalancedMeasurement	IMeasurement	Sets the measurement type that is used with balanced topologies.

BandwidthTarget	IMeasurement	The insertion loss value at which the bandwidth of a filter is measured.
BandwidthTracking	IMeasurement	Turns Bandwidth Tracking function ON and OFF.
CalibrationName	IMeasurement2	Returns the name of the current cal type.
CalibrationType	IMeasurement	Set or get the calibration type for the measurement.
CalibrationTypeID	IMeasurement2	Sets or returns the current cal type to the Cal Set ID (CLSID).
Center	IMeasurement2	Returns the stimulus value of the center point for the measurement.
channelNumber	IIMeasurement	Returns the channel number. Shared with the Channel Object
Domain	IMeasurement2	Returns the domain (frequency, time, power) for the measurement.
ElectricalDelay	IMeasurement	Sets electrical delay.
ElecDelayMedium	IMeasurement2	Sets or returns the characteristic of the electrical delay medium.
ErrorCorrection	IMeasurement	Set or get the state of error correction for the measurement.
FilterBW	IMeasurement	Returns the results of the SearchBandwidth method.
FilterCF	IMeasurement	Returns the Center Frequency result of the SearchBandwidth method.
FilterLoss	IMeasurement	Returns the Loss value of the SearchBandwidth method.
FilterQ	IMeasurement	Returns the Q (quality factor) result of the SearchBandwidth method.
Format	IMeasurement	Sets display format.
Gating	IMeasurement	Controls Time Domain Gating.
InterpolateCorrection	IMeasurement	Turns ON and OFF the calculation of new error terms when stimulus values change.
InterpolateNormalization	IMeasurement	Turns ON and OFF normalization interpolation when stimulus values change after receiver power cal of unratioed measurements.
IsSparameter	IMeasurement2	Returns true if measurement represents an S-Parameter.
LimitTest	IMeasurement	Collection for iterating through the Limit Segment objects (Limit Lines).
LimitTestFailed	IMeasurement	Returns the results of limit testing

LoadPort	IMeasurement	Returns the load port number associated with an S-parameter reflection measurement.
LogMagnitudeOffset	IMeasurement	Sets or returns the value that normalized, unratioed, receiver power measurement data will be shifted by
MagnitudeOffset	IMeasurment4	Offsets the magnitude of the entire data trace to a specified value.
MagnitudeSlopeOffset	IMeasurment4	Offsets the magnitude of the data trace to a value that changes linearly with frequency.
Marker	IMeasurement	Contains the methods and properties that control Markers.
MarkerFormat	IMeasurement	Sets or returns the format of all the markers in the measurement.
Marker State	IMeasurement3	Sets or returns the ON / OFF state of a marker.
Mean	IMeasurement	Returns the mean value of the measurement.
Name	IMeasurement	Sets or returns the name of the measurement.
NAWindow	IMeasurement	Controls the part of the display that contains the graticule, or what is written on the display.
Normalization	IMeasurement	Turns ON or OFF normalization for receiver power cal of unratioed measurements
Number	IMeasurement	Returns the number of the measurement.
NumberOfPoints	IMeasurement2	Returns the Number of Points of the measurement.
Parameter	IMeasurement	Returns the measurement Parameter.
PeakToPeak	IMeasurement	Returns the Peak to Peak value of the measurement.
PhaseOffset	IMeasurement	Sets the Phase Offset for the active channel.
ReceivePort	IMeasurement2	Returns the receiver port of the measurement.
ReferenceMarkerState	IMeasurement	Turns the reference marker ON or OFF
ShowStatistics	IMeasurement	Displays and hides the measurement statistics (peak-to-peak, mean, standard deviation) on the screen.
Smoothing	IMeasurement	Turns ON and OFF data smoothing.
SmoothingAperture	IMeasurement	Specifies or returns the amount of smoothing as a ratio of the number of data points in the measurement trace.
SourcePort	IMeasurement2	Returns the source port of the measurement.
Span	IMeasurement2	Returns the stimulus span (stop - start) for the measurement.

StandardDeviation	IMeasurement	Returns the standard deviation of the measurement.
Start	IMeasurement2	Returns the stimulus value of the first point for the measurement.
StatisticsRange	IMeasurement	Sets the User Range number for calculating measurement statistics.
Stop	IMeasurement2	Returns the stimulus value of the last point for the measurement.
Trace	IMeasurement	Controls scale, reference position, and reference line.
TraceMath	IMeasurement	Performs math operations on the measurement object and the trace stored in memory.
Transform	IMeasurement	Controls Time Domain transforms.
View	IMeasurement	Sets (or returns) the type of trace displayed on the screen
WGCutoffFreq	IMeasurement2	Sets or returns the value of the waveguide cut off frequency

IMeasurement History

Interface	Introduced with PNA Rev:
IMeasurement	1.0
IMeasurement2	3.0
IMeasurement3	4.0
IMeasurement4	4.2
IMeasurement5	5.0

IArrayTransfer Interface

Description

Contains methods for putting data in and getting data out of the analyzer using typed data. This interface transfers data more efficiently than the IMeasurement Interface.

Methods	Description
getComplex	Retrieves real and imaginary data from the specified buffer.
getNAComplex	Retrieves typed **NAComplex** data from the specified buffer.
getPairedData	Retrieves magnitude and phase data pairs from the specified buffer.
getScalar	Retrieves scalar data from the specified buffer.
putComplex	Puts real and imaginary data into the specified buffer.
putNAComplex	Puts typed **NAComplex** data into the specified buffer.
putScalar	Puts scalar data into the measurement result buffer.

Properties	Description
None	

IArrayTransfer History

Interface	Introduced with PNA Rev:
IArrayTransfer	1.0

Measurement Collection

Description

A collection object that provides a mechanism for iterating through the Application measurements.

Accessing the Measurements collection

```
Dim app As AgilentPNA835x.Application
Set app = CreateObject("AgilentPNA835x.Application", <analyzerName>)

Dim measments As Measurements
Set measments = app.Measurements
```

See Also:

Collections in the Analyzer

The PNA Object Model

Methods	Description
Add	Adds a Measurement to the collection.
Item	Use to get a handle on a measurement in the collection.
Remove	Removes a measurement from the measurements collection.
Properties	**Description**
Count	Returns the number of measurements in the analyzer.
Parent	Returns a handle to the current Application.

NAWindow Object

Description

The NAWindow object controls the part of the display that contains the graticule, or what is written on the display.

Accessing the NaWindow object

```
Dim app As AgilentPNA835x.Application
Set app = CreateObject("AgilentPNA835x.Application", <analyzerName>)

Dim window As NAWindow
Set window = app.NAWindows(1)
window.AutoScale
```

or

```
Dim app As AgilentPNA835x.Application
Set app = CreateObject("AgilentPNA835x.Application", "analyzerName")

app.NAWindows(1).AutoScale
```

See Also:

PNA Automation Interfaces

The PNA Object Model

(**Bold** Methods or Properties provide access to a child object)

Methods	Description
Autoscale	Autoscales all measurements in the window. Shared with the Trace Object
ShowMarkerReadout	Shows and Hides the Marker readout for the active marker in the upper-right corner of the window object.
ShowTable	Shows or Hides the specified table for the active measurement in the lower part of the window object.

Properties	Description
ActiveTrace	Sets a trace to the Active Trace.
MarkerReadout	Sets and reads the state of the Marker readout for the active marker in the upper-right corner of the window object.
MarkerReadoutSize	Specifies the size of font used when displaying Marker readout in the selected window.
OneMarkerReadoutPerTrace	Either show marker readout of only the active trace or all of the traces simultaneously.

Title	Writes or reads a custom title for the window.
TitleState	Turns ON and OFF the window title.
Trace	Collection for getting a handle to a trace or iterating through the traces in a window.
WindowNumber	Reads the number of the active window.
WindowState	Maximizes or minimizes a window. Shared with the Application Object

INaWindow History

Interface	Introduced with PNA Rev:
INaWindow	1.0

NAWindows Collection

Description

A collection object that provides a mechanism for iterating through the Application windows.

Accessing the NaWindows collection

```
Dim app As AgilentPNA835x.Application
Set app = CreateObject("AgilentPNA835x.Application", <analyzerName>)

Dim windows As NAWindows
Set windows = app.NAWindows
```

See Also:

Collections in the Analyzer

The PNA Object Model

Methods	Description
Add	Adds a window to the NAWindows collection.
Item	Use to get a handle to a window in the collection.
Remove	Removes a window from the NAWindows collection.

Properties	Description
Count	Returns the number of windows on the analyzer.
Parent	Returns a handle to the current Application.

Port Extension Object

Description

Contains the methods and properties that control Port Extensions.

Accessing a Port Extension object

```
Dim app As AgilentPNA835x.Application
Set app = CreateObject("AgilentPNA835x.Application", <analyzerName>)

Dim PortExt As PortExtension
Set PortExt = app.PortExtension
```

See Also:

PNA Automation Interfaces

The PNA Object Model

Methods
None

Property	Description
Input A	Sets the Input A extension value.
Input B	Sets the Input B extension value.
Input C	Sets the Input C extension value.
Port 1	Sets the Port 1 extension value.
Port 2	Sets the Port 2 extension value.
Port 3	Sets the Port 3 extension value.
State	Turns Port Extensions ON and OFF.

IPort Extension History

Interface	Introduced with PNA Rev:
IPort Extension	1.0

PowerLossSegment Object

Description

Contains the properties describing a segment of the power loss table used in source power calibration.

You can get a handle to one of these segments through the segments.Item Method of the PowerLossSegments collection.

Accessing the PowerLossSegment object

You can get a handle to one of these segments through PowerLossSegments.Item(n)

```
Dim app As AgilentPNA835x.Application
Set app = CreateObject("AgilentPNA835x.Application", <analyzerName>)

Dim PwrLossSeg As PowerLossSegment
Set PwrLossSeg = app.SourcePowerCalibrator.PowerLossSegments(1)
```

See Also:

PNA Automation Interfaces

The PNA Object Model

Methods
None

Properties	Description
Frequency	The frequency (Hz) associated with this segment. Shared with the PowerSensorCalFactorSegment Object
ss	The loss value (dB) associated with this segment.
SegmentNumber	Returns the number of this segment Shared with the PowerSensorCalFactorSegment Object

IPowerLossSegment History

Interface	Introduced with PNA Rev:
IPowerLossSegment	2.0

PowerLossSegments Collection

Description

A collection object that provides a mechanism for iterating through the segments of the power loss table used in source power calibration.

Accessing the PowerLossSegments collection

```
Dim app As AgilentPNA835x.Application
Set app = CreateObject("AgilentPNA835x.Application", <analyzerName>)

Dim PwrLossSegs As PowerLossSegments
Set PwrLossSegs = app.SourcePowerCalibrator.PowerLossSegments
```

See Also:

Collections in the Analyzer

The PNA Object Model

Methods	Description
Add	Adds a PowerLossSegment object to the collection.
Item	Use to get a handle to a PowerLossSegment object in the collection.
Remove	Removes an object from the collection.
Properties	**Description**
Count	Returns the number of objects in the collection.
Parent	Returns a handle to the Parent object (SourcePowerCalibrator) of this collection.

PowerSensor Object

Description

Each power sensor connected to the power meter associated with Source Power Calibration will have a PowerSensor object created to represent it. These PowerSensor objects reside in the PowerSensors collection within the SourcePowerCalibrator object. You cannot directly create PowerSensor objects, but can only retrieve existing ones from the PowerSensors collection.

The PowerSensorCalFactorSegment object is also accessed through the PowerSensor object. These are accessed through the CalFactorSegments collection in the PowerSensor object.

Accessing a PowerSensor object

```
Dim pna As AgilentPNA835x.Application
Set pna = CreateObject("AgilentPNA835x.Application", <analyzerName>)

Dim powerCalibrator as SourcePowerCalibrator
Dim powerSensor as PowerSensor
Dim calFactorSegment as PowerSensorCalFactorSegment

Set powerCalibrator = pna.SourcePowerCalibrator

' Specify GPIB address of the power meter.
powerCalibrator.PowerMeterGPIBAddress = 13

' Each time the PowerSensors collection is accessed, the power meter is queried to
determine which channels have sensors attached. The collection is updated
accordingly.

If powerCalibrator.PowerSensors.Count > 0 Then
' If channel B of the meter has a sensor attached but channel A does not, then
element 1 of the
' collection is sensor B.  Whenever channel A has a sensor, sensor A will be element
1.
Set powerSensor = powerCalibrator.PowerSensors(1)
' Insert one new PowerSensorCalFactorSegment at the beginning of the collection
(index 1).

powerSensor.CalFactorSegments.Add(1)
' Assign our variable to refer to that object.
Set calFactorSegment = powerSensor.CalFactorSegments(1)

' Set property values for that object.
calFactorSegment.Frequency = 300000
' frequency in Hz
calFactorSegment.CalFactor = 98
' cal factor in percent

End If
```

See Also:

PNA Automation Interfaces

The PNA Object Model

Methods
None

Properties	Description
CalFactorsSegments (collection)	
MinimumFrequency	Minimum usable frequency (Hz) specified for this power sensor.
MaximumFrequency	Maximum usable frequency (Hz) specified for this power sensor.
PowerMeterChannel	Identifies which power sensor this object corresponds to (or which channel of the power meter the sensor is connected to).
ReferenceCalFactor	Reference cal factor (%) associated with this power sensor.

IPowerSensor History

Interface	Introduced with PNA Rev:
IPowerSensor	2.0

PowerSensorCalFactorSegment Object

Description

Contains the properties describing a segment of a power sensor cal factor table.

Accessing the PowerSensorCalFactorSegment object

You can get a handle to one of these segments through CalFactorSegments.Item(n)

```
Dim app As AgilentPNA835x.Application
Set app = CreateObject("AgilentPNA835x.Application", <analyzerName>)

Dim calFactSeg As CalFactorSegments
Set calFactSeg = app.SourcePowerCalibrator.PowerSensors(1).CalFactorSegments(1)
```

See Also:

PNA Automation Interfaces

The PNA Object Model

Methods
None

Properties	**Description**
Frequency	The frequency (Hz) associated with this segment. Shared with the PowerLossSegment Object
CalFactor	The cal factor (%) associated with this segment.
SegmentNumber	Returns the number of this segment Shared with the PowerLossSegment Object

IPowerSensorCalFactorSegment History

Interface	**Introduced with PNA Rev:**
IPowerSensorCalFactorSegment	2.0

PowerSensors Collection

Description

A collection object that provides a mechanism for iterating through the PowerSensor objects which are connected to the power meter. Each time this collection object is accessed, the power meter is queried to determine how many sensors are connected to it. The collection size and order of objects is then adjusted accordingly before the requested method or property operation is performed. The power meter is specified by using the PowerMeterGPIBAddress property of the SourcePowerCalibrator object.

Accessing the PowerSensors Collection

```
Dim app As AgilentPNA835x.Application
Set app = CreateObject("AgilentPNA835x.Application", <analyzerName>)

Dim PwrSensors As PowerSensors
Set PwrSensors = app.SourcePowerCalibrator.PowerSensors
```

See Also:

Collections in the Analyzer

The PNA Object Model

Methods	Description
Item	Use to get a handle to a PowerSensor object in the collection.
Properties	**Description**
Count	Returns the number of objects in the collection.
Parent	Returns a handle to the Parent object (SourcePowerCalibrator) of this collection.

Preferences Object

Description

Sets the preferences for saving citifiles.

Accessing the Preferences object

```
Dim app As AgilentPNA835x.Application
Set app = CreateObject("AgilentPNA835x.Application", <analyzerName>)

Dim pref As Preferences
Set pref = app.Preferences
```

See Also:

Citifile Define Data Saves

PNA Automation Interfaces

The PNA Object Model

Method
None

Property	Description
CitiContents	Specifies the contents of subsequent citifile saves.
CitiFormat	Specifies the format of subsequent citifile saves.
SnPFormat	Specifies the format of subsequent .S1P, .S2P, .S3P file saves.

IPreferences History

Interface	Introduced with PNA Rev:
IPreferences	4.0

SCPIStringParser Object

Description

Provides the ability to send a SCPI command from within the COM command.

Accessing the Port Extension object

```
Dim app As AgilentPNA835x.Application
Set app = CreateObject("AgilentPNA835x.Application", <analyzerName>)

Dim SCPI As IScpiStringParser
Set SCPI = app.ScpiStringParser
```

See Also:

PNA Automation Interfaces

The PNA Object Model

Method	Interface	Description
Parse	ISCPIStringParser	Provides the ability to send a SCPI command from within the COM command.
Execute	**ISCPIStringParser2**	Does not convert scpi errors. Use :SYST:ERR?
Properties		
None		

History

Interface	Introduced with PNA Rev:
ISCPIStringParser	1.0
ISCPIStringParser2	3.0

Segment Object

Description

Contains the methods and properties that affect a sweep segment.

Note: All of these properties are shared with at least one of the following objects: Channel, Cal Set, PowerSensorCalFactorSegment, or PowerLossSegment.

Accessing a Segment object

You can get a handle to a sweep segment through the segments collection.

```
Dim app As AgilentPNA835x.Application
Set app = CreateObject("AgilentPNA835x.Application", <analyzerName>)

Dim segs As ISegments
Set segs = app.ActiveChannel.Segments

segs(2).NumberOfPoints = 30
```

See Also:

PNA Automation Interfaces

The PNA Object Model

Segment Sweep

Method
None

Property	Interface	Description
centerFrequency	ISegment	Sets or returns the center frequency of the segment. Shared with the Channel Object
DwellTime	ISegment	Dwell time value. Shared with the Channel Object
FrequencySpan	ISegment	Sets or returns the frequency span of the segment. Shared with the Channel Object
IFBandwidth	ISegment	Sets or returns the IF Bandwidth of the segment. Shared with the Channel Object and Cal Set object.
NumberOfPoints	ISegment	Sets or returns the Number of Points of the segment. Shared with the Channel Object
SegmentNumber	ISegment	Returns the number of the current segment.

StartFrequency	ISegment	Sets or returns the start frequency of the segment. Shared with the Channel Object
State	ISegment	Turns On or OFF a segment.
StopFrequency	ISegment	Sets or returns the stop frequency of the segment. Shared with the Channel Object
TestPortPower	ISegment	Sets or returns the RF power level of the segment. Shared with the Channel Object

ISegment History

Interface	Introduced with PNA Rev:
ISegment	1.0

Segments Collection

Description

A collection object that provides a mechanism for iterating through the sweep segments of a channel. Sweep segments are a potentially faster method of sweeping the analyzer through only the frequencies of interest. Learn more about Segment Sweep.

Accessing the Segments collection

```
Dim app As AgilentPNA835x.Application
Set app = CreateObject("AgilentPNA835x.Application", <analyzerName>)

Dim segs As ISegments
Set segs = app.ActiveChannel.Segments
```

See Also:

Collections in the Analyzer

The PNA Object Model

Methods	Interface See History	Description
Add	ISegments	Adds an item to either the Segments collection.
Item	ISegments	Use to get a handle to a segment in the collection..
Remove	ISegments	Removes an item from a collection of objects.
SetAllSegments	**ISegments2**	Uploads a segment table to the PNA.
Properties		**Description**
owArbitrarySegments	**ISegments3**	Enables the setup of arbitrary segment sweep
Count	ISegments	Returns the number of items in a collection of objects.
IF Bandwidth Option	ISegments	Enables the IFBandwidth to be set on individual sweep segments.
Parent	ISegments	Returns a handle to the current naNetworkAnalyzer application..
Source Power Option	ISegments	Enables setting the Source Power for a segment.

ISegments History

Interface	Introduced with PNA Rev:
ISegments	1.0
ISegments2	3.5
ISegments3	4.2

SMCType Object

Description

Contains the methods and properties to perform an Scalar Measurement Calibration for the Frequency Converter Application (option 083).

Accessing the SMCType object

See an example which creates and calibrates an SMC measurement.

See Also:

PNA Automation Interfaces

The PNA Object Model

Methods	
AcquireStep	Acquire the measurement data for the specified step in the calibration process.
GenerateErrorTerms	Generates the error terms for the calibration.
GenerateSteps	Returns the number of steps required to complete the calibration.
GetStepDescription	Returns the description of the specified step calibration process.
Initialize	Begins a calibration.

Property	Description
AutoOrient	Sets ECAL module automatic orientation ON or OFF.
CalibrationPort	Sets or returns the calibration source port for the calibration.
CalKitType	Sets and returns a calibration kit type for calibration.
CompatibleCalKits	Returns a list of cal kits that are compatible with the connector type for the specified port.
ConnectorType	Sets or queries the connector type for the specified port.
Do2PortEcal	Specify ECAL or Mechanical calibration.
EcalCharacterization	Specifies the characterization data within an ECal module to be used for the calibration.

EcalOrientation	Specifies which port of the ECal module is connected to which port of the PNA when the AutoOrient property = False.
OmitIsolation	Sets and returns whether Isolation portion of the calibration will be performed or not.
ThruCalMethod	Sets and returns the method for performing the thru portion of the calibration.
ValidConnectorTypes	Returns a list of connector types for which there are calibration kits.

ISMCType History

Interface	Introduced with PNA Rev:
ISMCType	3.5

SourcePowerCalibrator Object

Description

This object is a child of the Application object and is a vehicle for performing source power calibrations.

Accessing the SourcePowerCalibrator Object

```
Dim app As AgilentPNA835x.Application
Set app = CreateObject("AgilentPNA835x.Application", <analyzerName>)

Dim ispc As ISourcePowerCalibrator
Set ispc = app.SourcePowerCalibrator
```

See Also:

PNA Automation Interfaces

The PNA Object Model

Note: Interface **ISourcePowerCalibrator** is abbreviated as **ISPC** in the following table.

(**Bold** Methods or Properties provide access to a child object)

Methods	**Interface** See History	**Description**
AbortPowerAcquisition	ISPC	Aborts a source power cal acquisition sweep that is currently in progress.
AcquirePowerReadings	ISPC	Initiates a source power cal acquisition.
ApplyPowerCorrectionValues	ISPC	Applies correction values after completing a source power cal acquisition sweep.
CheckPower	ISPC2	Measures power at a specific frequency. Used to test power level before and/or after applying a source power calibration.
LaunchPowerMeterSettingsDialog	ISPC2	Launches the Power Meter Settings dialog on the PNA.
SetCalInfo	ISPC	OBSOLETE
SetCalInfo2	ISPC3	REPLACEMENT - Specifies the type of source power calibration about to be performed.
SetPowerAcquisitionDevice	ISPC3	Sets the power sensor channel (A or B) to be used. This method is ONLY necessary when performing an SMC calibration.

Properties	**Interface**	**Description**

CalPower	ISPC	Specifies the power level that is expected at the desired reference plane.
IterationsTolerance	ISPC3	Sets the maximum desired deviation from the sum of the test port power and the offset value.
MaximumIterationsPerPoint	ISPC3	Specifies maximum number of readings to take at each data point for iterating the source power.
PowerAcquisitionDevice	ISPC2	Specifies the power sensor channel (A or B) that is currently selected for use at a specific frequency.
PowerLossSegments (collection)	ISPC2	Collection for iterating through the segments of the power loss table used in source power calibration.
PowerMeterGPIBAddress	ISPC	Specifies the GPIB address of the power meter.
PowerSensors (collection)	ISPC2	Collection for iterating through the PowerSensor objects which are connected to the power meter for a source power cal.
ReadingsPerPoint	ISPC	Specifies the maximum power readings for power meter settling.
ReadingsTolerance	ISPC3	Power meter settling tolerance value.
UsePowerLossSegments	ISPC	Specifies if subsequent calls to the AcquirePowerReadings method will make use of the loss table (PowerLossSegments).
UsePowerSensorFrequencyLimits	ISPC	Specifies if subsequent calls to the AcquirePowerReadings method will make use of power sensor frequency checking capability.

ISourcePowerCalibrator History

Interface	Introduced with PNA Rev:
ISourcePowerCalibrator	2.0
ISourcePowerCalibrator2	3.5
ISourcePowerCalibrator3	4.0

Trace Object

Description

The Trace object controls how the measurement data is displayed. You can control scale, reference position, and value from the Trace Object.

Accessing a Trace object

There are several ways to get a handle to a trace.

```
Dim app As AgilentPNA835x.Application
Set app = CreateObject("AgilentPNA835x.Application", <analyzerName>)

Dim trace As Trace
```

Then you can do any of the following:

```
Set trace = app.NAWindows(1).traces(1)
```

```
set trace = app.NAWindows.item(1).ActiveTrace
```

```
set trace = app.ActiveNAWindow.traces.item(1)
```

```
set trace = app.ActiveNAWindow.ActiveTrace
```

```
Set trace = app.Measurements(1).trace
```

```
Set trace = app.ActiveMeasurement.trace
```

See Also:

PNA Automation Interfaces

The PNA Object Model

Traces, Channels, and Windows on the PNA

Methods	Description
Autoscale	Autoscales the trace or all of the traces in the selected window. Shared with the NAWindow Object

Property	Description
Name	Sets or returns the trace name
ReferencePosition	Sets or returns the Reference Position of the active trace.
ReferenceValue	Sets or returns the value of the Y-axis Reference Level of the active trace.
YScale	Sets or returns the Y-axis Per-Division value of the active trace.

ITrace History

Interface	Introduced with PNA Rev:
ITrace	1.0

Traces Collection

Description

Child of the **Application** Object. A collection that provides a mechanism for getting a handle to a trace or iterating through the traces in a window.

Accessing the Traces collection

Get a handle to the traces collection through the NaWindows collection. The following example sets the variable **trcs** to the collection of traces in window 1 of the NaWindows collection.

```
Dim app As AgilentPNA835x.Application
Set app = CreateObject("AgilentPNA835x.Application", <analyzerName>)

Dim trcs As traces
Set trcs = app.NAWindows(1).traces
```

See Also:

Collections in the Analyzer

The PNA Object Model

Methods	Description
Item	Use to get a handle to a trace
Properties	**Description**
Count	Returns the number of traces in the collection.
Parent	Returns a handle to the current Application.

Transform Object

Description

Contains the methods and properties that control Time Domain transforms.

Accessing the Transform Object

```
Dim app As AgilentPNA835x.Application
Set app = CreateObject("AgilentPNA835x.Application", <analyzerName>)

Dim trans As Transform
Set trans = app.ActiveMeasurement.Transform
```

See Also:

PNA Automation Interfaces

The PNA Object Model

Time Domain Topics

Methods	Interface	Description
SetFrequencyLowPass	ITransform	Sets low frequencies for low pass.
Property		**Description**
Center	ITransform	Sets or returns the Center time. Shared with the Gating Object
CoupledParameters	**ITransform2**	Select Transform parameters to couple
DistanceMarkerMode	**ITransform2**	Sets the measurement type in order to determine the correct marker distance.
DistanceMarkerUnit	**ITransform2**	Sets the unit of measure for the display of marker distance values.
pulseWidth	ITransform	Sets or returns the Impulse Width of Time Domain transform windows.
KaiserBeta	ITransform	Sets or returns the Kaiser Beta of Time Domain transform windows.
Mode	ITransform	Sets the type of transform.
Span	ITransform	Sets or returns the Span time. Shared with the Gating Object
Start	ITransform	Sets or returns the Start time. Shared with the Gating Object
State	ITransform	Turns an Object ON and OFF.
StepRiseTime	ITransform	Sets or returns the Rise time of the stimulus in Low Pass Step Mode.

Stop	ITransform	Sets or returns the Stop time. Shared with the Gating Object

ITransform History

Interface	Introduced with PNA Rev:
ITransform	1.0
ITransform2	4.2

TriggerSetup Object

Description

These properties setup Global triggering that effects the entire PNA application.

Accessing the TriggerSetup object

```
Dim app as AgilentPNA835x.Application
Dim trigSetup as ITriggerSetup
Set trigSetup = app.TriggerSetup
```

See Also:

PNA Automation Interfaces

The PNA Object Model

Triggering in the PNA

Methods	Interface See History (below)	Description
None		

Properties		Description
AcceptTriggerBeforeArmed	ITriggerSetup2	Allows a trigger signal to be remembered and then used when the PNA becomes armed (ready to be triggered).
ExternalTriggerConnectionBehavior	ITriggerSetup	Configures the external triggering signal for the PNA
Scope	ITriggerSetup	Determines whether a trigger signal affects a single channel or all channels in the PNA.
Source	ITriggerSetup	Sets or returns the source of triggering in the PNA.
TriggerOutputEnabled	ITriggerSetup2	Enables the PNA to send trigger signals out the rear-panel TRIGGER OUT connector.

ITriggerSetup History

Interface	Introduced with PNA Rev:
ITriggerSetup	4.0
ITriggerSetup2	4.2

VMC Type Object

Description

Contains the methods and properties to perform a Vector Measurement Calibration for the Frequency Converter Application (option 083).

Accessing the VMCType object

See an example which creates and calibrates a VMC measurement.

See Also:

PNA Automation Interfaces

The PNA Object Model

Methods	
AcquireStep	Acquire the measurement data for the specified step in the calibration process.
GenerateErrorTerms	Generates the error terms for the calibration.
GenerateSteps	Returns the number of steps required to complete the calibration.
GetStepDescription	Returns the description of the specified step in the calibration process.
Initialize	Begins a calibration.

Property	Description
AutoOrient	Sets ECAL module automatic orientation ON or OFF.
CalKitType	Sets and returns a calibration kit type for calibration.
CharacterizeMixerOnly	Sets and returns whether to perform a mixer characterization ONLY or full 2-port calibration.
CharFileName	Specifies the .S2P mixer characterization file name.
CharMixerReverse	Specifies the direction in which to characterize the calibration mixer.
CompatibleCalKits	Returns a list of cal kits that are compatible with the connector type for the specified port.
ConnectorType	Sets or queries the connector type for the specified port.
Do1PortEcal	Specify ECAL or Mechanical calibration for the mixer characterization portion of a VMC calibration.

Do2PortEcal	Specify ECAL or Mechanical calibration for the 2-port calibration portion of a VMC calibration.
EcalCharacterization	Specifies the characterization data within an ECal module to be used for the calibration.
EcalOrientation1Port	For Mixer Characterization ONLY - Specifies which port of the ECal module is connected to which port of the PNA
EcalOrientation2Port	For full 2-port VMC cal - Specifies which port of the ECal module is connected to which port of the PNA
LoadCharFromFile	Specifies and loads a mixer characterization (S2P) file.
OmitIsolation	Sets and returns whether Isolation portion of the calibration will be performed or not.
ThruCalMethod	Sets and returns the method for performing the thru portion of the calibration.
ValidConnectorTypes	Returns a list of connector types for which there are calibration kits.

IVMCType History

Interface	Introduced with PNA Rev:
IVMCType	3.5

About Trigger

Write/Read.

AcceptTriggerBeforeArmed Property

Description	Determines what happens to an EDGE trigger signal if it occurs before the PNA is ready to be triggered. (LEVEL trigger signals are always ignored.) For more information, see External triggering.
VB Syntax	*trigsetup*.**AcceptTriggerBeforeArmed** = *boolean*
Variable	**(Type) - Description**
trigsetup	A TriggerSetup2 **(object)**
boolean	Choose from: **0 - False -** A trigger signal is ignored if it occurs before the PNA is ready to be triggered. **1 - True -** A trigger signal is remembered and then used when the PNA becomes armed (ready to be triggered). The PNA remembers only one trigger signal.
Return Type	Boolean
Default	False

Examples

```
trigsetup.AcceptTriggerBeforeArmed = 1 'Write
```

```
atba = trigsetup.AcceptTriggerBeforeArmed 'Read
```

C++ Syntax	HRESULT get_AcceptTriggerBeforeArmed(BOOL *pVal); HRESULT put_AcceptTriggerBeforeArmed(BOOL newVal);
Interface	ITriggerSetup2

About Performing a Calibration

AcquisitionDirection Property

Description	Specifies the direction of each part of a 2-port calibration.
VB Syntax	*cal*.**AcquisitionDirection** = *value*
Variable	**(Type) - Description**
cal	A Calibrator **(object)**
value	**(enum NADirection)** - Choose from: **0 - naForward** - measures the forward direction **1 - naReverse** - measures the reverse direction
Return Type	Long Integer
Default	naForward
Examples	`cal.AcquisitionDirection = naForward`
C++ Syntax	HRESULT AcquisitionDirection(tagNADirection dir);
Interface	ICalibrator

About Calibration Kits

Read-only

ActiveCalKit Property

Description	Returns a handle to the Active CalKit object. You can either **(1)** use the handle directly to access CalKit properties and methods, or **(2)** set a variable to the CalKit object. The variable retains a handle to the original object if another CalKit becomes active.
VB Syntax	1) app.**ActiveCalKit.**<*setting*> **or** 2) Set *cKit* = *app*.**ActiveCalKit**
Variable	**(Type) - Description**
app	An Application **(object)**
<setting>	A CalKit property (or method) and arguments
cKit	**(object)** - A CalKit object
Return Type	CalKit object
Default	None
Examples	`Public cKit as calKit` `Set cKit = app.ActiveCalKit 'read`
C++ Syntax	HRESULT get_ActiveCalKit (ICalkit * kit)
Interface	IApplication

ActiveChannel Property

Description Returns a handle to the Active Channel object. You can either **(1)** use the handle directly to access channel properties and methods, or **(2)** set a variable to the channel object. The variable retains a handle to the original channel if another channel becomes active.

VB Syntax (1) *app.***ActiveChannel.***<setting>*
or
(2) Set *chan* = *app*.**ActiveChannel**

Variable	(Type) - Description
chan	A Channel **(object)**
app	An Application **(object)**
<setting>	A channel property (or method) and arguments

Return Type Channel object

Default Not applicable

Examples

```
1) app.ActiveChannel.Averaging = 1
2) Public chan as Channel
  Set chan = app.ActiveChannel
```

C++ Syntax HRESULT get_ActiveChannel(IChannel* *pVal)

Interface IApplication

ActiveMarker Property

Description	Returns a handle to the Active Marker object. You can either **(1)** use the handle directly to access Marker properties and methods, or **(2)** set a variable to the Marker object. The variable retains a handle to the original object if another Marker becomes active.
VB Syntax	1) *meas*.**ActiveMarker.**<*setting*> **or** 2) Set *mark* = *meas*.**ActiveMarker**
Variable	**(Type) - Description**
meas	**(object)** - An Measurement object
<*setting*>	A marker property (or method) and arguments
mark	**(object)** - A marker object
Return Type	marker object
Default	None
Examples	`Public mark as marker` `Set mark = meas.ActiveMarker`
C++ Syntax	HRESULT get_ActiveMarker(IMarker** marker)
Interface	IMeasurement

Read-only

ActiveMeasurement Property

Description	Returns a handle to the Active Measurement object. You can either **(1)** use the handle directly to access measurement properties and methods, or **(2)** set a variable to the measurement object. The variable retains a handle to the original measurement.
VB Syntax	1) *app.***ActiveMeasurement.***<setting>* **or** 2) Set *meas* = *app*.**ActiveMeasurement**
Variable	**(Type) - Description**
meas	A Measurement **(object)**
app	An Application **(object)**
<setting>	A measurement property (or method) and arguments
Return Type	Measurement object
Default	None

Examples	`1) app.ActiveMeasurement.Averaging = 1`
	`2) Public meas as Measurement`
	`  Set meas = app.ActiveMeasurement`

C++ Syntax	HRESULT get_ActiveMeasurement(IMeasurement **ppMeas)
Interface	IApplication

ActiveNAWindow Property

Description Returns a handle to the Active Window object. You can either **(1)** use the handle directly to access window properties and methods, or **(2)** set a variable to the window object. The variable retains a handle to the original window if another window becomes active.

VB Syntax 1) *app.***ActiveNAWindow***.<setting>*
or
2) Set *win* = *app*.**ActiveNAWindow**

Variable **(Type) - Description**

win A NAWindow **(object)**

app An Application **(object)**

<setting> A NAWindow property (or method) and arguments

Return Type A NAWindow object

Default Not applicable

Examples

```
Public win as NAWindow
Set win = app.ActiveWindow
```

C++ Syntax HRESULT get_ActiveNAWindow(INAWindow **ppWindow)

Interface IApplication

ActiveXAxisRange Property

Description Sets or returns the swept parameter to display on the X-axis for the selected FCA measurement.

VB Syntax *mixer*.**ActiveXAxisRange** = *value*

Variable **(Type) - Description**

mixer A Mixer **(object)**

value **(Enum as MixerStimulusRange)** - Parameter to display on the X-axis. Choose from:

0 - mixINPUT - Input frequency span

1 - mixOUTPUT - Output frequency span

2 - mixLO_1 - First LO frequency span

3 - mixLO_2 - Second LO frequency span

Return Type Enum

Default OUTPUT

Examples

```
mixer.ActiveXAxisRange = 1 'Write
variable = mixer.ActiveXAxisRange  'Read
```

C++ Syntax HRESULT get_ActiveXAxisRange(tagMixerStimulusRange *Val)

HRESULT put_ActiveXAxisRange(tagMixerStimulusRange newVal)

Interface IMixer3

ActiveTrace Property

Description Returns a handle to the Active Trace object. You can either **(1)** use the handle directly to access trace properties and methods, or **(2)** set a variable to the trace object. The variable retains a handle to the original trace if another trace becomes active.

VB Syntax 1) *win.***ActiveTrace***.<setting>*
or
2) Set *trce* = *win*.**ActiveTrace**

Variable **(Type) - Description**

trce A Trace **(object)**

win An NAWindow **(object)**

<setting> A trace property (or method) and arguments

Return Type An NAWindow object

Default None

Examples

```
1) win.ActiveTrace.Autoscale
2) Public trce as Trace
  Set trce = Application.ActiveNAWindow.ActiveTrace
```

C++ Syntax HRESULT get_ActiveTrace(ITrace* *pVal)

Interface INAWindow

Write/Read

AllowArbitrarySegments Property

Description Enables you to setup a segment sweep with arbitrary frequencies. The start and stop frequencies of each segment can overlap other segments. Also, each segment can have a start frequency that is greater than its stop frequency which causes a reverse sweep over that segment. Learn more about Arbitrary Segment Sweep.

VB Syntax *segs*.**AllowArbitrarySegments** = *value*

Variable **(Type) - Description**

segs A Segments collection **(object)**

value **(boolean)**
True - Allows the setup of arbitrary segment sweep.
False - Prevents the setup of arbitrary segment sweep.

Return Type Boolean

Default False

Examples

```
segs.AllowArbitrarySegments = True 'Write
```

```
AllowArbSegs = AllowArbitrarySegments 'Read
```

C++ Syntax HRESULT get_AllowArbitrarySegments(VARIANT_BOOL *pVal)

HRESULT put_AllowArbitrarySegments(VARIANT_BOOL newVal)

Interface ISegments3

AlternateSweep Property

Description Sets sweeps to either alternate or chopped.

VB Syntax *object*.**AlternateSweep** = *value*

Variable **(Type) - Description**

object Channel **(object)**

or

CalSet **(object) -** Read-only property

value **(boolean)** - Choose either:

False (0) - Sweep mode set to **Chopped** - reflection and transmission are measured on the same sweep.
True (1) - Sweep mode set to **Alternate** - reflection and transmission measured on separate sweeps. Improves Mixer bounce and Isolation measurements. Increases cycle time.

Return Type boolean

Default False (0)

Examples

```
chan.AlternateSweep = True 'Write
```

```
altSwp = chan.AlternateSweep 'Read
```

C++ Syntax HRESULT AlternateSweep(VARIANT_BOOL *pVal)
HRESULT AlternateSweep(VARIANT_BOOL newVal)

Interface IChannel

ICalSet3

Read-only

Application Property

Description	Returns the name of the Analyzer making measurements on the channel.
VB Syntax	*chan.***Application**
Variable	**(Type) - Description**
chan	A Channel **(object)**
Return Type	object
Default	None
Examples	`rfna = chan.Application 'returns the Analyzer name`
C++ Syntax	HRESULT get_Application(IApplication** Application)
Interface	IChannel

About Arrange Windows

Write/Read

ArrangeWindows Property

Description Sets or returns the arrangement of all the windows. Overlay, Stack2, Split3 and Quad4 will create windows.

To control the state of the one window you have a handle to, use app.WindowState.

VB Syntax *app.***ArrangeWindows** = *value*

Variable **(Type) - Description**

app An Application **(object)**

value **(enum NAWindowModes)** - Choose from:

0 - naTile
1 - naCascade
2 - naOverlay
3 - naStack2
4 - naSplit3
5 - naQuad4

Return Type NAWindowModes

Default naTile

Examples

```
app.ArrangeWindow = naTile 'Write
```

```
arrWin = app.ArrangeWindows 'Read
```

C++ Syntax HRESULT put_ArrangeWindows(tagNAWindowModes newVal)

Interface IApplication

AttenuatorMode Property

Description Sets or returns the mode of operation of the attenuator control for the specified port number. This command is automatically set to Manual when an Attenuator value is set.

VB Syntax *object*.**AttenuatorMode(***portNum***)** = *value*

Variable **(Type) - Description**

object Channel **(object)**

or

CalSet **(object) -** Read-only property

portNum **(long)** - Port number (**1** or **2**) of attenuator control to be changed.

value **(enum NAModes)** - Choose from:

0 - naAuto - Attenuator control set to automatic. The analyzer will set the attenuator control appropriately to deliver the specified power at the source.

1 - naManual - Specify the attenuator setting using chan.Attenuator (which automatically sets AttenuatorMode = naManual.

Return Type NAModes

Default 0 - Auto

Examples

```
chan.AttenuatorMode(1) = naAuto 'Write
```

```
attn = chan.AttenuatorMode(1) 'Read
```

C++ Syntax HRESULT get_AttenuatorMode(long port, tagNAModes* pVal)
HRESULT put_AttenuatorMode(long port, tagNAModes newVal)

Interface IChannel

ICalSet3

Attenuator Property

Description Sets or returns the value of the attenuator control for the specified port number. Sending this command automatically sets AttenuatorMode to Manual.

VB Syntax *object*.**Attenuator(***portNum***)** = *value*

Variable **(Type) - Description**

object Channel **(object)**

or

CalSet **(object)** - Read-only property

portNum **(long integer)** - Port number (**1** or **2**) of attenuator control to be changed.

value **(double)** - Attenuator value in dB in 10dB steps.

N**ote:** To determine the maximum attenuator value, use MaximumSourceStepAttenuator Property

If an invalid value is entered, the analyzer will select the next lower valid value. For example, if 19.9 is entered the analyzer will select 10 dB attenuation.

Return Type Double

Default 20 dB

Examples

```
chan.Attenuator(1) = 20 'Write
```

```
attn = chan.Attenuator(cnum) 'Read
```

C++ Syntax HRESULT get_Attenuator(long port, double *pVal)
HRESULT put_Attenuator(long port, double newVal)

Interface IChannel

ICalSet3

About ECAL orientation

AutoOrient Property

Description	Sets ECAL module automatic orientation ON or OFF.
VB Syntax	*obj*.**AutoOrient** = *bool*
Variable	**(Type) - Description**
obj	SMCType (object) or VMCType (object)
bool	(Boolean) 1 - Set AutoOrientation ON 0 - Set AutoOrientation OFF
Return Type	Boolean
Default	1
Examples	`Smc.AutoOrient = 1`
C++ Syntax	HRESULT put_AutoOrient(VARIANT_BOOL bAutoOrient); HRESULT get_AutoOrient(VARIANT_BOOL *bAutoOrient);
Interface	SMCType VMCType

About Averaging

AveragingCount Property

Description Returns the number of sweeps that have been acquired and averaged into the measurements on this channel. AveragingFactor specifies the number of sweeps to average. AveragingCount indicates the progress toward that goal.

VB Syntax *value* = *chan*.**AveragingCount**

Variable **(Type) - Description**

chan A Channel **(object)**

value **(Long Integer)** - Variable to store the returned count

Return Type Long Integer

Default Not Applicable

Example

```
avgcount = chan.AveragingCount
```

C++ Syntax HRESULT get_AveragingCount(long* count)

Interface IChannel

About Averaging

Write/Read

AveragingFactor Property

Description	Specifies the number of measurement sweeps to combine for an average. Must also turn averaging ON by setting *chan*.Averaging = 1. Averaging is only allowed on ratioed measurements; not on single input measurements.
VB Syntax	*chan*.**AveragingFactor** = *value*
Variable	**(Type) - Description**
chan	A Channel **(object)**
value	**(Long Integer)** - Number of measurement sweeps to average. Choose any number between **1** and **1024.**
Return Type	Long Integer
Default	1
Examples	`chan.AveragingFactor = 5 'Write` `avgfact = chan.AveragingFactor ' doesn't work -Read`
C++ Syntax	HRESULT get_AveragingFactor(long *pVal) HRESULT put_AveragingFactor(long newVal)
Interface	IChannel

Averaging Property

Description	Turns trace averaging ON or OFF for all measurements on the channel. Averaging is only allowed on ratioed measurements; not on single input measurements.
VB Syntax	*chan*.**Averaging** = *state*
Variable	**(Type) - Description**
chan	A Channel **(object)**
state	**(boolean)** **0** - Turns averaging OFF **1** - Turns averaging ON
Return Type	Boolean
Default	0
Examples	`chan.Average = 1 'Write` `averg = chan.Averaging 'Read`
C++ Syntax	HRESULT get_Averaging(BOOL *pVal) HRESULT put_Averaging(BOOL newVal)
Interface	IChannel

About Avoid Spurs Feature

AvoidSpurs Property

Description Sets and returns the state of the avoid spurs feature.

VB Syntax *mixer*.**AvoidSpurs =** *boolean*

Variable **(Type) - Description**

mixer A Mixer **(object)**

value **(Boolean)** - State of avoid spurs feature. Choose from

0 Avoid spurs OFF

1 Avoid spurs ON

Return Type Boolean

Default 0

Examples

```
mixer.AvoidSpurs = 1 'Write
variable = mixer.AvoidSpurs  'Read
```

C++ Syntax

HRESULT get_AvoidSpurs(Bool *bVal)

HRESULT put_AvoidSpurs(Bool newVal)

Interface IMixer3

[About Marker Search](#)

Write/Read

BandwidthTarget Property

Description	Sets the insertion loss value at which the bandwidth of a filter is measured (using BandwidthTracking or SearchFilterBandwidth). For example, if you want to determine the filter bandwidth 3 db below the bandpass peak value, set BandwidthTarget to **-3**.
VB Syntax	*meas*.**BandwidthTarget** = *value*
Variable	**(Type) - Description**
meas	A Measurement **(object)**
value	**(single)** - Target value. Choose any number between **-500** and **500**
Return Type	Single
Default	-3
Examples	`meas.BandwidthTarget = -3 'Write` `fbw = meas.BandwidthTarget 'Read`
C++ Syntax	HRESULT put_BandwidthTarget(float target) HRESULT get_BandwidthTarget(float* target)
Interface	IMeasurement

BandwidthTracking Property

Descriaption Searches continually (every sweep) for the current BandwidthTarget (default is -3). To search the filter bandwidth for ONE SWEEP only (not continually), use meas.SearchFilterBandwidth.

This feature uses markers 1-4. To turn off these markers, either turn them off individually or DeleteAllMarkers.

The bandwidth statistics are displayed on the analyzer screen. To get the bandwidth statistics, use either GetFilterStatistics or FilterBW, FilterCF , FilterLoss ,or FilterQ.

The analyzer screen will show either Bandwidth statistics OR Trace statistics; not both.

To restrict the search to a UserRange with the bandwidth search, first activate marker 1 and set the desired UserRange. Then send the SearchFilterBandwidth command. The user range used with bandwidth search only applies to marker 1 searching for the max value. The other markers may fall outside the user range.

VB Syntax *meas*.**BandwidthTracking** = *value*

Variable **(Type) - Description**

meas A Measurement **(object)**

value **(boolean)**
1 - Turns bandwidth tracking ON
0 - Turns bandwidth tracking OFF

Return Type Boolean

Default 0 - OFF

Examples

```
meas.BandwidthTracking = 1 'Write
```

```
bwtrack = meas.BandwidthTracking 'Read
```

C++ Syntax HRESULT put_BandwidthTracking(VARIANT_BOOL state)
HRESULT get_BandwidthTracking(VARIANT_BOOL* state)

Interface IMeasurement

BeginResponse Property

Description When constructing a limit line, specifies the amplitude value of the start of a limit segment.

VB Syntax *limtseg*.**BeginResponse** = *value*

Variable **(Type) - Description**

limtseg A LimitSegment **(object)**

value **(double)** - Amplitude value. No units

Return Type Double

Default 0

Examples

```
Set limtseg = meas.LimitTest(1)
limtseg.BeginResponse = 10 'Write
```

```
BegResp = limtseg.BeginResponse 'Read
```

C++ Syntax HRESULT get_BeginResponse(double *pVal)
HRESULT put_BeginResponse(double newVal)

Interface ILimitSegment

BeginStimulus Property

Description When constructing a limit line, specifies the beginning X-axis value.

VB Syntax *limtseg*.**BeginStimulus** = *value*

Variable **(Type) - Description**

limtseg A LimitSegment **(object)**

value **(double) -** Stimulus value. No units

Return Type Double

Default 0

Examples

```
Set limtseg = meas.LimitTest(1)
 limtseg.Type = naLimitSegmentType_Maximum
 limtseg.BeginStimulus = 3e9
 limtseg.EndStimulus = 4e9
 limtseg.BeginResponse = 10
 limtseg.EndResponse = 10
```

```
BegStim = limtseg.BeginStimulus 'Read
```

C++ Syntax
HRESULT get_BeginStimulus(double *pVal)
HRESULT put_BeginStimulus(double newVal)

Interface ILimitSegment

Write/Read

BucketNumber Property

Description Sets or returns the bucket number (data point) for the active marker.

VB Syntax *mark*.**BucketNumber** = *value*

Variable **(Type) - Description**

mark A Marker **(object)**

value **(long integer)** - Data point. Choose any number between 0 and the measurement's number of data points - 1.

For example, with Number of points = 201, choose between 0 and 200

Return Type Long Integer

Default The first marker is set to the middle of the span. Subsequent markers are set to the bucket number of the previously active marker.

Examples

```
mark.BucketNumber = 100 'moves the active marker to data point 100 -
Write
```

```
pointNumber = mark.BucketNumber 'returns the data point number the
active marker is currently on. -Read
```

C++ Syntax HRESULT get_BucketNumber(long *pVal)
HRESULT put_BucketNumber(long newVal)

Interface IMarker

Write/Read

C0 Property

Description Sets and Returns the C0 (C-zero) value (the first capacitance value) for the calibration standard.

To set the other capacitance values, use C1, C2, C3

VB Syntax *calstd*.**C0** = *value*

Variable **(Type) - Description**

calstd A CalStandard **(object).** Use calKit.GetCalStandard to get a handle to the standard.

value **(single)** - Value for C0 in picofarads

Return Type Single

Default Not Applicable

Examples

```
calstd.C0 = 15 'Write the value of C0 to 15picofarads
```

```
cap0 = calstd.C0 'Read the value of C0
```

C++ Syntax HRESULT get_C0(float *pVal)
HRESULT put_C0(float newVal)

Interface ICalStandard

C1 Property

Description	Sets and Returns the C1 value (the second capacitance value) for the calibration standard. To set the other capacitance values, use C0, C2, C3
VB Syntax	*calstd*.**C1** = *value*
Variable	**(Type) - Description**
calstd	A CalStandard **(object).** Use calKit.GetCalStandard to get a handle to the standard.
value	**(single)** - Value for C1 in picofarads
Return Type	Single
Default	Not Applicable
Examples	`calstd.C1 = 15 'Write the value of C1 to 15picofarads` `cap1 = calstd.C1 'Read the value of C1`
C++ Syntax	HRESULT get_C1(float *pVal) HRESULT put_C1(float newVal)
Interface	ICalStandard

Write/Read

C2 Property

Description Sets and Returns the C2 value (the third capacitance value) for the calibration standard.

To set the other capacitance values, use C0, C1, C3

VB Syntax *calstd*.**C2** = *value*

Variable **(Type) - Description**

calstd A CalStandard **(object).** Use calKit.GetCalStandard to get a handle to the standard.

value **(single)** - Value for C2 in picofarads

Return Type Single

Default Not Applicable

Examples

```
calstd.C2 = 15 'Write the value of C2 to 15picofarads
```

```
cap2 = calstd.C2 'Read the value of C2
```

C++ Syntax HRESULT get_C2(float *pVal)
HRESULT put_C2(float newVal)

Interface ICalStandard

C3 Property

Description Sets and Returns the C3 value (the fourth capacitance value) for the calibration standard.

To set the other capacitance values, use C0, C1, C2

VB Syntax *calstd*.**C3** = *value*

Variable **(Type) - Description**

calstd A CalStandard **(object).** Use calKit.GetCalStandard to get a handle to the standard.

value **(single)** - Value for C3 in picofarads

Return Type Single

Default Not Applicable

Examples

```
calstd.C3 = 15 'Write the value of C3 to 15picofarads
```

```
cap3 = calstd.C3 'Read the value of C3
```

C++ Syntax HRESULT get_C3(float *pVal)
HRESULT put_C3(float newVal)

Interface ICalStandard

CalFactor Property

Description Sets or returns the cal factor value associated with a power sensor cal factor segment.

VB Syntax *calFactSeg*.**CalFactor** = *value*

Variable **(Type) - Description**

powerCalibrator **(object) -** A PowerSensorCalFactorSegment (object)

value **(double)** – Cal factor in percent. Choose any value between 1 and 150

Return Type Double

Default 0

Examples

```
calFactSeg.CalFactor = 98 'Write
factor = calFactSeg.CalFactor 'Read
```

C++ Syntax HRESULT put_CalFactor(Double newVal);

HRESULT get_CalFactor(Double *pVal);

Interface IPowerSensorCalFactorSegment

CalibrationType Property

Description Specifies the type of calibration to perform or apply to the measurement.

VB Syntax *meas*.**CalibrationType** = *type*

Variable **(Type) - Description**

meas A Measurement **(object)**

type **(enum NACalType) -** Calibration type. Choose from:

0 - naCalType_Response_Open
1 - naCalType_Response_Short
2 - naCalType_Response_Thru
3 - naCalType_Response_Thru_And_Isol
4 - naCalType_OnePort
5 - naCalType_TwoPort_SOLT
6 - naCalType_TwoPort_TRL
7 - naCalType_None
8 - naCalType_ThreePort_SOLT

Return Type **NACalType**

Default naCalType_None

Examples

```
meas.CalibrationType = naCalType_Response_Open 'Write

meascal = meas.CalibrationType 'Read
```

C++ Syntax HRESULT put_CalibrationType (tagNACalType CalType)
HRESULT get_CalibrationType (tagNACalType* pCalType)

Interface IMeasurement

Read only

CalibrationName Property

Description	Get the name of the current cal type.
VB Syntax	*value* = *meas*.**CalibrationName**
Variable	**(Type) - Description**
value	**(string)** - Variable to store the returned value.
meas	A Measurement **(object)**
Return Type	String
Default	Not Applicable
Examples	`ct = meas.CalibrationName`
C++ Syntax	HRESULT get_CalibrationName([out,retval] BSTR* CalibrationName);
Interface	IMeasurement2

Read / Write

CalibrationPort Property

Description Specifies which SMC port to calibrate.

VB Syntax *SMC*.**CalibrationPort** = *value*

Variable **(Type) - Description**

SMC SMCType (object)

value (String) Port number to be calibrated. Choose from:

- **1** - SMC forward
- **2** - SMC reverse
- **Both**

Return Type String

Default 1

Examples

```
value = SMC.CalibrationPort = "Both"
```

C++ Syntax

```
HRESULT put_CalibrationPort(BSTR port);
HRESULT get_CalibrationPort(BSTR *port);
```

Interface SMCType

VMCType

Write/Read

Learn about SMC cal types

CalibrationTypeID Property

Description Sets or returns the current cal type for custom measurements (FCA) using a Cal Type Name.

You can also use the CLSID or GUID associated with the cal type.

To select cal types for standard measurements, see Calibration Type Property.

VB Syntax *meas*.**CalibrationTypeID =** *id*

Variable **(Type) - Description**

meas A Measurement **(object)**

id (String) Cal type identifier Name. Use one of the following:

Name - Choose one of the following:

- "VMC"
- “SMC_2P” or “SMC”
- “SMCRsp+IN”
- “SMCRsp+OUT”
- “SMCRsp”

You can also use a ClassID or GUID. The following are **examples**:

CLSID - "VectorMixerCal.VCMCType"

GUID - "{2061767B-0FE2-4F6F-86D0-9AB332B18DA5}"

Return Type String

Default Not Applicable

Examples

```
Dim pna
Dim m

 Set pna = CreateObject("AgilentPNA835x.Application")
 Set m = pna.ActiveMeasurement
 m.CalibrationTypeID = "VMC"
 m.ErrorCorrection = True
 MsgBox   m.CalibrationName
```

C++ Syntax HRESULT get_CalibrationTypeID([out, retval] BSTR* CalibrationTypeID);

HRESULT put_CalibrationTypeID([in] BSTR CalibrationTypeID);

Interface IMeasurement2

CalKitType Property

Description Sets and returns a calibration kit type for calibration or to be used for kit modification. To get a handle to this kit, use app.ActiveCalKit.

There is also a CalKitType property on the GuidedCalibration object.

VB Syntax *object*.**CalKitType** = *value*

Variable **(Type) - Description**

object calkit (object) or

Application (object)

Note: app.CalKitType and **calkit**.calKitType perform exactly the same function.

value **(enum naCalKit)** - Calibration Kit type. Choose from:

1 - naCalKit_User1
2 - naCalKit_User2
3 - naCalKit_User3
4 - naCalKit_User4
..
..
..
49 - naCalKit_User49
50 - naCalKit_User50

These enumerated values correspond with the calibration kit ID on the Advanced Cal Kit Modify dialog box.

To change the cal kit name, use Name property.

Return Type NACalKit

Default Not Applicable

Examples

```
calkit.CalKitType = naCalKit_User27
```

```
kitype = app.CalKitType
```

C++ Syntax HRESULT get_CalKitType(tagNACalKit *pVal);
HRESULT put_CalKitType(tagNACalKit newVal);

Interface IApplication
ICalKit

Write/Read

CalKitType Property

Description Sets and returns a calibration kit type for the specified port number.

VB Syntax *object*.**CalKitType** *(port)* = *value*

Variable **(Type) - Description**

object GuidedCalibration (object) **or**

SMCType (object) **or**

VMCType (object)

port (Long) Port number the cal kit will be assigned to.

value **(string)** - Calibration Kit type.

Use CompatibleCalKits property for a list of valid Cal Kits.

Return Type String

Default Not Applicable

Examples

```
SMC.CalKitType(1) = naCalKit_User27
```

```
value = smc.CalKitType(1)
```

C++ Syntax

Interface SMCType

VMCType

Read-only

About Source Power Cal

CalPower Property

Description Specifies the power level that is expected at the desired reference plane (DUT input or output). This is not used for segment sweep with independent power levels or power sweeps.

Note: Although this command still works, it is recommended that you specify cal power by setting the test port power and offset value.

VB Syntax *value* = *powerCalibrator*.**CalPower** *(chan, sourcePort)*

Variable **(Type) - Description**

value **(double)** - Variable to store the returned Cal power value in dBm.

powerCalibrator **(object)** - A SourcePowerCalibrator object

chan **(long integer)** - Channel number of the PNA.

sourcePort **(long integer)** - Source port number

Return Type None

Default 0

Examples

```
Set powerCalibrator = pna.SourcePowerCalibrator
powerCalibrator.CalPower = -10 'Write

power = powerCalibrator.CalPower 'Read
```

C++ Syntax HRESULT get_CalPower(long channel, long sourcePort, double *pVal);

Interface ISourcePowerCalibrator

Write/Read

Center Property

Description	Sets or returns the Center time of either Gating or Time Domain transform windows
VB Syntax	*object*.**Center** = *value*
Variable	**(Type) - Description**
object	**(object)** As Gating **or** **(object)** As Transform
value	**(double)** - Center time in seconds. Choose any number between: **± (points-1) / frequency span**
Return Type	Double
Default	0

Examples	`trans.Center = 4.5e-9 'sets the Center time of a transform window -Write`
	`gate.Center = 4.5e-9 'sets the Center time of a gating window - Write`
	
	`cnt = trans.Center 'Read`

C++ Syntax	HRESULT get_Center(double *pVal) HRESULT put_Center(double newVal)
Interface	ITransform IGating

Read-only

Center Property

Description	Returns the stimulus value of the center data point for the measurement. This function does NOT work for segment sweep measurements. To understand how this property is useful, see IMeasurement2 Interface.
VB Syntax	*value* = *meas*.**Center**
Variable	**(Type) - Description**
value	**(Double)** - Variable to store the returned value.
meas	A Measurement **(object)**
Return Type	Double
Default	Not Applicable
Examples	`Print meas.Center 'prints the center data point`
C++ Syntax	HRESULT get_Center(double * Val);
Interface	IMeasurement2

Write/Read

CenterFrequency Property

Description	Sets or returns the center frequency of the channel **or** Sets or returns the center frequency of the segment. see also Measurement2 interface
VB Syntax	*object*.**centerFrequency** = *value*
Variable	**(Type) - Description**
object	A Channel **(object)** **or** A Segment **(object)**
value	**(double)** - Center frequency in Hertz. Choose any number between the **minimum** and **maximum** frequencies of the analyzer.
Return Type	Double
Default	Center of the frequency range

Examples	`chan.centerFrequency = 4.5e9 'sets the center frequency of a linear sweep for the channel object -Write`
	
	`centfreq = chan.centerFrequency 'Read`

C++ Syntax	HRESULT get_CenterFrequency(double *pVal) HRESULT put_CenterFrequency(double newVal)
Interface	IChannel ISegment

About Channels

Read-only

ChannelNumber Property

Description	Returns the Channel number of the Channel or Measurement object.
VB Syntax	*object.***ChannelNumber**
Variable	**(Type) - Description**
object	A Channel **(object)** **or** A Measurement **(object)**
Return Type	Long Integer
Default	Not applicable
Examples	`chanNum = chan.ChannelNumber 'returns the channel number` `chanNum = meas.ChannelNumber 'returns the channel number of the measurement`
C++ Syntax	HRESULT get_ChannelNumber(long *pVal)
Interface	IChannel IMeasurement

Read/Write

CharacterizeMixerOnly Property

Description	Sets and returns whether to perform ONLY a mixer characterization.
VB Syntax	*VMC*.**CharacterizeMixerOnly** = *bool*
Variable	**(Type) - Description**
VMC	VMCType (object)
bool	(Boolean) 1 - Perform ONLY mixer characterization. 0 - Perform both mixer characterization and calibration.
Return Type	Boolean
Default	0
Examples	`value = VMC.CharacterizeMixerOnly`
C++ Syntax	HRESULT put_CharacterizeMixerOnly(VARIANT_BOOL bCharMixerOnly); HRESULT get_CharacterizeMixerOnly(VARIANT_BOOL *bCharMixerOnly);
Interface	VMCType

Write/Read

CharFileName Property

Description	Specifies the mixer characterization (.S2P) file and immediately loads the file. Also specify the use of a characterization file with LoadCharFromFile Property
VB Syntax	*VMC*.**CharFileName** = *value*
Variable	**(Type) - Description**
VMC	VMCType (object)
value	(String) Full path and file name of the mixer characterization file.
Return Type	Not Applicable
Default	"C:\Program Files\Agilent\Network Analyzer\Documents\default.S2P"
Examples	`VMC.CharFileName = "C:\Program Files\Agilent\Network Analyzer\Documents\default.S2P"`
C++ Syntax	HRESULT put_CharFileName(BSTR filename); HRESULT get_CharFileName(BSTR *filename);
Interface	VMCType

Read-Write

CharMixerReverse Property

Description Specifies the direction in which to characterize the calibration mixer. Learn more about the calibration mixer.

VB Syntax *VMC*.**CharMixerReverse** = *bool*

Variable **(Type) - Description**

VMC VMCType (object)

bool (Boolean)

0 - Characterize the calibration mixer in the SAME direction as that specified in the mixer setup.

1 - Characterize the calibration mixer in the REVERSE direction as that specified in the mixer setup.

Return Type Boolean

Default 0

Examples

```
VMC.CharMixerReverse = 0
```

C++ Syntax HRESULT put_CharMixerReverse(VARIANT_BOOL bcharReverse);

HRESULT get_CharMixerReverse(VARIANT_BOOL *bcharReverse);

Interface VMCType2

About Citifiles

Write/Read

CitiContents Property

Description	Specifies the contents of subsequent citifile saves using app.SaveCitiDataData or app.SaveCitiFormattedData
VB Syntax	*pref*.**CitiContents** = *value*
Variable	**(Type) - Description**
pref	A Preferences **(object)**
value	**(string) -** Contents that will be saved with subsequent save commands. Choose from: **"Single"** - Single trace **"Displayed"** - All displayed traces **"Auto"** - All displayed traces
Return Type	String
Default	"Auto"
Examples	`pref.CitiContents = "Single" 'Write` `content = pref.CitiContents 'Read`
C++ Syntax	HRESULT get_CitiContents(BSTR *Contents) HRESULT put_CitiContents(BSTR Contents)
Interface	IPreferences

About Citifiles

Write/Read

CitiFormat Property

Description	Specifies the format of subsequent citifile saves using app.SaveCitiFormattedData
VB Syntax	*pref*.**CitiFormat** = *value*
Variable	**(Type) - Description**
pref	A Preferences **(object)**
value	**(string)** - Format in which the citifile will be saved with subsequent save commands. Choose from: **"MA"** - Linear Magnitude / degrees **"DB"** - Log Mag / degrees **"RI" -** Real / Imaginary **"Auto"** - Format in which the trace is already displayed. If other than Log Mag, Linear Magnitude, or Real/Imag, then the format will be in Real/Imag.
Return Type	String
Default	"Auto"
Examples	`pref.CitiFormat = "MA" 'Write` `format = pref.CitiFormat 'Read`
C++ Syntax	HRESULT get_CitiFormat(BSTR *Format) HRESULT put_CitiFormat(BSTR Format)
Interface	IPreferences

Read-only

CompatibleCalKits Property

Description	Returns a list of cal kits that are compatible with the connector type for the specified port.
VB Syntax	*value* = *obj*.**CompatibleCalKits** *(port)*
Variable	**(Type) - Description**
value	(Variant) Variable to store the returned list of CalKits.
obj	Any of the following: GuidedCalibration (object) SMCType (object) VMCType (object)
port	(Long) Port number for which you want compatible kits.
Return Type	Variant
Default	Not Applicable
Examples	`Dim kits As Variant` `kits = MySMC.CompatibleCalKits(1)`
C++ Syntax	HRESULT get_CompatibleCalKits(long port, VARIANT* Kits);
Interface	IGuidedCalibration SMCType VMCType

Read / Write

ConnectorType Property

Description Sets or queries the connector type for the specified port.

VB Syntax *obj*.**ConnectorType** *(port)* = *value*

Variable **(Type) - Description**

obj Any of the following:

GuidedCalibration (object)

SMCType (object)

VMCType (object)

port (Long) Port number of the connector type

value (String) - Connector type.

Use ValidConnectorType Property to list connector types.

Return Type String

Default None

Examples

```
Dim value As String
Value = MySMC.ConnectorType
```

C++ Syntax HRESULT get_ConnectorType(long port, BSTR *connector)

HRESULT put_ConnectorType(long port, BSTR connector)

Interface IGuidedCalibration

SMCType

VMCType

Read-only

Count Property

Description Returns the number of items in a collection of objects.

VB Syntax *object.***Count**

Variable **(Type) - Description**

object Any of the following **(objects):**

Cal Sets collection
CalFactorSegments collection
Channels collection
LimitTest collection
Measurements collection
NAWindows collection
PowerLossSegments collection
PowerSensors collection
Segments collection
Traces collection

Return Type Long Integer

Default Not applicable

Examples

```
numofchans = chans.Count 'return the number of channels
-Read
```

C++ Syntax HRESULT get_Count(long *p<interface>)

Interface
ICal Sets
ICalFactorSegments
IChannels
ILimitTest
IMeasurements
INAWindows
IPowerLossSegments
IPowerSensors
ISegments
ITraces

CouplePorts Property

Description Turns ON and OFF port power coupling. ON means the power level is the same for both ports. OFF means the power level may be set independently for each port.

VB Syntax *object*.**CouplePorts** = *value*

Variable **(Type) - Description**

object Channel **(object)**

or

CalSet **(object) -** Read-only property

value **(enum NAStates)** Choose from:
0 - NaOff - Turns coupling OFF
1 - NaOn - Turns coupling ON

Retaurn Type Long Integer
1 - ON
0 - OFF

Default NaON (1)

Examples

```
chan.CouplePorts = NaOff 'Write
```

```
couplport = chan.CouplePorts 'Read
```

C++ Syntax HRESULT get_CouplePorts(tagNAStates *pState)
HRESULT put_CouplePorts(tagNAStates newState)

Interface IChannel

|CalSet3

CoupleChannelParams Property

Description Turns ON and OFF Time Domain Trace Coupling. All of the measurements in the specified channel are coupled.

- To select Transform parameters to couple, use Trans.CoupledParameters Property
- To select Gating parameters to couple, use Gate.CoupledParameters Property

VB Syntax *chan*.**CoupleChannelParams** = *state*

Variable **(Type) - Description**

chan A Channel **(object)**

state **(boolean)**
0 - Turns Trace Coupling OFF
1 - Turns Trace Coupling ON

Return Type Boolean

Default 1 - ON

Examples

```
chan.CoupleChannelParams = 1 'Write
```

```
couple = chan.CoupleChannelParams 'Read
```

C++ Syntax HRESULT get_CoupleChannelParams(VARIANT_BOOL *isCoupled);

HRESULT put_CoupleChannelParams(VARIANT_BOOL isCoupled);

Interface IChannel5

Write/Read

About Coupled Markers

CoupledMarkers Property

Description	Sets and Reads the state of Coupled Markers (ON and OFF)
VB Syntax	*app*.**CoupledMarkers** = *state*
Variable	**(Type) - Description**
app	**(object) -** An Application object
state	**(boolean)** **False (0)** - Turns Coupled Markers OFF **True (1)** - Turns Coupled Markers ON
Return Type	Boolean **0** - OFF **1** - ON
Default	OFF (0)

Examples	`app.CoupledMarkers = True 'Write`
	
	`coupl = app.CoupledMarkers 'Read`

C++ Syntax	HRESULT put_CoupledMarkers(VARIANT_BOOL bState) HRESULT get_CoupledMarkers(VARIANT_BOOL *bState)
Interface	IApplication

About Time Domain Trace Coupling

Write/Read

CoupledParameters Property (Gating)

Description Specifies the time domain gating parameters to be coupled. The settings for those parameters will be copied from the active measurement to all other measurements on the channel.

To turn coupling ON and OFF, use CoupleChannelParams Property

To specify Transform parameters to couple, use Transform.CoupledParameters Property

VB Syntax *gate*.**CoupledParameters** = *value*

Variable **(Type) - Description**

trans A Gating **(object)**

value **(Enum As NAGatingCoupledParams)** - Parameters to couple. To specify more than one parameter, add the numbers. Choose from:

1 - naGatingStimulusCoupled (Start, Stop, Center, and Span TIME settings.)

2 - naGateStateCoupled (ON / OFF)

4 - naGatingShapeCoupled (Minimum, Normal, Wide, and Maximum)

8 - naGatingTypeCoupled (Bandpass and Notch)

Return Type Enum

Default 29

Examples

```
gate.CoupledParameters = 15 'Couple all parameters
```

```
CP = gate.CoupledParameters 'Read
```

C++ Syntax HRESULT get_CoupledParameters(long *lParams);

HRESULT put_CoupledParameters(long lParams);

Interface IGating2

Write/Read

CoupledParameters Property (Transform)

Description Specifies the time domain transform parameters to be coupled. The settings for those parameters will be copied from the active measurement to all other measurements on the channel.

To turn coupling ON and OFF, use CoupleChannelParams Property

To specify Gating parameters to couple, use Gate.CoupledParameters Property

VB Syntax *trans*.**CoupledParameters** = *value*

Variable **(Type) - Description**

trans A Transform **(object)**

value **(Enum As NATransformCoupledParams)** - Parameters to couple. To specify more than one parameter, add the numbers. Choose from:

1 - naTransformStimulusCoupled (Start, Stop, Center, and Span TIME settings.)

2 - naTransformStateCoupled (ON / OFF)

4 - naTransformWindowCoupled (Kaiser Beta / Impulse Width)

8 - naTransformModeCoupled (Low Pass Impulse, Low Pass Step, Band Pass)

16 - naTransformDistMkrUnitCoupled (Distance maker Units)

Return Type Enum

Default 29

Examples

```
trans.CoupledParameters = 31 'Couple all parameters
```

```
CP = trans.CoupledParameters 'Read
```

C++ Syntax HRESULT get_CoupledParameters(long *lParams);

HRESULT put_CoupledParameters(long lParams);

Interface ITransform2

Write/Read

About CW Frequency

CW Frequency Property

Description	Set the Continuous Wave (CW) frequency. Must first send chan.SweepType = naCWTimeSweep
VB Syntax	*object*.**CWFrequency** = *value*
Variable	**(Type) - Description**
object	Channel **(object)** **or** CalSet **(object) -** Read-only property
value	**(double)** CW frequency. Choose any number between: the **minimum** and **maximum** frequency limits of the analyzer Units are Hz
Return Type	Double
Default	1e9
Examples	`chan.CWFrequency = 5e9 'Write` `cwfreq = chan.CWFrequency 'Read`
C++ Syntax	HRESULT put_CWFrequency(double newVal) HRESULT get_CWFrequency(double *pVal)
Interface	IChannel |CalSet3

About Modifying Cal Kits

Write/Read

Delay Property

Description	Sets and Returns the electrical delay value for the calibration standard.
VB Syntax	*calstd*.**Delay** = *value*
Variable	**(Type) - Description**
calstd	A CalStandard **(object).** Use calKit.GetCalStandard to get a handle to the standard.
value	**(single)** - Electrical delay in seconds
Return Type	Single
Default	Not Applicable

Exaamples	`calstd.Delay = .00015 'Write the Delay .00015 seconds`
	
	`stdDelay = calstd.Delay 'Read the value of Delay`

C++ Syntax	HRESULT get_Delay(float *pVal) HRESULT put_Delay(float newVal)
Interface	ICalStandard

DeltaMarker Property

Description	Sets a marker as a delta marker. The reference marker must already be turned ON. See meas.ReferenceMarkerState
VB Syntax	*mark*.**DeltaMarker** = *state*
Variable	**(Type) - Description**
app	A Marker **(object)**
state	**(boolean)** - ON (1) marker is a delta marker OFF (0) marker is NOT a delta marker
Return Type	Boolean
Default	OFF (0)

Examples	`mark.DeltaMarker = True 'Write`
	
	`delta = mark.DeltaMarker 'Read`

C++ Syntax	HRESULT get_DeltaMarker(VARIANT_BOOL bState) HRESULT put_DeltaMarker(VARIANT_BOOL *bState)
Interface	IMarker

Description Property

Description	Sets or returns the descriptive string assigned to the Cal Set. Change this string so that you can easily identify each Cal Set constructed.
VB Syntax	*CalSet*.**Description** = *value*
Variable	**(Type) - Description**
CalSet	**(object) -** A Cal Set object
value	**(string)** – Description of the Cal Set
Return Type	String
Default	"CalSet_n" where n is an integer number.
Examples	`CalSet.Description = "My Cal Set" 'Write` `desc = CalSet.Description 'Read`
C++ Syntax	HRESULT get_Description(BSTR *pVal) HRESULT put_Description(BSTR newVal);
Interface	ICalSet

About Data Format

Write/Read

Format Property

Description	Sets or returns the display format of the measurement.
VB Syntax	*meas*.**Format** = *value*
Variable	**(Type) - Description**
meas	A Measurement **(object)**
value	**(enum NADataFormat)** - Choose from: 0 - naDataFormat_LinMag 1 - naDataFormat_LogMag 2 - naDataFormat_Phase 3 - naDataFormat_Polar 4 - naDataFormat_Smith 5 - naDataFormat_Delay 6 - naDataFormat_Real 7 - naDataFormat_Imaginary 8 - naDataFormat_SWR
Return Type	Long Integer
Default	**1** - naDataFormat_LogMag
Examples	`meas.Format = naDataFormat_Real 'Write` `fmt = meas.Format 'Read`
C++ Syntax	HRESULT get_Format(tagDataFormat *pVal) HRESULT put_Format(tagDataFormat newVal)
Interface	IMeasurement

Write-Read

DisplayAutomationErrors Property

Description Enables or disables automation error messages from being displayed on the screen.

VB Syntax app.**DisplayAutomationErrors** = value

Variable **(Type) - Description**

app An application**(object)**

value (Boolean)

True allows error to show on display,

False turns error off from display.

Return Type Boolean

Default True

Examples

```
Dim app As Application
Set app = New Application
app.DisplayAutomationErrors = False    'Turns off display
print app.DisplayAutomationErrors      'prints False
```

C++ Syntax HRESULT get_DisplayAutomationErrors(VARIANT_BOOL * Val);

HRESULT put_DisplayAutomationErrors(VARIANT_BOOL Val);

Interface IApplication2

Write-Read

DisplayGlobalPassFail Property

Description Shows or hides the dialog which displays global pass/fail results. Learn more about Global Pass/Fail.

VB Syntax *app*.**DisplayGlobalPassFail** = *value*

Variable **(Type) - Description**

app An Application **(object)**

value (Boolean)

True (1) - displays the pass/fail dialog.

False(0) - hides the pass/fail dialog.

Return Type Boolean

Default False

Examples

```
Dim app As Application
Set app = New Application
app.DisplayGlobalPassFail = true  'shows dialog
```

C++ Syntax HRESULT get_DisplayGlobalPassFail(VARIANT_BOOL * Val);

HRESULT put_DisplayGlobalPassFail(VARIANT_BOOL Val);

Interface IApplication6

Distance Property

Description Set or query marker distance on a time domain trace.

The Write command moves the marker to the specified distance value. Once moved, you can read the Y axis value or read the X-axis time value. (Distance is calculated from the X-axis time value.)

The Read command reads the distance of the marker.

If the marker is set as delta, the WRITE and READ data is relative to the reference marker.

VB Syntax *mark*.**Distance** = *value*

Variable **(Type) - Description**

mark A Marker **(object)**

value **(double)** - Marker distance in the unit of measure specified with DistanceMarkerUnit Property

Return Type Double

Default Not Applicable

Examples

```
mark.Distance = 3e9 'Write
```

```
XVal = mark.Distance 'Read
```

C++ Syntax HRESULT get_Distance(double *pVal);

HRESULT put_Distance(double newVal);

Interface IMarker2

About Distance MarkerSettings

DistanceMarkerMode Property

Description Specifies the measurement type in order to determine the correct marker distance.

- Select Auto for S-Parameter measurements.
- Select Reflection or Transmission for arbitrary ratio or unratioed measurements.

This settings affects the display of ALL markers for only the ACTIVE measurement.

VB Syntax *trans*.**DistanceMarkerMode** = *value*

Variable **(Type) - Description**

trans A Transform **(object)**

value **(enum As NADistanceMarkerMode)** - Choose from:

0 - naDistanceMarkerModeAuto

1 - naDistanceMarkerModeReflection

2 - naDistanceMarkerModeTransmission

Return Type Enum

Default 0 - naDistanceMarkerModeAuto

Examples

```
trans.DistanceMarkerMode = naDistanceMarkerModeReflection 'Write
```

```
DMM = trans.DistanceMarkerMode 'Read
```

C++ Syntax HRESULT get_DistanceMarkerMode(tagNADistanceMarkerMode *pVal);

HRESULT put_DistanceMarkerMode(tagNADistanceMarkerMode newVal);

Interface ITransform2

Write/Read

DistanceMarkerUnit Property

Description Specifies the unit of measure for the display of marker distance values. This settings affects the display of ALL markers for only the ACTIVE measurement (unless Distance Maker Units are coupled using CoupledParameters Property.

VB Syntax *trans*.**DistanceMarkerUnit** = *value*

Variable **(Type) - Description**

trans A Transform **(object)**

value **(Enum As NADistanceMarkerUnit)** - Distance Marker Units. Choose from

0 - naDistanceMarkerUnitMeter

1 - naDistanceMarkerUnitFeet

2 - naDistanceMarkerUnitInch

Return Type Enum

Default 0 - naDistanceMarkerUnitMeter

Examples

```
trans.DistanceMarkerUnit = naDistanceMarkerUnitFeet 'sets the
```

```
U = trans.DistanceMarkerUnit 'Read
```

C++ Syntax HRESULT get_DistanceMarkerUnit(tagNADistanceMarkerUnit *pVal);

HRESULT put_DistanceMarkerUnit(tagNADistanceMarkerUnit newVal);

Interface ITransform2

Read-Write

Do1PortEcal Property

Description Specify ECAL or Mechanical calibration for the mixer characterization portion of a VMC calibration.

VB Syntax *VMC*.**Do1PortEcal** = *bool*

Variable **(Type) - Description**

VMC VMCType (object)

bool (Boolean)

1 - ECAL

0 - Mechanical

Return Type Boolean

Default 0

Examples

```
value = VMC.Do1PortECal
```

C++ Syntax HRESULT put_Do1PortEcal(VARIANT_BOOL bDoEcal);

HRESULT get_Do1PortEcal(VARIANT_BOOL *bDoEcal);

Interface VMCType

Read-Write

Do2PortEcal Property

Description Specify ECAL or Mechanical calibration. For VMC, this selection only applies to the 2-port calibration portion. For mixer characterization (VMC), use Do1PortEcal Property

VB Syntax *object*.**Do2PortEcal** = *bool*

Variable **(Type) - Description**

object SMCType (object) or

VMCType (object)

bool (Boolean)

1 - ECAL

0 - Mechanical

Return Type Boolean

Default 0

Examples

```
value = VMC.Do2PortECal
```

C++ Syntax HRESULT put_Do2PortEcal(VARIANT_BOOL bDoEcal);

HRESULT get_Do2PortEcal(VARIANT_BOOL *bDoEcal);

Interface SMCType

VMCType

Read-only

Domain Property

Description	Returns the domain (frequency,time, power) of the measurement. To understand how this property is useful, see IMeasurement2 Interface.
VB Syntax	*value* = *meas*.**Domain**
Variable	**(Type) - Description**
value	(**Enum** as NADomainType) - variable to store the returned value 0 - Frequency 1 - Time 2 - Power
meas	A Measurement **(object)**
Return Type	Enum as NADomainType
Default	Not Applicable
Examples	`Print meas.Domain 'prints the value of the domain enum`
C++ Syntax	HRESULT get_Domain(tagNADomainType * Val);
Interface	IMeasurement2

Write/Read

About Dwell Time

DwellTime Property

Description Sets or returns the dwell time at the start of each sweep point for all measurements in a channel. Dwell time is only available with Chan.SweepGenerationMode **= naSteppedSweep** (not naAnalogSweep).

Sets or returns the dwell time of a specified sweep segment.

VB Syntax *object*.**DwellTime** = *value*

Variable **(Type) - Description**

object A Channel (object)

or CalSet (object) - Read-only property

or Segment (object)

value **(double)** - Dwell Time in seconds. Choose any number between **0** and **86400**

Return Type Double

Default 0

Examples

```
chan.DwellTime = 3e-3 'sets the dwell time for the channel -Write
segs(3).DwellTime = 1e9 'sets the dwell time of segment 3 -Write

dwell = chan.DwellTime 'Read
```

C++ Syntax HRESULT get_DwellTime(double *pVal)
HRESULT put_DwellTime(double newVal)

Interface IChannel
ISegment

|CalSet3

Write/Read

ECALCharacterization Property - Obsolete

Description

Note: This property is replaced by ECALCharacterizationEx Property.

Specifies the characterization data within an ECal module to be used for the calibration.

Learn more about ECal User Characterization.

VB Syntax *cal*.**ECALCharacterization**(*mod) = value*

Variable **(Type) - Description**

cal Calibrator (object)

module 1- ECal module

value **(Long)** – Characterization data within the ECal module to be used for ECal operations. Choose from:

0 – Factory Characterization

1 – UserCharacterization1

2 – UserCharacterization2

3 – UserCharacterization3

4 – UserCharacterization4

5 – UserCharacterization5

Return Type Long

Default **0** - Factory Characterization

Interface ICalibrator2

Write/Read

ECALCharacterization Property

Description Specifies the characterization data within an ECal module to be used for the SMC calibration.

Learn more about ECal User Characterization.

VB Syntax *SMC*.**ECALCharacterization**(*mod) = value*

Variable **(Type) - Description**

SMC SMCType (object)

module 1- ECal module

value **(Long)** – Characterization data within the ECal module to be used for ECal operations. Choose from:

0 – Factory Characterization

1 – UserCharacterization1

2 – UserCharacterization2

3 – UserCharacterization3

4 – UserCharacterization4

5 – UserCharacterization5

Return Type Long

Default **0** - Factory Characterization

Examples

```
SMC.ECALCharacterization(1)= 2
```

C++ Syntax HRESULT put_ECALCharacterization(long moduleNumber, long characterization);

HRESULT get_ECALCharacterization(long moduleNumber, long* characterization);

Interface ICalibrator2

Write/Read

ECALCharacterization Property

Description Specifies the characterization data within an ECal module to be used, and the portion of the VMC calibration.

Learn more about ECal User Characterization.

VB Syntax *VMC*.**ECALCharacterization** (*module,port) = value*

Variable **(Type) - Description**

VMC VMCType (object)

module **(long integer)**

1 - ECAL module

port **(boolean)**

True - 2-port calibration portion of the VMC

False - 1-port (mixer characterization portion of the VMC cal

value **(Long)** – Characterization data within the ECal module to be used for ECal operations. Choose from:

0 – Factory Characterization

1 – UserCharacterization1

2 – UserCharacterization2

3 – UserCharacterization3

4 – UserCharacterization4

5 – UserCharacterization5

Return Type Long

Default **0** - Factory Characterization

Examples

```
VMC.ECALCharacterization (1,True) = 4
```

C++ Syntax HRESULT put_ECALCharacterization(long moduleNumber, long characterization);

HRESULT get_ECALCharacterization(long moduleNumber, long* characterization);

Interface ICalibrator2

Write/Read

ECALCharacterizationEx Property

Description This property replaces ECALCharacterization Property.

Specifies the characterization data within an ECal module to be used for the calibration.

Learn more about ECal User Characterization.

VB Syntax *cal*.**ECALCharacterizationEx** (*module) = value*

Variable **(Type) - Description**

cal Calibrator (object)

module **(long integer)** Optional argument. ECal module.

Choose from modules **1** through **8**

Use IsECALModuleFoundEx to determine the number of modules connected to the PNA

Use GetECALModuleInfoEx to returns the model and serial number of each module.

value **(Long)** – Characterization data within the ECal module to be used for ECal operations. Choose from:

0 – Factory Characterization

1 – UserCharacterization1

2 – UserCharacterization2

3 – UserCharacterization3

4 – UserCharacterization4

5 – UserCharacterization5

Return Type Long

Default **0** - Factory Characterization

Examples

```
cal.ECALCharacterizationEx (4) = 2
```

C++ Syntax HRESULT put_ECALCharacterizationEx(long moduleNumber, long characterization);

HRESULT get_ECALCharacterizationEx(long moduleNumber, long* characterization);

Interface ICalibrator4

Read-only

ECALCharacterizationIndexList Property

Description Returns a list of characterizations stored in the specified ECal module. Learn more about ECal User characterization.

VB Syntax *clist* = *cal*.**ECALCharacterizationIndexList** *(moduleNum)*

Variable **(Type) - Description**

clist Variable to store the returned list of characterization index numbers.

cal Calibrator (object)

module ECal module from which to read user characterization numbers.

Return Type Variant

Default Not Applicable

Examples

```
'Module 1 contains User Characterizations 1 and 3.
clist = cal.ECALCharacterizationIndexList(1)
'Returns the following (0 always indicates the factory
characterization):
0,1,3
```

C++ Syntax HRESULT get_ECALCharacterizationIndexList(long moduleNumber, VARIANT *characterizations);

Interface ICalibrator6

ECALIsolation Property

Description Specifies whether the acquisition of the ECal calibration should include isolation or not.

Note: This command is ignored on PNA release 4.26 and greater. Isolation is never performed with ECal as the inherent isolation of the PNA is better than that achieved with ECal.

VB Syntax *cal*.**ECALIsolation** = *value*

Variable **(Type) - Description**

cal A Calibrator **(object)**

value **(boolean)**

False (0) - Exclude Isolation

True (1) - Include Isolation

Return Type Boolean

Default False (0)

Examples

```
Dim oPNA as AgilentPNA835x.Application
Dim oCal as Calibrator
Set oPNA = CreateObject("AgilentPNA835x.Application",
"MachineName")
Set oCal = oPNA.ActiveChannel.Calibrator
' Uncomment the following line to have the cal include isolation
' oCal.ECALIsolation = True
' Uncomment the following line to have the cal omit isolation
'oCal.ECALIsolation = False
oCal.DoECAL2Port ' Do the cal
```

C++ Syntax HRESULT put_ECALIsolation (VARIANT_BOOL bIsolationState);

HRESULT get_ECALIsolation (VARIANT_BOOL *bIsolationState);

Interface Calibrator

Read-only

ECALModuleNumberList Property

Description Returns a list of index numbers to be used for referring to the ECal modules that are currently attached to the PNA.

VB Syntax *clist* = *cal*.**ECALModuleNumberList**

Variable **(Type) - Description**

clist Variable to store the returned list of index numbers.

cal Calibrator (object)

Return Type Variant

Default Not Applicable

Examples

```
clist = cal.ECALModuleNumberList
'If 2 modules are attached to the PNA
'then the returned list will be:
1,2
```

C++ Syntax HRESULT get_ECALModuleNumberList(VARIANT *modules);

Interface ICalibrator6

Read/Write

EcalOrientation Property

Description Specifies which port of the ECal module is connected to which port of the PNA when the AutoOrient property = False.

VB Syntax *SMC*.**EcalOrientation** *(mod)* = *value*

Variable **(Type) - Description**

SMC SMCType **(object)**

mod (Long)

1 - Use ECAL Module for the calibration.

value **(string) -**Format this parameter in the following manner:

Aw,Bx,Cy,Dz

where

- A, B, C, and D are literal ports on the ECAL module
- w,x,y, and z are substituted for PNA port numbers to which the ECAL module port is connected.

Ports of the module which are not used are omitted from the string.

For example, on a 4-port ECal module with

- port A connected to PNA port 2
- port B connected to PNA port 3
- port C not connected
- port D connected to PNA port 1

the string would be: A2,B3,D1

Return Type String

Default "A1,B2"

Examples

```
SMC.EcalOrientation (1) = "A2,B1"
```

C++ Syntax HRESULT put_EcalOrientation(long lModuleNum, BSTR orientation);

HRESULT get_EcalOrientation(long lModuleNum, BSTR *orientation);

Interface SMCType

Read/Write

EcalOrientation1Port Property

Description	**For Mixer Characterization ONLY** Specifies which port of the ECal module is connected to which port of the PNA for the Do1PortECAL property when the AutoOrient property = False.
VB Syntax	*VMC*.**EcalOrientation1Port** *(mod)* = *value*
Variable	**(Type) - Description**
VMC	VMCType **(object)**
mod	(Long) **1** - Use ECAL Module for the calibration.
value	**(string) -** Choose from: **"A1"** - ECAL module port A is connected to PNA port 1 **"B1" -** ECAL module port A is connected to PNA port 1
Return Type	String
Default	"A1" If anything other than port 1 is specified, "B1" will be used. For example, if "A2" is specified, "B1" is used.
Examples	`VMC.EcalOrientation1Port(1) = "B1"`
C++ Syntax	HRESULT put_EcalOrientation1Port(long lModuleNum, BSTR orientation); HRESULT get_EcalOrientation1Port(long lModuleNum, BSTR
Interface	VMCType

Read/Write

EcalOrientation2Port Property

Description Specifies which port of the ECal module is connected to which port of the PNA for the Do2PortECAL property when the AutoOrient property = False.

VB Syntax *VMC*.**EcalOrientation2Port** *(mod) = value*

Variable **(Type) - Description**

VMC VMCType **(object)**

mod (Long) Module being used for the calibration.

Choose from 1 or 2.

value **(string) -**Format this parameter in the following manner:

Aw,Bx,Cy,Dz

where

- A, B, C, and D are literal ports on the ECAL module
- w,x,y, and z are substituted for PNA port numbers to which the ECAL module port is connected.

Ports of the module which are not used are omitted from the string.

For example, on a 4-port ECal module with

- port A connected to PNA port 2
- port B connected to PNA port 3
- port C not connected
- port D connected to PNA port 1

the string would be: A2,B3,D1

Return Type String

Default "A1,B2"

Examples

```
VMC.EcalOrientation1Port(1) = "A2,B1"
```

C++ Syntax HRESULT put_EcalOrientation2Port(long lModuleNum, BSTR orientation);

HRESULT get_EcalOrientation2Port(long lModuleNum, BSTR *orientation);

Interface VMCType

Write/Read

ECALPortMapEx Property

Description This property replaces ECALPortMap Property

Specifies which ports of the ECal module are connected to which ports of the PNA for the DoECAL1PortEx and DoECAL2PortEx methods when the OrientECALModule property = False.

VB Syntax *cal*.**ECALPortMapEx** *(module)* = *value*

Variable **(Type) - Description**

cal A Calibrator **(object)**

module **(long integer)** Optional argument. ECal module.

Choose from modules **1** through **8**

Use IsECALModuleFoundEx to determine the number of modules connected to the PNA

Use GetECALModuleInfoEx to return the model and serial number of each module.

value **(string) -**Format this parameter in the following manner:

Aw,Bx,Cy,Dz

where

- A, B, C, and D are literal ports on the ECAL module
- w,x,y, and z are substituted for PNA port numbers to which the ECAL module port is connected.

Ports of the module which are not used are omitted from the string.

For example, on a 4-port ECal module with

- port A connected to PNA port 2
- port B connected to PNA port 3
- port C not connected
- port D connected to PNA port 1

the string would be: A2,B3,D1

DoECAL1PortEx or **DoECAL2PortEx** methods will fail if the port numbers passed to those methods are not in the string of this property and OrientECALModule property = False.

Return Type String

Default Not Applicable

Examples

```
Dim cal As Calibrator
Dim sPortMap As String
Set cal = PNAapp.ActiveChannel.Calibrator
cal.ECALPortMapEx = “a2,b1” 'Write
sPortMap = cal.ECALPortMap 'Read
```

C++ Syntax

HRESULT put_ECALPortMapEx(long moduleNumber, BSTR strPortMap);

HRESULT get_ECALPortMapEx(long moduleNumber, BSTR *strPortMap);

Interface

ICalibrator4

Write/Read

ElecDelayMedium Property

Description Sets or returns the electrical delay medium.

VB Syntax *meas*.**ElecDelayMedium =** *value*

Variable **(Type) - Description**

meas A Measurement **(object)**

value **(enum NACalStandardMedium)** choose from

0 - naCoax

1 - naWaveGuide

Return Type NACalStandardMedium

Default Not Applicable

Examples

```
Print meas.ElecDelayMedium 'prints the value of the electrical delay medium
```

C++ Syntax HRESULT get_ElecDelayMedium(tagNACalStandardMedium *pVal);

HRESULT put_ElecDelayMedium(tagNACalStandardMedium newVal);

Interface IMeasurement2

ElectricalDelay Property

Description	Sets the Electrical Delay for the active channel.
VB Syntax	*meas*.**ElectricalDelay** = *value*
Variable	**(Type) - Description**
meas	A Measurement **(object)**
value	**(double)** - Electrical Delay in seconds. Choose any number between **-9.99** and **9.99**
Return Type	Double
Default	0
Examples	`meas.ElectricalDelay = 1e-3 'Write`
	`edelay = meas.ElectricalDelay 'Read`
C++ Syntax	HRESULT get_ElectricalDelay(double *pVal) HRESULT put_ElectricalDelay(double newVal)
Interface	IMeasurement

ErrorCorrection Property

Description Sets (or returns) error correction ON or OFF for the measurement.

VB Syntax *meas*.**ErrorCorrection** = *value*

Variable **(Type) - Description**

meas A Measurement **(object)**

value **(boolean)**
0 - Turns error correction OFF
1 - Turns error correction ON

Return Type Boolean

Default Not Applicable

Examples

```
meas.ErrorCorrection = 1 'Write
```

```
errcorr = meas.ErrorCorrection 'Read
```

C++ Syntax HRESULT put_ErrorCorrection (VARIANT_BOOL bState)
HRESULT get_ErrorCorrection (VARIANT_BOOL *bState)

Interface IMeasurement

Write/Read

ExternalALC Property

Description	Sets or returns the source of the analyzer leveling control.
VB Syntax	*app.***ExternalALC** = *value*
Variable	**(Type) - Description**
app	An Application **(object)**
value	**(boolean)** - Choose from: **True (or 1)** - Leveling control supplied through the rear panel. **False (or 0)** - Leveling control supplied inside the analyzer
Return Type	Boolean
Default	0

Examples	`app.ExternalALC = True 'Write`
	
	`extALC = app.ExternalALC 'Read`

C++ Syntax	HRESULT get_ExternalALC(VARIANT_BOOL *pVal) HRESULT put_ExternalALC(VARIANT_BOOL newVal)
Interface	IApplication

Write/Read.

ExternalTriggerConnectionBehavior Property

Description Configures the external triggering signal for the PNA. To receive trigger signals from an external source, TriggerSouce Property must be set to External.

Edge triggering is only available on some PNA models. For more information, see External Triggering.

VB Syntax *trigsetup.***ExternalTriggerConnectionBehavior** (*conn*) = *value*

Variable **(Type) - Description**

trigsetup A TriggerSetup **(object)**

conn **(enum NATriggerConnection)** Rear Panel connector to send or receive trigger signals. Choose from:

Only one of the input connectors is active at a time. When a command is sent to one, the PNA automatically makes the other INACTIVE.

0 - **naTriggerConnectionAUXT** Trigger IN from external source (AUX IO connector Pin 19)

1 - **naTriggerConnectionBNC1** Trigger IN from external source (Trigger IN BNC connector)

2 - **naTriggerConnectionBNC2** Trigger OUT (Trigger OUT BNC connector). Only useful in point sweep mode.

value **(enum NAExternalTriggerBehavior)** -

0 - **naTriggerInactive** - Disables the specified connector.

Choose from ONLY 1 through 4 when <conn> is set to either naTriggerConnection**BNC1** or naTriggerConnection**AUXT**

1 - **naTriggerInEdgeNegative** - Triggers the PNA when receiving a negative going signal

2 - **naTriggerInEdgePositive** - Triggers the PNA when receiving a positive going signal

3 - **naTriggerInLevelLow** - Triggers the PNA when receiving a low level signal

4 - **naTriggerInLevelHigh** - Triggers the PNA when receiving a High-level signal

Choose from ONLY 5 through 8 when <conn> is set to naTriggerConnection**BNC2.**

In addition to sending this command, you must also use TriggerOutputEnabled Property to enable the BNC2 output.

5 - **naTriggerOutPulsePositiveAfter** - Sends a POSITIVE going TTL pulse at the END of each point during the sweep.

6 - **naTriggerOutPulsePositiveBefore** - Sends a POSITIVE going TTL pulse at the START of each point during the sweep.

7 - **naTriggerOutPulseNegativeAfter** - Sends a NEGATIVE going TTL pulse at the END of each point during the sweep.

8 - **naTriggerOutPulseNegativeBefore** - Sends a NEGATIVE going TTL pulse at the START of each point during the sweep.

Return Type Enum as NAExternalTriggerBehavior

Default BNC1 = **naTriggerInactive**

BNC2 = **naTriggerInactive**

AUXT = **naTriggerInLevelHigh**

When Output is enabled

BNC1 = **naTriggerInactive**

BNC2 = **naTriggerOutPulsePositiveAfter**

AUXT = **naTriggerInLevelHigh**

Examples

```
trigsetup.ExternalTriggerConnectionBehavior (naTriggerConnectionAUXT) =
naTriggerInLevelLow 'Write
```

```
trigBehav = trigsetup.ExternalTriggerConnectionBehavior
(naTriggerConnectionAUXT) 'Read
```

C++ Syntax HRESULT get_ExternalTriggerConnectionBehavior(tagNATriggerConnection connection,tagNAExternalTriggerBehavior *trigger);

HRESULT put_ExternalTriggerConnectionBehavior(tagNATriggerConnection connection,tagNAExternalTriggerBehavior trigger);

Interface ITriggerSetup

Write/Read

ExternalTriggerDelay Property

Description	Sets and reads the trigger delay for all measurements in the CHANNEL. This delay is only applied while in app.Source = naTriggerSourceExternal and trigsetup.Scope = naChannelTrigger . After an external trigger is applied, the start of the sweep is delayed for the specified delay value plus any inherent latency. To apply a trigger delay for all channels (Global), use TriggerDelay Property.
VB Syntax	*chan.***ExternalTriggerDelay** = *value*
Variable	**(Type) - Description**
chan	A Channel **(object)**
value	**Double**- Trigger delay value in seconds. Range is from 0 to 107 seconds
Return Type	Double
Default	0
Examples	`chan.ExternalTriggerDelay = .003  'Write` `delay = chan.ExternalTriggerDelay 'Read`
C++ Syntax	HRESULT get_ExternalTriggerDelay(double *delay); HRESULT put_ExternalTriggerDelay(double delay)
Interface	IChannel6

About Marker Search

Read-only

FilterBW Property

Description	Returns the results of the SearchBandwidth method.
VB Syntax	*filtBW* = *meas*.**FilterBW**
Variable	**(Type) - Description**
filtBW	**(single) -** Variable to store bandwidth data
meas	A Measurement **(object)**
Return Type	Single
Default	Not applicable
Examples	`filterBW = meas.FilterBW 'Read`
C++ Syntax	HRESULT get_FilterBW(float* bw)
Interface	IMeasurement

Read-only

FilterCF Property

Description	Returns the Center Frequency result of the SearchBandwidth method.
VB Syntax	*filtCF* = *meas*.**FilterCF**
Variable	**(Type) - Description**
filtCF	**(double) -** Variable to store bandwidth CF data
meas	A Measurement **(object)**
Return Type	Double
Default	Not applicable
Examples	`filtCF = meas.FilterCF 'Read`
C++ Syntax	HRESULT get_FilterCF(double* centerFrequency)
Interface	IMeasurement

About Marker Search

Read-only

FilterLoss Property

Description	Returns the Loss value of the SearchBandwidth method.
VB Syntax	*filtLoss* = *meas*.**FilterLoss**
Variable	**(Type) - Description**
filtLoss	**(single) -** Variable to store bandwidth Loss data
meas	A Measurement **(object)**
Return Type	Single
Default	Not applicable
Examples	`filterLoss = meas.FilterLoss 'Read`
C++ Syntax	HRESULT get_FilterLoss(float* loss)
Interface	IMeasurement

About Marker Search

Read-only

FilterQ Property

Description	Returns the Q (quality factor) result of the SearchBandwidth method.
VB Syntax	*filtQ* = *meas*.**FilterQ**
Variable	**(Type) - Description**
filtQ	**(single)** - Variable to store bandwidth Q data
meas	A Measurement **(object)**
Return Type	Single
Default	Not applicable
Examples	`filtQ = meas.FilterQ 'Read`
C++ Syntax	HRESULT get_FilterQ(float* quality)
Interface	IMeasurement

FirmwareMajorRevision Property

Description	Returns the major firmware revision number as an integer. For example, given a firmware revision number A.03.30, this command returns 3.
VB Syntax	*value* **=** *cap*.**FirmwareMajorRevision**
Variable	**(Type) - Description**
value	(Long) - Variable to store the returned integer value of the firmware revision number.
cap	A Capabilities **(object)**
Return Type	Long
Default	Not Applicable
Examples	`value = cap.FirmwareMajorRevision 'Read`
C++ Syntax	HRESULT get_FirmwareMajorRevision(long * majorRev);
Interface	ICapabilities

FirmwareMinorRevision Property

Description Returns the minor firmware revision number as an integer. For example, given a firmware revision number A.03.30, this command returns 30.

VB Syntax *value* **=** *cap*.**FirmwareMinorRevision**

Variable **(Type) - Description**

value (Long) - Variable to store the returned decimal value of the firmware revision number.

cap A Capabilities **(object)**

Return Type Long

Default Not Applicable

Examples

```
value = cap.FirmwareMinorRevision 'Read
```

C++ Syntax HRESULT get_FirmwareMinorRevision(long * minorRev);

Interface ICapabilities

FirmwareSeries Property

Description	Returns the alpha portion of the firmware revision number. For example, given a firmware revision number A.03.30, this command returns A.
VB Syntax	*value* **=** *cap*.**FirmwareSeries**
Variable	**(Type) - Description**
value	(String) **-** Variable to store the returned alpha value of the firmware revision number.
cap	A Capabilities **(object)**
Return Type	String
Default	Not Applicable
Examples	`value = cap.FirmwareSeries 'Read`
C++ Syntax	HRESULT get_FirmwareSeries(BSTR * series);
Interface	ICapabilities

Read-only

FootSwitch Property

Description Reads the Footswitch Input (pin 20 of the AUX IO connector).

VB Syntax *value* = *AuxIO*.**Footswitch**

Variable **(Type) - Description**

value **(boolean) -** Variable to store the returned value

False (0) -foot switch is released

True (1) - footswitch is depressed

AuxIO **(object) -** A Hardware Aux I/O object

Return Type Boolean

Default True (1)

Examples

```
fs = aux.Footswitch
```

C++ Syntax HRESULT get_FootSwitch (VARIANT_BOOL* State);

Interface IHWAuxIO

FootswitchMode Property

Description Determines what occurs when the footswitch is pressed. For more information see the FootSwitch In pin description in the Auxiliary IO connector.

VB Syntax *AuxIo.***FootSwitchMode** = *value*

Variable **(Type) - Description**

value **(enum NAFootSwitchMode)**

0 - naIgnoreFootswitch - Footswitch presses are ignored.

1 - naSweepTrigger - Footswitch presses trigger a sweep. The PNA must be in Manual Trigger Mode.

2 - naRecallNextState - Footswitch presses recall an instrument state. When more than one state is available, then each footswitch press recalls the next state, then starts over from the beginning. It is possible for a recalled state to override the current mode. If the recalled state is IGNore, then mode changes and additional footswitch presses are ignored.

3 - naRunMacro - Footswitch presses load and run a macro. When more than one macro is available, then each footswitch press loads and runs the next macro, then starts over from the beginning. It is possible for a Macro to override the current mode. If the macro contains a Preset, then the mode changes to the default setting IGNore and additional footswitch presses are ignored.

AuxIO **(object) -** A Hardware Aux I/O object

Return Type **NAFootSwitchMode**

Default **0 - naIgnoreFootswitch**

Examples

```
auxIo.FootSwitchMode = naIgnoreFootSwitch 'Write
```

C++ Syntax HRESULT get_FootSwitchMode(NAFootSwitchMode *pFootSwitchMode)

HRESULT put_FootSwitchMode(NAFootSwitchMode newFootSwitchMode)

Interface IHWAuxIO3

Format Property (marker)

Description Sets (or returns) the format of the marker.

VB Syntax *mark*.**Format** = *value*

Variable **(Type) - Description**

mark A Marker **(object)**

value **(enum NAMarkerFormat)** - Choose from:

0 - naMarkerFormat_LinMag
1 - naMarkerFormat_LogMag
2 - naMarkerFormat_Phase
3 - naMarkerFormat_Delay
4 - naMarkerFormat_Real
5 - naMarkerFormat_Imaginary
6 - naMarkerFormat_SWR
7 - naMarkerFormat_LinMagPhase
8 - naMarkerFormat_LogMagPhase
9 - naMarkerFormat_RealImaginary
10 - naMarkerFormat_ComplexImpedance
11 - naMarkerFormat_ComplexAdmittance

Return Type NAMarkerFormat

Default **1** - naMarkerFormat_LogMag

Examples

```
mark.Format = naMarkerFormat_SWR 'Write
```

```
fmt = mark.Format 'Read
```

C++ Syntax HRESULT get_Format(tagNAMarkerFormat *pVal)
HRESULT put_Format(tagNAMarkerFormat newVal)

Interface IMarker

About Source Power Cal

Frequency Property

Description Sets or returns the frequency associated with a PowerSensorCalFactorSegment

or

Sets or returns the frequency associated with a PowerLossSegment.

VB Syntax *object.***Frequency** = *value*

Variable **(Type) - Description**

object **(object) -** PowerSensorCalFactorSegment or PowerLossSegment

value **(double)** – Frequency in units of Hz. This can be any non-negative value (limited by the maximum value of double).

Return Type Double

Default 0

Examples

```
seg.Frequency = 6e9 'Write
freq = seg.Frequency 'Read
```

C++ Syntax HRESULT put_Frequency(double newVal);

HRESULT get_Frequency(double *pVal);

Interface IPowerSensorCalFactorSegment

IPowerLossSegment

About Frequency Range

FrequencySpan Property

Description Sets or returns the frequency span of the channel.

Sets or returns the frequency span of the segment.

VB Syntax *object*.**FrequencySpan** = *value*

Variable **(Type) - Description**

object A Channel **(object)**
or
A Segment **(object)**

value **(double)** - Frequency span in Hertz. Choose any number between the **minimum** and **maximum** frequencies of the analyzer.

Return Type Double

Default Full frequency span of the analyzer

Examples

```
chan.FrequencySpan = 4.5e9 'sets the frequency span of a linear sweep
for the channel object -Write
```

```
freqspan = chan.FrequencySpan 'Read
```

C++ Syntax HRESULT get_FrequencySpan(double *pVal)
HRESULT put_FrequencySpan(double newVal)

Interface IChannel
ISegment

FrequencyOffsetDivisor Property

Description Specifies (along with FrequencyOffsetMultiplier) the value to multiply by the stimulus.

See other Frequency Offset properties

VB Syntax *object*.**FrequencyOffsetDivisor** = *value*

Variable **(Type) - Description**

object Channel **(object)**

or

CalSet **(object) -** Read-only property

value **(Double)** - Divisor value. Range is 1 to 1000

Return Type Double

Default 1

Examples

```
chan.FrequencyOffsetDivisor = 2 'Write
```

```
fOffsetDiv = chan.FrequencyOffsetDivisor 'Read
```

C++ Syntax HRESULT get_FrequencyOffsetDivisor(double*pval)

HRESULT put_FrequencyOffsetDivisor(double newVal)

Interface IChannel2

|CalSet3

FrequencyOffsetFrequency Property

Description	Specifies an absolute offset frequency in Hz. For mixer measurements, this would be the LO frequency. See other Frequency Offset properties.
VB Syntax	*object*.**FrequencyOffsetFrequency** = *value*
Variable	**(Type) - Description**
object	Channel **(object)** **or** CalSet **(object) -** Read-only property
value	**(Double)** - Offset value. Range is +/- 1000 GHz. (Offsets can be positive or negative.)
Return Type	Double
Default	0 Hz
Examples	`chan.FrequencyOffsetFrequency = 2 'Write` `fOffsetFreq = chan.FrequencyOffsetFrequency 'Read`
C++ Syntax	HRESULT get_FrequencyOffsetFrequency(double*pval) HRESULT put_FrequencyOffsetFrequency(double newVal)
Interface	IChannel2 |CalSet3

FrequencyOffsetMultiplier Property

Description Specifies (along with FrequencyOffsetDivisor) the value to multiply by the stimulus. See other Frequency Offset properties.

VB Syntax *object*.**FrequencyOffsetMultiplier** = *value*

Variable **(Type) - Description**

object Channel **(object)**

or

CalSet **(object) -** Read-only property

value **(Double)** - Multiplier value. Range is 1 to 1000

Return Type Double

Default 1

Examples

```
chan.FrequencyOffsetMultiplier = 2 'Write
```

```
fOffsetMult = chan.FrequencyOffsetMultiplier 'Read
```

C++ Syntax HRESULT get_FrequencyOffsetMultiplier (double*pval);
HRESULT put_FrequencyOffsetMultiplier (double newVal);

Interface IChannel2

|CalSet3

FrequencyOffsetCWOverride Property

Description	Establishes a fixed (CW) stimulus frequency while measuring the Response over a swept frequency range. For example, a fixed-frequency PNA stimulus may be applied to the RF input of a mixer whose local oscillator (LO) is being swept. Because the IF output of the mixer will be swept, the PNA receivers must also be swept. See other Frequency Offset properties.
VB Syntax	*object*.**FrequencyOffsetCWOverride** = *value*
Variable	**(Type) - Description**
object	Channel **(object)** **or** CalSet **(object) -** Read-only property
value	**(Enum as NaStates)** - Choose from: naOFF (0) - Turns CW override OFF naON (1) - Turns CW override ON
Return Type	Enum
Default	0 Hz
Examples	`chan.FrequencyOffsetCWOverride = 1 'Write` `fOffsetOV = chan.FrequencyOffsetCWOverride 'Read`
C++ Syntax	HRESULT get_FrequencyOffsetCWOverride (tagNAStates *pstate) HRESULT put_FrequencyOffsetCWOverride (tag NAStates newState)
Interface	IChannel2 |CalSet3

FrequencyOffsetState Property

Description Enables Frequency Offset on ALL measurements that are present on the active channel. This immediately causes the source and receiver to tune to separate frequencies. The receiver frequencies are specified with other channel and offset settings. To make the stimulus settings, use Channel Start, Stop Frequency properties. See other Frequency Offset properties.

Tip: To avoid unnecessary errors, first make other frequency offset settings. Then turn Frequency Offset ON.

VB Syntax *object*.**FrequencyOffsetState** = *value*

Variable **(Type) - Description**

object Channel **(object)**

or

CalSet **(object) -** Read-only property

value **(Enum as NaStates)** - Choose from:

naOFF (0) - Turns Frequency Offset OFF

naON (1) - Turns Frequency Offset ON

Return Type Enum

Default naOFF (0)

Examples

```
chan.FrequencyOffsetState = True 'Write
```

```
Foffset = chan.FrequencyOffsetState 'Read
```

C++ Syntax HRESULT FrequencyOffsetState (tag NAStates *pState);
HRESULT FrequencyOffsetState (tag NAStates newState)

Interface IChannel2

|CalSet3

Shape Property

Description Specifies the shape of the gate filter.

VB Syntax *gat*.**Shape** = *value*

Variable **(Type) - Description**

gat A Gating **(object)**

value **(enum** NAGateShape**) -** Choose from:

0 - naGateShapeMaximum
1 - naGateShapeWide
2 - naGateShapeNormal
3 - naGateShapeMinimum

Return Type NAGateShape

Default 2 - Normal

Examples

```
gat.Shape = naGateShapeMaximum 'Write
```

```
filterShape = gat.Shape 'Read
```

C++ Syntax HRESULT get_Shape(tagNAGateShape *pVal)
HRESULT put_Shape(tagNAGateShape newVal)

Interface IGating

Write/Read

Type Property

Description Specifies the type of gate filter used.

VB Syntax *gat*.**Type** = *value*

Variable **(Type) - Description**

gat A Gating **(object)**

value **(enum NAGateType)** - Choose from:
0 - naGateTypeBandpass - Includes (passes) the range between the start and stop times.
1 - naGateTypeNotch - Excludes (attenuates) the range between the start and stop times.

Return Type NAGateType

Default Bandpass

Examples

```
gate.Type = naGateTypeNotch 'Write
```

```
filterType = gate.Type 'Read
```

C++ Syntax HRESULT get_Type(tagNAGateType *pVal)
HRESULT put_Type(tagNAGateType newVal)

Interface IGating

GPIBAddress Property

Description	Sets and returns the PNA GPIB address bus.
VB Syntax	*app.***GPIBAddress** (*bus)* = *value*
Variable	**(Type) - Description**
app	An Application **(object)**
bus	(Short Integer) GPIB bus. MUST be set to 0.
value	(Short Integer) GPIB Address on the PNA. Choose a value between 0 and 30.
Return Type	Short Integer
Default	16
Examples	`address=app.GPIBAddress(0) 'Read` `app.GPIBAddress(0)=16 'Write`
C++ Syntax	HRESULT get_GPIBAddress(short busIndex, short* address); HRESULT put_GPIBAddress(short busIndex,short address);
Interface	IApplication8

Write/Read

GPIBMode Property

Description Changes the analyzer to a GPIB system controller or a talker/listener on the bus. The analyzer must be the controller if you want to use it to send commands to other instruments. The analyzer must be a talker/listener if you want to send it commands from another PC.

VB Syntax *app.***GPIBMode** *value*

Variable **(Type) - Description**

app An Application **(object)**

value **(enum NAGPIBMode)** -Choose either:
0 - naTalkerListener - the analyzer is a talker / listener
1 - naSystemController - the analyzer is the system controller

Return Type Long Integer

Default 0 - naTalkerListener

Examples

```
app.GPIBMode = naTalkerListener 'Write
```

```
mode = app.GPIBMode 'Read
```

C++ Syntax HRESULT get_GPIBMode(tagGPIBModeEnum* eGpibMode)
HRESULT put_GPIBMode(tagGPIBModeEnum eGpibMode)

Interface IApplication

Read-only

IDString Property

Description	Returns the ID of the analyzer, including the Model number, Serial Number, and the Software revision number.
VB Syntax	*value* = *app.***IDString**
Variable	**(Type) - Description**
app	An Application **(object)**
value	**(string)** - variable to contain the returned ID string
Return Type	String
Default	Not Applicable
Examples	`id = app.IDString`
C++ Syntax	HRESULT IDString(BSTR* IDString)
Interface	IApplication

Write/Read

IFBandwidthOption Property

Description	Enables the IFBandwidth to be set on individual sweep segments. This property must be set True **before** *seg*.IFBandwidth = *value* is sent. Otherwise, this command will be ignored.
VB Syntax	*segs*.**IFBandwidthOption** = *value*
Variable	**(Type) - Description**
segs	A Segments collection **(object)**
value	**(boolean)** **True** - Enables variable IFBandwidth setting for segment sweep **False** - Disables variable IFBandwidth setting for segment sweep
Return Type	Boolean
Default	False

Examples	`segs.IFBandwidthOption = True 'Write`
	
	`IFOption = IFBandwidthOption 'Read`

C++ Syntax	HRESULT get_IFBandwidthOption(VARIANT_BOOL *pVal) HRESULT put_IFBandwidthOption(VARIANT_BOOL newVal)
Interface	ISegments

Write/Read

About IF Bandwidth

IFBandwidth Property

Description	Sets or returns the IF Bandwidth of the channel. Sets or returns the IF Bandwidth of the segment. Returns the IF Bandwidth used in the Cal Set
VB Syntax	*object*.**IFBandwidth** = *value*
Variable	**(Type) - Description**
object	Channel **(object)** **or** Segment **(object)** **or** CalSet **(object) -** Read-only property
value	**(double)** - IF Bandwidth in Hz. The list of valid IF Bandwidths is different depending on the PNA model. (Click to see the lists.) If an invalid number is specified, the analyzer will round up to the closest valid number.
Return Type	Double
Default	See Preset IFBW for your PNA model.

Examples	`chan.IFBandwidth = 3e3 'sets the IF Bandwidth of for the channel object to 3 kHz. -Write`
	`seg.IFBandwidth = 5 'sets the IF Bandwidth of the segment to 5 Hz. -Write`
	
	`ifbw = chan.IFBandwidth -Read`

C++ Syntax	HRESULT get_IFBandwidth(double *pVal); HRESULT put_IFBandwidth(double newVal);
Interface	IChannel ISegment ICalSet3

IFDenominator Property

Description	Sets or returns the denominator value of the IF Fractional Multiplier. Only applies to 2 stage mixers!
VB Syntax	*mixer*.**IFDenominator =** *value*
Variable	**(Type) - Description**
mixer	A Mixer **(object)**
value	**(long)** IF Denominator value.
Return Type	Long
Default	1
Examples	`Print mixer.IFDenominator 'prints the value of the IFDenominator`
C++ Syntax	HRESULT get_IFDenominator(long *pVal) HRESULT put_IFDenominator(long newVal)
Interface	IMixer

IFFilterSampleCount Property

Description Sets or retrieves the number of taps in the IF filter when the IFFilterSource property is set to naIFFilterSourceManual.

Note: An error will occur if this command is used on a PNA without option H08 installed.

VB Syntax *IfConfig*.**IFFilterSampleCount** = *value*

Variable **(Type) - Description**

IfConfig IFConfiguration (object)

value **(long)** – The IF filter sample count. The minimum and maximum allowed values for this property can vary by model number, but can be queried using the MinimumIFFilterSampleCount and MaximumIFFilterSampleCount properties.

Return Type Long Integer

Default PNA Model number dependent

Examples

```
App.ActiveChannel.IFConfiguration.IFFilterSampleCount = 200 'Write
```

```
variable = App.ActiveChannel.IFConfiguration.IFFilterSampleCount 'Read
```

See an example program

C++ Syntax HRESULT get_IFFilterSampleCount(long * pSampleCount);

HRESULT put_IFFilterSampleCount (long sampleCount);

Interface IIFConfiguration2

IFFilterSamplePeriod Property

Description Sets or returns the IF filter sample period time. This time is only used if the IFFilterSamplePeriodMode is set to naManual.

Note: An error will occur if this command is used on a PNA without option H08 installed.

VB Syntax *IfConfig*.**IFFilterSamplePeriod** = *value*

Variable **(Type) - Description**

IfConfig IFConfiguration (object)

value **(double)** – The sample period time in seconds. Valid sample period times can be queried using the IFFilterSamplePeriodList property.

Return Type Double

Default PNA Model number dependent.

Examples

```
App.ActiveChannel.IFConfiguration.IFFilterSamplePeriod = .000006 'Write
```

```
variable = App.ActiveChannel.IFConfiguration.IFFilterSamplePeriod 'Read
```

See an example program

C++ Syntax HRESULT get_IFFilterSamplePeriod(double * pSamplePeriod);

HRESULT put_IFFilterSamplePeriod (double samplePeriod);

Interface IIFConfiguration2

IFFilterSamplePeriodList Property

Description	Retrieves the list of available IF filter sample periods for the instrument. **Note:** An error will occur if this command is used on a PNA without option H08 installed.
VB Syntax	*variable* = *IfConfig*.**IFFilterSamplePeriodList**
Variable	**(Type) - Description**
variable	(Array) An array of permissible values that can be passed to the IFFilterSamplePeriod property.
IfConfig	IFConfiguration (object)
Return Type	Array
Default	Not applicable
Examples	See below

```
Dim Variable
Variable =
App.ActiveChannel.IFConfiguration.IFFilterSamplePeriodList 'Read
MsgBox “First IF Sample Period Value: “ & Variable(0)
```

See an example program

C++ Syntax	HRESULT get_IFFilterSamplePeriodList(SAFEARRAY ** ppSamplePeriodList);
Interface	IIFConfiguration2

IFFilterSamplePeriodMode Property

Description Sets or returns the IF filter sample period mode.

Note: An error will occur if this command is used on a PNA without option H08 installed.

VB Syntax *IfConfig*.**IFFilterSamplePeriodMode** = *value*

Variable **(Type) - Description**

IfConfig IFConfiguration (object)

value **(enum NAModes)** -

0 - naAuto - IF filter sample period is chosen automatically.

1 - naManual - the IF filter sample period is the value specified by the IFSamplePeriod property.

Return Type NAModes

Default 0 - naAuto

Examples

```
App.ActiveChannel.IFConfiguration.IFFilterSamplePeriodMode = naAuto
'Write
```

```
variable = App.ActiveChannel.IFConfiguration.IFFilterSamplePeriodMode
'Read
```

See an example program

C++ Syntax HRESULT get_IFFilterSamplePeriodMode(tagNAModes * pMode);

HRESULT put_IFFilterSamplePeriodMode (tagNAModes mode);

Interface IIFConfiguration2

IFFilterSource Property

Description Sets or retrieves type of IF filter to be used.

Note: An error will occur if this command is used on a PNA without option H08 installed.

VB Syntax *IfConfig*.**IFFilterSource** = *value*

Variable **(Type) - Description**

object IFConfiguration (object)

value **(NAIFFilterSource)** – The type of filter to use. Choose from:

naIFFilterSourceAuto – The IF filter type is automatically chosen.

naIFFilterSourceManual – The IF filter is a predetermined shape where the IFFilterSampleCount determines the number of taps in the filter.

Return Type NAIFFilterSource

Default naIFFilterSourceAuto

Examples

```
App.ActiveChannel.IFConfiguration.IFFilterSource =
naIFFilterSourceManual 'Write
```

```
Variable = App.ActiveChannel.IFConfiguration.IFFilterSource 'Read
```

See an example program

C++ Syntax HRESULT get_IFFilterSource(tagNAIFFilterSource * pFilterSource);

HRESULT put_IFFilterSource(tagNAIFFilterSource filterSource);

Interface IIFConfiguration2

IFGainLevel Property

Description Manually sets the gain level for the specified receiver.

VB Syntax *IfConfig*.**IFGainLevel** *(id)* **=** *value*

Variable **(Type) - Description**

IfConfig An IFConfiguration **(object)**

id Receiver for which to set the gain level. Choose from: 'A', 'B', 'R1', 'R2'.

Note: The A and R1 receivers are always switched together. B and R2 are also always switched together. For example, if you specify 'A', R1 will also be switched.

value (long Integer) Gain Level. Choose from:

0 - Lowest gain setting
1
2 - Highest gain setting

Return Type Long Integer

Default 0 (Lowest setting)

Examples

```
IfConfig.IFGainLevel('A') = 1
```

C++ Syntax HRESULT get_IFGainLevel (BSTR IDString, *GainLevel)

HRESULT put_IFGainLevel (BSTR IDString, GainLevel)

Interface IFConfiguration

IFGainMode Property

Description Sets the gain state for ALL receivers to Auto or Manual.

VB Syntax *IfConfig*.**IFGainMode** *(id)* **=** *value*

Variable **(Type) - Description**

IfConfig An IFConfiguration **(object)**

id Receivers for which to set the state. Choose 'ALL'.

value **(enum as NAModes)**

Choose from:

0 - naAUTO

1 - naMANUAL (use IFGainLevel Property to manually set gain level)

Return Type NAModes

Default 0 - naAUTO

Examples

```
IfConfig.IFGainMode("ALL") = naAUTO
```

C++ Syntax HRESULT get_IFGainMode (BSTR IDString, NAModes *gainMode)

HRESULT put_IFGainMode (BSTR IDString, NAModes gainMode)

Interface IIFConfiguration

IFGateEnable Property

Description Sets or retrieves the state of the IF Gate.

Note: An error will occur if this command is used on a PNA without option H08 installed.

VB Syntax *IfConfig*.**IFGateEnable** = *value*

Variable **(Type) - Description**

IfConfig IFConfiguration (object)

value **(Boolean**) – The state of the IF Gate.

True – The IF Gate is in use.

False – The IF Gate is not in use.

Return Type Boolean

Default False

Examples

```
App.ActiveChannel.IFConfiguration.IFGateEnable = True 'Write
variable = App.ActiveChannel.IFConfiguration.IFGateEnable 'Read
```

See an example program

C++ Syntax HRESULT get_IFGateEnable(VARIANT_BOOL * pGateEnabled);

HRESULT put_IFGateEnable(VARIANT_BOOL gateEnabled);

Interface IIFConfiguration

IFNumerator Property

Description	Sets or returns the numerator value of the IF Fractional Multiplier. Only applies to 2 stage mixers!
VB Syntax	*mixer*.**IFNumerator =** *value*
Variable	**(Type) - Description**
mixer	A Mixer **(object)**
value	**(Long)**
Return Type	long
Default	Not Applicable
Examples	`Print mixer.IFNumerator 'prints the value of the IFNumerator`
C++ Syntax	HRESULT get_IFNumerator(long *pVal) HRESULT put_IFNumerator(long newVal)
Interface	IMixer

IFSideband Property

Description	Sets or returns the value of the IF sideband, high or low. Only applies to 2 stage mixers!
VB Syntax	*mixer*.**IFSideband =***value*
Variable	**(Type) - Description**
mixer	A Mixer **(object)**
value	**(enum as FCASideBand)** - Choose from: 0 - LOW 1 - HIGH
Return Type	FCASideBand
Default	0 - LOW
Examples	`Print mixer.IFSideband 'prints the value of the IFSideband`
C++ Syntax	HRESULT get_IFSideband(FCASideBand *pVal) HRESULT put_IFSideband(FCASideBand newVal)
Interface	IMixer

IFSourcePath Property

Description Sets the source path of the specified receiver. An error is returned if *<id>* is not valid, or if *<value>* is not valid for the specified *<id>*.

VB Syntax *IfConfig*.**IFSourcePath** *(id)* **=** *value*

Variable **(Type) - Description**

IfConfig An IFConfiguration **(object)**

id Receiver for which to set the gain level. Choose from: 'A', 'B', 'R1', 'R2'..

Note: The A and R1 receivers are always switched together. B and R2 are also always switched together. For example, if you specify "A", R1 will also be switched.

value **(Enum as NAIFSourcePath)** Source path. Choose from:

0 - **naNormalIFPath** - the PNA decides the appropriate IF input paths.

1 **- naExternalIFPath -** always use the rear panel IF inputs.

Return Type NAIFSourcePath

Default 0 - naNormalIFPath

Examples

```
IfConfig.IFSourcePath('A') = naNormalIFPath
```

C++ Syntax HRESULT get_IFSourcePath (BSTR IDString, naIFSourcePath *IFSourcePath)

HRESULT put_IFSourcePath (BSTR IDString, naIFSourcePath IFSourcePath)

Interface IIFConfiguration

IFStartFrequency Property

Description	Sets or returns the start frequency value of the mixer IF frequency. Only applies to 2 stage mixers!
VB Syntax	*mixer*.**IFStartFrequency =** *value*
Variable	**(Type) - Description**
mixer	A Mixer **(object)**
value	**(double)** - Frequency in Hertz.
Return Type	Double
Default	Not Applicable
Examples	`Print mixer.IFStartFrequency 'prints the value of the IFStartFrequency`
C++ Syntax	HRESULT get_IFStartFrequency(double *pVal) HRESULT put_IFStartFrequency(double newVal)
Interface	IMixer

IFStopFrequency Property

Description Sets or returns the stop frequency value of the mixer IF frequency.

Only applies to 2 stage mixers!

VB Syntax *mixer*.**IFStopFrequency =** *value*

Variable **(Type) - Description**

mixer A Mixer **(object)**

value **(double)** - IF stop frequency in Hertz.

Return Type Double

Default Not Applicable

Examples

```
Print mixer.IFStopFrequency 'prints the value of the
IFStopFrequency
```

C++ Syntax HRESULT get_IFStopFrequency(double *pVal)

HRESULT put_IFStopFrequency(double newVal)

Interface IMixer

Write/Read

ImpulseWidth Property

Description Sets or returns the Impulse Width of Time Domain transform windows

VB Syntax *trans*.**ImpulseWidth** = *value*

Variable **(Type) - Description**

trans A Transform **(object)**

value **(double)** - Impulse Width in seconds. Range of settings depends on the frequency range of your analyzer.

Return Type Double

Default .98 / Default Span

Examples

```
trans.ImpulseWidth = 200e-12 'sets the Impulse width of a transform window -Write
```

```
IW = trans.ImpulseWidth 'Read
```

C++ Syntax HRESULT get_ImpulseWidth(double *pVal)
HRESULT put_ImpulseWidth(double newVal)

Interface ITransform

IndexState Property

Description Determines the control of Material Handler connector Pin 20.

VB Syntax *handler*.**IndexState** = *value*

Variable **(Type) - Description**

handler **(object) -** A Handler I/O object

value (**boolean**)

0 (False) **-** Pin 20 is controlled by Output Port B6

1(True) **-** Pin 20 is controlled by the Index signal

Return Type Boolean

Default

Examples

```
handler.IndexState = False 'Write
bState = handler.IndexState 'Read
```

C++ Syntax HRESULT put_IndexState (BOOL *pVal);

HRESULT get_IndexState (BOOL newVal);

Interface IHWMaterialHandlerIO2

InputA Property

Description Sets a Port Extension value for Receiver A

VB Syntax *portExt*.**InputA** = *value*

Variable **(Type) - Description**

portExt A Port Extension **(object)**

value **(double)** - Port Extension value in seconds. Choose any number between **-10** and **10**

Return Type Double

Default 0

Examples

```
portExt.InputA = 10e-6 'Write
```

```
inA = portExt.InputA 'Read
```

C++ Syntax HRESULT get_InputA(double *pVal)
HRESULT put_InputA(double newVal)

Interface IPortExtension

About Port Extensions

Write/Read

InputB Property

Description	Sets the Port Extension value for Receiver B
VB Syntax	*portExt*.**InputB** = *value*
Variable	**(Type) - Description**
portExt	A Port Extension **(object)**
value	**(double)** - Port Extension value in seconds. Choose any number between **-10** and **10**
Return Type	Double
Default	0
Examples	`portExt.InputB = 10e-6 'Write`
	`inB = portExt.InputB 'Read`
C++ Syntax	HRESULT get_InputB(double *pVal) HRESULT put_InputB(double newVal)
Interface	IPortExtension

About Port Extensions

Write/Read

InputC Property

Description	Sets the Port Extension value for Receiver C
VB Syntax	*portExt*.**InputC** = *value*
Variable	**(Type) - Description**
portExt	A Port Extension **(object)**
value	**(double)** - Port Extension value in seconds. Choose any number between **-10** and **10**
Return Type	Double
Default	0
Examples	`portExt.InputC = 10e-6 'Write`
	`inC = portExt.InputC 'Read`
C++ Syntax	HRESULT get_InputC(double *pVal) HRESULT put_InputC(double newVal)
Interface	IPortExtension

InputDenominator Property

Description	Sets or returns the denominator value of the Input Fractional Multiplier.
VB Syntax	*mixer*.**InputDenominator =** *value*
Variable	**(Type) - Description**
mixer	A Mixer **(object)**
value	**(Long)** - Input denominator value.
Return Type	Long
Default	1
Examples	`Print mixer.InputDenominator 'prints the value of the InputDenominator`
C++ Syntax	HRESULT get_InputDenominator(long *pVal) HRESULT put_InputDenominator(long newVal)
Interface	IMixer

IsInputGreaterThanLO Property

Description Specifies whether to use the Input frequency that is greater than the LO or less than the LO. To learn more, see the mixer setup dialog box help.

VB Syntax *mixer*.**IsInputGreaterThanLO** *(LO)* **=** *bool*

Variable **(Type) - Description**

mixer A Mixer **(object)**

LO (Integer) - LO stage number

Choose from 1 (default) or 2

bool **(Boolean)** -

True (1) - Use the Input that is Greater than the specified LO.

False (0) - Use the Input that is Less than the specified LO.

Return Type Boolean

Default True

Examples

```
mixer.IsInputGreaterThanLO(1) = True
```

C++ Syntax HRESULT get_IsInputGreaterThanLO(VARIANT_BOOL * val);

HRESULT put_IsInputGreaterThanLO(VARIANT_BOOL val);

Interface IMixer2

InputNumerator Property

Description Sets or returns the numerator value of the Input Fractional Multiplier.

VB Syntax *mixer*.**InputNumerator =** *value*

Variable **(Type) - Description**

mixer A Mixer **(object)**

value **(Long)** - Input numerator value.

Return Type Long

Default 1

Examples

```
Print mixer.InputNumerator 'prints the value of the
InputNumerator
```

C++ Syntax HRESULT get_InputNumerator(long *pVal)

HRESULT put_InputNumerator(long newVal)

Interface IMixer

InputPower Property

Description	Sets or returns the value of the Input Power.
VB Syntax	*mixer*.**InputPower =** *value*
Variable	**(Type) - Description**
mixer	A Mixer **(object)**
value	**(double)** - Input power in dBm.
Return Type	Double
Default	-17dBm
Examples	`Print mixer.InputPower 'prints the value of the InputPower`
C++ Syntax	HRESULT get_InputPower(double *pVal) HRESULT put_InputPower(double newVal)
Interface	IMixer

InputStartFrequency Property

Description	Sets and returns the start frequency value of the mixer Input frequency.
VB Syntax	*mixer*.**InputStartFrequency =** *value*
Variable	**(Type) - Description**
mixer	A Mixer **(object)**
value	**(double)** - Input start frequency in Hertz.
Return Type	Double
Default	Start frequency of the PNA
Examples	`mixer.InputStartFrequency = Start_Freq`
C++ Syntax	HRESULT get_InputStartFrequency(double *pVal) HRESULT put_InputStartFrequency(double newVal)
Interface	IMixer

InputStopFrequency Property

Description	Sets and returns the stop frequency value of the mixer Input frequency.
VB Syntax	*mixer*.**InputStopFrequency =** *value*
Variable	**(Type) - Description**
mixer	A Mixer **(object)**
value	**(double)** - Input stop frequency in Hertz.
Return Type	Double
Default	Stop frequency of the PNA
Examples	`mixer.InputStopFrequency = Stop_Freq`
C++ Syntax	HRESULT get_InputStopFrequency(double *pVal) HRESULT put_InputStopFrequency(double newVal)
Interface	IMixer

InternalTestsetPortCount Property

Description	Returns the number of PNA test ports on the internal testset.
VB Syntax	*value* **=** *cap*.**InternalTestsetPortCount**
Variable	**(Type) - Description**
value	(Long) - Variable to store the returned number of PNA test ports.
cap	A Capabilities **(object)**
Return Type	Long
Default	Not Applicable
Examples	`value = cap.InternalTestsetPortCount 'Read`
C++ Syntax	HRESULT get_InternalTestsetPortCount(long * numPorts);
Interface	ICapabilities

About Interpolation

Write/Read

Interpolate Correction Property

Description Turns ON and OFF correction interpolation which calculates new error terms when stimulus values change after calibration.

When this property is ON and error correction is being applied, the calibration subsystem attempts to interpolate the error terms whenever the stimulus parameters are changed.

When this property is OFF under the same circumstances, error correction is turned OFF.

VB Syntax *meas*.**InterpolateCorrection** = *value*

Variable **(Type) - Description**

meas A Measurement **(object)**

value **(boolean)** - Choose from:

True - Turns correction interpolation ON
False - Turns correction interpolation OFF

Return Type Boolean

Default True

Examples

```
meas.InterpolateCorrection = False
```

```
calInterpolate = InterpolateCorrection 'Read
```

C++ Syntax HRESULT get_InterpolateCorrection(boolean *pVal)
HRESULT put_InterpolateCorrection(boolean newVal)

Interface IMeasurement

Write/Read

InterpolateNormalization Property

Description Turns ON and OFF normalization interpolation which calculates new divisor data when stimulus values change after normalization.

When this property is ON and normalization is being applied, the Normalization algorithm attempts to interpolate the divisor data whenever the stimulus parameters are changed.

When this property is OFF under the same circumstances, normalization is turned OFF.

Normalization is currently supported only on measurements of unratioed power for the purpose of performing a receiver power calibration.

VB Syntax *meas*.**InterpolateNormalization** = *value*

Variable **(Type) - Description**

meas **(object) -** A Measurement object

value **(boolean)**

0 – Turns normalization interpolation OFF

1 – Turns normalization interpolation ON

Return Type Boolean

Default 0 -OFF

Examples

```
meas.InterpolateNormalization = 1 'Write
```

```
normalized = meas.InterpolateNormalization 'Read
```

C++ Syntax HRESULT put_InterpolateNormalization(VARIANT_BOOL bState);

HRESULT get_InterpolateNormalization(VARIANT_BOOL *bState);

Interface IMeasurement

About the ExtTestSetIO connector

Interrupt Property

Description Reads the boolean that represents the state of the Interrupt In line (pin 13) on the external test set connector.

VB Syntax *value* = *ExtIO*.**Interrupt**

Variable **(Type) - Description**

value **(boolean)** - Variable to store the returned data

ExtIO **(object) -** An ExternalTestSetIO object

Return Type Boolean

False (0) - indicates the line is being held at a TTL High

True (1) - indicates the line is being held at a TTL Low

Default Not Applicable

Examples

```
value = ExtIO.Interrupt
```

C++ Syntax HRESULT get_Interrupt(VARIANT_BOOL* bValue);

Interface IHWExternalTestSetIO

IsContinuous Property

Description	Returns whether or not a channel is in continuous mode. To set the channel to continuous mode, use Continuous Method.
VB Syntax	*value* = *chan*.**IsContinuous**
Variable	**(Type) - Description**
value	**(boolean)** - Choose either: **False (0)** - Channel trigger is NOT set to continuous. **True (1)** - Channel trigger IS set to continuous.
chan	Channel **(object)**
Return Type	boolean
Default	Not Applicable
Examples	`trig = chan.IsContinuous 'Read`
C++ Syntax	HRESULT get_IsContinuous (VARIANT_BOOL *bContinuous);
Interface	IChannel3

IsECALModuleFoundEx Property - OBSOLETE

Description

Note: This command is replaced by ECALModuleNumberList and ECALCharacterizationIndexList

Returns true or false depending on whether communication was established between the PNA and the specified ECal module.

VB Syntax *modFound* = *cal*.**IsECALModuleFoundEx** *(module)*

Variable **(Type) - Description**

modFound **(boolean) -** Variable to store the returned test result.

True - The PNA identified the presence of the specified ECal module.

False - The PNA did NOT identify the presence of the specified ECal module.

cal **(object) -** A Calibrator object

module **(long integer)** ECal module.

Choose from modules **1** through **8**

Use GetECALModuleInfoEx to return the model and serial number of each module.

Return Type Boolean

Default Not applicable

Examples

```
Set cal = pna.ActiveChannel.Calibrator
moduleFound = cal.IsECALModuleFoundEx(1)
```

C++ Syntax HRESULT get_IsECALModuleFoundEx(long moduleNumber, VARIANT_BOOL *bModuleFound);

Interface ICalibrator4

IsFrequencyOffsetPresent Property

Description	Returns a value indicating the presence of Frequency Offset Option 080 in the remote PNA.
VB Syntax	*value* **=** *cap*.**IsFrequencyOffsetPresent**
Variable	**(Type) - Description**
value	(Boolean) - Variable to store the returned value (True or False).
cap	A Capabilities **(object)**
Return Type	Boolean
Default	Not Applicable
Examples	`value = cap.isFrequencyOffsetPresent(1) 'Read`
C++ Syntax	HRESULT get_IsFrequencyOffsetPresent(VARIANT_BOOL * present);
Interface	ICapabilities

IsHold Property

Description Returns whether or not a channel is in hold mode. To set the channel to hold mode, use Hold Method.

VB Syntax *value* = *chan*.**IsHold**

Variable **(Type) - Description**

value **(boolean)** - Choose either:

False (0) - Channel trigger is NOT set to hold.

True (1) - Channel trigger IS set to hold.

chan Channel **(object)**

Return Type boolean

Default Not Applicable

Examples

```
trig = chan.IsHold 'Read
```

C++ Syntax HRESULT get_IsHold (VARIANT_BOOL *bHold);

Interface IChannel3

IsReceiverStepAttenuatorPresent Property

Description	Returns a value indicating the presence of Receiver step attenuators in the remote PNA.
VB Syntax	*value* **=** *cap*.**IsReceiverStepAttenuatorPresent** *(n)*
Variable	**(Type) - Description**
value	**(Boolean)** - Variable to store the returned value.
cap	A Capabilities **(object)**
n	**(Long) -** port number to query for step attenuators
Return Type	Boolean
Default	Not Applicable
Examples	`value = cap.IsReceiverStepAttenuatorPresent(1)'Read`
C++ Syntax	HRESULT get_IsReceiverStepAttenuatorPresent(long portNumber, VARIANT_BOOL * present);
Interface	ICapabilities

IsReferenceBypassSwitchPresent Property

Description	Returns a value indicating the presence of a Reference Bypass Switch in the remote PNA.
VB Syntax	*value* **=** *cap*.**IsReferenceBypassSwitchPresent** *(n)*
Variable	**(Type) - Description**
value	(Boolean) - Variable to store the returned value (True or False).
cap	A Capabilities **(object)**
n	**(Long) -** port number to query for reference bypass switch
Return Type	Boolean
Default	Not Applicable
Examples	`value = cap.IsReferenceBypassSwitchPresent(1) 'Read`
C++ Syntax	HRESULT get_IsReferenceBypassSwitchPresent(long portNumber, VARIANT_BOOL * present);
Interface	ICapabilities

Read-only

IsSParameter Property

Description	Returns true if measurement represents an S-Parameter
VB Syntax	*value* = *meas*.**IsSparameter**
Variable	**(Type) - Description**
meas	A Measurement **(object)**
value	**(Boolean)** 1 True - measurement is an S-Parameter 0 False - measurement is NOT an S-Parameter
Return Type	Boolean
Default	True
Examples	`print app.IsSparameter`
C++ Syntax	HRESULT IsSparameter([out, retval] VARIANT_BOOL * bVal);
Interface	IMeasurement2

Write / Read

IterationsTolerance Property

Description Sets the maximum desired deviation from the sum of the test port power and the offset value. Power readings will continue to be made, and source power adjusted, until a reading is within this tolerance value or the max number of readings has been met. The last value to be read is the valid reading for that data point.

VB Syntax *pwrCal*.**IterationsTolerance** = *value*

Variable **(Type) - Description**

pwrCal **(object) -** A SourcePowerCalibrator (object)

value **(Double)** – Tolerance value in dBm. Choose any number between 0 and 5

Return Type Double

Default .05 dB

Examples

```
Set powerCalibrator = pna.SourcePowerCalibrator
powerCalibrator.IterationsTolerance = .1 'Write
ReadTol = powerCalibrator.IterationsTolerance 'Read
```

C++ Syntax HRESULT get_IterationsTolerance(double *pVal);

HRESULT put_IterationsTolerance(double newVal);

Interface ISourcePowerCalibrator3

Write/Read

KaiserBeta Property

Description	Sets or returns the Kaiser Beta of Time Domain transform windows
VB Syntax	*trans*.**KaiserBeta** = *value*
Variable	**(Type) - Description**
trans	A Transform **(object)**
value	**(single)** - Kaiser Beta. Choose any number between **0** and **13.**
Return Type	Single
Default	0

Examples

```
trans.KaiserBeta = 6 'sets the Kaiser Beta of a transform window -Write
```

```
KB = trans.KaiserBeta  'Read
```

C++ Syntax

HRESULT get_KaiserBeta(float *pVal)
HRESULT put_KaiserBeta(float newVal)

Interface ITransform

L0 Property

Description Sets and Returns the L0 (L-zero) value (the first inductance value) for the calibration standard.

To set the other inductance values, use L1, L2, L3

VB Syntax *calstd*.**L0** = *value*

Variable **(Type) - Description**

calstd A CalStandard **(object).** Use calKit.GetCalStandard to get a handle to the standard.

value **(single)** - Value for L0 in picohenries

Return Type Single

Default Not Applicable

Examples

```
calstd.L0 = 15 'Write the value of L0 = 15picohenries
```

```
Induct0 = calstd.L0 'Read the value of L0
```

C++ Syntax HRESULT get_L0(float *pVal)
HRESULT put_L0(float newVal)

Interface ICalStandard

Write/Read

L1 Property

Description	Sets and Returns the L1 value (the second inductance value) for the calibration standard. To set the other inductance values, use L0, L2, L3
VB Syntax	*calstd*.**L1** = *value*
Variable	**(Type) - Description**
calstd	A CalStandard **(object).** Use calKit.GetCalStandard to get a handle to the standard.
value	**(single)** - Value for L1 in picohenries
Return Type	Single
Default	Not Applicable
Examples	`calstd.L1 = 15 'Write the value of L1 = 15picohenries` `Induct1 = calstd.L1 'Read the value of L1`
C++ Syntax	HRESULT get_L1(float *pVal) HRESULT put_L1(float newVal)
Interface	ICalStandard

Write/Read

L2 Property

Description Sets and Returns the L2 value (the third inductance value) for the calibration standard.

To set the other inductance values, use L0, L1, L3

VB Syntax *calstd*.**L2** = *value*

Variable **(Type) - Description**

calstd A CalStandard **(object).** Use calKit.GetCalStandard to get a handle to the standard.

value **(single)** - Value for L2 in picohenries

Return Type Single

Default Not Applicable

Examples

```
calstd.L2 = 15 'Write the value of L2 to 15picohenries
```

```
Induct2 = calstd.L2 'Read the value of L2
```

C++ Syntax HRESULT get_L2(float *pVal)
HRESULT put_L2(float newVal)

Interface ICalStandard

Write/Read

L3 Property

Description	Sets and Returns the L3 value (the third inductance value) for the calibration standard. To set the other inductance values, use L0, L1, L2
VB Syntax	*calstd*.**L3** = *value*
Variable	**(Type) - Description**
calstd	A CalStandard **(object).** Use calKit.GetCalStandard to get a handle to the standard.
value	**(single)** - Value for L3 in picohenries
Return Type	Single
Default	Not Applicable
Examples	`calstd.L3 = 15 'Write the value of L3 to 15picohenries` `Induct3 = calstd.L3 'Read the value of L3`
C++ Syntax	HRESULT get_L3(float *pVal) HRESULT put_L3(float newVal)
Interface	ICalStandard

Write/Read

Label Property

Description	Sets and Returns the label for the calibration standard. The label is used to prompt the user to connect the specified standard.
VB Syntax	*calstd*.**Label** = *value*
Variable	**(Type) - Description**
calstd	A CalStandard **(object).** Use calKit.GetCalStandard to get a handle to the standard.
value	**(string) -** between 1 and 12 characters long. Cannot begin with a numeric.
Return Type	String
Default	Not Applicable
Examples	`calstd.Label = "Short" 'Write` `stdLabel = calstd.Label 'Read`
C++ Syntax	HRESULT get_Label(BSTR *pVal) HRESULT put_Label(BSTR newVal)
Interface	ICalStandard

LastModified Property

Description	Returns the time stamp of the last modification to this CalSet
VB Syntax	*value* = *Object*.**LastModified**
Variable	**(Type) - Description**
object	Channel **(object)** **or** CalSet **(object) -** Read-only property
value	**(Variant)** – Variable to store the time stamp.
Return Type	Variant
Default	Not Applicable
Examples	`date = CalSet.LastModified 'Read`
C++ Syntax	HRESULT get_LastModified(VARIANT* datetime)
Interface	ICalSet3

Read-only

LimitTestFailed Property

Description	Returns the results of limit testing for the measurement.
VB Syntax	*testFailed* = *meas*.**LimitTestFailed**
Variable	**(Type) - Description**
testFailed	**(boolean)** Variable to store the returned value **False (0**) - Limit Test Passed **True (1**) - Limit Test Failed
meas	A Measurement **(object)**
Return Type	Boolean
Default	False returned if there is no testing in progress

Examples	`Dim testRes As Boolean`
	`testRes = meas.LimitTestFailed`
	`MsgBox (testRes)`

C++ Syntax	HRESULT get_LimitTestFailed(VARIANT_BOOL* trueIfFailed)
Interface	IMeasurement

Write/Read

EndStimulus Property

Description	When constructing a limit line, specifies the stimulus value for the end of the segment.
VB Syntax	*limtseg*.**EndStimulus** = *value*
Variable	**(Type) - Description**
limtseg	A LimitSegment **(object)**
value	**(double)** - End Stimulus X-axis value. No units
Return Type	Double
Default	0

Examples

```
Set limtseg = meas.LimitTest(1)
limtseg.EndStimulus = 8e9 'Write
```

```
EndStim = limtseg.EndStimulus 'Read
```

C++ Syntax	HRESULT get_EndStimulus(double *pVal) HRESULT put_EndStimulus(double newVal)
Interface	ILimitSegment

About Limits

Write/Read

EndResponse Property

Description	When constructing a limit line, specifies the amplitude value at the end of the limit segment.
VB Syntax	*limtseg*.**EndResponse** = *value*
Variable	**(Type) - Description**
limts	A LimitSegment **(object)**
value	**(double)** - Y-axis value of the End Response limit. No units
Return Type	Double
Default	0

Examples

```
Set limtseg = meas.LimitTest(1)
limtseg.EndResponse = 10 'Write
```

```
EndResp = limtseg.EndResponse 'Read
```

C++ Syntax	HRESULT get_EndResponse(double *pVal) HRESULT put_EndResponse(double newVal)
Interface	ILimitSegment

Write/Read

Type (limit) Property

Description Specifies the Limit Line type.

VB Syntax *limt(index)*.**Type** = value

Variable **(Type) - Description**

limt A LimitSegment **(object)**

index **(variant)** - Limit line number in the LimitTest collection

value **(enum NALimitSegmentType)** - Limit Line type. Choose from:
0 - naLimitSegmentType_OFF - turns limit line OFF
1 - naLimitSegmentType_Maximum - limit line fails with a data point ABOVE the line
2 - naLimitSegmentType_Minimum - limit line fails with a data point BELOW the line

Return Type Long Integer

Default 0 - OFF

Examples

```
Set limts = meas.LimitTest
limts.Type = naLimitSegmentType_Maximum 'Write
```

```
limitType = limts.Type 'Read
```

C++ Syntax HRESULT put_Type(tagNALimitSegmentType *pVal)
HRESULT get_Type(tagNALimitSegmentType newVal)

Interface ILimitSegment

LineDisplay Property

Description	Turns the display of limit lines ON or OFF. To turn limit TESTING On and OFF, use State Property. **Note**: Trace data must be ON to view limit lines
VB Syntax	*limitst.***LineDisplay** = *state*
Variable	**(Type) - Description**
limitst	A LimitTest **(object)**
state	**(boolean)** **0** - Turns the display of limit lines OFF **1** - Turns the display of limit lines ON
Return Type	Long Integer
Default	**1** - ON
Examples	`Limttest.LineDisplay = 1 'Write` `lineDsp = Limttest.LineDisplay 'Read`
C++ Syntax	HRESULT get_LineDisplay(VARIANT_BOOL *pVal) HRESULT put_LineDisplay(VARIANT_BOOL newVal)
Interface	ILimitTest

Read/Write

LoadCharFromFile Property

Description Sets and returns whether a Mixer characterization file is to be loaded. Specify and load the filename with CharFileName Property

VB Syntax *VMC*.**LoadCharFromFile** = *bool*

Variable **(Type) - Description**

VMC VMCType (object)

bool (Boolean)

1 - Load from file

0 - Perform mixer characterization

Return Type Boolean

Default 0

Examples

```
value = VMC.LoadCharFromFile
```

C++ Syntax HRESULT put_LoadCharFromFile(VARIANT_BOOL bLoadCharFromFile);

HRESULT get_LoadCharFromFile(VARIANT_BOOL *bLoadCharFromFile);

Interface VMCType

Read-only

LoadPort Property

Description	Returns the load port number associated with an S-parameter reflection measurement. If the measurement is not a reflection S-parameter, the number returned by this property will have no meaning.
VB Syntax	*loadPort* = *meas.***LoadPort**
Variable	**(Type) - Description**
loadPort	**(long integer) -** The reflection measurement's load port number.
meas	A Measurement **(object)**
Return Type	Long Integer
Default	Not Applicable
Examples	`Set meas = pna.ActiveMeasurement` `loadPort = meas.LoadPort`
C++ Syntax	HRESULT get_LoadPort(long *pPortNumber);
Interface	IMeasurement

Write/Read

LocalLockoutState Property

Description	Prevents use of the mouse, keyboard, and front panel while your program is running. Use of these controls while this property is set TRUE causes an error message on the PNA display. To prevent these messages, see About Error Messages.
VB Syntax	*app*.**LocalLockoutState** = *bool*
Variable	**(Type) - Description**
app	An Application **(object)**
bool	**(boolean)** -Choose either: **0 (False) -** User Interface is NOT locked out. **1 (True) -** User Interface IS locked out.
Return Type	Boolean
Default	0 - False
Examples	`app.LocalLockoutState = True 'Write` `block = app.LocalLockoutState 'Read`
C++ Syntax	HRESULT get_LocalLockoutState(VARIANT_BOOL *State); HRESULT put_LocalLockoutState(VARIANT_BOOL *State);
Interface	IApplication4

LODenominator Property

Description	Sets or returns the denominator value of the LO Fractional Multiplier.
VB Syntax	*mixer*.**LODenominator** *(n)* **=** *value*
Variable	**(Type) - Description**
mixer	A Mixer **(object)**
value	**(Long)** - LO denominator value
n	**(Long)** - LO stage number Choose from 1 or 2
Return Type	Long
Default	1
Examples	`Print mixer.LODenominator(1) 'prints the value of the first LODenominator`
C++ Syntax	HRESULT get_LODenominator(long *pVal) HRESULT put_LODenominator(long newVal)
Interface	IMixer

LOFixedFrequency Property

Description	Sets or returns the LO frequency value
VB Syntax	*mixer*.**LOFixedFrequency** *(n)* = *value*
Variable	**(Type) - Description**
mixer	A Mixer **(object)**
value	**(double)** - LO Frequency in Hertz.
n	**(Long)** - LO stage number Choose from 1 or 2
Return Type	Double
Default	0 Hz
Examples	`Print mixer.LOFixedFrequency(1) 'prints the value of the first LO fixed frequency`
C++ Syntax	HRESULT get_LOFixedFrequency(double *pVal) HRESULT put_LOFixedFrequency(double newVal)
Interface	IMixer

LogMagnitudeOffset Property

Description Sets or returns the power offset value in dBm that the normalized unratioed power measurement data will be shifted by. The unratioed power measurement is effectively calibrated to the power level specified by the value of LogMagnitudeOffset as soon as the Normalization property is set to ON after the DataToDivisor method has been called.

VB Syntax *meas*.**LogMagnitudeOffset** = *value*

Variable **(Type) - Description**

meas **(object) -** A Measurement object

value **(double) -** Power offset in dBm. No limits are enforced on this value, but the PNA receivers themselves have maximum and minimum power specifications. This value must comply with those limits for a valid receiver power calibration

Return Type Double

Default 0

Examples

```
meas.LogMagOffset = -10 'Write (-10 dBm)
```

```
calpower = meas.LogMagOffset 'Read
meas.DataToDivisor 'Store meas data as measurement divisor
meas.Normalize = 1 'Measurement is now calibrated to –10 dBm
```

C++ Syntax HRESULT put_LogMagOffset(double newVal);

HRESULT get_LogMagOffset(double *pVal);

Interface IMeasurement

LOName Property

Description Sets or returns the name of the external LO source.

VB Syntax *mixer*.**LOName** *(n)* **=***value*

Variable **(Type) - Description**

mixer A Mixer **(object)**

n **(Long)** - LO stage number

Choose from 1 or 2

value **(string)** - External LO Source name. Must be a valid source name already configured on the PNA. The source must also be connected properly..

Return Type String

Default "Not controlled"

Examples

```
mixer.LOName(1) = "MySource"
```

C++ Syntax HRESULT get_LOName(string *pVal)

HRESULT put_LOName(string newVal)

Interface MIXer

LONumerator Property

Description Sets or returns the numerator value of the LO Fractional Multiplier.

VB Syntax *mixer*.**LONumerator** *(n)* = *value*

Variable **(Type) - Description**

mixer A Mixer **(object)**

value **(Long)** - LO denominator value

n **(Long)** - LO stage number

Choose from 1 or 2

Return Type Long

Default 1

Examples

```
Print mixer.LONumerator(1) 'prints the value of the
first LO Numerator
```

C++ Syntax HRESULT get_LONumerator(long *pVal)

HRESULT put_LONumerator(long newVal)

Interface IMixer

LOPower Property

Description Sets or returns the value of LO Power.

VB Syntax *mixer*.**LOPower** *(n)* **=** *value*

Variable **(Type) - Description**

mixer A Mixer **(object)**

n **(Long)** - LO stage number

Choose from 1 or 2

value **(double)** - LO Power in dBm.

Return Type Double

Default -10dBm

Examples

```
Print mixer.LOPower(1) 'prints the value of the LO Power
```

C++ Syntax HRESULT get_LOPower(double *pVal)

HRESULT put_LOPower(double newVal)

Interface IMixer

LORangeMode Property

Description	Sets or returns the LO sweep mode.
VB Syntax	*mixer*.**LORangeMode** *(n)* **=** *value*
Variable	**(Type) - Description**
mixer	A Mixer **(object)**
n	**(Long)** - LO stage number. Choose from 1 or 2
value	**(Enum as MixerRangeMode)** - LO sweep mode. Choose from: 0 - **mixSwept** 1 - **mixFixed**
Return Type	Enum
Default	0 - mixSwept
Examples	`mixer.LORangeMode(1)=mixSwept`
C++ Syntax	HRESULT get_LORangeMode(long *pVal) HRESULT put_LORangeMode(long newVal)
Interface	IMixer3

Loss Property

Description Sets and Returns the insertion loss for the calibration standard.

VB Syntax *calstd*.**loss** = *value*

Variable **(Type) - Description**

calstd A CalStandard **(object).** Use calKit.GetCalStandard to get a handle to the standard.

value **(single)** - Insertion loss in Mohms / sec. (MegaOhms per second of electrical delay)

Return Type Single

Default Not Applicable

Examples

```
calstd.loss = 3.5e9 'Write
```

```
stdLoss = calstd.loss 'Read the value of Loss
```

C++ Syntax HRESULT get_Loss(float *pVal)
HRESULT put_Loss(float newVal)

Interface ICalStandard

Loss (Source Power Cal) Property

Description Sets or returns the loss value associated with a PowerLossSegment.

VB Syntax *lossSeg*.**Loss** = *value*

Variable **(Type) - Description**

lossSeg **(object) -** PowerLossSegment

value **(double)** – Loss value in dB. This can be any value between 0 and 200.

Return Type Double

Default 0

Examples

```
lossSeg.Loss = 0.5 'Write
lossVal = lossSeg.Loss 'Read
```

C++ Syntax HRESULT put_Loss(Double newVal);

HRESULT get_Loss(Double *pVal);

Interface IPowerLossSegment

LOStage Property

Description	Sets or returns the number of LO stages present in the mixer.
VB Syntax	*mixer*.**LOStage =** *value*
Variable	**(Type) - Description**
mixer	A Mixer **(object)**
value	**(Long)** - Number of LO stages. Choose from **1** or **2**
Return Type	Long
Default	1
Examples	`mixer.LOStage = 1 'sets the LO Stage value to 1`
C++ Syntax	HRESULT get_LOStage(long *pVal) HRESULT put_LOStage(long newVal)
Interface	IMixer

LOStartFrequency Property

Description	Sets or returns the LO start frequency value. This command can only be used with SMC (not VMC) measurements.
VB Syntax	*mixer*.**LOStartFrequency** *(n)* = *value*
Variable	**(Type) - Description**
mixer	A Mixer **(object)**
value	**(double)** - LO Start Frequency in Hertz.
n	**(Long)** - LO stage number Choose from 1 or 2
Return Type	Double
Default	
Examples	`Print mixer.LOStartFrequency(1) 'prints the value of the first LO start frequency`
C++ Syntax	HRESULT get_LOStartFrequency(long id, double *pVal) HRESULT put_LOStartFrequency(long id,double newVal)
Interface	IMixer3

LOStopFrequency Property

Description	Sets or returns the LO stop frequency value. This command can only be used with SMC (not VMC) measurements.
VB Syntax	*mixer*.**LOStopFrequency** *(n)* = *value*
Variable	**(Type) - Description**
mixer	A Mixer **(object)**
value	**(double)** - LO Stop Frequency in Hertz.
n	**(Long)** - LO stage number Choose from 1 or 2
Return Type	Double
Default	
Examples	`Print mixer.LOStopFrequency(1) 'prints the value of the first LO stop frequency`
C++ Syntax	HRESULT get_LOStopFrequency(long id, double *pVal) HRESULT put_LOStopFrequency(long id,double newVal)
Interface	IMixer3

MagnitudeOffset Property

Description Offsets the data trace magnitude by the specified value.

To offset the data trace magnitude to a slope value that changes with frequency, use MagnitudeSlopeOffset Property.

To implement a Receiver Cal offset, use LogMagnitudeOffset property.

VB Syntax *meas*.**MagnitudeOffset** = *value*

Variable **(Type) - Description**

meas **(object) -** A Measurement object

value **(double) -** Offset value in dB.

Return Type Double

Default 0

Examples

```
meas.MagnitudeOffset = 4 'Write
```

```
offs = meas.MagnitudeOffset 'Read
```

C++ Syntax HRESULT put_MagnitudeOffset(double newVal);

HRESULT get_MagnitudeOffset(double *pVal);

Interface IMeasurement4

MagnitudeSlopeOffset Property

Description Offsets the data trace magnitude to a value that changes linearly with frequency. The offset slope begins at 0 Hz.

To offset the entire data trace magnitude by a specified value, use MagnitudeOffset Property.

VB Syntax *meas*.**MagnitudeSlopeOffset** = *value*

Variable **(Type) - Description**

meas **(object) -** A Measurement object

value **(double) -** Offset slope value in dB / 1GHz.

Return Type Double

Default 0

Examples

```
meas.MagnitudeSlopeOffset = 4 'Writes a slope offset of 4dB/1GHz.
```

```
offs = meas.MagnitudeSlopeOffset 'Read
```

C++ Syntax HRESULT put_MagnitudeSlopeOffset(double newVal);

HRESULT get_MagnitudeSlopeOffset(double *pVal);

Interface IMeasurement4

About Marker Format

Write/Read

MarkerFormat Property

Description Sets (or returns) the format of all the markers in the measurement. To override this setting for an individual marker, use **mark.Format**

VB Syntax *meas*.**MarkerFormat** = *value*

Variable **(Type) - Description**

meas A Measurement **(object)**

value **(enum NAMarkerFormat)** - Choose from:

0 - naMarkerFormat_**LinMag**
1 - naMarkerFormat_**LogMag**
2 - naMarkerFormat_**Phase**
3 - naMarkerFormat_**Delay**
4 - naMarkerFormat_**Real**
5 - naMarkerFormat_**Imaginary**
6 - naMarkerFormat_**SWR**
7 - naMarkerFormat_**LinMagPhase**
8 - naMarkerFormat_**LogMagPhase**
9 - naMarkerFormat_**RealImaginary**
10 - naMarkerFormat_**ComplexImpedance**
11 - naMarkerFormat_**ComplexAdmittance**

Return Type Long Integer

Default **1** - naMarkerFormat_LogMag

Examples

```
meas.MarkerFormat = naMarkerFormat_SWR 'Write
```

```
fmt = mark.Format 'Read
```

C++ Syntax HRESULT put_MarkerFormat(tagNAMarkerFormat NewFormat)

Interface IMeasurement

Write-only

InterpolateMarkers Method

Description Turns **All** Marker Interpolation ON and OFF for the measurement. Marker interpolation enables X-axis resolution between the discrete data values. The analyzer will calculate the x and y-axis data values between discrete data points. To override this property for individual markers, use the Interpolated property.

VB Syntax *meas*.**Interpolate** = *value*

Variable **(Type) - Description**

meas A Measurement **(object)**

value **(boolean)**
False - Turns interpolation OFF for all markers in the measurement
True - Turns interpolation ON for all markers in the measurement

Return Type Boolean

Default True (ON)

Examples

```
meas.Interpolate = 1
```

C++ Syntax HRESULT InterpolateMarkers(VARIANT_BOOL bNewVal)

Interface IMeasurement

Interpolated Property

Description Turns marker Interpolation ON and OFF. Marker interpolation enables X-axis resolution beyond the discrete data values. The analyzer will calculate the x and y-axis data values between discrete data points. Use meas.Interpolate to change interpolation of **all** markers in a measurement. This command will override the measurement setting.

VB Syntax *mark*.**Interpolated** = *value*

Variable **(Type) - Description**

mark A Marker **(object)**

value **(boolean)**
False - Turns interpolation OFF
True - Turns interpolation ON

Return Type Boolean

Default True (ON)

Examples

```
mark.Interpolated = 1 'Write
```

```
interpolate = mark.Interpolated 'Read
```

C++ Syntax HRESULT get_Interpolated(VARIANT_BOOL *pVal)
HRESULT put_Interpolated(VARIANT_BOOL newVal)

Interface IMarker

Number Property

Description	Returns the number of the marker.
VB Syntax	*marknum* = *mark*.**Number**
Variable	**(Type) - Description**
marknum	**(long)** - Variable to store marker number
mark	A Marker **(object)**
Return Type	Long Integer
Default	Not applicable
Examples	`marknum = mark.Number 'Read`
C++ Syntax	HRESULT get_Number(long *pVal)
Interface	IMarker

MarkerReadout Property

Description Enables or disables the readout of markers in the window. To show the marker on the screen use ShowMarkerReadout Method.

VB Syntax *win*.**MarkerReadout** = *state*

Variable **(Type) - Description**

win A NAWindow (object)

state **(boolean)**

True (1) - enables marker readout
False (0) - disables marker readout

Return Type Boolean

Default True

Examples

```
win.MarkerReadout = True 'Write
State = app.ActiveNAWindow.MarkerReadout 'Read
```

C++ Syntax HRESULT get_MarkerReadout(VARIANT_BOOL *pVal)
HRESULT put_MarkerReadout(VARIANT_BOOL newVal)

Interface INAWindow

Write/Read

MarkerReadoutSize Property

Description Specifies the size of font used when displaying Marker Readout in the selected window.

VB Syntax *win*.**MarkerReadoutSize** = *value*

Variable **(Type) - Description**

win A NAWindow (object)

value **(enum NAFontSize)**

0 - naDefault - marker readout appears in default font size

1 - naLarge - marker readout appears in large font size

Return Type Long Integer

Default naDefault

Examples

```
win.MarkerReadoutSize = naDefault 'write default size for marker readout
```

```
Dim Size As NAFontSize
Size = app.ActiveNAWindow.MarkerReadoutSize 'Read
```

C++ Syntax HRESULT get_MarkerReadoutSize(tagNAFontSize *pVal)

HRESULT put_MarkerReadoutSize(tagNAFontSize newVal)

Interface INAWindow

MarkerState Property

Description Sets or returns the ON / OFF state of the specified marker.

VB Syntax *meas*.**MarkerState** *(n)* = *state*

Variable **(Type) - Description**

meas A Measurement **(object)**

n (Long Integer) Marker number to turn on or off.

state (boolean) -
ON (1) turns the specified marker ON
OFF (0) turns the specified marker OFF

Return Type Boolean

Default 0 - OFF

Examples

```
meas.MarkerState(1) = True
```

```
reference = meas.MarkerState(2)
```

C++ Syntax HRESULT get_MarkerState(long markerNum, VARIANT_BOOL bState)
HRESULT put_MarkerState(long markerNum, VARIANT_BOOL* bState)

Interface IMeasurement3

Type (Marker) Property

Description Sets and reads the marker type.

VB Syntax *mark*.**Type** = *value*

Variable **(Type) - Description**

chan A Marker **(object)**

value **(enum NAMarkerType)** - Marker Type. Choose from:
0 - naMarkerType_Normal - the X-axis value for a normal marker will always be determined by the measurement data of the marker.

1 - naMarkerType_Fixed - retains and keeps its x-axis value at the time the marker type is set.

Return Type Long Integer

Default naMarkerType_Normal

Examples

```
mark.Type = naMarkerType_Normal 'Write
```

```
MrkType = mark.Type 'Read
```

C++ Syntax HRESULT get_Type(tagNAMarkerType *pVal)
HRESULT put_Type(tagNAMarkerType newVal)

Interface IMarker

Stimulus Property

Description Sets and reads the X-Axis value of the marker. If the marker is a delta marker, the value will be relative to the reference marker.

VB Syntax *mark*.**Stimulus** = *value*

Variable **(Type) - Description**

mark A Marker **(object)**

value **(double)** - X-Axis value. Choose any number within the full span of the channel or User Range (if set).

Return Type Double

Default First activated Marker turns ON in the middle of the X-axis range. Subsequent markers turn ON at the position of the most recently active marker.

Examples

```
mark.Stimulus = 3e9 'Write
```

```
XVal = mark.Stimulus 'Read
```

C++ Syntax HRESULT get_Stimulus(double *pVal)
HRESULT put_Stimulus(double newVal)

Interface IMarker

Read-only

Value Property

Description Reads the Y-Axis value of the marker. If the marker is a delta marker, the value will be relative to the reference marker.

You cannot set the Y-axis value of a marker. The marker remains at the position at the time you set marker.Type.

VB Syntax *YValue* = *mark*.**Value** *(format)*

Variable **(Type) - Description**

YValue A variable to store the Y-axis value

mark A Marker **(object)**

format **(enum NAMarkerFormat)** - The format you would like the marker's Y-axis value. The number in parenthesis following the format is the number of values that are returned in a variant array. Choose from:

0 - naMarkerFormat_**LinMag** (1)
1 - naMarkerFormat_**LogMag** (1)
2 - naMarkerFormat_**Phase** (1)
3 - naMarkerFormat_**Delay** (1)
4 - naMarkerFormat_**Real** (1)
5 - naMarkerFormat_**Imaginary** (1)
6 - naMarkerFormat_**SWR** (1)
7 - naMarkerFormat_**LinMagPhase** (2)
8 - naMarkerFormat_**LogMagPhase** (2)
9 - naMarkerFormat_**RealImaginary** (2)
10 - naMarkerFormat_**ComplexImpedance** (3)
11 - naMarkerFormat_**ComplexAdmittance** (3)

Return Type Variant - The previous list of formats indicates the number of values that are returned in a variant array

Default Not applicable

Examples

```
YVal = mark.Value 'Read
```

C++ Syntax HRESULT get_Value(tagNAMarkerFormat format, VARIANT *pVal)

Interface IMarker

MaximumFrequency Property

Description Sets and Returns the maximum frequency for the calibration standard.

VB Syntax *calstd*.**MaximumFrequency** = *value*

Variable **(Type) - Description**

calstd A CalStandard **(object).** Use calKit.GetCalStandard to get a handle to the standard.

value **(double)** - Maximum frequency in Hertz.

Return Type Double

Default Not Applicable

Examples

```
calstd.MaximumFrequency = 9e9 'Write
```

```
maxFrequency = calstd.MaximumFrequency 'Read
```

C++ Syntax HRESULT get_MaximumFrequency(double *pVal)
HRESULT put_MaximumFrequency(double newVal)

Interface ICalStandard

MaximumFrequency Property

Description	Returns the maximum frequency of the remote PNA.
VB Syntax	*value* **=** *cap*.**MaximumFrequency**
Variable	**(Type) - Description**
value	(Double) - Variable to store the returned maximum frequency of the PNA.
cap	A Capabilities **(object)**
Return Type	Double
Default	Not Applicable
Examples	`value = cap.MaximumFrequency 'Read`
C++ Syntax	HRESULT get_MaximumFrequency(&frequencyMax);
Interface	**I**Capabilities

MaximumFrequency (Source Power Cal) Property

Description	Maximum usable frequency specified for the power sensor.
VB Syntax	*pwrSensor*.**MaximumFrequency** = *value*
Variable	**(Type) - Description**
pwrSensor	**(object) -** A PowerSensor (object)
value	**(double) -**Frequency in Hertz.
Return Type	Double
Default	0

Examples

```
Set powerCalibrator = pna.SourcePowerCalibrator
powerCalibrator.PowerSensors(1).MaximumFrequency = 6e9
'Write

MaxFreq = powerCalibrator.PowerSensors(1).MaximumFrequency
'Read
```

C++ Syntax

HRESULT put_MaximumFrequency(double newVal);

HRESULT get_MaximumFrequency(double *pVal);

Interface IPowerSensor

MaximumIFFilterSampleCount Property

Description Returns the maximum allowed value for the IFFilterSampleCount property for the queried PNA.

Note: An error will occur if this command is used on a PNA without option H08 installed.

VB Syntax *IfConfig*.**MaximumIFFilterSampleCount** = *value*

Variable **(Type) - Description**

IfConfig IFConfiguration (object)

value **(long)** The maximum allowed value that can be applied to the IFFilterSampleCount property.

Return Type Long Integer

Default PNA Model number dependent

Examples

```
variable =
App.ActiveChannel.IFConfiguration.MaximumIFFilterSampleCount
'Read
```

See an example program

C++ Syntax HRESULT get_MaximumIFFilterSampleCount(long * pMaxSampleCount);

Interface IIFConfiguration2

About Source Power Cal

Write / Read

MaximumIterationsPerPoint Property

Description Sets the maximum number of readings to take at each data point for iterating the source power. Power readings will continue to made, and source power adjusted, until a reading is within the IterationsTolerance value or this max number of readings has been met. The last value to be read is the valid reading for that data point.

VB Syntax *pwrCal*.**MaximumIterationsPerPoint** = *value*

Variable **(Type) - Description**

pwrCal **(object) -** A SourcePowerCalibrator (object)

value **(Long)** – Maximum number of readings. Choose any number between 1 and 100.

Return Type Long Integer

Default 5

Examples

```
Set powerCalibrator = pna.SourcePowerCalibrator
powerCalibrator.MaximumIterationsPerPoint = 5 'Write
MaxReads = powerCalibrator.MaximumIterationsPerPoint 'Read
```

C++ Syntax HRESULT get_MaximumIterationsPerPoint(long *pVal);

HRESULT put_MaximumIterationsPerPoint(long newVal);

Interface ISourcePowerCalibrator3

MaximumNumberOfPoints Property

Description	Returns the maximum possible number of data points.
VB Syntax	*value* **=** *cap*.**MaximumNumberOfPoints**
Variable	**(Type) - Description**
value	**(Long)** - Variable to store the returned maximum value for number of points.
cap	A Capabilities **(object)**
Return Type	Long
Default	Not Applicable
Examples	`value = cap.MaximumNumberOfPoints 'Read`
C++ Syntax	HRESULT get_MaximumNumberOfPoints(double * maximumNumberOfPoints);
Interface	ICapabilities

MaximumReceiverStepAttenuator Property

Description	Returns the maximum amount of receiver attenuation.
VB Syntax	*value* **=** *cap*.**MaximumReceiverStepAttenuator** *(n)*
Variable	**(Type) - Description**
value	**(Double)** - Variable to store the returned value of maximum receiver attenuation.
cap	A Capabilities **(object)**
n	**(Long) -** port number to query for step attenuators
Return Type	Double
Default	Not Applicable
Examples	`value = cap.MaximumReceiverStepAttenuator'Read`
C++ Syntax	HRESULT get_MaximumReceiverStepAttenuator(long portNumber, double * attenuation);
Interface	ICapabilities

MaximumSourceALCPower Property

Description	Returns a value indicating the maximum amount of source ALC power.
VB Syntax	*value* **=** *cap*.**MaximumSourceALCPower** *(n)*
Variable	**(Type) - Description**
value	**(Double)** - Variable to store the returned value for the maximum amount of source ALC power.
cap	A Capabilities **(object)**
n	**(Long) -** source number to query for maximum ALC power
Return Type	Double
Default	Not Applicable
Examples	`value = cap.MaximumSourceALCPower'Read`
C++ Syntax	HRESULT get_MaximumSourceALCPower(long sourceNum, double * power);
Interface	ICapabilities

MaximumSourceStepAttenuator Property

Description	Returns a value for the maximum amount of source attenuation.
VB Syntax	*value* **=** *cap*.**MaximumSourceStepAttenuator** *(n)*
Variable	**(Type) - Description**
value	**(Double)** - Variable to store the returned value for the maximum amount of source attenuation.
cap	A Capabilities **(object)**
n	**(Long) -** port number to query for the maximum amount of source attenuation
Return Type	Double
Default	Not Applicable
Examples	`value = cap.MaximumSourceStepAttenuator'Read`
C++ Syntax	HRESULT get_MaximumSourceStepAttenuator(long portNumber, double * attenuation);
Interface	ICapabilities

Read-only

Mean Property

Description	Returns the mean value of the measurement . To retrieve all 3 statistics value at the same time, use meas.GetTraceStatistics
VB Syntax	*average* = *meas*.**Mean**
Variable	**(Type) - Description**
average	**(single) -** Variable to store mean value
meas	A Measurement **(object)**
Return Type	Single
Default	Not applicable
Examples	`Dim average as Single` `average = meas.Mean 'Read`
C++ Syntax	HRESULT get_Mean(float* mean)
Interface	IMeasurement

Medium Property

Description	Sets and Returns the media type of the calibration standard.
VB Syntax	*calstd*.**Medium** = *value*
Variable	**(Type) - Description**
calstd	A CalStandard **(object).** Use calKit.GetCalStandard to get a handle to the standard.
value	**(enum** NACalStandardMedium**) -** Medium of the transmission line of the standard. Choose from: **0 - naCoax** - Coaxial Cable **1 - naWaveGuide**
Return Type	Long Integer
Default	Not Applicable

Examples	`calstd.Medium = naCoax 'Write`
	
	`stdMedium = calstd.Medium 'Read`

C++ Syntax	HRESULT get_Medium(tagNACalStandardMedium *pVal) HRESULT put_Medium(tagNACalStandardMedium newVal)
Interface	ICalStandard

Write/Read

MinimumFrequency Property

Description Sets and Returns the minimum frequency for the calibration standard.

VB Syntax *calstd*.**MinimumFrequency** = *value*

Variable **(Type) - Description**

calstd A CalStandard **(object).** Use calKit.GetCalStandard to get a handle to the standard.

value **(double)** -Minimum frequency in Hertz.

Return Type Double

Default Not Applicable

Examples

```
calstd.MinimumFrequency = 300e3 'Write
```

```
minFrequency = calstd.MinimumFrequency 'Read
```

C++ Syntax HRESULT get_MinimumFrequency(double *pVal)
HRESULT put_MinimumFrequency(double newVal)

Interface ICalStandard

MinimumFrequency Property

Description	Returns the minimum frequency of the remote PNA.
VB Syntax	*value* **=** *cap*.**MinimumFrequency**
Variable	**(Type) - Description**
value	(Double) - Variable to store the returned minimum frequency of the PNA.
cap	A Capabilities **(object)**
Return Type	Double
Default	Not Applicable
Examples	`value = cap.MinimumFrequency 'Read`
C++ Syntax	HRESULT get_MinimumFrequency(double *pVal)
Interface	ICapabilities

MinimumFrequency (Source Power Cal) Property

Description Minimum usable frequency specified for the power sensor.

VB Syntax *pwrSensor*.**MinimumFrequency** = *value*

Variable **(Type) - Description**

pwrSensor **(object) -** A PowerSensor (object)

value **(double) -**Frequency in Hertz.

Return Type Double

Default 0

Examples

```
Set powerCalibrator = pna.SourcePowerCalibrator
powerCalibrator.PowerSensors(1).MinimumFrequency = 300e3
'Write

MinFreq = powerCalibrator.PowerSensors(1).MinimumFrequency
'Read
```

C++ Syntax HRESULT put_MinimumFrequency(double newVal);

HRESULT get_MinimumFrequency(double *pVal);

Interface IPowerSensor

MinimumIFFilterSampleCount Property

Description Returns the Minimum allowed value for the IFFilterSampleCount property for the queried PNA.

Note: An error will occur if this command is used on a PNA without option H08 installed.

VB Syntax *IfConfig*.**MinimumIFFilterSampleCount** = *value*

Variable **(Type) - Description**

IfConfig IFConfiguration (object)

value **(long)** The minimum allowed value that can be applied to the IFFilterSampleCount property.

Return Type Long Integer

Default PNA Model number dependent

Examples

```
variable =
App.ActiveChannel.IFConfiguration.MinimumIFFilterSampleCount
'Read
```

See an example program

C++ Syntax HRESULT get_MinimumIFFilterSampleCount(long * pMinSampleCount);

Interface IIFConfiguration2

MinimumNumberOfPoints Property

Description	Returns the minimum possible number of data points for a data trace.
VB Syntax	*value* **=** *cap*.**MinimumNumberOfPoints**
Variable	**(Type) - Description**
value	(Long) - Variable to store the returned minimum value for number of points.
cap	A Capabilities **(object)**
Return Type	Long
Default	Not Applicable
Examples	`value = cap.MinimumNumberOfPoints 'Read`
C++ Syntax	HRESULT get_MinimumNumberOfPoints(double * minimumNumberOfPoints);
Interface	ICapabilities

MinimumReceiverStepAttenuator Property

Description	Returns a value indicating the minimum amount of receiver attenuation.
VB Syntax	*value* **=** *cap*.**MinimumReceiverStepAttenuator** *(n)*
Variable	**(Type) - Description**
value	**(Double)** - Variable to store the returned minimum value of receiver attenuation.
cap	A Capabilities **(object)**
n	**(Long) -** port number to query for minimum value of receiver attenuation
Return Type	Double
Default	Not Applicable
Examples	`value = cap.MinimumReceiverStepAttenuator'Read`
C++ Syntax	HRESULTget_ MinimumReceiverStepAttenuator(long portNumber, double * attenuation);
Interface	ICapabilities

MinimumSourceALCPower Property

Description	Returns a value indicating the minimum amount of source ALC power.
VB Syntax	*value* **=** *cap*.**MinimumSourceALCPower** *(n)*
Variable	**(Type) - Description**
value	**(Double)** - Variable to store the returned minimum value of source ALC power.
cap	A Capabilities **(object)**
n	**(Long) -** source number to query for the minimum value of source ALC power
Return Type	Double
Default	Not Applicable
Examples	`value = cap.MinimumSourceALCPower'Read`
C++ Syntax	HRESULT get_MinimumSourceALCPower(long sourceNum, double * power);
Interface	ICapabilities

Write/Read

Name (Measurement) Property

Description Sets (or returns) the Name of the measurement. Measurement names must be unique among the set of measurements. Measurement names cannot be an empty string.

Note: This is the same name as trace.Name; when one changes, the other changes.

VB Syntax *meas*.**Name** = *value*

Variable **(Type) - Description**

meas A Measurement **(object)**

value **(string)** - A user defined name of the measurement

Return Type String

Default "CH1_S11_1" - name of the default measurement

Examples

```
meas.Name = "Filter BPass" 'Write
```

```
MName = meas.Name 'Read
```

C++ Syntax HRESULT get_Name(BSTR *pVal)
HRESULT put_Name(BSTR newVal)

Interface IMeasurement

Write/Read

Name (trace) Property

Description	Sets or returns the name of the Trace. Use the trace name to identify the trace and refer to the trace in the collection. **Note:** This is the same name as meas.Name; when one changes, the other changes.
VB Syntax	*trac*.**Name** = *value*
Variable	**(Type) - Description**
trac	A Trace **(object)**
value	**(String)** Trace name
Return Type	String
Default	"CH1_S11_1" - name of the default measurement
Examples	`trace.Name = "myTrace" 'Write` `traceName = Name.Trace 'Read`
C++ Syntax	HRESULT put_Name(BSTR name) HRESULT get_Name(BSTR *name)
Interface	ITrace

Write/Read

Name (CalKit) Property

Description Sets and Returns a name for the selected calibration kit.

VB Syntax *calKit*.**Name** = *value*

Variable **(Type) - Description**

calKit A CalKit **(object).**

value **(string)** -Calibration Kit name. Any string name, can include numerics, period, and spaces; any length (although the dialog box display is limited to about 30 characters).

Return Type String

Default Not Applicable

Examples

```
calKit.Name = "MyCalKit" 'Write
```

```
KitName = calKit.Name 'Read
```

C++ Syntax HRESULT get_Name(BSTR *pVal)
HRESULT put_Name(BSTR newVal)

Interface ICalKit

NominalIncidentPowerState Property

Description Toggles the Nominal Incident Power setting ON and OFF. This setting is ONLY to be used with SMC measurements, not VMC. Learn more about Nominal Incident Power.

VB Syntax *mixer*.**NominalIncidentPowerState =** *bool*

Variable **(Type) - Description**

mixer A Mixer **(object)**

bool **(boolean)** - Nominal Incident Power State. Choose from:

1 -(True) Turn nominal incident power ON

0 -(False) Turn nominal incident power OFF

Return Type Boolean

Default 0 -(False)

Examples

```
mixer.NominalIncidentPowerState = True 'sets
NominalIncidentPowerState to ON
```

C++ Syntax HRESULT get_NominalIncidentPowerState(VARIANT_BOOL *pVal)

HRESULT put_NominalIncidentPowerState(VARIANT_BOOL val);

Interface IMixer4

Normalization Property

Description Sets or returns normalization ON or OFF for the measurement. Normalization is currently supported only on measurements of unratioed power for the purpose of performing a receiver power calibration. If this property is set to ON for a ratioed measurement (such as S-parameter), it will return an error. This property will also return an error when set to ON if the divisor buffer doesn't yet exist.

VB Syntax *meas*.**Normalization** = *value*

Variable **(Type) - Description**

meas **(object) -** A Measurement object

value **(boolean)**

0 – Turns normalization OFF

1 – Turns normalization ON

Return Type Boolean

Default 0 -OFF

Examples

```
meas.Normalization = 1 'Write
```

```
normalized = meas.Normalization 'Read
```

C++ Syntax HRESULT put_Normalization(VARIANT_BOOL bState);

HRESULT get_Normalization(VARIANT_BOOL *bState);

Interface IMeasurement

Read-only

Number (Measurement) Property

Description Returns the Number of the measurement. Measurement numbers are assigned internally.

Note: Measurement numbers are NOT the same as their number in the Measurements collection. Measurement number is used to identify the measurement associated with an event.

This property is used to identify measurements when events occur through the OnMeasurementEvent callback. For example:

```
OnMeasurementEvent (naEventId_MSG_LIMIT_FAILED, 3)
```

VB Syntax *measNum* = *meas*.**Number**

Variable **(Type) - Description**

measNum **(long) -** variable to store the measurement number

meas A Measurement **(object)**

Return Type Long Integer

Default "1" - number of the default measurement

Examples

```
measNum = meas.Number
```

C++ Syntax HRESULT get_Number(long *MeasurementNumber)

Interface IMeasurement

About Number of Points

Write/Read

NumberOfPoints Property

Description Sets or returns the Number of Points of the channel.
Sets or returns the Number of Points of the segment.

see also Measurement2 interface

VB Syntax *object*.**NumberOfPoints** = *value*

Variable **(Type) - Description**

object Channel **(object)**

or

CalSet **(object) -** Read-only property

value **(long)** - Number of Points.

For channel, choose any number from **1** to **16001.**

For segment, the total number of points in all segments cannot exceed **16001.** A segment can have as few as 1 point.

Return Type Long Integer

Default 201 for channel

21 for segment

Examples

```
chan.NumberOfPoints = 201 'sets the number of points for all
measurements in the channel. -Write
```

```
numofpts = chan.NumberOfPoints 'Read
```

C++ Syntax HRESULT get_NumberOfPoints(long *pVal)
HRESULT put_NumberOfPoints(long newVal)

Interface IChannel
ISegment

|CalSet3

Read-only

NumberOfPoints Property

Description	Returns the number of data points of the measurement. To understand how this property is useful, see IMeasurement2 Interface.
VB Syntax	*value* = *meas*.**NumberOfPoints**
Variable	**(Type) - Description**
value	(**Long)** - variable to store the returned value
meas	A Measurement **(object)**
Return Type	Long Integer
Default	Not Applicable
Examples	`Print meas.NumberOfPoints 'prints the number of data points`
C++ Syntax	HRESULT get_NumberOfPoints(long *pVal);
Interface	IMeasurement2

Read-only

NumberOfPorts Property

Description	Returns the number of hardware source ports on the PNA.
VB Syntax	*value* = *app.***NumberOfPorts**
Variable	**(Type) - Description**
app	An Application **(object)**
value	**(long integer)** - variable to contain the returned value
Return Type	(long integer)
Default	Not Applicable
Examples	`iNumPorts = app.NumberOfPorts`
C++ Syntax	HRESULT NumberOfPorts(long* NumPorts)
Interface	IApplication

Read/Write

OmitIsolation Property

Description Sets and returns whether Isolation portion of the calibration will be performed or not.

VB Syntax *obj*.**OmitIsolation** = *bool*

Variable **(Type) - Description**

obj SMCType (object)

or

VMCType (object)

bool (Boolean)

1 - Isolation is NOT performed

0 - Isolation is performed

Return Type Boolean

Default 1

Examples

```
value = SMC.OmitIsolation
```

C++ Syntax HRESULT put_OmitIsolation (VARIANT_BOOL bState)
HRESULT get_OmitIsolation (VARIANT_BOOL *bState)

Interface SMCType

VMCType

About Marker Readout

Write/Read

OneReadoutPerTrace Property

Description Either show marker readout of only the active trace or all of the traces simultaneously.

VB Syntax *win*.**OneReadoutPerTrace** = *state*

Variable **(Type) - Description**

win A NAWindow (object)

value **(boolean)**

True (1) - show a single marker per trace
False (0) **-** show up to 4 markers per active trace

Return Type Boolean

Default False (0)

Examples

```
win.OneReadoutPerTrace = True 'Write
```

```
State = app.ActiveNAWindow.OneReadoutPerTraceBegResp
'Read
```

C++ Syntax HRESULT get_OneReadoutPerTrace(VARIANT_BOOL *pVal)
HRESULT put_OneReadoutPerTrace(VARIANT_BOOL newVal)

Interface INAWindow

Read-only

About Options

Options Property

Description	Returns a string identifying the analyzer option configuration.
VB Syntax	*value* = *app.***Options**
Variable	**(Type) - Description**
app	An Application **(object)**
value	**(string)** - variable to contain the returned string
Return Type	String
Default	Not Applicable
Examples	`availOptions = app.Options`
C++ Syntax	HRESULT Options(BSTR* OptionString)
Interface	IApplication

Write/Read

OrientECALModule Property

Description Specifies if the PNA should perform orientation of the ECal module during calibration. Orientation is a technique by which the PNA automatically determines which ports of the module are connected to which ports of the PNA. Orientation begins to fail at very low power levels or if there is much attenuation in the path between the PNA and the ECal module.

VB Syntax cal.OrientECALModule = value

Variable **(Type) - Description**

cal A Calibrator **(object)**

value **(boolean)**

False (0) – DoECAL1Port and DoECAL2Port methods will use the value of the ECALPortMap property to determine the port connections.

True (1) - DoECAL1Port and DoECAL2Port methods will use Orientation technique to determine port connections.

Return Type Boolean

Default True (1)

Examples

```
Dim cal As Calibrator
Dim bOrient As Boolean
Set cal = PNAapp.ActiveChannel.Calibrator
cal.OrientECALModule = False 'Write
bOrient = cal.OrientECALModule 'Read
```

C++ Syntax

```
HRESULT put_OrientECALModule(VARIANT_BOOL bOrient);
HRESULT get_OrientECALModule(VARIANT_BOOL *bOrient);
```

Interface ICalibrator3

OutputFixedFrequency Property

Description Sets or returns the mixer output fixed frequency value. This command can only be used with SMC (not VMC) measurements.

VB Syntax *mixer*.**OutputFixedFrequency** = *value*

Variable **(Type) - Description**

mixer A Mixer **(object)**

value **(double)** - Output Fixed Frequency in Hertz.

Return Type Double

Default Not Applicable

Examples

```
Print mixer.OutputFixedFrequency 'prints the output fixed
frequency value of the mixer.
```

C++ Syntax HRESULT get_OutputFixedFrequency(double *pVal)

HRESULT put_OutputFixedFrequency(double newVal)

Interface IMixer3

OutputSideband Property

Description Sets or returns the value of the output sideband, high or low.

VB Syntax *mixer*.**OutputSideband =** *value*

Variable **(Type) - Description**

mixer A Mixer **(object)**

value **(FCASideBand) -**

Choose from:

0 - LOW

1 - HIGH

Return Type FCASideBand

Default LOW

Examples

```
Print mixer.OutputSideband 'prints the value of the
OutputSideband
```

C++ Syntax HRESULT get_OutputSideband(FCASideBand *pVal)

HRESULT put_OutputSideband(FCASideBand newVal)

Interface IMixer

OutputStartFrequency Property

Description	Sets or returns the mixer output start frequency.
VB Syntax	*mixer*.**OutputStartFrequency =** *value*
Variable	**(Type) - Description**
mixer	A Mixer **(object)**
value	**(double)** - Output Start Frequency in Hertz.
Return Type	Double
Default	Not Applicable
Examples	`Print mixer.OutputStartFrequency 'prints the value of the OutputStartFrequency`
C++ Syntax	HRESULT get_OutputStartFrequency(double *pVal) HRESULT put_OutputStartFrequency(double newVal)
Interface	IMixer

OutputStopFrequency Property

Description	Sets or returns the mixer Output Stop frequency.
VB Syntax	*mixer*.**OutputStopFrequency =** *value*
Variable	**(Type) - Description**
mixer	A Mixer **(object)**
value	**(double)** - Output stop frequency in Hertz.
Return Type	Double
Default	Not Applicable
Examples	`Print mixer.OutputStopFrequency 'prints the value of the OutputStopFrequency`
C++ Syntax	HRESULT get_OutputStopFrequency(double *pVal) HRESULT put_OutputStopFrequency(double newVal)
Interface	IMixer

Read-only

Parameter Property

Description	Returns the measurement Parameter. To change the parameter, use meas.ChangeParameter
VB Syntax	*measPar* = *meas*.**Parameter**
Variable	**(Type) - Description**
measPar	**(string)** - Variable to store Parameter string
meas	A Measurement **(object)**
Return Type	String
Default	Not applicable
Examples	`measPar = meas.Parameter 'Read`
C++ Syntax	HRESULT get_Parameter(BSTR *pVal)
Interface	IMeasurement

Read-only

Parent Property

Description Returns a handle to the parent object of the collection object being refered to in the statement. The parent property allows the user to traverse from an object back up the object hierarchy.

VB Syntax *object.***Parent**

Variable **(Type) - Description**

object Channels collection
Channel object
Measurements collection
NAWindows collection
Traces collection
Segments collection
PowerSensors collection
CalFactorSegments collection
PowerLossSegments collection

Return Type Object

Default Not Applicable

Examples

```
parentobj = chans.Parent 'returns a handle to the parent object
(Application) of the chans collection. -Read
```

C++ Syntax HRESULT get_Parent(IApplication* *pApplication); //IChannels, IChannel, IMeasurements and INAWindows

HRESULT get_Parent(IChannel* *pChannel); //ITraces

HRESULT get_Parent(INAWindow* *pWindow); //ISegments

HRESULT get_Parent(IPowerSensor* *pSensor); //ICalFactorSegments

HRESULT get_Parent(ISourcePowerCalibrator* *pCalibrator); //IPowerSensors and IPowerLossSegments

Interface IChannels

IChannel

IMeasurements

INAWindows

ITraces

ISegments

IPowerSensors

ICalFactorSegments

IPowerLossSegments

Read/Write

PassFailLogic Property

Description Sets and reads the logic of the PassFail line on the HANDLER IO connector (pin 33) and AUX IO connector (pin 12). Learn more about Global Pass/Fail.

Note: This line is connected to both the Handler IO and Aux IO in the PNA. Therefore, this command will affect both of these connectors in the same way.

VB Syntax *object*.**PassFailLogic** = *value*

Variable	**(Type) - Description**
object	**(object)** - An Aux I/O or Handler I/O object
value	**(enum as NARearPanelIOLogic)** Choose from: **0 - naPositiveLogic -** Causes the PassFail line to have positive logic (high = pass, low = fail). **1 - naNegativeLogic** - Causes the PassFail line to have negative logic (high = fail, low = pass).

Return Type Long Integer

Default naPositiveLogic

Examples

```
aux.PassFailLogic = naNegativeLogic 'Write
Text1.Text = aux.PassFailLogic 'Read
```

C++ Syntax HRESULT put_PassFailLogic (tagNARearPanelIOLogic Mode);

HRESULT get_PassFailLogic (tagNARearPanelIOLogic* Mode);

Interface IHWAuxIO

IHWMaterialHandlerIO

Read/Write

PassFailMode Property

Description	Sets and reads the mode of the PassFail line on the HANDLER IO connector (pin 33) and AUX IO connector (pin 12). Learn more about Global Pass/Fail. **Note**: This line is connected to both the Handler IO and Aux IO in the PNA. Therefore, this command will affect both of these connectors in the same way.
VB Syntax	*object*.**PassFailMode** = *value*
Variable	**(Type) - Description**
object	**(object) -** An Aux I/O or Handler I/O object
value	**(enum as NAPassFailMode)**.Choose from: **0 - naDefaultPassNoWaitMode**- the line stays in PASS state. When a device fails, then the line goes to fail IMMEDIATELY. **1 - naDefaultPassWaitMode** - the line stays in PASS state. When a device fails, then the line goes to fail after the Sweep End line is asserted. **2 - naDefaultFailWaitMode**- the line stays in FAIL state. When a device passes, then the line goes to PASS state after the Sweep End line is asserted.
Return Type	Long Integer
Default	0 - naDefaultPassNoWaitMode
Examples	`HWAuxIO.PassFailMode = naDefaultPassNoWaitMode 'Write` `mode = HWAuxIO.PassFailMode 'Read`
C++ Syntax	HRESULT put_PassFailMode (tagNAPassFailMode Mode); HRESULT get_PassFailMode (tagNAPassFailMode* Mode);
Interface	IHWAuxIO IHWMaterialHandlerIO

PassFailPolicy Property

Description	Sets the policy used to determine how global pass/fail is computed. Learn more about Global Pass/Fail.
VB Syntax	*object*.**PassFailPolicy** = *value*
Variable	**(Type) - Description**
object	**(object)** - An Aux I/O or Handler I/O object
value	**(enum as NAPassFailPolicy)** Choose from: 0 - **naPolicyAllTests - -** Pass/Fail Status returns PASS if all tests on all measurements pass. 1 - **naPolicyAllMeas** - Pass/Fail Status returns PASS if all measurements have associated tests, and all tests pass. FAIL is returned if even one measurement has no associated limit test. Only those measurements which are not in HOLD mode contribute to the pass/fail result.
Return Type	Long Integer
Default	naPolicyAllTests
Examples	`matHndler.PassFailPolicy = naPolicyAllTests 'Write` `policy = aux.PassFailPolicy 'Read`
C++ Syntax	HRESULT put_PassFailPolicy (tagNAPassFailPolicy Policy); HRESULT get_PassFailPolicy (tagNARearPanelIOLogic* Policy);
Interface	IHWAuxIO4 IHWMaterialHandlerIO2

PassFailScope Property

Description Sets and reads the Scope of the PassFail line on the HANDLER IO connector (pin 33) and AUX IO connector (pin 12). Learn more about Global Pass/Fail.

Note: The PassFail line is connected to both the Handler IO and Aux IO in the PNA. Therefore, this command will affect both of these connectors in the same way.

VB Syntax *object*.**PassFailScope** = *value*

Variable **(Type) - Description**

object **(object)** - An Aux I/O or Handler IO object

value **(enum NAPassFailScope)** Choose from:

0 - naChannelScope - The PassFail line returns to its default state before sweeps on the next channel start. (A channel measurement may require several sweeps.)

1 - naGlobalScope - The PassFail line returns to its default state before the sweeps for the next **triggerable** channel start.

The default state of the PassFail line before a measurement occurs and after a failure occurs is set by the PassFailMode property.

Return Type enum NAPassFailScope

Default 1 - naGlobalScope

Examples

```
HWAuxIO.PassFailScope = naGlobalScope 'Write
scope = HWAuxIO.PassFailScope 'Read
```

C++ Syntax HRESULT put_PassFailScope (tagNAPassFailScope Scope);

HRESULT get_PassFailScope (tagNAPassFailScope* Scope);

Interface IHWAuxIO

IHWMaterialHandlerIO

Read-Only

PassFailStatus Property

Description Returns the most recent pass/fail status value. Use this command as follows:

1. Set the PNA trigger scope to GLOBAL
2. Set the PNA trigger source to MANUAL or EXTERNAL.
3. Configure and enable Limit Testing
4. Trigger the PNA.
5. Use the *OPC? (with SCPIStringParser object) to determine when the sweep is complete.
6. Use the **PassFailStatus** property to obtain the global pass/fail result.

Learn more about Global Pass/Fail.

VB Syntax *var* = *object*.**PassFailStatus**

Variable **(Type) - Description**

var (enum as NAPassFailStatus) Variable to store returned status. One of the following will be returned:

0 - naStatusFail - all measurements not in HOLD mode have been swept, and one or more limit tests failed according to the specified Pass/Fail policy.

1 - naStatusPass - all measurements not in HOLD mode have been swept, and all associated limit tests have passed.

2 - naStatusNone - status cannot be determined because measurements are in progress.

object **(object)** - An Aux I/O or Handler I/O object

Return Type Long Integer

Default Not Applicable

Examples

```
status = aux.PassFailStatus 'Read
```

C++ Syntax HRESULT get_PassFailPolicy (tagNAPassFailStatus* status);

Interface IHWAuxIO4

IHWMaterialHandlerIO2

Write/Read

PeakExcursion Property

Description	Sets and reads the peak excursion value for the specified marker. The Excursion value determines what is considered a "peak".
VB Syntax	*mark*.**PeakExcursion** = *value*
Variable	**(Type) - Description**
mark	A Marker **(object)**
value	**(single)** - Peak Excursion. Choose any number between **-500** and **500**
Return Type	Single
Default	3
Examples	`mark.PeakExcursion = 1 'Write` `PkExcur = mark.PeakExcursion 'Read`
C++ Syntax	HRESULT get_PeakExcursion(float *pVal) HRESULT put_PeakExcursion(float newVal)
Interface	IMarker

About Marker Search

Write/Read

PeakThreshold Property

Description Sets peak threshold for the specified marker. If a peak (using the criteria set with PeakExcursion) is below this reference value, it will not be considered when searching for peaks.

VB Syntax *mark*.**PeakThreshold** = *value*

Variable **(Type) - Description**

mark A Marker **(object)**

value **(single) -** Peak Threshold. Choose any number between:
-500 and **500**

Return Type Single

Default -100db

Examples

```
mark.PeakThreshold = 1 'Write
```

```
PkThresh = mark.PeakThreshold 'Read
```

C++ Syntax HRESULT get_PeakThreshold(float *pVal)
HRESULT put_PeakThreshold(float newVal)

Interface IMarker

PeakToPeak Property

Description	Returns the Peak to Peak value of the measurement.To retreive all 3 statistics value at the same time, use meas.GetTraceStatistics
VB Syntax	*pp* = *meas*.**PeakToPeak**
Variable	**(Type) - Description**
pp	**(single)** - Variable to store peak-to-peak value
meas	A Measurement **(object)**
Return Type	Single
Default	Not applicable
Examples	`pp = meas.PeakToPeak 'Read`
C++ Syntax	HRESULT get_PeakToPeak(float* pp)
Interface	IMeasurement

Write/Read

PhaseOffset Property

Description Sets the Phase Offset for the active channel.

VB Syntax *meas*.**PhaseOffset** = *value*

Variable **(Type) - Description**

meas A Measurement **(object)**

value **(double)** - PhaseOffset in degrees. Choose any number between: **-360** and **+360**

Return Type Double

Default 0

Examples

```
meas.PhaseOffset = 25 'Write

poffset = meas.PhaseOffset 'Read
```

C++ Syntax

HRESULT get_PhaseOffset(double *pVal)
HRESULT put_PhaseOffset(double newVal)

Interface IMeasurement

About Port Extensions

Write/Read

Port1 Property

Description	Sets a Port Extension value for Port 1
VB Syntax	*portExt*.**Port1** = *value*
Variable	**(Type) - Description**
portExt	A Port Extension **(object)**
value	**(double)** - Port Extension value in seconds. Choose any number between **-10** and **10**
Return Type	Double
Default	0
Examples	`portExt.Port1 = 10e-6 'Write`
	`prt1 = portExt.Port1 'Read`
C++ Syntax	HRESULT get_Port1(double *pVal) HRESULT put_Port1(double newVal)
Interface	IPortExtension

Port2 Property

Description	Sets a Port Extension value for Port 2
VB Syntax	*portExt*.**Port2** = *value*
Variable	**(Type) - Description**
portExt	A Port Extension **(object)**
value	**(double)** - Port Extension value in seconds. Choose any number between **-10** and **10**
Return Type	Double
Default	0

Examples

```
portExt.Port2 = 10e-6 'Write
```

```
prt2 = portExt.Port2 'Read
```

C++ Syntax

HRESULT get_Port2(double *pVal)
HRESULT put_Port2(double newVal)

Interface IPortExtension

About Port Extensions

Write/Read

Port3 Property

Description	Sets a Port Extension value for Port 3
VB Syntax	*portExt*.**Port3** = *value*
Variable	**(Type) - Description**
portExt	A Port Extension **(object)**
value	**(double)** - Port Extension value in seconds. Choose any number between **-10** and **10**
Return Type	Double
Default	0
Examples	`portExt.Port3 = 10e-6 'Write`
	`prt3 = portExt.Port3 'Read`
C++ Syntax	HRESULT get_Port3(double *pVal) HRESULT put_Port3(double newVal)
Interface	IPortExtension

Read/Write

PortCLogic Property

Description Sets and reads the logic mode of Port C on the AUX IO connector and the Handler IO connector.

Note: Port C lines are connected to both the Handler IO and Aux IO in the PNA. Therefore, this command will affect both of these connectors in the same way.

VB Syntax *AuxIO*.**PortCLogic** = *value*

Variable **(Type) - Description**

AuxIO **(object) -** A Hardware Aux I/O object

value (**Enum as NaRearPanelIOLogic**) - Choose from:

0 - naPositiveLogic - The associated data line goes **HIGH** when writing a 1 to a PortC bit.

1 - naNegativeLogic - The associated data line goes **LOW** when writing a 1 to a PortC bit.

When Port C is in Output/Write mode, a change in logic causes the output lines to change state immediately. For example, Low levels change to High levels.

When Port C is in Input/Read mode, a change in logic will not cause the lines to change, but data read from Port C will reflect the change in logic.

Return Type Enum

Default 1 - naNegativeLogic

Examples

```
auxIO.PortCLogic = value 'Write
value = auxIo.PortCLogic 'Read
```

C++ Syntax HRESULT put_PortCLogic (tagNARearPanelIOLogic Mode);

HRESULT get_PortCLogic (tagNARearPanelIOLogic* Mode);

Interface IHWAuxIO

Read/Write

PortCMode Property

Description Sets and reads whether Port C is setup for writing or reading data on the AUX IO connector and the Handler IO connector.

Note: Port C lines are connected to both the Handler IO and Aux IO in the PNA. Therefore, this command will affect both of these connectors in the same way.

VB Syntax *AuxIO*.**PortCMode** = *value*

Variable **(Type) - Description**

AuxIO **(object) -** A Hardware Aux I/O object

value **(enum as NaPortMode)** - Choose from:

0 - naInput - set the port for reading

1 - naOutput - set the port for writing

Return Type Enum as NaPortMode

Default 1 - naInput

Examples

```
auxIo.get_PortCMode = naInput 'Write
value = auxIo.get_PortCMode 'Read
```

C++ Syntax HRESULT get_PortCMode(tagNAPortMode* pMode);

HRESULT put_PortCMode(tagNAPortMode pMode);

Interface IHWAuxIO

PortLabel Property

Description	Sets and returns the label on the calibration kit Port for the calibration wizard.
VB Syntax	*calKit*.**Portlabel** (*portNum*) = *value*
Variable	**(Type) - Description**
calKit	A CalKit **(object)**
portNum	**(long integer) -** number of the port to be labeled. Choose either **1** or **2**
value	**(string) -** Label that is visible in the calibration wizard.
Return Type	String
Default	Depends on the Cal Kit.
Examples	`calKit.PortLabel = "MyCalKit" 'Write` `kitLabel = calKit.PortLabel 'Read`
C++ Syntax	HRESULT get_PortLabel(long port, BSTR *pVal) HRESULT put_PortLabel(long port, BSTR newVal)
Interface	ICalKit

Read/Write

PortLogic Property

Description Sets and returns the logic mode of data ports A-H on the HandlerIO connector. Port C of the Handler IO is connected internally to the Port C of the Aux IO connector. Therefore, it will have the same logic mode.

VB Syntax *handler*.**PortLogic** = *value*

Variable	(Type) - Description
handler	**(object) -** A HandlerI/O object
value	(**enum as NaRearPanelIOLogic**) - Choose from: **0 - naPositiveLogic** - When a value of one is written, the associated line goes High **1 - naNegativeLogic** - When a value of one is written, the associated line goes Low For ports that are in output (write) mode, a change in logic causes the output lines to change state immediately. For example, Low levels change immediately to High levels. For ports that are in input (read) mode (C,D,E only), a change in logic will be reflected when data is read from that port. For example, if a line read 0, the next read after a logic change will read 1.

Return Type Long Integer

Default 1 - naNegativeLogic

Examples

```
handler.PortLogic = value 'Write
value = handler.PortLogic 'Read
```

C++ Syntax HRESULT put_PortLogic(tagNARearPanelIOLogic Mode);

HRESULT get_PortLogic(tagNARearPanelIOLogic* Mode);

Interface IHWMaterialHandlerIO

PortMode Property

Description Sets and returns whether Port C or Port D is used for writing or reading data on the Handler IO connector. The Handler IO Port C is connected internally to the Port C of the Aux IO connector. Therefore, the Aux IO connector will have the same input/output mode.

VB Syntax *handler*.**PortMode** *(port)* = *value*

Variable **(Type) - Description**

handler **(object) -** A Handler I/O object

port **(enum as NAMatHandlerPort)** Port to be changed. Choose from:

2 -naPortC

3- naPortD

value **(enum as NaPortMode)** - Choose from:

0 - naInput - set the port for reading

1 - naOutput - set the port for writing

Return Type Long Integer

Default 1 - naInput

Examples

```
handler.PortMode(naPortC) = naInput 'Write
value = handler.PortMode(naPortD) 'Read
```

C++ Syntax HRESULT put_PortMode (tagNAMatHandlerPort Port, tagNAPortMode Mode);

HRESULT get_PortMode (tagNAMatHandlerPort Port, tagNAPortMode* Mode);

Interface IHWMaterialHandlerIO

PowerSlope Property

Description	Sets or returns the Power Slope value. Power Slope function increases or decreases the output power over frequency. Units are db/GHz. For example: PowerSlope = 2 will increase the power 2db/1GHZ.
VB Syntax	*object*.**PowerSlope** = *value*
Variable	**(Type) - Description**
object	Channel **(object)** **or** CalSet **(object) -** Read-only property
value	**(double)** - Power Slope. Choose any number between -2 and 2. **No slope = 0**
Return Type	Double
Default	0
Examples	`chan.PowerSlope = 2 'Write` `pwrslp = chan.PowerSlope 'Read`
C++ Syntax	HRESULT get_PowerSlope(double *pVal) HRESULT put_PowerSlope(double newVal)
Interface	IChannel |CalSet3

PowerAcquisitionDevice Property

Description Returns the power sensor channel (A or B) that is currently selected for use at a specific frequency.

If UsePowerSensorFrequencyLimits is set to False, this property will return the sensor channel last used for a source power calibration. This setting corresponds to the **Use this sensor only** checkbox in the Power Sensor Settings dialog.

When performing an SMC calibration, use SetPowerAcquisitionDevice Method to set the power sensor channel.

VB Syntax *sensor* = *pwrCal*.**PowerAcquisitionDevice***(dFreq)*

Variable **(Type) - Description**

sensor (**enum NAPowerAcquisitionDevice**) The currently selected sensor channel for the specified frequency. Choose from:

0 – naPowerSensor_A

1 – naPowerSensor_B

pwrCal A SourcePowerCalibrator (object)

dFreq (double) Frequency (Hz) for the power reading of interest.

Return Type enum NAPowerAcquisitionDevice

Default Not Applicable

Examples

```
selectedSensor = pwrCal.PowerAcquisitionDevice(1.E9) 'Write
```

C++ Syntax HRESULT get_PowerAcquisitionDevice(double dFreq, tagNAPowerAcquisitionDevice* enumAcqDevice);

Interface ISourcePowerCalibrator2

Read-only

PowerMeterChannel Property

Description Identifies which channel of the power meter the power sensor is connected to.

VB Syntax *chan* = *powerSensor*.**PowerMeterChannel**

Variable **(Type) - Description**

chan **(enum NAPowerAcquisitionDevice)** – Power meter channel identifier for sensor. Choose from:

0 – naPowerSensor_A

1 – naPowerSensor_B

pwrSensor **(object) -** A PowerSensor (object)

Return Type NAPowerAcquisitionDevice

Default Not Applicable

Examples

```
Set pwrCal = pna.SourcePowerCalibrator
meterChannel = pwrCal.PowerSensors(1).PowerMeterChannel
```

C++ Syntax HRESULT PowerMeterChannel(tagNAPowerAcquisitionDevice *pSensor);

Interface IPowerSensor

About Source Power Cal

Write / Read

PowerMeterGPIBAddress Property

Description Specifies the GPIB address of the power meter that will be referenced by the SourcePowerCalibrator object.

When performing a source power cal, the PNA will search VISA interfaces that are configured in the Agilent IO LIbraries on the PNA.

VB Syntax *powerCalibrator*.**PowerMeterGPIBAddress** = *value*

Variable **(Type) - Description**

powerCalibrator **(object) -** A SourcePowerCalibrator (object)

value **(long integer)** – Power meter GPIB address. Choose any number between 0 and 30.

Return Type Long integer

Default 13

Examples

```
Set powerCalibrator = pna.SourcePowerCalibrator
powerCalibrator.PowerMeterGPIBAddress = 13 'Write

pwrMtrAddress = powerCalibrator.PowerMeterGPIBAddress 'Read
```

C++ Syntax HRESULT put_PowerMeterGPIBAddress(long newVal);

HRESULT get_PowerMeterGPIBAddress(long *pVal);

Interface ISourcePowerCalibrator

R1inputPath Property

Description PNA models with option 081 have a switch in the test set that allows access to the port 1 reference receiver through the front panel Reference 1 connectors. This command throws that switch between the internal path to the receiver, or through the external connectors. You could use this feature to make converter measurements relative to a reference ("golden") mixer.

See other Frequency Offset properties.

VB Syntax chan.**R1InputPath** = *value*

Variable **(Type) - Description**

chan A Channel **(object)**

value **(Enum as naInputPath)** - Choose from:naPathInternal -

(0) - internal path to the reference receiver naPathExternal

(1) - path through external connectors

Return Type Enum

Default naPathInternal - (0)

Examples

```
chan.R1InputPath = naPathInternal 'Write
```

```
Inpath = chan.R1InputPath 'Read
```

C++ Syntax HRESULT get_R1InputPath (tag NAInputPath *pPath);

HRESULT put_R1InputPath (tag NAInputPath newPath);

Interface IChannel2

Write / Read

ReadingsPerPoint Property

Description Specifies the maximum number of power readings that are taken at each stimulus point to allow for power meter settling. Each reading is averaged with the previous readings at that stimulus point. When this average meets the ReadingsTolerance value or this number of readings has been made, the average is returned as the valid reading.

VB Syntax *pwrCal*.**ReadingsPerPoint** = *value*

Variable **(Type) - Description**

pwrCal **(object) -** A SourcePowerCalibrator (object)

value **(long integer)** – Number of power readings. Choose any number between 3 and 100.

Return Type Long Integer

Default 3

Examples

```
Set powerCalibrator = pna.SourcePowerCalibrator
powerCalibrator.ReadingsPerPoint = 3 'Write

numReadings = powerCalibrator.ReadingsPerPoint 'Read
```

C++ Syntax HRESULT put_ReadingsPerPoint(long newVal);

HRESULT get_ReadingsPerPoint(long *pVal);

Interface ISourcePowerCalibrator

About Source Power Cal

ReadingsTolerance Property

Description Each power reading is averaged with the previous readings at each stimulus point. When the average meets this tolerance value or the maximum ReadingsPerPoint has been made, the average is returned as the valid reading.

VB Syntax *pwrCal*.**ReadingsTolerance** = *value*

Variable **(Type) - Description**

pwrCal **(object) -** A SourcePowerCalibrator (object)

value **(Double)** – Power meter settling tolerance value in dB. Choose any number between 0 and 5.

Return Type Double

Default .05 dB

Examples

```
Set powerCalibrator = pna.SourcePowerCalibrator
powerCalibrator.ReadingsTolerance = .1 'Write
ReadTol = powerCalibrator.ReadingsTolerance 'Read
```

C++ Syntax HRESULT get_ReadingsTolerance(double *pVal);

HRESULT put_ReadingsTolerance(double newVal);

Interface ISourcePowerCalibrator3

ReadyForTriggerState Property

Description Determines the control of Material Handler connector Pin 21.

VB Syntax *handler*.**ReadyForTriggerState** = *value*

Variable **(Type) - Description**

handler **(object) -** A Handler I/O object

value (**boolean**)

0 (False) **-** Pin 21 is controlled by Output Port B7

1(True) **-** Pin 21 is controlled by the Ready for Trigger signal

Return Type Boolean

Default

Examples

```
handler.ReadyForTriggerState = False 'Write
bState = handler.ReadyForTriggerState 'Read
```

C++ Syntax HRESULT put_ReadyForTriggerState (BOOL *pVal);

HRESULT get_ReadyForTriggerState (BOOL newVal);

Interface IHWMaterialHandlerIO2

ReceiverAttenuator Property

Description Sets or returns the value of the specified receiver attenuator control.

VB Syntax *object*.**ReceiverAttenuator(***rec***)** = *value*

Variable **(Type) - Description**

object Channel **(object)**

or

CalSet **(object) -** Read-only propertyA Channel **(object)**

rec **(long integer)** - Receiver with attenuator control to be changed. Choose from:
1 - Receiver A
2 - Receiver B

value **(double)** - Attenuator value in dB. Choose any Long Integer between 0 and 35 in 5dB steps:

If an invalid value is entered, the analyzer will select the next lower valid value. For example, if 19.9 is entered the analyzer will select 15 dB attenuation.

Return Type Double

Default 0 db

Examples

```
chan.ReceiverAttenuator(1) = 5 'Write
```

```
attn = chan.ReceiverAttenuator(2) 'Read
```

C++ Syntax HRESULT get_ReceiverAttenuator(long lport, double *pVal)
HRESULT put_ReceiverAttenuator(long lport, double newVal)

Interface IChannel

|CalSet3

ReceiverCount Property

Description	Returns the number of receivers in the remote PNA.
VB Syntax	*value* **=** *cap*.**ReceiverCount**
Variable	**(Type) - Description**
value	(Long) - Variable to store the returned number of receivers.
cap	A Capabilities **(object)**
Return Type	Long
Default	Not Applicable
Examples	`value = cap.ReceiverCount 'Read`
C++ Syntax	HRESULT get_ReceiverCount(long * receiverCount);
Interface	ICapabilities

ReceiverStepAttenuatorStepSize Property

Description Returns a value indicating the step size of the attenuator.

VB Syntax *value* **=** *cap*.**ReceiverStepAttenuatorStepSize** *(n)*

Variable **(Type) - Description**

value **(Double)** - Variable to store the returned value of the attenuator step size.

cap A Capabilities **(object)**

n **(Long) -** port number to query for the value of the attenuator step size.

Return Type Double

Default Not Applicable

Examples

```
value = cap.ReceiverStepAttenuatorStepSize'Read
```

C++ Syntax HRESULT get_ReceiverStepAttenuatorStepSize(long portNumber, double * stepSize);

Interface ICapabilities

Read-only

ReceivePort Property

Description	Returns the receiver (response) port number of measurement. To understand how this property is useful, see IMeasurement2 Interface.
VB Syntax	*value* = *meas*.**ReceivePort**
Variable	**(Type) - Description**
meas	A Measurement **(object)**
value	**(Long) -** Variable to store the returned value
Return Type	Long Integer
Default	Not Applicable
Examples	`rp = meas.ReceivePort`
C++ Syntax	HRESULT ReceivePort([out, retval] Long* rcvPort);
Interface	IMeasurement2

ReduceIFBW Property

Description	Sets or returns the state of the Reduced IF Bandwidth at Low Frequencies setting.
VB Syntax	*chan*.**ReduceIFBW** = *state*
Variable	**(Type) - Description**
chan	A Channel **(object)**
state	**(boolean)** **0 -** Turns Reduce IFBW **OFF** **1 -** Turns Reduce IFBW **ON**
Return Type	Boolean
Default	1

Examples	`chan.ReduceIFBW = 1 'Write`
	
	`reduce = chan.ReduceIFBW 'Read`

C++ Syntax	HRESULT get_ReduceIFBW(BOOL *pVal) HRESULT put_ReduceIFBW(BOOL newVal)
Interface	IChannel

Read-only

ReferenceCalFactor Property

Description	Reference cal factor (%) associated with this power sensor. This property and the CalFactorSegments collection are used to perform source power calibration **only** if the power sensor does not contain cal factors in EPROM (for example, HP/Agilent 848x sensors).
VB Syntax	*powerSensor*.**ReferenceCalFactor** = *value*
Variable	**(Type) - Description**
pwrSensor	**(object) -** A PowerSensor (object)
value	**(double)** – Cal factor in units of percent. This can be any value between 1 and 150.
Return Type	Double
Default	100

Examples

```
Set powerCalibrator = pna.SourcePowerCalibrator
powerCalibrator.PowerSensors(1).ReferenceCalFactor = 99 'Write
```

```
RefFact = powerCalibrator.PowerSensors(1).ReferenceCalFactor
'Read
```

C++ Syntax	HRESULT put_ReferenceCalFactor(double newVal); HRESULT get_ReferenceCalFactor(double *pVal);
Interface	IPowerSensor

ReferenceMarkerState Property

Description	Turn ON or OFF the reference marker.
VB Syntax	*meas*.**ReferenceMarkerState** = *state*
Variable	**(Type) - Description**
app	A Measurement **(object)**
state	(boolean) - ON (1) turns the reference marker ON OFF (0) turns the reference marker OFF
Return Type	Boolean
Default	0 - OFF

Examples	`meas.ReferenceMarkerState = True`
	
	`reference = meas.ReferenceMarkerState`

C++ Syntax	HRESULT get_ReferenceMarkerState(VARIANT_BOOL bState) HRESULT put_ReferenceMarkerState(VARIANT_BOOL* bState)
Interface	IMeasurement

About Reference Level

Write/Read

ReferenceValue Property

Description	Sets or returns the value of the Y-axis Reference Level of the active trace.
VB Syntax	*trce*.**ReferenceValue** = *value*
Variable	**(Type) - Description**
trce	A Trace **(object)**
value	**(double)** - Reference Value. Units and range depend on the current data format.
Return Type	Double
Default	Not applicable
Examples	`meas.ReferenceValue = 0 'Write` `rlev = meas.ReferenceValue 'Read`
C++ Syntax	HRESULT get_ReferenceValue(double *pVal) HRESULT put_ReferenceValue(double newVal)
Interface	ITrace

About Reference Position

Write/Read

ReferencePosition Property

Description	Sets or returns the Reference Position of the active trace.
VB Syntax	*trce*.**ReferencePosition** = *value*
Variable	**(Type) - Description**
trce	A Trace **(object)**
value	**(double)** - Reference position on the screen measured in horizontal graticules from the bottom of the screen. Choose from any number between: **0** and **10.**
Return Type	Double
Default	0
Examples	`meas.ReferencePosition = 5 'Middle of the screen -Write` `rpos = meas.ReferencePosition -Read`
C++ Syntax	HRESULT get_ReferencePosition(double *pVal) HRESULT put_ReferencePosition(double newVal)
Interface	ITrace

Write/Read.

Scope Property

Description	Sets or returns the scope of a trigger signal. This determines whether a trigger signal affects a single channel or all channels in the PNA. **Note:** naGlobalTrigger is not compatible with TriggerMode = naTriggerModePoint. If you set Scope = naGlobalTrigger, any channel in naTriggerModePoint will be set to naTriggerModeMeasurement .
VB Syntax	*trigsetup.***Scope** = *value*
Variable	**(Type) - Description**
trigsetup	A TriggerSetup **(object)**
value	**(enum NATriggerType)** - Trigger type. Choose from: **0 - naGlobalTrigger** - a trigger signal is applied to all triggerable channels **1 - naChannelTrigger -** a trigger signal is applied to the current channel. The next trigger signal will be applied to the next channel;not necessarily the next channel in numeric sequence (1-2-3-4 and so forth).
Return Type	Long Integer
Default	naGlobalTrigger
Examples	`trigsetup.Scope = naGlobalTrigger 'Write` `trigtyp = trigsetup.Scope 'Read`
C++ Syntax	HRESULT get_Scope(tagNATriggerType *pTrigger) HRESULT put_Scope(tagNATriggerType trigger)
Interface	ITriggerSetup

SearchFunction Property

Description Emulates the Tracking function in the marker search dialog box. The value you choose for SearchFunction will determine the type of search that takes place when the Tracking property is set true.

The tracking function finds the selected search function every sweep. In effect, turning Tracking ON is the same as executing one of the "Search..." methods (such as SearchMin, SearchMax) for every sweep.

VB Syntax *mark*.**SearchFunction** = *value*

Variable **(Type) - Description**

mark A Marker **(object)**

value **(enum NAMarkerFunction)** - search function. Choose from:

0 - naMarkerFunction_**None**
1 - naMarkerFunction_**Min**
2 - naMarkerFunction_**Max**
3 - naMarkerFunction_**Target**
4 - naMarkerFunction_**NextPeak**
5 - naMarkerFunction_**PeakRight**
6 - naMarkerFunction_**PeakLeft**

Return Type Long Integer

Default **0 -** naMarkerFunction_None

Examples

```
mark.SearchFunction = naMarkerFunction_Target 'When this marker is set
to track, it will track the Target value.
```

```
searchfunction = mark.SearchFunction 'Read
```

C++ Syntax HRESULT get_SearchFunction(tagNAMarkerFunction *pVal)
HRESULT put_SearchFunction(tagNAMarkerFunction newVal)

Interface IMarker

About Security Level

Write/Read

SecurityLevel Property

Description	Controls the display of frequency information on the PNA screen and printouts.
VB Syntax	*app*.**SecurityLevel** *value*
Variable	**(Type) - Description**
app	An Application **(object)**
value	**(enum NASecurityLevel)** -Choose either: **0 - None -** ALL frequency information is displayed **1 - Low -** NO frequency information is displayed. Frequency information can be redisplayed using the Security Setting dialog box or this command.. **2- High -** NO frequency information is displayed. Frequency information can be redisplayed ONLY by performing a Preset or recalling an instrument state with None or Low security settings, or using this command.
Return Type	Long Integer
Default	0 - None
Examples	`app.SecurityLevel = None 'Write` `level = app.SecurityLevel 'Read`
C++ Syntax	HRESULT get_NASecurityLevel(tagNASecurityLevel *level); HRESULT put_NASecurityLevel(tagNASecurityLevel level);
Interface	IApplication4

Read-only

SegmentNumber Property

Description Returns the number of the current segment, PowerSensorCalFactorSegment or PowerLossSegment object.

VB Syntax *seg.***SegmentNumber**

Variable **(Type) - Description**

seg **(object) -** A Segment, PowerSensorCalFactorSegment or PowerLossSegment. Get a handle to the object by referring to the item in the appropriate collection (Segments, CalFactorSegments or PowerLossSegments).

Return Type Long Integer

Default Not Applicable

Examples

```
segNum = seg.SegmentNumber 'returns the segment number -Read
```

C++ Syntax HRESULT get_SegmentNumber(long *pVal)

Interface ISegment
IPowerSensorCalFactorSegment
IPowerLossSegment

ShowStatistics Property

Description Displays and hides the measurement (Trace) statistics (peak-to-peak, mean, standard deviation) on the screen. To display measurement statistics for a narrower band of the X-axis, use StatisticsRange.

The analyzer will display either measurement statistics or Filter Bandwidth statistics; not both.

VB Syntax *meas*.**ShowStatistics** = *value*

Variable **(Type) - Description**

meas A Measurement **(object)**

value **(boolean)** - Boolean value:

1 - Show statistics
0 - Hide statistics

Return Type Boolean

Default 0 - Hide

Examples

```
meas.ShowStatistics = True 'Write
```

```
showstats = meas.ShowStatistics 'Read
```

C++ Syntax HRESULT put_ShowStatistics(VARIANT_BOOL bState)

Interface IMeasurement

SICL Property

Description

Allows you to control the PNA via SICL (standard instrument control library). In this mode, the analyzer can receive SCPI commands from the LAN interface or from a program residing on the PNA itself. This command performs the same function as the SICL / GPIB dialog box - **SICL Enabled** checkbox. See Configuring the analyzer for SICL/VISA.

With this method you can augment a test program written using SICL that resides on the PNA so that it will run unattended. An automation script can be written to start the PNA, enable SICL (using the SICL property), and then start the SICL based program.

VB Syntax

app.**SICL** *value*

Variable

(Type) - Description

app

An Application **(object)**

value

(Boolean) Choose from:

True - enable SICL

False - disable SICL

Return Type

Boolean

Default

False

Examples

```
Dim Pna as AgilentPNA835x.Application
Dim siclState as Boolean
Set Pna = CreateObject("AgilentPNA835x.Application")
Pna.SICL = true          'write
```

```
siclState = Pna.SICL      'Read
```

C++ Syntax

HRESULT get_SICL(VARIANT_BOOL *pVal)

HRESULT put_SICL(VARIANT_BOOL newVal)

Interface

IApplication5

SICLAddress Property

Description Sets and returns the PNA SICL address. This is the address used for SICL over LAN.

VB Syntax *app*.**SICLAddress** = *value*

Variable **(Type) - Description**

app An Application **(object)**

value (Integer) SICL Address of the PNA. Choose a value between 0 and 30.

Return Type Short Integer

Default 16

Examples

```
address=app.SICLAddress 'Read
```

```
app.SICLAddress=16 'Write
```

C++ Syntax HRESULT get_SICLAddress(short busIndex, short* address);

HRESULT put_SICLAddress(short busIndex,short address);

Interface IApplication8

About Performing a Calibration

Read / Write

Simultaneous2PortAcquisition Property

Description Specifies whether a 2-port calibration will be done with a single set of standards (one port at a time) or with two sets of standards (simultaneously).

The AcquireCalStandard2 command uses the same standard index for each calibration class. To specify the calibration standard gender for each port, you must first ensure that the order of calibration class accurately reflects the configuration of your DUT. For example, for a DUT with a male connector on port 1 and a female connector on port 2, order the devices within the S11 classes (A, B, and C) such that the MALE standards are first in the list. Then order the S22 classes specifying the FEMALE standards as the first in the list.

VB Syntax *cal*.**Simultaneous2PortAcquisition** = *state*

Variable **(Type) - Description**

cal A Calibrator **(object)**

state **(boolean)** - Choose from:

True - measures 2 ports simultaneously
False - measures 1 port at a time

Return Type Boolean

Default False

Examples

```
cal.Simultaneous2PortAcquisition = True
```

C++ Syntax HRESULT put_Simultaneous2PortAcquisition(VARIANT_BOOL bTwoSetsOfStandards)

HRESULT Simultaneous2PortAcquisition(VARIANT_BOOL *bTwoSetsOfStandards)

Interface ICalibrator

About Smoothing

Write/Read

SmoothingAperture Property

Description	Specifies or returns the amount of smoothing as a ratio of the number of data points in the measurement trace.
VB Syntax	*meas*.**SmoothingAperture** = *value*
Variable	**(Type) - Description**
meas	A Measurement **(object)**
value	**(double)** - Smoothing Aperture. A ratio of (aperture points / trace points)/100 Choose any number between **.01** and **.25**.
Return Type	Double
Default	.25
Examples	`meas.SmoothingAperture = .10 'Write` `saperture = meas.SmoothingAperture 'Read`
C++ Syntax	HRESULT get_SmoothingAperture(double *pVal) HRESULT put_SmoothingAperture(double newVal)
Interface	IMeasurement

Smoothing Property

Description Turns ON and OFF data smoothing.

VB Syntax *meas*.**Smoothing** = *state*

Variable **(Type) - Description**

meas A Measurement **(object)**

state **(boolean)**
1 - Turns smoothing ON
0 - Turns smoothing OFF

Return Type Boolean

Default 0

Examples

```
meas.Smoothing = 1 'Write
```

```
smooth = meas.Smoothing 'Read
```

C++ Syntax HRESULT get_Smoothing(VARIANT_BOOL *pVal)
HRESULT put_Smoothing(VARIANT_BOOL newVal)

Interface IMeasurement

About SnP files

Write/Read

SnPFormat Property

Description Specifies the format of .S1P, .S2P, .S3P data.

Use either app.Save (saves data to file) or meas.GetSnPData (reads data into variant array).

VB Syntax *pref*.**SnPFormat** = *value*

Variable **(Type) - Description**

pref A Preferences **(object)**

value **(string)** - Format of the .S1P, .S2P, .S3P data. Choose from:

"MA" - Linear Magnitude / degrees

"DB" - Log Mag / degrees

"RI" - Real / Imaginary

"Auto" - Format in which the trace is already displayed. If other than Log Mag, Linear Magnitude, or Real/Imag, then the format will be in Real/Imag.

Return Type String

Default "Auto"

Examples

```
pref.SnPFormat = "MA” 'Write
```

```
format = pref.SnPFormat 'Read
```

C++ Syntax HRESULT get_SnPFormat(BSTR *Format)
HRESULT put_SnPFormat(BSTR Format)

Interface IPreferences

About Limits

Write/Read

SoundOnFail Property

Description	Turns ON or OFF the audio indicator for limit failures.
VB Syntax	*limitst.***SoundOnFail** = *state*
Variable	**(Type) - Description**
limitst	A LimitTest **(object)**
state	**(boolean)** **0 -** Turns the sound OFF **1 -** Turns the sound ON
Return Type	Long Integer
Default	**1 -** ON
Examples	`Limttest.SoundOnFail = 1 'Write` `sound = Limttest.SoundOnFail 'Read`
C++ Syntax	HRESULT get_SoundOnFail(VARIANT_BOOL *pVal) HRESULT put_SoundOnFail(VARIANT_BOOL newVal)
Interface	ILimitTest

SourceCount Property

Description	Returns the number of sources in the remote PNA.
VB Syntax	*value* **=** *cap*.**SourceCount**
Variable	**(Type) - Description**
value	(Long) - Variable to store the returned number of sources.
cap	A Capabilities **(object)**
Return Type	Long
Default	Not Applicable
Examples	`value = cap.SourceCount 'Read`
C++ Syntax	HRESULT get_SourceCount(long * sourceCount);
Interface	ICapabilities

Read-only

SourcePort Property

Description Returns the source port of measurement. To understand how this property is useful, see IMeasurement2 Interface.

VB Syntax *value* = *meas*.**SourcePort**

Variable **(Type) - Description**

meas A Measurement **(object)**

value **(Long) -** Variable to store the returned value

Return Type Long Integer

Default Not Applicable

Examples

```
sp = meas.SourcePort
```

C++ Syntax HRESULT SourcePort([out, retval] Long* srcPort);

Interface IMeasurement2

Source Property

Description Sets or returns the source of triggering in the PNA.

VB Syntax *app.***Source** = *value*

Variable **(Type) - Description**

app A TriggerSetup **(object)**

value **(enum NATriggerSource)** - Choose from:

0 - naTriggerSourceInternal - free run

1 - naTriggerSourceManual - manual trigger source; use app.ManualTrigger to send a trigger signal.

2 - **naTriggerSouceExternal -** a trigger signal is generated when a trigger signal is sensed on the external trigger pin of the Aux IO connector. Use ExternalTriggerConnectionBehavior to configure the characteristics of the external trigger signal.

Return Type Long Integer

Default naTriggerSourceInternal

Examples

```
app.Source = naTriggerSourceInternal 'Write
```

```
trigsource = app.Source 'Read
```

C++ Syntax HRESULT get_Source(tagNATriggerSource *pTrigger);

HRESULT put_Source(tagNATriggerSource trigger);

Interface ITriggerSetup

Write / Read

SourcePowerCalPowerOffset Property

Description Sets or returns a power level offset from the PNA test port power. This can be a gain or loss value (in dB) to account for components you connect between the source and the reference plane of your measurement. For example, specify 10 dB to account for a 10 dB amplifier at the input of your DUT.

Cal power is the sum of the test port power setting and this offset value. Following the calibration, the PNA power readouts are adjusted to the cal power.

This property performs the same function as the power offset argument on the SetCalInfo2 Method

VB Syntax *chan*.**SourcePowerCalPowerOffset** *(sourcePort) = value*

Variable **(Type) - Description**

chan **(object) -** A Channel object

sourcePort **(long integer)** - The source port for which to set this power offset value.

value **(double)** - Gain or loss value in dB. Choose a value between -200 and 200

Return Type Double

Default 0 dB

Examples

```
chan.SourcePowerCalPowerOffset(1) = 10 'Write
offset = chan.SourcePowerCalPowerOffset(2) 'Read
```

C++ Syntax HRESULT get_SourcePowerCalPowerOffset(long sourcePort, double *pVal);

HRESULT put_SourcePowerCalPowerOffset(long sourcePort, double newVal);

Interface IChannel4

SourcePowerCorrection Property

Description Sets source power correction ON or OFF for a specific source port on this channel, or returns the current ON or OFF state of correction for that source port.

VB Syntax *chan*.**SourcePowerCorrection**(sourcePort) = *value*

Variable **(Type) - Description**

chan **(object)** – A Channel object

sourcePort **(long integer)** – Source port for which to set or return the ON or OFF state of source power correction.

value **(boolean)**

False (0) – Turns source power correction OFF for the source port.

True (1) – Turns source power correction ON for the source port.

Return Type Boolean

Default False (0) - Source power correction will turn correction ON

Examples

```
chan.SourcePowerCorrection(1) = 1 'Write
calOnPort2 = chan.SourcePowerCorrection(2) 'Read
```

C++ Syntax HRESULT put_SourcePowerCorrection(VARIANT_BOOL bState);

HRESULT get_SourcePowerCorrection(VARIANT_BOOL *bState);

Interface IChannel

Write/Read

SourcePowerOption Property

Description Enables the source power to be set on individual sweep segments. This property must be set True **before** seg.TestPortPower = *value* is sent. Otherwise, the test port power command will be ignored.

VB Syntax *segs*.**SourcePowerOption** = *state*

Variable **(Type) - Description**

segs A Segments collection **(object)**

state **(boolean)**
True (1) - Enables variable TestPortPower to be set segment sweep
False (0) - Disables variable TestPortPower to be set segment sweep

Return Type Boolean
True - Enabled
False - Disabled

Default False

Examples

```
segs.SourcePowerOption = True 'Write
```

```
powerOption = SourcePowerOption 'Read
```

C++ Syntax HRESULT get_SourcePowerOption(VARIANT_BOOL *pVal)
HRESULT put_SourcePowerOption(VARIANT_BOOL newVal)

Interface ISegments

SourcePowerState Property

Description Turns Source Power ON and OFF.

See note about source power state with instrument state save and recall.

VB Syntax *app*.**SourcePowerState** = *state*

Variable **(Type) - Description**

app An Application **(object)**

state **(boolean)**
False (0) - Turns Source Power OFF
True (1) - Turns Source Power ON

Return Type Boolean
0 - Power OFF
1 - Power ON

Default ON (1)

Examples

```
app.SourcePowerState = True 'Write
```

```
pwr = app.SourcePowerState 'Read
```

C++ Syntax HRESULT get_SourcePowerState(VARIANT_BOOL *pVal)
HRESULT put_SourcePowerState(VARIANT_BOOL newVal)

Interface IApplication

Write/Read

Span Property

Description Sets or returns the Span time of either Gating or Time Domain transform windows

VB Syntax *object*.**Span** = *value*

Variable **(Type) - Description**

object **(object)** As Gating
or
(object) As Transform

value **(double)** - Span time in seconds. Choose any number between: **2*[(number of points-1) / frequency span]** and **0**

Return Type Double

Default 20ns

Examples

```
Trans.Span = 4.5e-9 'sets the time span of a transform window -Write
Gate.Span = 4.5e-9 'sets the Span time of a gating window -Write
```

```
span = Trans.Span 'Read
```

C++ Syntax HRESULT get_Span(double *pVal)
HRESULT put_Span(double newVal)

Interface ITransform
IGating

Read-only

Span Property

Description Returns the stimulus span of the measurement (stop-start data points). To understand how this property is useful, see IMeasurement2 Interface.

VB Syntax *value* = *meas*.**Span**

Variable **(Type) - Description**

value **(Double)** - Variable to store the returned value.

meas A Measurement **(object)**

Return Type Double

Default Not Applicable

Examples

```
Print meas.Span 'prints the span of the measurement
```

C++ Syntax HRESULT get_Span(double * Val);

Interface IMeasurement2

StandardDeviation Property

Description Returns the standard deviation of the measurement.
To retreive all 3 statistics value at the same time, use meas.GetTraceStatistics

VB Syntax *stdev* = *meas*.**StandardDeviation**

Variable **(Type) - Description**

stdev **(single)** - Variable to store standard deviation value

meas A Measurement **(object)**

Return Type Single

Default Not applicable

Examples

```
stdev = meas.StandardDeviation 'Read
```

C++ Syntax HRESULT get_StandardDeviation(float* stdDeviation)

Interface IMeasurement

Write/Read

StandardForClass Property

Description Sets a standard to a calibration class. Does NOT set or dictate the order for measuring the standards.

VB Syntax *calKit*.**StandardForClass***(class, portNum) = value*

Variable **(Type) - Description**

calKit A CalKit **(object).** Use calKit.GetCalStandard to get a handle to the standard.

class **(enum NACalClass)** Standard. Choose from:

1 - naClassA

2 - naClassB

3 - naClassC

4 - naClassD

5 - naClassE

6 - naReferenceRatioLine

7 - naReferenceRatioThru

SOLT Standards

1 - naSOLT_Open

2 - naSOLT_Short

3 - naSOLT_Load

4 - naSOLT_Thru

5 - naSOLT_Isolation

TRL Standards

1 - naTRL_Reflection

2 - naTRL_Line_Reflection

3 - naTRL_Line_Tracking

4 - naTRL_Thru

5 - naTRL_Isolation

portNum **(long) -** The port number the standard will be connected to. For example, you may have a 3.5mm connector designated for port 1, and Type N designated for port 2.

value **(long)** - Calibration class number. Choose a number between **1** and **8.** The <*value*> numbers are associated with the following calibration classes:

<*value*>	Class	Description
1	S11A	Reflection standard
2	S11B	Reflection standard
3	S11C	Reflection standard
4	S21T	Thru standard
5	S22A	Reflection standard
6	S22B	Reflection standard
7	S22C	Reflection standard
8	S21T	Thru standard

Return Type Long Integer

Default Not Applicable

Examples

```
calKit.StandardForClass(naSOLT_Short, 1) = 1
```

```
Kclass = calKit.StandardForClass(naSOLT_Short, 1)
```

C++ Syntax HRESULT put_StandardForClass (NACalClass item, long pNum);

HRESULT get_StandardForClass (NACalClass* item, long *pNum);

Interface ICalKit

Write/Read

StartFrequency Property

Description Sets or returns the start frequency of the channel
or
Sets or returns the start frequency of the segment.

see also Measurement2 interface

VB Syntax *object*.**StartFrequency** = *value*

Variable **(Type) - Description**

object Channel **(object)**

or

CalSet **(object) -** Read-only property

value **(double)** - Start frequency in Hertz. Choose any number between the **minimum** and **maximum** frequencies of the analyzer.

Return Type Double

Default Channel - Minimum frequency of the analyzer
Segment - 0

Examples

```
chan.StartFrequency = 4.5e9 'sets the start frequency of a linear sweep for the channel object -Write
```

```
startfreq = Chan.StartFrequency 'Read
```

C++ Syntax HRESULT get_StartFrequency(double *pVal)
HRESULT put_StartFrequency(double newVal)

Interface IChannel
ISegment

|CalSet3

StartPower Property

Description	Sets the start power of the analyzer when sweep type is set to Power Sweep. Frequency of the measurement is set with chan.CWFrequency.
VB Syntax	*object*.**StartPower** = *value*
Variable	**(Type) - Description**
object	Channel **(object)** **or** CalSet **(object) -** Read-only property
value	**(double)** - Start Power in dBm. **Note:** The range of settable power values depends on the PNA model and if source attenuators are installed. To determine the range of values, use cap.MaximumSourceALCPower and cap.MinimumSourceALCPower Auto attenuation is not allowed in Power Sweep.
Return Type	Double
Default	0
Examples	`Chan.StartPower = -10 'Write` `strtpwr = Chan.StartPower 'Read`
C++ Syntax	HRESULT get_StartPower(double *pVal) HRESULT put_StartPower(double newVal)
Interface	IChannel |CalSet3

About Time Domain

Write/Read

Start Property

Description Sets or returns the start time of either Gating or Time Domain transform windows

VB Syntax *object*.**Start** = *value*

Variable **(Type) - Description**

object **(object)** As Gating
or
(object) As Transform

value **(double)** - Start time in seconds. Choose any number between:
± (number of points-1) / frequency span

Return Type Double

Default -10ns

Examples

```
Trans.Start = 4.5e-9 'sets the start time of a transform window -Write
Gate.Start = 4.5e-9 'sets the start time of a gating window -Write
```

```
strt = Trans.Start 'Read
```

C++ Syntax HRESULT get_Start(double *pVal)
HRESULT put_Start(double newVal)

Interface ITransform
IGating

Read-only

Start Property

Description	Returns the stimulus value of the first data point for the measurement. To understand how this property is useful, see IMeasurement2 Interface.
VB Syntax	*value* = *meas*.**Start**
Variable	**(Type) - Description**
value	**(Double)** - Variable to store the returned value
meas	A Measurement **(object)**
Return Type	Double
Default	Not Applicable
Examples	`Print meas.Start 'prints the stimulus value of the first data point`
C++ Syntax	HRESULT get_Start (double * Val);
Interface	IMeasurement2

Write/Read

State Property

Description	Turns an Object ON and OFF.
VB Syntax	*object.***State** = *value*
Variable	**(Type) - Description**
object	Applies to any of the following: Gating (object) LimitTest (object) Port Extension (object) Segment (object) Transform (object) **Note:** For **LimitTest.State** - If using Global Pass/Fail status, trigger the PNA AFTER turning Limit testing ON.
value	**(boolean)** - **0** - Turns *obj* OFF **1** - Turns *obj* ON
Return Type	Long Integer
Default	Depends on the object: **0** - Gating **0** - LimitTest **0** - Port Extension **1** - Segment **0** - Transform
Examples	`Seg.State = 1 'Turns the segment object ON -Write` `tran = Trans.State 'returns the state of Transform -Read`
C++ Syntax	HRESULT get_State(VARIANT_BOOL *pVal) HRESULT put_State(VARIANT_BOOL newVal)
Interface	ISegment ITransform IGating ILimitTest IPortExtension

About User Ranges

Write/Read

Statistics Range Property

Description	Sets the User Range number for calculating measurement statistics. Set the start and stop values for a User Range with chan.UserRangeMin and chan.UserRangeMax. There are 9 User Ranges per channel. User ranges are applied independently to any measurement.
VB Syntax	*meas*.**StatisticsRange** = *value*
Variable	**(Type) - Description**
meas	A Measurement **(object)**
value	**(long integer)** - Range Number. Choose any number between 0 and 9. **1 - 9** are user-defined ranges **0** is Full Span
Return Type	Long Integer
Default	0
Examples	`meas.StatisticsRange = 2 'Write` `statrange = meas.StatisticsRange 'Read`
C++ Syntax	HRESULT get_StatisticsRange(long* rangeNumber) HRESULT put_StatisticsRange(long rangeNumber)
Interface	IMeasurement

Write/Read

StepRiseTime Property

Description Sets or returns the Rise time of the stimulus in Low Pass Step Mode.

VB Syntax *trans*.**StepRiseTime** = *value*

Variable **(Type) - Description**

trans A Transform **(object)**

value **(double)** - Rise time in seconds. Choose any number between **5.0e-13** and **1.63e-14.**

Return Type Double

Default 0

Examples

```
trans.StepRiseTime = 1.0e-14 'sets the step rise time to 100
psec. -Write
```

```
rt = trans.StepRiseTime 'Read
```

C++ Syntax HRESULT get_StepRiseTime(double *pVal)
HRESULT put_StepRiseTime(double newVal)

Interface ITransform

StimulusValues Property

Description	Returns multiple X-axis frequency arrays (source and response) needed by frequency offset measurements. The arrays contain one frequency value for each data point.
VB Syntax	*value* = *object*.**StimulusValues** *(range)*
Variable	**(Type) - Description**
object	Channel **(object)** **or** CalSet **(object) -** Read-only property
range	**(Long)** – Specify **0** to return source frequencies. Specify **1** to return response frequencies.
Return Type	1-dimensional variant array
Default	Not Applicable
Examples	`array = CalSet.StimulusValues 'Read`
C++ Syntax	HRESULT get_StimulusValues (long range, VARIANT* vals)
Interface	ICalSet3

Write/Read

StopFrequency Property

Description	Sets or returns the stop frequency of the channel **or** Sets or returns the stop frequency of the segment. see also Measurement2 interface
VB Syntax	*object*.**StopFrequency** = *value*
Variable	**(Type) - Description**
object	Channel **(object)** **or** CalSet **(object) -** Read-only property
value	**(double)** - Stop frequency in Hertz. Choose any number between the **minimum** and **maximum** frequencies of the analyzer.
Return Type	Double
Default	Channel **-** Maximum frequency of the analyzer Segment - 0
Examples	`chan.StopFrequency = 4.5e9 'sets the stop frequency of a linear sweep for the channel object -Write` `stopfreq = Chan.StopFrequency 'Read`
C++ Syntax	HRESULT get_StopFrequency(double *pVal) HRESULT put_StopFrequency(double newVal)
Interface	IChannel ISegment |CalSet3

Write/Read

About Power Sweep

StopPower Property

Description	Sets the Stop Power of the analyzer when sweep type is set to Power Sweep. Frequency of the measurement is set with chan.CWFrequency
VB Syntax	*object*.**StopPower** = *value*
Variable	**(Type) - Description**
object	Channel **(object)** **or** CalSet **(object) -** Read-only property
value	**(double)** - Stop Power in dB. Start Power in dB. **Note:** The range of settable power values depends on the PNA model and if source attenuators are installed. To determine the range of values, use cap.MaximumSourceALCPower and cap.MinimumSourceALCPower Auto attenuation is not allowed in Power Sweep.
Return Type	Double
Default	0
Examples	`Chan.StopPower = -10 'Write` `stppwr = Chan.StopPower 'Read`
C++ Syntax	HRESULT get_StopPower(double *pVal) HRESULT put_StopPower(double newVal)
Interface	IChannel |CalSet3

About Time Domain

Stop Property

Description Sets or returns the Stop time of either Gating or Time Domain transform windows

VB Syntax *object*.**Stop** = *value*

Variable **(Type) - Description**

object **(object)** As Gating
or
(object) As Transform

value **(double)** - Start time in seconds. Choose any number between:
± (number of points-1) / frequency span

Return Type Double

Default 10 ns

Examples

```
Trans.Stop = 4.5e-9 'sets the stop time of a transform window - Write
Gate.Stop = 4.5e-9 'sets the stop time of a gating window -Write
```

```
stp = Trans.Stop 'Read
```

C++ Syntax HRESULT get_Stop(double *pVal)
HRESULT put_Stop(double newVal)

Interface ITransform
IGating

Read- only

Stop Property

Description	Returns the stimulus value of the last data point for the measurement. To understand how this property is useful, see IMeasurement2 Interface.
VB Syntax	*value* = *meas*.**Stop**
Variable	**(Type) - Description**
value	**(Double)** Variable to store the returned value
meas	A Measurement **(object)**
Return Type	Double
Default	Not Applicable
Examples	`Print meas.Stop 'prints the stimulus value of the last data point`
C++ Syntax	HRESULT get_Stop(double * Val);
Interface	IMeasurement2

Read/Write

SweepEndMode Property

Description	Sets and reads the event that will cause the Sweep End line to go to a low state. The line will return to a high state after the appropriate calculations are complete. **Note**: This line is connected to the following pins on the HANDLER IO connector and AUX IO connector in the PNA. Therefore, this command will affect both of these connectors in the same way.
VB Syntax	*object*.**SweepEndMode** = *value*
Variable	**(Type) - Description**
object	**(object) -** A HandlerIO or AuxIO object
value	(enum as NASweepEndMode) Choose from: 0 - naSweep - the line goes low when each sweep is complete 1 - naChannelSweep - the line goes low when all the sweeps for each channel is complete. 2 - naGlobalSweep - the line goes low when all sweeps for all triggerable channels are complete.
Return Type	Long Integer
Default	0 - naSweep
Examples	`HWAuxIO.PassFailMode = naSweep 'Write` `value = HWAuxIO.PassFailMode 'Read`
C++ Syntax	HRESULT put_SweepEndMode (tagNASweepEndMode Mode); HRESULT get_SweepEndMode (tagNASweepEndMode* Mode);
Interface	IHWAuxIO IHWMaterialHandlerIO

About the ExtTestSetIO connector

Read-only

SweepHoldOff Property

Description Returns a boolean that represents the state of SweepHoldoff line (pin2) of the External Test Set connector.

VB Syntax *value* = *ExtIO*.**SweepHoldOff**

Variable **(Type) - Description**

value **(boolean)** - Variable to store the returned data

ExtIO **(object) -** An External IO object

Return Type Boolean

False (0) - indicates the line is being held at a TTL Low

True (1) - indicates the line is being held at a TTL High

Default Not Applicable

Examples

```
value = ExtIO.SweepHoldOff
```

C++ Syntax HRESULT get_SweepHoldOff(VARIANT_BOOL* bValue);

Interface IHWExternaTestSetIO

SweepGenerationMode Property

Description	Sets the method used to generate a sweep: continuous ramp (analog) or discrete steps (stepped).
VB Syntax	*object*.**SweepGenerationMode** = *value*
Variable	**(Type) - Description**
object	Channel **(object)** **or** CalSet **(object) -** Read-only property
value	**(enum NASweepGenerationModes)** - **Choose either:** **0 - naSteppedSweep** - source frequency is CONSTANT during measurement of eah displayed point. More accurate than Analog. Dwell time can be set in this mode. **1 - naAnalogSweep** - source frequency is continuously RAMPING during measurement of each displayed point. Faster than Stepped. Sweep time (not dwell time) can be set in this mode.
Return Type	Long Integer
Default	Analog
Examples	`Chan.SweepGenerationMode = naAnalogSweep 'Write` `swpgen = Chan.SweepGenerationMode 'Read`
C++ Syntax	HRESULT get_SweepGenerationMode(tagNASweepGenerationModes* pVal) HRESULT put_SweepGenerationMode(tagNASweepGenerationModes newVal)
Interface	IChannel |CalSet3

SweepTime Property

Description Sets the Sweep time of the analyzer. If sweep time accuracy is critical, use ONLY the values that are attained using the up and down arrows next to the sweep time entry box. See Sweep Time.

VB Syntax *object*.**SweepTime** = *value*

Variable **(Type) - Description**

object Channel **(object)**

or

CalSet **(object) -** Read-only property

value **(double)** - Sweep time in seconds. The maximum sweep time of the PNA is 86400 seconds (1 day).

To set the fastest sweep speed possible, set this value to 0.

Return Type Double

Default 0

Examples

```
chan.SweepTime = 3e-3 'Write
```

```
swptme = chan.SweepTime 'Read
```

C++ Syntax HRESULT get_SweepTime(double *pVal)
HRESULT put_SweepTime(double newVal)

Interface IChannel

CalSet3

About Sweep Types

Write/Read

SweepType Property

Description	Sets the type of X-axis sweep that is performed on a channel.
VB Syntax	*object*.**SweepType** = *value*
Variable	**(Type) - Description**
object	Channel **(object)** **or** CalSet **(object) -** Read-only property
value	**(enum NASweepTypes)** - Choose from: **0 - naLinearSweep** **1 - naLogSweep** **2 - naPowerSweep** **3 - naCWTimeSweep** **4 - naSegmentSweep** **Note**: Sweep type cannot be set to Segment sweep if there are no segments turned ON. A segment is automatically turned ON when a application is created.
Return Type	Long Integer
Default	naLinearSweep
Examples	`chan.SweepType = naPowerSweep 'Write` `swptyp = chan.SweepType 'Read`
C++ Syntax	HRESULT get_SweepType(tagNASweepTypes* pVal) HRESULT put_SweepType(tagNASweepTypes newVal)
Interface	IChannel |CalSet3

Write/Read

SystemImpedanceZ0 Property

Description	Sets and returns the impedance for the analyzer.
VB Syntax	*app*.**SystemImpedanceZ0** = *value*
Variable	**(Type) - Description**
app	An Application **(object)**
value	(double) Analyzer Impedance. Choose any number between 0 and 1000 ohms.
Return Type	Double
Default	50
Examples	`app.SystemImpedanceZ0 = 75 'Write`
	`z0 = app.SystemImpedanceZ0 'Read`
C++ Syntax	HRESULT get_SystemImpedanceZ0(double dSystemZ0) HRESULT put_SystemImpedanceZ0(double *pdSystemZ0)
Interface	IApplication

SystemName Property

Description	Returns the computer name of the PNA.
VB Syntax	*name* = *app.***SystemName**
Variable	**(Type) - Description**
name	(String) Variable to store the returned computer name.
app	An Application **(object)**
Return Type	String
Default	Not Applicable
Examples	`name = app.SystemName`
C++ Syntax	HRESULT SystemName(BSTR* computerName)
Interface	IApplication

TargetValue Property

Description Sets the target value for the marker when doing Target Searches (SearchTargetLeft, SearchTarget, SearchTargetRight).

VB Syntax *mark*.**TargetValue** = *value*

Variable **(Type) - Description**

mark A Marker **(object)**

value **(single)** - Target value. Choose any number between: **-500** and **500**

Return Type Single

Default 0

Examples

```
mark.TargetValue = 10.5 'Write
```

```
target = mark.TargetValue 'Read
```

C++ Syntax HRESULT get_TargetValue(float *pVal)
HRESULT put_TargetValue(float newVal)

Interface IMarker

TestPortPower Property

Description Sets or returns the RF power level for the channel
or
Sets or returns the RF power level of the segment.

VB Syntax *object*.**TestPortPower***(portNum) = value*

Variable **(Type) - Description**

object A Channel **(object)**

or

A CalSet **(object)**

or

A Segment **(object) - See also** Segs.SetAllSegments

portNum **(long integer)** - Port number for which to set source power.

If chan.CouplePorts = True, then **chan.TestPortPower (1) = 0** sets power level at ALL ports to 0 dBm.

value **(double)** - RF Power in dBm.

Note: The range of settable power values depends on the PNA model and if source attenuators are installed. To determine the range of values, use cap.MaximumSourceALCPower and cap.MinimumSourceALCPower

Actual achievable leveled power depends on frequency.

Return Type Double

Default 0

Examples

```
chan.TestPortPower(1) = 5 'sets the port 1 RF power level for the channel object -Write
```

```
powerlev = Chan.TestPortPower(1) 'Read
```

C++ Syntax HRESULT get_TestPortPower(long port, double *pVal)
HRESULT put_TestPortPower(long port, double newVal)

Interface IChannel
ICalSet3
ISegment

Read/Write

ThruCalMethod Property

Description	Sets and returns the method for performing the thru portion of the calibration.
	Note: There is also a ThruCalMethod Property on the GuidedCalibration object.
VB Syntax	*obj*.**ThruCalMethod** = *value*
Variable	**(Type) - Description**
obj	SMCType (object)
	or
	VMCType (object)
value	(String) Specifies the Thru method. Case insensitive - include spaces. Choose from:
	"Default"
	"Flush Thru" **or** "FLUSH"
	"Unknown Thru" **or** "UNKN"
	"Adapter Removal" **or** "ADAP"
Return Type	String
Default	**Default**
Examples	`SMC.ThruCalMethod = "UNKN"`
C++ Syntax	HRESULT put_ThruCalMethod(enum NAThruCalMethod thruMethod);
	HRESULT get_ThruCalMethod(enum NAThruCalMethod *thruMethod);
Interface	SMCType
	VMCType

Read/Write

ThruCalMethod Property

Description	Sets and returns the method for performing the THRU portion of the calibration.
VB Syntax	*guidedCal*.**ThruCalMethod** = *value*
Variable	**(Type) - Description**
guidedCal	GuidedCalibration (object)
value	(Enum as NAThruCalMethod) Specifies the THRU method. Choose from: **0 - naDefaultCalMethod** - allow the PNA to choose the best possible method (from the following) depending on whether the device or ECal module is insertable or non-insertable and given the model number of the PNA. (default selection if omitted.) **1 - naAdapterRemoval** - Perform Adapter removal calibration. **2 - naFlushThru** - Perform Flush Thru calibration. **3 - naDefinedThru -** Perform Defined Thru calibration. If performing an ECal, this is the Thru standard in the ECal module. **4 - naUnknownThru** - Perform Unknown Thru calibration.
Return Type	Enum
Default	0 - naDefaultCalMethod
Examples	`guided.ThruCalMethod = naDefinedThru`
C++ Syntax	HRESULT get_ThruCalMethod(enum NAThruCalMethod *thruMethod); HRESULT put_ThruCalMethod(enum NAThruCalMethod thruMethod);
Interface	IGuidedCalibration

Read/Write

ThruPortList Property

Description

Note: Available only on PNA releases 5.0 and greater.

Sets and returns the thru connection port pairs for the calibration.

- For 3-port cals, specify at least two pairs.
- For 4-port cals, specify at least three pairs.
- For highest accuracy, specify more than the minimum pairs.
- You can query (not set) the thru port pairs for one and two port calibrations.

Learn more about Thru method and port pairings.

See an example of a 4-port guided calibration using COM.

VB Syntax *guidedCal*.**ThruPortList** = *t1a, t1b, t2a, t2b, t3a, t3b*

Variable **(Type) - Description**

guidedCal GuidedCalibration (object)

t1a, t1b... **(Variant)** Port numbers in pairs:

t1a, t1b (Thru1 - port A and port B)

,t2a, t2b (Thru2 - port A and port B)

,t3a, t3b (Thru3 - port A and port B)

Return Type **Variant -** Returns comma-separated pairs.

Default Not Applicable

Example

```
guided.ThruPortList = 1,2,1,3,1,4
```

```
'Sets the following three thru connections for a 4-port
calibration:
Thru 1 - ports 1 and 2
Thru 2 - ports 1 and 3
Thru 3 - ports 1 and 4
```

C++ Syntax HRESULT get_ThruPortList(VARIANT* portList);

HRESULT put_ThruPortList(VARIANT portList);

Interface IGuidedCalibration

Write/Read

Title Property

Description	Writes or reads a custom title for the window. Newer entries replace (not append) older entries.Turn the title ON and OFF with TitleState
VB Syntax	*win.***Title** = *string*
Variable	**(Type) - Description**
win	A NaWindow **(object)**
string	**(long)** - Title limited to 50 characters.
Return Type	String
Default	Null
Examples	`win.Title = "Hello World" 'Write` `titl = win.Title 'Read`
C++ Syntax	HRESULT get_Title(BSTR *title) HRESULT put_Title(BSTR title)
Interface	INAWindow

TitleState Property

Description Turns ON and OFF the window title. Write a window title with Title

VB Syntax *win.***TitleState** = *state*

Variable **(Type) - Description**

win A NaWindow **(object)**

state **(boolean)**
True (1) - Title ON
False (0) - Title OFF

Return Type Long Integer
0 - Title OFF
1 - Title ON

Default 0 - OFF

Examples

```
win.TitleState = True 'Write
```

```
titlestate = win.TitleState 'Read
```

C++ Syntax HRESULT get_TitleState(VARIANT_BOOL* bState)
HRESULT put_TitleState(VARIANT_BOOL bState)

Interface INAWindow

Write/Read

TraceMath Property

Description Performs math operations on the measurement object and the trace stored in memory. (There MUST be a trace stored in Memory to perform math. See Meas.DataToMemory method.)

VB Syntax *meas.***TraceMath** = *value*

Variable **(Type) - Description**

meas A measurement **(object)**

value **(enum NAMathOperation)** - Choose from:

0 - naDataNormal
1 - naDataMinusMemory
2 - naDataPlusMemory
3 - naDataDivMemory
4 - naDataTimesMemory

Return Type NAMathOperation

Default Normal (0)

Examples

```
meas.TraceMath = naDataMinusMemory 'Write
```

```
mathOperation = meas.TraceMath 'Read
```

C++ Syntax HRESULT get_TraceMath(tagNAMathOperation* pMathOp)
HRESULT put_TraceMath(tagNAMathOperation mathOp)

Interface IMeasurement

Tracking Property

Description This property, when on, executes the search function (marker.SearchFunction) every sweep.

In effect, turning Tracking ON is the same as executing one of the immediate, one-time, "Search..." methods (such as SearchMin, SearchMax) for every sweep.

VB Syntax *mark*.**Tracking** = *state*

Variable **(Type) - Description**

mark A Marker **(object)**

state **(boolean)** - Tracking state. Choose from:
ON (1)
OFF (0)

Return Type Boolean
0 - Tracking OFF
1 - Tracking ON

Default 0 - OFF

Examples

```
mark.Tracking = 1 'Write
```

```
markTracking = mark.Type 'Read
```

C++ Syntax HRESULT put_Tracking(VARIANT_BOOL bOn)
HRESULT get_Tracking(VARIANT_BOOL * pbOn)

Interface IMarker

About Time Domain

Mode Property

Description Sets the type of transform.

VB Syntax *trans*.**Mode** = *value*

Variable **(Type) - Description**

trans A Transform **(object)**

value **(enum NATransformMode)** - Choose from:

0 - naTransformBandpassImpulse
1 - naTransformLowpassImpulse
2 - naTransformLowpassStep

Return Type NATransformMode

Default **0** - naTransformBandpassImpulse

Examples

```
trans.Mode = naTransformLowpassStep 'Write
```

```
transmode = trans.Mode 'Read
```

C++ Syntax HRESULT get_Mode(tagNATransformMode *pVal)
HRESULT put_Mode(tagNATransformMode newVal)

Interface ITransform

Write/Read

TriggerDelay Property

Description Sets and reads the trigger delay. This delay is only applied while in External Trigger mode. After an external trigger is applied, the start of the sweep is delayed for the specified delay value plus any inherent latency.

VB Syntax *app.***TriggerDelay** = *value*

Variable **(Type) - Description**

app An Application **(object)**

value **Double**- Trigger delay value. Range is from 0 to 1 second

Return Type Double

Default 0

Examples

```
app.TriggerDelay = .003  'Write
```

```
delay = app.TriggerDelay 'Read
```

C++ Syntax HRESULT get_TriggerDelay(double *delay);

HRESULT put_TriggerDelay(double delay)

Interface IApplication

TriggerOutputEnabled Property

Description	Enables the PNA to send trigger signals out the rear-panel TRIGGER OUT connector. For more information, see External triggering.
VB Syntax	*trigsetup.***TriggerOutputEnabled** = *boolean*
Variable	**(Type) - Description**
trigsetup	A TriggerSetup2 **(object)**
boolean	Choose from: **0 - False -** PNA does NOT send output trigger signals. **1 - True -** PNA sends output trigger signals.
Return Type	Boolean
Default	False
Examples	`trigsetup.TriggerOutputEnabled = 1 'Write` `atba = trigsetup.TriggerOutputEnabled 'Read`
C++ Syntax	HRESULT get_TriggerOutputEnabled(BOOL *pVal); HRESULT put_TriggerOutputEnabled(BOOL newVal);
Interface	ITriggerSetup2

TriggerMode Property

Description Each trigger signal will cause either:
all measurements in the channel to be made **or**
only a single data point in the channel at a time.

VB Syntax *chan.***TriggerMode** = *value*

Variable **(Type) - Description**

chan A Channel **(object)**

value **(enum NATriggerMode)** - Choose from:

0 - naTriggerModePoint - a single data point is measured with each trigger signal the channel receives. Subsequent trigger signals continue to go to the channel in Point mode until the channel measurements are complete.

1 - naTriggerModeMeasurement - all measurements in the channel are made with each trigger signal the channel receives.

Note: Point Mode is not compatible when TriggerType is set to naGlobalTrigger. If you change any channel to TriggerModePoint, TriggerType will be set to naChannelTrigger.

Return Type Long Integer

Default 0 - naTriggerModeMeasurement

Examples

```
chan.TriggerMode = naTriggerModePoint 'Write
```

```
trigtyp = chan.TriggerMode 'Read
```

C++ Syntax HRESULT get_TriggerMode (tagNATriggerMode *pMode)
HRESULT put_TriggerMode (tagNATriggerMode newMode)

Interface IChannel

About Trigger Source

Write/Read

TriggerSignal Property - Obsolete

Description

Note: This command has been replaced by Source Property

Sets or returns the trigger source.

VB Syntax *app.***TriggerSignal** = *value*

Variable **(Type) - Description**

app An Application **(object)**

value **(enum NATriggerSignal)** - Choose from:

0 - naTriggerInternal - free run

1 - naTriggerExternalPositive - a trigger signal is generated when a TTL high is sensed on the external trigger pin of the Aux IO connector

2 - naTriggerExternalNegative - a trigger signal is generated when a TTL low is sensed on the external trigger pin of the Aux IO connector.

3 - naTriggerManual - manual trigger source; use `app.ManualTrigger` to send a trigger signal.

4 - **naTriggerExternalHigh -** a trigger signal is generated when a TTL high is sensed on the external trigger pin of the Aux IO connector

5 - **naTriggerExternalLow -** a trigger signal is generated when a TTL low is sensed on the external trigger pin of the Aux IO connector

Return Type Long Integer

Default naTriggerInternal

Examples

```
app.TriggerSignal = naTriggerExternalPositive 'Write
```

```
trigsign = app.TriggerSignal 'Read
```

C++ Syntax

HRESULT get_TriggerSignal(tagNATriggerSignal *pSignal)
HRESULT put_TriggerSignal(tagNATriggerSignal signal)

Interface IApplication

Write/Read.

TriggerType Property - Obsolete

Description	**Note**: This property has been replaced with Scope Property. Sets or returns the trigger type which determines the scope of a trigger signal.
VB Syntax	*app.***TriggerType** = *value*
Variable	**(Type) - Description**
app	An Application **(object)**
value	**(enum NATriggerType)** - Trigger type. Choose from: **0 - naGlobalTrigger** - a trigger signal is applied to all triggerable channels **1 - naChannelTrigger -** a trigger signal is applied to the current channel. The next trigger signal will be applied to the next channel; not necessarily channel 1-2-3-4.
Return Type	Long Integer
Default	naGlobalTrigger
Examples	`app.TriggerType = naGlobalTrigger 'Write` `trigtyp = app.TriggerType 'Read`
C++ Syntax	HRESULT get_TriggerType(tagNATriggerType *pTrigger) HRESULT put_TriggerType(tagNATriggerType trigger)
Interface	IApplication

Type (calstd) Property

Description Sets and Returns the type of calibration standard.

VB Syntax *calstd*.**Type** = *value*

Variable **(Type) - Description**

calstd A CalStandard **(object).** Use calKit.GetCalStandard to get a handle to the standard.

value **(enum** NACalStandardType**)** -Choose from:
0 - naOpen
1 - naShort
2 - naLoad
3 - naThru

Return Type Long Integer

Default Not Applicable

Examples

```
calstd.Type = naOpen 'Write
```

```
standardtype = calstd.Type 'Read
```

C++ Syntax HRESULT get_Type(tagNACalStandardType *pVal)
HRESULT put_Type(tagNACalStandardType newVal)

Interface ICalStandard

Write/Read

TZImag Property

Description	Sets and Returns the TZImag value (the **I**maginary Terminal **I**mpedance value) for the calibration standard. Only applicable when "Type" is set to **naArbitraryImpedance.** To set the other resistance values, use TZReal
VB Syntax	*calstd*.**TZImag** = *value*
Variable	**(Type) - Description**
calstd	A CalStandard **(object).** Use calKit.GetCalStandard to get a handle to the standard.
value	**(single)** - Value for TZImag in Ohms
Return Type	Single
Default	Not Applicable
Examples	`calstd.TZImag = 15 'Write the value of TZImag to 15 Ohms` `imp0 = calstd.TZImag 'Read the value of TZImag`
C++ Syntax	HRESULT get_TZImag(float *pVal); HRESULT put_TZImag(float newVal);
Interface	ICalStandard2

Write/Read

TZReal Property

Description	Sets and Returns the TZReal value (the real Terminal Impedance value) for the calibration standard. Only applicable when "Type" is set to **naArbitraryImpedance.** To set the other resistance values, use TZImag
VB Syntax	*calstd*.**TZReal** = *value*
Variable	**(Type) - Description**
calstd	A CalStandard **(object).** Use calKit.GetCalStandard to get a handle to the standard.
value	**(single)** - Value for TZReal in Ohms
Return Type	Single
Default	Not Applicable
Examples	`calstd.TZReal = 15 'Write the value of TZReal to 15 Ohms` `imp0 = calstd.TZReal 'Read the value of TZReal`
C++ Syntax	HRESULT get_TZReal(float *pVal); HRESULT put_TZReal(float newVal);
Interface	ICalStandard2

Read-only

UnusedChannelNumbers Property

Description Returns an array of channel numbers that are NOT in use. An unused channel has NO measurements subscribed to it.

VB Syntax *chanNumbers* **=** *chans.***UnusedChannelNumbers** *(NumberOfChannels)*

Variable	**(Type) - Description**
chanNumbers	Variable array to store the returned channel numbers
chans	A Channel collection **(object)**
NumberOfChannels	**(Long Integer)** Number of channels that you are requesting.

Return Type One-dimensional array of long integers. The size of the array is specified by the *NumberOfChannels* parameter.

Default Not Applicable

Examples

```
chanNumbers = chans.UnusedChannelNumbers(5)
```

C++ Syntax HRESULT get_UnusedChannelNumbers(long numberRequested,VARIANT* channelNumbers);

Interface IChannels2

Read-only

UsedChannelNumbers Property

Description	Returns an array of channel numbers that are in use. A used channel has at least one measurement subscribed to it
VB Syntax	*chanNumbers* **=** *chans.***UsedChannelNumbers**
Variable	**(Type) - Description**
chanNumbers	Variable array to store the returned channel numbers
chans	A Channel collection **(object)**
Return Type	One-dimensional array of long integers
Default	Not Applicable
Examples	`chanNumbers = chans.UsedChannelNumbers`
C++ Syntax	HRESULT get_UsedChannelNumbers(VARIANT* channelNumbers);
Interface	IChannels2

UsePowerLossSegments Property

Description	Specifies if subsequent calls to the AcquirePowerReadings method will make use of the loss table (PowerLossSegments).
VB Syntax	*pwrCal*.**UsePowerLossSegments** = *value*
Variable	**(Type) - Description**
pwrCal	**(object)** – A SourcePowerCalibrator (object)
value	**(boolean) -** **False (0)** – Do not use loss table **True (1)** – Use loss table
Return Type	Boolean
Default	False

Examples	`Set powerCalibrator = pna.SourcePowerCalibrator`
	`powerCalibrator.UsePowerLossSegments = 1 'Write`
	`lossTableState = powerCalibrator.UsePowerLossSegments 'Read`

C++ Syntax	HRESULT put_UsePowerLossSegments(VARIANT_BOOL bState); HRESULT get_UsePowerLossSegments(VARIANT_BOOL *bState);
Interface	ISourcePowerCalibrator

Write / Read

UsePowerSensorFrequencyLimits Property

Description Specifies if subsequent calls to the AcquirePowerReadings method will observe frequency values of the MinimumFrequency and MaximumFrequency properties.

VB Syntax *pwrCal*.**UsePowerSensorFrequencyLimits** = *value*

Variable **(Type) - Description**

pwrCal **(object)** – A SourcePowerCalibrator (object)

value **(boolean) -**

False (0) – Do not use power sensor frequency limits. An acquisition will use just one power sensor for the entire sweep, regardless of frequency.

True (1) – Use power sensor frequency limits. A requested acquisition will only succeed for those frequency points which fall between the MinimumFrequency and MaximumFrequency values of that PowerSensor. An acquisition will pause in mid-sweep if the frequency is about to exceed the MaximumFrequency value. When the sweep is paused in this manner, a sensor connected to the other channel input of the power meter can be connected to the measurement port in place of the previous sensor, and then the sweep completed by another call to AcquirePowerReadings. However, the MaximumFrequency specified for the second sensor would need to be sufficient for the sweep to complete.

Return Type Boolean

Default False (0)

Examples

```
Set powerCalibrator = pna.SourcePowerCalibrator
powerCalibrator.UsePowerSensorFrequencyLimits = 1 'Write
FreqCheck = powerCalibrator.UsePowerSensorFrequencyLimits 'Read
```

C++ Syntax HRESULT put_UsePowerSensorFrequencyLimits(VARIANT_BOOL bState);

HRESULT get_UsePowerSensorFrequencyLimits(VARIANT_BOOL *bState);

Interface ISourcePowerCalibrator

UserRange Property

Description Assigns the marker to the specified User Range. This restricts the marker's x-axis travel to the User Range span, specified with Start and Stop values.

- Each channel has 10 user ranges.
- Markers and trace statistics can be restricted to any user range.
- More than one marker can occupy a user range.
- User ranges can overlap. For example:
  - User range 1 - 3GHz to 5GHz
  - User range 2 - 4GHz to 6GHz

Note: User ranges are especially useful in restricting marker searches to specific areas of the measurement.

VB Syntax *mark*.**UserRange** = *value*

Variable **(Type) - Description**

mark A Marker **(object)**

value **(long integer)** - User Range. Choose any number between 0 and 9 (0=Full Span)

Return Type Long Integer

Default 0 - Full Span

Examples

```
mark.UserRange = 1 'Write
```

```
UseRnge = mark.UserRange 'Read
```

C++ Syntax

HRESULT get_UserRange(long *pRangeNumber)
HRESULT put_UserRange(long lRangeNumber)

Interface IMarker

UserRangeMax Property

Description Sets the stimulus stop value for the specified User Range.

This property uses different arguments for the channel and marker objects.

VB Syntax *chan.***UserRangeMax(***domainType,Mnum***)** = *value*
or
mark.**UserRangeMax(***rnum***)** = *value*

Variable **(Type) - Description**

chan A Channel **(object)**

mark A Marker **(object)**
To assign a marker to a User Range, use the UserRange Property.

Note: The Marker object does not require the "DomainType" argument.

domainType **(enum NADomainType)** - Choose from:

0 - naDomainFrequency
1 - naDomainTime
2 - naDomainPower

Mnum **(long integer)** - User Range number. Choose any number between 1 and 9 (0=Full Span)

value **(double)** - Stop value. Choose any number within the full span of the channel

Return Type Double

Default The current stimulus setting for the channel

Examples

```
mark.UserRangeMax(1) = 3e9 'Write
chan.UserRangeMax(naDomainFrequency,1) = 3e9 'Write
```

```
UseRngeMax = mark.UserRangeMax 'Read
UseRngeMax = chan.UserRangeMax 'Read
```

C++ Syntax HRESULT put_UserRangeMax(tagNADomainType domain, long rangeNumber, double maxValue)
HRESULT get_UserRangeMax(tagNADomainType domain, long rangeNumber, double *maxValue)

Interface IChannel

UserRangeMin Property

Description	Sets the stimulus start value for the specified User Range. This property uses different arguments for the channel and marker objects.
VB Syntax	*chan.***UserRangeMin(***domainType,range***)** = *value* **or** *mark*.**UserRangeMin(***range***)** = *value*
Variable	**(Type) - Description**
chan	A Channel **(object)**
mark	A Marker **(object)** To assign a marker to a User Range, use the UserRange Property. **Note:** The Marker object does not require the DomainType argument
domainType	**(enum NADomainType)** Type of sweep currently implemented on the channel - Choose from: **0 - naDomainFrequency** **1 - naDomainTime** **2 - naDomainPower**
range	**(long)** - User Range number. Choose any number between **1** and **9** (0=Full Span)
value	**(double)** - Start value. Choose any number within the full span of the analyzer
Return Type	Double
Default	The current stimulus setting for the channel

Examples

```
mark.UserRangeMin(1) = 3e9 'Write
chan.UserRangeMin(naDomainFrequency,1) = 3e9 'Write
```

```
UseRngeMin = mark.UserRangeMin 'Read
UseRngeMin = chan.UserRangeMin 'Read
```

C++ Syntax	HRESULT put_UserRangeMin(tagNADomainType domain, long rangeNumber, double minValue) HRESULT get_UserRangeMin(tagNADomainType domain, long rangeNumber, double *minValue)
Interface	IChannel

UserPresetEnable Property

Description 'Checks' and 'clears' the enable box on the User Preset dialog box. This only affects subsequent Presets from the front panel user interface.

Regardless of the state of the User Preset Enable checkbox, the app.Preset command will always preset the PNA to the factory preset settings, and app.UserPreset will always perform a User Preset.

VB Syntax *app.*__UserPresetEnable__ = *state*

Variable **(Type) - Description**

app An Application **(object)**

state **(boolean)** Front Panel User Preset State. Choose from:

0 (False) – User Preset OFF

1 (True) – User Preset ON

Return Type Boolean

Default OFF (0)

Examples

```
app.UserPresetEnable = True 'Write
```

```
upreset = app.UserPresetEnable 'Read
```

C++ Syntax HRESULT get_UserPresetEnable(VARIANT_BOOL *pVal)
HRESULT put_UserPresetEnable(VARIANT_BOOL newVal)

Interface IApplication6

Read only

ValidConnectorTypes Property

Description Returns a list of connector types for which there are calibration kits.

VB Syntax *value* = *obj*.**ValidConnectorTypes**

Variable **(Type) - Description**

value (Variant) List of connector types

obj Any of the following:

GuidedCalibration (object)

SMCType (object)

VMCType (object)

Return Type Variant

Default Not Applicable

Examples

```
value = SMC.ValidConnectorTypes
```

C++ Syntax HRESULT get_ValidConnectorTypes(VARIANT* connectorTypes);

Interface IGuidedCalibration

SMCType

VMCType

About Port Extensions

Write/Read

VelocityFactor Property

Description	Sets the velocity factor to be used with Electrical Delay, Port Extensions, and Time Domain marker distance calculations.
VB Syntax	*app.***VelocityFactor** = *value*
Variable	**(Type) - Description**
app	An Application **(object)**
value	**(double)** - Velocity factor. Choose a number between: **0** and **10** (.66 polyethylene dielectric; .7 teflon dielectric)
Return Type	Double
Default	1

Examples	`app.VelocityFactor = .66 'Write`
	
	`RelVel = app.VelocityFactor 'Read`

C++ Syntax	HRESULT get_VelocityFactor(double *pVal) HRESULT put_VelocityFactor(double newVal)
Interface	IApplication

About Math Operations

View Property

Description Sets (or returns) the type of trace displayed on the screen.

VB Syntax *meas*.**View** = *value*

Variable **(Type) - Description**

meas A measurement **(object)**

value **(enum NAView)** - Type of trace. Choose from:

0 - naData
1 - naDataAndMemory
2 - naMemory
3 - naNoTrace

Note: The **naData** trace may reflect the result of a TraceMath operation.

Return Type NAView

Default naData

Examples

```
meas.View = naData 'Write
```

```
trceview = meas.View 'Read
```

C++ Syntax HRESULT get_View(tagNAView* pView)
HRESULT put_View(tagNAView newView)

Interface IMeasurement

Write/Read

Visible Property

Description Makes the Network Analyzer application visible or not visible. In the Not Visible state, the analyzer cycle time for making measurements can be significantly faster because the display does not process data.

VB Syntax *app*.**Visible** = *state*

Variable **(Type) - Description**

app An Application **(object)**

state **(boolean)**
0 - Network Analyzer application **NOT** visible
1 - Network Analyzer application **IS** visible

Return Type Boolean
0 - Not visible
1 - Visible

Default 1

Examples

```
app.Visible = 0 'Write
```

```
vis = app.Visible 'Read
```

C++ Syntax HRESULT get_Visible(VARIANT_BOOL * bVisible)
HRESULT put_Visible(VARIANT_BOOL bVisible)

Interface IApplication

Write-Read

WGCutoffFreq Property

Description	Sets or returns the value of the waveguide cut off frequency.
VB Syntax	*meas*.**WGCutoffFreq =** *value*
Variable	**(Type) - Description**
meas	A Measurement **(object)**
value	**(double)** - Frequency in Hertz.
Return Type	Double
Default	Not Applicable
Examples	`Print meas.WGCutoffFreq 'prints the value of the waveguide cut off frequency`
C++ Syntax	HRESULT get_WGCutoffFreq(double *pVal); HRESULT put_WGCutoffFreq(double newVal);
Interface	IMeasurement2

Read-only

WindowNumber Property

Description	Returns the window number. You might use this property to identify a particular window so that you can create a new Measurement in that window.
VB Syntax	*value* = *win*.**WindowNumber**
Variable	**(Type) - Description**
win	A NAWindow (object)
value	**(long integer) -** Variable to store the returned window number
Return Type	Long Integer
Default	Not Applicable
Examples	`value = app.ActiveNAWindow.WindowNumber`
C++ Syntax	HRESULT (long* windowNumber);
Interface	INAWindow

About Arranging Windows

Write/Read

WindowState Property

Description Sets or returns the window setting of Maximized, Minimized, or Normal. To arrange all of the windows, use app.ArrangeWindows.

VB Syntax *object*.**WindowState** = *value*

Variable **(Type) - Description**

object An Application **(object)** - main window
or
A NaWindow **(object)** - data windows

value **(enum NAWindowStates)** - The window state. Choose from:
0 - naMinimized - Minimizes the window to an Icon on the lower toolbar
1 - naMaximized - Maximizes the window
2 - naNormal - changes the window size to the user defined setting (between Max and Min).

Return Type Long Integer

Default naMaximized

Examples

```
app.WindowState = naMinimized 'changes the Network Analyzer application
window to an icon. -Write
win.WindowState = naNormal 'changes the window defined by the win object
variable to user defined settings. -Write
```

```
winstate = app.WindowState 'Read
```

C++ Syntax HRESULT get_WindowState(tagNAWindowStates *pVal)
HRESULT put_WindowState(tagNAWindowStates newVal)

Interface INAWindow
IApplication

About X-Axis Spacing

Write/Read

XAxisPointSpacing Property

Description	Sets X-axis Point Spacing for the displaytraces measured with segment sweeps on the active channel.
VB Syntax	*chan*.**XAxisPointSpacing** = *value*
Variable	**(Type) - Description**
chan	A Channel **(object)**
value	**(Enum as naStates)** - Choose from: **0 - naOFF** - Turns X-axis Point Spacing OFF **1 - naON -** Turns X-axis Point Spacing ON
Return Type	Enum
Default	0 - naOFF
Examples	`chan.XAxisPointSpacing = naOFF 'Write` `xspac = chan.XAxisPointSpacing 'Read`
C++ Syntax	HRESULT get_XAxisPointSpacing (tagNAStates *pState); HRESULT put_XAxisPointSpacing (tagNAStates newState);
Interface	IChannel2

Write/Read

About Scale

YScale Property

Description	Sets or returns the Y-axis Per-Division value of the active trace.
VB Syntax	*trace*.**YScale** = *value*
Variable	**(Type) - Description**
trace	A Trace **(object)**
value	**(double)** - Scale /division number. Units and range depend on the current data format.
Return Type	Double
Default	10 (db)
Examples	`trac.YScale = 5 'Write` `yscl = trac.YScale 'Read`
C++ Syntax	HRESULT get_YScale(double *pVal) HRESULT put_YScale(double newVal)
Interface	ITrace

About Modifying Cal Kits

Write/Read

Z0 Property

Description	Sets and Returns the characteristic impedance for the calibration standard.
VB Syntax	*calstd*.**Z0** = *value*
Variable	**(Type) - Description**
calstd	A CalStandard **(object).** Use calKit.GetCalStandard to get a handle to the standard.
value	**(single)** -Impedance in Ohms
Return Type	Single
Default	Not Applicable
Examples	`calstd.Z0 = 50 'Write` `impedance = calstd.Z0 'Read`
C++ Syntax	HRESULT get_Z0(float *pVal) HRESULT put_Z0(float newVal)
Interface	ICalStandard

Abort Method

Description Ends the current measurement sweep on the channel.

VB Syntax *chan.***Abort** [*sync*]

Variable **(Type) - Description**

chan **(object)** - A Channel object

sync **(boolean)** - wait (or not) for the analyzer to stop before processing subsequent commands. Optional argument; if unspecified, value is set to False. Choose from:
True - synchronize **-** the analyzer will not process subsequent commands until the current measurement is aborted.
False - continue processing commands immediately

Return Type None

Default None

Examples

```
chan.abort True
chan.abort
```

C++ Syntax HRESULT Abort(VARIANT_BOOL bSynchronize);

Interface IChannel

AbortPowerAcquisition Method

About Source Power Cal

Description	Aborts a source power cal acquisition sweep that is currently in progress.
VB Syntax	*powerCalibrator*.**AbortPowerAcquisition**
Variable	**(Type) - Description**
powerCalibrator	**(object)** - A SourcePowerCalibrator object
Return Type	None
Default	Not Applicable
Examples	`powerCalibrator.AbortPowerAcquisition`
C++ Syntax	HRESULT AbortPowerAcquisition();
Interface	ISourcePowerCalibrator

Write-only

AcquireCalStandard Method - Obsolete

Description **Note**: This command has been replaced by AcquireCalStandard2 Method, which provides for acquisition of sliding load standards. All other functionality is identical.

VB Syntax *cal*.**AcquireCalStandard** *std[,index]*

Variable **(Type) - Description**

cal A Calibrator **(object)**

std **(enum NACalClass)** Standard to be measured. Choose from:

1 - naClassA

2 - naClassB

3 - naClassC

4 - naClassD

5 - naClassE

6 - naReferenceRatioLine

7 - naReferenceRatioThru

SOLT Standards

1 - naSOLT_Open

2 - naSOLT_Short

3 - naSOLT_Load

4 - naSOLT_Thru

5 - naSOLT_Isolation

TRL Standards

1 - naTRL_Reflection

2 - naTRL_Line_Reflection

3 - naTRL_Line_Tracking

4 - naTRL_Thru

5 - naTRL_Isolation

index **(long integer)** number of the standard. Optional argument - Used if there is more than one standard required to cover the necessary frequency range. If unspecified, value is set to 0.

Return Type None

Default Not Applicable

Examples `Cal.AcquireCalStandard naSOLT_Thru 'Write`

C++ Syntax HRESULT AcquireCalStandard(tagNACalClass enumClass, short standardNumber)

Interface ICalibrator

AcquireCalStandard2 Method

Description Measures the specified standard from the selected calibration kit. The calibration kit is selected using app.CalKitType.

For 2-port calibration, it is also necessary to specify direction with AcquisitionDirection.

To omit Isolation from a 2-port calibration, do not Acquire a cal standard for naSOLT_Isolation.

For using two sets of standards, see Simultaneous2PortAcquisition Property.

Note: This command replaces AcquireCalStandard. This command provides for the acquisition of a sliding load cal. All other functionality is identical.

VB Syntax *cal*.**AcquireCalStandard2** *std[,index][,slide]*

Variable **(Type) - Description**

cal A Calibrator **(object)**

std **(enum NACalClass)** Standard to be measured. Choose from:

1 - naClassA

2 - naClassB

3 - naClassC

4 - naClassD

5 - naClassE

6 - naReferenceRatioLine

7 - naReferenceRatioThru

SOLT Standards

1 - naSOLT_Open

2 - naSOLT_Short

3 - naSOLT_Load

4 - naSOLT_Thru

5 - naSOLT_Isolation

TRL Standards

1 - naTRL_Reflection

2 - naTRL_Line_Reflection

3 - naTRL_Line_Tracking

4 - naTRL_Thru

5 - naTRL_Isolation

[index]	**(long integer)** Number of the standard. Optional argument **-** Used if there is more than one standard required to cover the necessary frequency range. If unspecified, value is set to 1.
[slide]	(**enum** as NACalStandardSlidingState) Optional argument. State of the sliding load. The slide should be set a minimum of five times. Seven is the maximum that can be stored. Choose from: 0 - **naNotSlidingStd** - not using a sliding load - Default if not specified. 1 - **naSlideIsSet** - slide is set for acquisition 2 - **naSlideIsDone** - this next acquisition will be the last. Calculations will then be performed.
Return Type	None
Default	Not Applicable

Examples	`Cal.AcquireCalStandard2 naSOLT_Thru`
	`Cal.AcquireCalStandard2  naSOLT_Thru,2,naNotSlidingStd`
	`'measures the second standard listed in the class of naSOLT_Thru`

C++ Syntax	HRESULT AcquireCalStandard2(NACalClass enumClass, long standardPosition, NACalStandardSlidingState slidingStandardState)
Interface	ICalibrator

AcquireCalConfidenceCheckECALEx Method

Description This method replaces AcquireCalConfidenceCheckECAL

Transfers confidence data from the specified ECal module into the measurement's memory trace. The data is transferred to the specified S-parameter on the same channel as this Calibrator object.

The characterization within the ECal module that the confidence data will be read from is specified by **ECALCharacterizationEx.** The default value is 0.

VB Syntax *cal*.**AcquireCalConfidenceCheckECALEx** *Sparam [,ecalModule]*

Variable **(Type) - Description**

cal A Calibrator **(object)**

Sparam **(String)** S-parameter to transfer confidence data to. This parameter must be present on the same channel as the calibrator object.

ecalModule **(Integer) –** Optional argument. ECal module.

Choose from modules **1** through **8**

Use IsECALModuleFoundEx to determine the number of modules connected to the PNA

Use GetECALModuleInfoEx to return the model and serial number of each module.

Return Type None

Default Not applicable

Examples

```
Cal.AcquireCalConfidenceCheckECALEx "S11", 2
```

C++ Syntax HRESULT AcquireCalConfidenceCheckECALEx(BSTR strParameter, long moduleNumber = 1);

Interface ICalibrator4

AcquirePowerReadings Method

Description	Initiates a source power cal acquisition.
VB Syntax	*powerCalibrator*.**AcquirePowerReadings** *device [,sync]*
Variable	**(Type) - Description**
powerCalibrator	**(object)** - A SourcePowerCalibrator object
device	**(enum NAPowerAcquisitionDevice)** The specific device (sensor on the power meter) to be used for the acquisition. Choose from: **0 – naPowerSensor_A** **1 – naPowerSensor_B** To use the sensor that currently corresponds to the frequency of interest, use the value from the PowerAcquisitionDevice property.
sync	**(boolean)** Optional argument. If not specified, value is set to False. Choose from: **True (1)** – The method does not return until this acquisition has completed (the program calling this method is halted while waiting for the method to return). **False (0)** – The method initiates an acquisition then returns immediately (while the acquisition still proceeds). The program calling this method can then perform other operations during the acquisition.
Return Type	None
Default	Not Applicable
Examples	`powerCalibrator.AcquirePowerReadings naPowerSensor_A, True`
C++ Syntax	HRESULT AcquirePowerReadings(tagNAPowerAcquisitionDevice enumAcqDevice, VARIANT_BOOL bSync);
Interface	ISourcePowerCalibrator

Write-only

AcquireStep Method

Description	Acquire the measurement data for the specified step in the calibration process.
VB Syntax	*obj*.**AcquireStep** *(n)*
Variable	**(Type) - Description**
obj	Any of the following: GuidedCalibration (object) SMCType (object) VMCType (object)
n	Step number in the calibration process. Use GenerateSteps to determine the total number of steps. Use GetStepDescription to read the description of each step.
Return Type	Not Applicable
Default	Not Applicable
Examples	`VMC.AcquireStep (3)`
C++ Syntax	HRESULT put_AcquireStep([in] long step);
Interface	SMCType VMCType IGuidedCalibration

Write-only

Activate Method

Description	Makes an object the Active Object. When making a measurement active, the channel and window the measurement is contained in becomes the active channel and active window. In order to change properties on any of the active objects, you must first have a "handle" to the active object using the **Set** command. For more information, See Getting a Handle to an Object. You do not have to make an object "Active" to set or read its properties remotely. But an object must be "Active" to change its values from the front panel.
VB Syntax	*object.***Activate**
Variable	**(Type) - Description**
object	Measurement **(object)** **or** Marker **(object)**
Return Type	Not Applicable
Default	Not Applicable
Examples	`meas.Activate` `mark.Activate`
C++ Syntax	HRESULT Activate()
Interface	IMeasurement IMarker

ActivateMarker Method

Description	Makes a marker the Active Marker. Use meas.ActiveMarker to read the number of the active marker.
VB Syntax	*meas.***ActivateMarker(***Mnum***)**
Variable	**(Type) - Description**
meas	A Measurement **(object)**
Mnum	**(long integer)** - the number of the marker to make active. Choose any marker number from **1** to **9**.
Return Type	None
Default	Not Applicable
Examples	`meas.ActivateMarker(1)'Write`
C++ Syntax	HRESULT ActivateMarker(long lMarkerNumber)
Interface	IMeasurement
Remarks	Use ReferenceMarkerState to control the Reference marker.

ActivateWindow Method

Description	Makes a window object the Active Window. In order to change properties on any of the active objects, you must first have a "handle" to the active object using the **Set** command. For more information, See Programming the Analyzer Object Model. You do not have to make an object "Active" to set or read its properties remotely. But an object must be "Active" to change its values from the front panel.
VB Syntax	*app.***ActivateWindow** *n*
Variable	**(Type) - Description**
app	An Application **(object)**
n	**(long)** Number of the window to make active
Return Type	Window Object
Default	Not Applicable
Examples	`app.ActivateWindow 4`
C++ Syntax	HRESULT ActivateWindow(long WindowNumber)
Interface	IApplication

Add (channels) Method

Description	Creates a channel and returns a handle to it. If the channel already exists, it returns the handle to the existing channel.
VB Syntax	*chans.***Add** *(item)*
Variable	**(Type) - Description**
chans	A Channel collection **(object)**
item	**(variant)** - Channel number.
Return Type	Channel
Default	Not Applicable
Examples	`chans.Add 3 'Creates channel 3`
C++ Syntax	HRESULT Add(VARIANT numVal, IChannel** pChannel)
Interface	IChannels

Write-only

Add (measurement) Method

Description Adds a Measurement to the collection.

VB Syntax *meas.***Add** *channel,param,source[,window]*

meas A Measurements collection **(object)**

channel **(long)** - Channel number of the new measurement.

param **(string)** - Parameter of the new measurement. Choose from:

- **"S11"**
- **"S22"**
- **"S21"**
- **"S12"**
- **"A"**
- **"B"**
- **"R1"**
- **"R2"**

or

- **combine 2 of (A,B,R1,R2) in this format: "A/R1"**

source **(long integer)** - Source port number; if unspecified, value is set to 1. Only used for non-s-parameter measurements; ignored if s-parameter.

window **(long integer) -** Optional argument. Window number of the new measurement. Choose 1 to 16. If unspecified, the S-Parameter will be created in the Active Window.

Return Type None

Default None

Examples

```
meass.Add 3,"A/R1",1,1 'Adds A/R1 measurement to channel 3 in window 1
```

C++ Syntax HRESULT Add(long ChannelNum, BSTR strParameter, long srcPort, VARIANT_BOOL bNewWindow)

Interface IMeasurements

Write-only

Add (NAWindows) Method

Description	Add a window to the display. Does not add a measurement. The window number must not already exist.
VB Syntax	*wins.***Add** *[item]*
Variable	**(Type) - Description**
wins	A NAWindow collection **(object)**
item	**(variant)** - optional argument; Window number. Range between 1 - 4
Return Type	Object
Default	Not Applicable
Examples	`wins.Add 3 'Creates a window number 3`
C++ Syntax	HRESULT Add(long windowNumber)
Interface	INAWindows

Write-only

Add (PowerLossSegment) Method

Description	Adds a PowerLossSegment to the PowerLossSegments collection. To ensure predictable results, it is best to remove all segments before defining a new list of segments. For each segment in the collection, do a seg.Remove.
VB Syntax	*segs*.**Add** *(item [size])*
Variable	**(Type) - Description**
segs	**(object) -** A PowerLossSegments collection (object)
item	**(variant)** - Number of the new segment. If it already exists, a new segment is inserted at the requested position.
size	**(long integer) -** Optional argument. The number of segments to add, starting with item. If unspecified, value is set to 1.
Return Type	None
Default	Not Applicable
Examples	`segs.Add 1, 4 'Adds segments 1,2,3 and 4`
C++ Syntax	HRESULT Add(VARIANT index, long size);
Interface	IPowerLossSegments

Write-only

Add (PowerSensorCalFactorSegment) Method

Description Adds a PowerSensorCalFactorSegment to the CalFactorSegments collection.

To ensure predictable results, it is best to remove all segments before defining a new list of segments. For each segment in the collection, do a seg.Remove.

VB Syntax *segs***.Add** *(item [size])*

Variable **(Type) - Description**

segs **(object) -** A CalFactorSegments collection (object)

item **(variant)** - Number of the new segment. If it already exists, a new segment is inserted at the requested position.

size **(long integer) -** Optional argument. The number of segments to add, starting with item. If unspecified, value is set to 1.

Return Type None

Default Not Applicable

Examples

```
segs.Add 1, 4 'Adds segments 1,2,3 and 4
```

C++ Syntax HRESULT Add(VARIANT index, long size);

Interface ICalFactorSegments

Write-only

Add (segment) Method

Description	Adds segments to the Segments collection, but does not turn the segments ON.
VB Syntax	*segs.***Add** *(item, [size])*
segs	A segments collection **(object)**
item	**(variant)** Number of the new segment. If it already exists, a new segment is inserted at the requested position.
size	**(long integer)** Optional argument. The number of segments to add, starting with *item*. If unspecified, value is set to 1.
Return Type	None
Default	None
Examples	`Segs.Add 1, 4 'Adds segments 1,2,3,and 4. (does NOT automatically turn segments ON)`
C++ Syntax	HRESULT Add(VARIANT index, long size);
Interface	ISegments
Remarks	To ensure predictable results, it is best to remove all segments before defining a segment list. For each segment in the collection, do a seg.Remove.

About Analyzer Events

Write/Read

AllowAllEvents Method

Description	Sets event filtering to monitor all events in the analyzer. This is the default setting when subscribing to events. This could slow the measurement speed of the analyzer significantly.
VB Syntax	*app*.**AllowAllEvents**
Variable	**(Type) - Description**
app	An Application **(object)**
Return Type	Not Applicable
Default	Not Applicable
Examples	`app.AllowAllEvents`
C++ Syntax	HRESULT AllowAllEvents()
Interface	IApplication

AllowEventCategory Method

Description Sets event filtering to monitor a category of event.

VB Syntax *app*.**AllowEventCategory,** *category, state*

Variable **(Type) - Description**

app An Application **(object)**

category Category to monitor. Choose from list in Working with the Analyzer's Events

state **(boolean)**
True - monitor
False - do not monitor

Return Type Not Applicable

Default Not Applicable

Examples

```
app.AllowEventCategory
```

C++ Syntax HRESULT AllowEventCategory(tagNAEventCategory category, VARIANT_BOOL bAllow)

Interface IApplication

About Analyzer Events

Write/Read

AllowEventMessage Method

Description	Sets event filtering to monitor specific events.
VB Syntax	*app*.**AllowEventMessage** *event*
Variable	**(Type) - Description**
app	An Application **(object)**
event	Event to monitor. Refer to list in Working with the Analyzer's Events
state	**(boolean)** **True** - monitor **False** - do not monitor
Return Type	Not Applicable
Default	Not Applicable
Examples	`app.AllowEventMessage`
C++ Syntax	HRESULT AllowEventMessage(tagNAEventID eventID, VARIANT_BOOL bAllow)
Interface	IApplication

Write/Read

AllowEventSeverity Method

Description	Sets event filtering to monitor levels of severity.
VB Syntax	*app*.**AllowEventSeverity** *severity,state*
Variable	**(Type) - Description**
app	An Application **(object)**
severity	**(enum naEventSeverity)** Choose from:naEventSeverityERROR naEventSeverityINFORMATIONAL naEventSeveritySUCCESS naEventSeverityWARNING
state	**(boolean)** **True** - monitor **False** - do not monitor
Return Type	Not Applicable
Default	Not Applicable
Examples	`app.AllowEventSeverity`
C++ Syntax	HRESULT AllowEventSeverity(tagNAEventSeverity severity, VARIANT_BOOL bAllow)
Interface	IApplication

Write only

Apply Method

Description	Applies the mixer setup to the mixer object and turns the channel ON. (Performs the same function as the Apply button on the mixer setup dialog box.)
VB Syntax	*mxr*.**Apply**
Variable	**(Type) - Description**
mxr	Mixer Interface pointer to the Measurement (object)
Return Type	Not Applicable
Default	Not Applicable
Examples	`mxr.Apply`
C++ Syntax	HRESULT Apply()
Interface	IMixer3

ApplyPowerCorrectionValues Method

Description	Applies the array of power correction values to the channel memory and turns correction ON. Perform after completing a source power cal acquisition sweep. This command does NOT save the correction values. To save correction values, save an instrument / calibration state (*.cst file) after performing a source power cal.
VB Syntax	*powerCalibrator*.ApplyPowerCorrectionValues
Variable	**(Type) - Description**
powerCalibrator	**(object)** - A SourcePowerCalibrator object
Return Type	None
Default	Not Applicable
Examples	`powerCalibrator.ApplyPowerCorrectionValues`
C++ Syntax	HRESULT ApplyPowerCorrectionValues();
Interface	ISourcePowerCalibrator

Autoscale Method

Description Trace Object - Autoscales only the **ONE** trace on which Autoscale is being called.

NAWindow Object - Autoscales **ALL** of the traces in the Window on which Autoscale is being called.

VB Syntax *object*.**Autoscale**

Variable **(Type) - Description**

object Trace **(object)**
or
NAWindow **(object)**

Return Type Not Applicable

Default Not Applicable

Examples

```
Trac.Autoscale 'Autoscales the trace
Win.Autoscale 'Autoscales all the traces in the window -Write
```

C++ Syntax HRESULT AutoScale()

Interface INAWindow
ITrace

AveragingRestart Method

Description	Clears and restarts averaging of the measurement data.
VB Syntax	*chan*.**AveragingRestart**
Variable	**(Type) - Description**
chan	A Channel **(object)**
Return Type	Not Applicable
Default	Not Applicable
Examples	`chan.AveragingRestart`
C++ Syntax	HRESULT AveragingRestart()
Interface	IChannel

Write-only

BuildHybridKit Method

Description	Use this method when you have different port connectors. This is a convenient way to combine two kits that match the connectors on your DUT.
VB Syntax	*app.***BuildHybridKit** *port1Kit,p1sex,port2Kit,p2sex,adapter,user kit*
Variable	**(Type) - Description**
app	An Application **(object)**
port1Kit *port2Kit*	**(enum NACalKit)** - Specifies the two kits to be used to build the hybrid kit. Choose from: naCalKit_85032F_N50 naCalKit_85033E_3_5 naCalKit_85032B_N50 naCalKit_85033D_3_5 naCalKit_85038A_7_16 naCalKit_85052C_3_5_TRL naCalKit_User7 naCalKit_User8 naCalKit_User9 naCalKit_User10
p1sex *p2sex*	**(enum NAPortSex)** - Specifies the sex of the connector at that port. Choose from: **naMale** **naFemale** **naDon'tCare**
adapter	**(enum NAAdapter)** -Choose from: **naUserkit -** the electrical length of the adapter in the userKit specifications **naZeroLength** - no adapter
userKit	**(enum NACalKit)** - The Hybrid kit **-** Choose from the previous list of kits
Return Type	Not Applicable
Default	Not Applicable
Examples	`app.BuildHybridKit naCalKit_85033E_3_5,naMale,naCalKit_85038A_7_16 ,naFemale,naUserkit,naCalKit_User8`
C++ Syntax	HRESULT BuildHybridKit\(tagNACalKit port1Kit, tagNAPortSex port1Sex, tagNACalKit port2Kit, tagNAPortSex port2Sex, tagNAAdapter adapter, tagNACalKit userKit)
Interface	IApplication

Write-only

CalculateErrorCoefficients Method

Description	This method is the final call in a calibration process. It calculates error-correction terms and turns error-correction ON. Do NOT use this command during an ECAL.
VB Syntax	*cal*.**CalculateErrorCoeffecients**
Variable	**(Type) - Description**
cal	Calibrator **(object)**
Return Type	Not Applicable
Default	Not Applicable
Examples	`Cal.CalculateErrorCoeffecients`
C++ Syntax	HRESULT CalculateErrorCoefficients()
Interface	ICalibrator

Write only

Calculate Method

Description Calculates the Input or Output frequencies of the mixer setup, applies the mixer setup to the mixer object, and turns the channel ON.

VB Syntax *mxr*.**Calculate** (*port)*

Variable **(Type) - Description**

mxr Mixer Interface pointer to the Measurement (object)

port (enum as MixerCalculation) Port of the mixer for which to calculate start and stop frequencies. Choose from:

	enum	1st or only stage requires:	In addition, 2nd stage requires:
0	mixCalculateINPUT	• Output Start and Stop frequencies • LO frequency • Output sideband (High or Low)	• IF Start and Stop frequencies • 2nd LO frequency • IF sideband (High or Low)
1	mixCalculateINPUT AndOUTPUT (2 stage mixers ONLY)	NA	• IF Start and Stop frequencies • Both LO frequencies
2	mixCalculateOUTPUT	• Input Start and Stop frequencies • LO frequency • Output sideband (High or Low)	• IF Start and Stop frequencies • 2nd LO frequency • IF sideband (High or Low)
3	mixCalculateLO1	• Input Start and Stop frequencies • Output frequency • Output sideband (High or Low)	• IF Start and Stop frequencies • 2nd LO frequency • IF sideband (High or Low)

4	mixCalculateLO2	NA	• Input Start and stop frequencies • 1st LO start and stop frequencies • Output frequency • IF sideband(High or Low • Output sideband(High or Low)

Return Type Not Applicable

Default Not Applicable

Examples

```
mxr.Calculate (mixCalculateOUTPUT)
```

C++ Syntax HRESULT Calculate()

Interface IMixer

Write-only

ChangeParameter Method

Description Changes the parameter of the measurement.

VB Syntax *meas.***ChangeParameter**(*param,lPort*)

Variable **(Type) - Description**

meas A Measurement **(object)**

param **(string)** - New parameter. Choose from:

S11 | S22 | S21 | S12

Additionally, for 3-port analyzers only:

S33 | S13 | S31 | S23 | S32

For non-ratioed measurements:
A | B | R1 | R2
C - 3-port analyzers only

For ratioed measurements:

A/B

A/C - 3 port analyzers only

B/A

B/C - 3 port analyzers only

C/A - 3 port analyzers only

C/B - 3 port analyzers only

A/R1

B/R1

C/R1 - 3 port analyzers only

A/R2

B/R2

R1/A

R2/A

R1/B

R2/B

R1/C - 3 port analyzers only

R2/R1

R1/R2

lPort	**(long integer)** Load port if *param* is a reflection S-Parameter Ignored if *param* is a transmission S-Parameter Source port if *param* is anything other than an S-parameter
Return Type	Not Applicable
Default	Not Applicable
Examples	`meas.ChangeParameter "S11",1`
C++ Syntax	HRESULT ChangeParameter(BSTR parameter, long lPort)
Interface	IMeasurement

CheckPower Method

Description Measures power at a specified frequency. Use this method to test power level before and/or after applying a source power calibration.

VB Syntax *pow = pwrCal.***CheckPower** *(device, freq* [,*unit*])

Variable **(Type) - Description**

pow **(double)** Variable to store power value returned by this method.

pwrCal A SourcePowerCalibrator **(object)**

device **(enum NAPowerAcquisitionDevice)** The specific sensor on the power meter to be used for the acquisition. Choose from:

0 – naPowerSensor_A

1 – naPowerSensor_B

To use the sensor that currently corresponds to the frequency of interest, use the value from the PowerAcquisitionDevice property.

freq **(double)** Frequency (Hz) at which the sensor is to read the power.

unit **(enum NAPowerUnit)**

Optional argument. Choose from:

naDBM – Returns the power in dBm.(default)

naWATT – Returns the power in Watts.

Return Type Double

Default Not Applicable

Examples

```
watt = powerCalibrator.CheckPower(naPowerSensor_A, 1E9, naWATT)
```

C++ Syntax HRESULT put_CheckPower(tagNAPowerAcquisitionDevice enumAcqDevice, double dFreq, tagNAPowerUnit enumPowerUnit, double *pdPower);

Interface ISourcePowerCalibrator2

Close CalSet Method

Description Closes read/write access to the Cal Set.

See OpenCalSet for an explanation of gaining access to the Cal Set.

When you are finished reading and writing data from or to the Cal Set, close the Cal Set. Subsequent read/writes will require a new OpenCal Set call.

Reading and writing Cal Set data is performed with the PutStandard, GetStandard, PutErrorTerm, GetErrorTerm method calls. These methods are provided by the ICal Set and ICalData2 interfaces.

VB Syntax *CalSet*.**CloseCalSet**

Variable **(Type) - Description**

CalSet **(object) -** A Cal Set object

Return Type Not Applicable

Default Not Applicable

Examples

```
Cal Set.CloseCalSet
```

C++ Syntax HRESULT CloseCalSet

Interface ICalSet

Write-only

ComputeErrorTerms Method

Description Computes error terms for the caltype specified by a preceding OpenCal Set call.

The Cal Set must first be opened using OpenCalSet. If this call has not been made, the following error is issued:

E_NA_Cal Set_ACCESS_DENIED

The standards data required for the CalType must be available in the Cal Set or this error will be returned: E_NA_STANDARD_NOT_FOUND.

Note: Error term computation requires data for the actual calibration kit standards from the current kit definition. ComputeErrorTerms assumes that the standards were acquired using only one standard per class.

VB Syntax *CalSet*.**ComputeErrorTerms**

Variable **(Type) - Description**

CalSet **(object) -** A Cal Set object

Return Type Not Applicable

Default Not Applicable

Examples

```
CalSet.ComputeErrorTerms
```

C++ Syntax HRESULT ComputeErrorTerms( )

Interface ICalSet

ConfigNarrowBand3 Method

Description

Note: This method replaces ConfigNarrowBand2 Method. The BW argument now returns 0 if no solution is found for the specified PRF and BW. In addition, adjustments were made to the filter finder algorithm

This subroutine determines, then returns, the proper configuration for pulsed measurements on the PNA using the spectral nulling technique. The configuration returned needs to be sent to the PNA and any other related external equipment such as pulse generators. The routine will take a desired Pulse Repetition Frequency (PRF) and measurement IFBW and return a possibly modified PRF and IFBW for proper pulsed operation on the PNA. The routine will also return the Sample Rate, Number of Taps, and Offset that must be sent to the PNA to configure it in pulsed mode using the spectral nulling technique.

Although the example below uses COM programming to communicate with the PNA, these commands can be replaced with SCPI equivalents.

Note: The pulsed application may set the offset frequency (option 080) of the PNA to some value other than zero (the default value). If the stop frequency is set to the maximum of the PNA model, then an error message may appear on the PNA stating that the response frequency has exceeded the maximum allowed frequency. To fix this, set the stop frequency to a value that is at least 2 KHz less than the maximum allowed. For example, if you have a 20 GHz PNA, and the stop frequency is set to 20 GHz, and the error message appears, then set the stop frequency to 19.999998 GHz

VB Syntax

*Pulsed.***ConfigNarrowBand** *(PRF, NumTaps, BW, OffSet, SampleRate, Precision, FixedPRF, PG81110)*

Variable	(Type) - Description
Pulsed	**(interface)** An interface to the agilentpnapulsed.dll application interface.
PRF	**(Double)** The Pulse Repetition Frequency. **[out]** The pulse repetition frequency that has been optimized for use with the PNA. NOTE: This value may be different from the value requested. **[in]** The desired pulse repetition frequency.
NumTaps	**(Long)** The number of taps to send to the PNA for pulsed operation.
BW	**(Long)** The PNA IF Bandwidth. **[out]** The PNA IF bandwidth that has been optimized for use with the PNA. NOTE: This value may be different from the value requested. Zero (0) is returned if no solution is found for the specified *PRF* and *BW*. **[in]** The desired PNA IF bandwidth.
Offset	**(Double)** The offset value to send to the PNA for pulsed operation. The offset value is used to adjust the PNA for the two different possible sample rates that may be returned.
SampleRate	**(Double)** The sample rate to send to the PNA for pulsed operation.

Precision **(Double)** The precision variables sets the precision that will be used to decrement the PRF when running the configuration routines. This variable can be set to the precision required by the external pulse generators so that the configuration routine will not return a PRF that is not within the precision limits of the pulse generators.

FixedPRF **(Boolean)**

1 (True) Signals the .DLL routine to NOT adjust the PRF value; rather adjust ONLY the IF Bandwidth. This is the default setting.

0 (False) Adjust both the PRF and IF Bandwidth values as necessary.

PG81110 **(Boolean)**

1 (True) You are using an Agilent 81110 as the pulse generator. This allows increased accuracy in adjustments for offset and PRF.

0 (False) Not using an Agilent 81110.

Return Type Not Applicable

Default Not Applicable

Example See an example using this command.

C++ Syntax HRESULT ConfigNarrowBand(double *pPRF, long *pNumTaps, long *pBW, double *pOffset, double *pSampleRate, int Precision)

Interface AgilentPNAPulsed.Application

Write-only

Continuous Method

Description The channel continuously responds to trigger signals.

Note: This command does **NOT** change TriggerSignal to Continuous.

VB Syntax *chan*.**Continuous**

Variable **(Type) - Description**

chan A Channel **(object)**

Return Type Not Applicable

Default Not Applicable

Examples

```
chan.Continuous
```

C++ Syntax HRESULT Continuous()

Interface IChannel

Copy Method

Description Creates a new Cal Set and copies the current Cal Set data into it. Therefore, you now have a clone Cal Set with a different ID. Use this command to manipulate data on a Cal Set without corrupting the original cal data.

VB Syntax *CalSet*.**Copy**

Variable **(Type) - Description**

CalSet **(object) -** A Cal Set object

Return Type Not Applicable

Default Not Applicable

Examples

```
Dim mgr As CalManager
Dim ocalset As CalSet
Dim newcalset As CalSet
Set mgr = pna.GetCalManager
'Create a new (empty) Cal Set.
Set ocalset = mgr.CreateCalSet(1)
ocalset.Description = "original calset"
pna.Channel(1).SelectCalSet ocalset.GetGUID, True

'Launch the cal wizard and allow the user to perform the
calibration.
If pna.LaunchCalWizard(False) Then
'If the Launch returns true then the calibration finished.
ocalset.Save

'Copy the Cal Set to the new one.
Set newcalset = ocalset.Copy
newcalset.Description = "copy of original calset"

Else
'If the cal doesn't finish, delete the old Cal Set
'so it isn't taking up unnecessary memory.
mgr.DeleteCalSet ocalset.GetGUID
End If
```

As a result, the programmer can manipulate the data in the new Cal Set and always revert back to the old Cal Set as needed.

C++ Syntax HRESULT Copy(ICalSet** pCalSet);

Interface ICalSet

Write-only

CopyToChannel Method

Description Sets up another channel as a copy of this object's channel.

VB Syntax chan.**CopyToChannel**(lChanNum)

Variable **(Type) - Description**

chan A Channel **(object)**

lChanNum **(long integer)** – Number of the channel to become a copy of this channel.

Return Type None

Default Not Applicable

Examples

```
Dim chan As Channel
Set chan = PNAapp.ActiveChannel
Const lNEW_CHAN_NUM As Long = 2
chan.CopyToChannel(lNEW_CHAN_NUM)
```

C++ Syntax HRESULT CopyToChannel(long lChanNum);

Interface IChannel2

CreateSParameterEx Method

Description Creates a new S-Parameter measurement in an existing or new window and specifies the load port for 3-port devices.

VB Syntax *app*.**CreateSParameter** *chan,recvr,source[,loadPort][,window]*

Variable **(Type) - Description**

app Application **(object)**

chan **(long integer)** - Channel number of the new measurement

recvr **(long integer)** - Port number of the receiver

source **(long integer)** - Port number of the source

loadPort **(long integer)** - Port number of the load. Required for reflection measurements of 3-port devices on 3-port PNAs.

window **(long integer)** - Optional argument. Window number of the new measurement. Choose **1** to **4**. If unspecified, the S-Parameter will be created in the Active Window.

Return Type Not Applicable

Default Not Applicable

Examples

```
app.CreateSParameter 1,2,1,1 'Creates a new S21 measurement in
channel 1 and New window(1)
app.CreateSParameter 2,1,1,3,1 'Creates a new S11 measurement on
channel 2 with port 3 as the load. Create in the active window
```

C++ Syntax HRESULT CreateSParameter(long ChannelNum, long RcvPort, long SrcPort, long LoadPort, long windowNumber)

Interface IApplication

Write-only

CreateCalSet Method

Description Creates a new Cal Set.

The new cal set is initialized with the stimulus settings from the channel whose number is passed as the argument to this method. Stimulus settings include frequency, bandwidth, number of points, and so forth.

Use this method when you want to manually upload data to the Cal Set using the returned ICal Set interface handle..

The channel number does not restrict the usage of this Cal Set on any other channel. It simply provides a link to the originating channel so that the stimulus values can be stored in the Cal Set.

Note: Be sure to SAVE the CalSet you are creating. Use ICalSet::Save.

VB Syntax *calMgr.***CreateCalSet** *(chan)*

Variable **(Type) - Description**

calMgr **(object) -** A CalManager object

chan **(long) -** channel number of the new Cal Set.

Return Type ICal Set Interface

Default Not Applicable

Example

```
calMgr.CreateCalSet 1
```

C++ Syntax HRESULT CreateCalSet(long ChannelNumber, ICal Set** pCal Set);

Interface ICalManager

Write-only

CreateCustomMeasurementEx Method

Description Creates a new custom measurement.

VB Syntax *app.***CreateCustomMeasurementEx** *chanNum,ID,initData [,window]*

Variable **(Type) - Description**

app **(object) -** An Application object

chanNum **(long)** -Channel number used by the new measurement; can exist or be a new channel.

ID **(string) -** Identifier of the new custom measurement object. The new custom measurement must be installed and registered on the PNA.

Choose from the following Program IDs:

- FCA.VMCMeas
- SMC_Forward.SMC_ForwardMeas
- SMC_Reverse.SMC_ReverseMeas

InitData **(variant)** Measurement specific data for passing initialization data to the new measurement. The variant type and content of the initialization value is defined by the type of measurement being created.

Program ID	Choose from:
FCA.VMCMeas	"S11" "VC21" "S22"
SMC_Forward.SMC_ForwardMeas	"S11" "SC21" "S22" "IPwr" "OPwr"
SMC_Reverse.SMC_ReverseMeas	"S11" "SC12" "S22" "IPwr" "OPwr"

window **(long)** Optional argument. Number of the window the new custom measurement will be placed in. Choose **1** to **16**. If unspecified, the measurement is placed in the active window.

Return Type IMeasurement

Default Not Applicable

Examples

```
'To create a forward scalar mixer measurement in channel 2:
Dim MyMeas as Agilent835x.Measurement
Set MyMeas = app.CreateCustomMeasurementEx ( 2,
"SMC_Forward.SMC_ForwardMeas", "SC21" )
```

```
'To create a reverse scalar mixer measurement in channel 2:
Dim MyMeas as Agilent835x.Measurement
Set MyMeas = app.CreateCustomMeasurementEx ( 2,
"SMC_Reverse.SMC_ReverseMeas", "SC12" )
```

```
'To create a vector mixer measurement in channel 2:
Dim MyMeas as Agilent835x.Measurement
Set MyMeas = app.CreateCustomMeasurementEx ( 2, "FCA.VMCMeas",
"VC21" )
```

C++ Syntax HRESULT put_CreateCustomMeasurementEx (long ChannelNum, BSTR guid, VARIANT initData, long windowNumber, IMeasurement** ppMeasurement);

Interface IApplication3

Write-only

CreateCustomCal Method

Description	Creates a custom cal object.
VB Syntax	*calmgr.***CreateCustomCal**(*CalType*)
Variable	**(Type) - Description**
calMgr	Cal Manager **(Object)**
CalType	**(String)** Name of the calibration. Choose from: **"VMC"** or **"VectorMixerCal.VMCType"** **"SMC"** or **"ScalarMixerCal.SMCType"**
Return Type	Not Applicable
Default	Not Applicable

Examples

```
Dim CalMgr As ICalManager2
Dim SMC As ISMCType
Set SMC = CreateCustomCal("SMC")
```

See SMC and VMC examples using this command.

C++ Syntax	HRESULT CreateCustomCal(BSTR CustomCal)
Interface	ICalManager2

Write-only

CreateMeasurement Method

Description	Creates a new measurement
VB Syntax	*app.***CreateMeasurement** *chanNum,param,lPort[,window]*
Variable	**(Type) - Description**
app	Application **(object)**
chanNum	**(long)** - Channel number of the new measurement; can exist or be a new channel
param	**((string) -** New parameter. Choose from:

S11 | S22 | S21 | S12

Additionally, for 3-port analyzers only:

S33 | S13 | S31 | S23 | S32

For non-ratioed measurements:
A | B | R1 | R2
C -3-port analyzers only

For ratioed measurements:

A/B

A/C - 3 port analyzers only

B/A

B/C - 3 port analyzers only

C/A - 3 port analyzers only

C/B - 3 port analyzers only

A/R1

B/R1

C/R1 - 3 port analyzers only

A/R2

B/R2

R1/A

R2/A

R1/B

R2/B

R1/C - 3 port analyzers only

R2/R1

R1/R2

lPort **(long)** -
Load port if *param* is a reflection S-Parameter

Ignored if *param* is a transmission S-Parameter

Source port if *param* is anything other than an S-parameter

window **(long)** Optional argument. Window number of the new measurement. Choose **1** to **16**. If unspecified, the measurement will be created in the Active Window.

Return Type Not Applicable

Default Not Applicable

Examples

```
app.CreateMeasurement(1,"A/R1",1,0)
```

C++ Syntax HRESULT CreateMeasurement(long ChannelNum, BSTR strParameter, long lPort, long windowNumber)

Interface IApplication

Write-only

DataToDivisor Method

Description	Stores the measurement's data to the measurement's "divisor" buffer for use by the Normalization data processing algorithm. Normalization is currently supported only on measurements of unratioed power, for purpose of receiver power calibration. If DataToDivisor is called on a ratioed measurement (such as an S-parameter), it will return an error.
VB Syntax	*meas*.**DataToDivisor**
Variable	**(Type) - Description**
meas	**(object) -** A Measurement object
Return Type	Not Applicable
Default	Not Applicable
Examples	`meas.DataToDivisor`
C++ Syntax	HRESULT DataToDivisor();
Interface	IMeasurement

DataToMemory Method

Description	Stores the active measurement data into memory creating a memory trace. The memory can then be displayed or used in calculations with the measurement data.
VB Syntax	*meas.***DataToMemory**
Variable	**(Type) - Description**
meas	A Measurement **(object)**
Return Type	Not Applicable
Default	Not Applicable
Examples	`meas.DataToMemory`
C++ Syntax	HRESULT DataToMemory()
Interface	IMeasurement

Write-only

Delete Method

Description	Deletes the measurement.
VB Syntax	*meas.***Delete**
Variable	**(Type) - Description**
meas	The Measurement object to delete **(object)**
Return Type	Not Applicable
Default	Not Applicable
Examples	`meas.Delete`
C++ Syntax	HRESULT Delete()
Interface	IMeasurement

Write-only

DeleteMarker Method

Description	Deletes a marker from the measurement.
VB Syntax	*meas.***DeleteMarker(***Mnum***)**
Variable	**(Type) - Description**
meas	A Measurement **(object)**
Mnum	**(long)** - Any existing marker number in the measurement
Return Type	Not Applicable
Default	Not Applicable
Examples	`meas.DeleteMarker(1)`
C++ Syntax	HRESULT DeleteMarker(long lMarkerNumber)
Interface	IMeasurement

DeleteAllMarkers Method

Description	Deletes all of the markers from the measurement.
VB Syntax	*meas.***DeleteAllMarkers**
Variable	**(Type) - Description**
meas	**(object) -** The Measurement object from which markers will be deleted.
Return Type	Not Applicable
Default	Not Applicable
Examples	`meas.DeleteAllMarkers`
C++ Syntax	HRESULT DeleteAllMarkers()
Interface	IMeasurement

DeleteCalSet Method

Description Deletes a Cal Set from the set of available Cal Sets. This method immediately updates the Cal Set file on the hard drive. If the Cal Set is currently being used by a channel, this request will be denied and an error is returned.

Errors returned by this method:

E_NA_CAL_SET_IN_USE
E_NA_Cal Set_NOT_FOUND
E_NA_Cal Set_SAVE_FAILED

Using the Cal Sets collection is a convenient way to manage Cal Sets.

VB Syntax *calMgr.***DeleteCalSet** *(GUID)*

Variable **(Type) - Description**

calMgr **(object) -** A CalManager object

GUID **(string)** - GUID number of the Cal Set to be deleted

Return Type Not Applicable

Default Not Applicable

Example

```
dim cs As CalSet   ' the collection
dim strGUID as string

strGUID = cs.GetGUID
calMgr.DeleteCalSet  strGUID
```

C++ Syntax HRESULT DeleteCalSet(BSTR strGUID);

Interface ICalManager

Write-only

DeleteShortCut Method

Description	Removes a macro from the list of macros in the analyzer. Does not remove the file. **Note:** There are always 12 macro positions. They do not have to be sequential. For example, you can have number 7 but no numbers 1 to 6.
VB Syntax	*app*.**DeleteShortCut** *item*
Variable	**(Type) - Description**
app	An Application **(object)**
item	**(long integer)** number of the macro to be deleted.
Return Type	Not Applicable
Default	Not Applicable
Examples	`app.DeleteShortCut 2`
C++ Syntax	HRESULT DeleteShortcut(long Number)
Interface	IApplication

DisallowAllEvents Method

Description	Sets event filtering to monitor NO eventst.
VB Syntax	*app*.**DisallowAllEvents**
Variable	**(Type) - Description**
app	An Application **(object)**
Return Type	Not Applicable
Default	Not Applicable
Examples	`app.DisallowAllEvents`
C++ Syntax	HRESULT DisallowAllEvents()
Interface	IApplication

Write-only

DoPrint Method

Description	Prints the screen to the default Printer.
VB Syntax	*app.*__DoPrint__
Variable	**(Type) - Description**
app	An Application **(object)**
Return Type	Not Applicable
Default	Not Applicable
Examples	`app.DoPrint`
C++ Syntax	HRESULT DoPrint()
Interface	IApplication

DoECAL1PortEx Method

Description This method replaces DoECAL1Port Method.

Does a 1-Port calibration using an ECAL module. You must first have a 1-port measurement active to perform the calibration.

The characterization within the ECal module that will be used for the calibration is specified by **ECALCharacterizationEx.** The default value is 0.

VB Syntax *cal.***DoECAL1PortEx** *[port][,module]*

Variable	(Type) - Description
cal	A Calibrator **(object)**
port	**(long integer)** Optional argument - Port number to calibrate. Choose from: 1 - Calibrate port 1 (default if unspecified) 2 - Calibrate port 2
module	**(long integer)** Optional argument. ECal module. Choose from modules **1** through **8** Use IsECALModuleFoundEx to determine the number of modules connected to the PNA Use GetECALModuleInfoEx to returns the model and serial number of each module.
Return Type	None
Default	Not Applicable

Examples

```
cal.DoECAL1PortEx,2,2
```

C++ Syntax HRESULT DoECAL1PortEx(long port, long moduleNumber = 1);

Interface ICalibrator4

DoECAL2PortEx Method

Description	This method replaces DoECAL2Port Method. Does a 2-Port calibration using an ECAL module. You must first have a 2-port measurement active to perform the calibration. The characterization within the ECal module that will be used for the calibration is specified by **ECALCharacterizationEx**. The default value is 0.
VB Syntax	*cal*.**DoECAL2PortEx** *[portA][,portB][,module]*
Variable	**(Type) - Description**
cal	A Calibrator **(object)**
portA	**(long integer)** Optional argument - Number of the receive port to calibrate. Choose from: 1 - Calibrate port 1 (default, if unspecified) 2 - Calibrate port 2 3 - Calibrate port 3 (if the PNA has 3 ports)
portB	**(long integer)** Optional argument - Number of the source port to calibrate. Choose from: 1 - Calibrate port 1 (default, if unspecified) 2 - Calibrate port 2 3 - Calibrate port 3 (if the PNA has 3 ports)
module	**(long integer)** Optional argument. ECal module. Choose from modules **1** through **8** Use IsECALModuleFoundEx to determine the number of modules connected to the PNA Use GetECALModuleInfoEx to returns the model and serial number of each module.
Return Type	None
Default	Not Applicable
Examples	`cal.DoECAL2PortEx,1,2,3`
C++ Syntax	HRESULT DoECAL2PortEx(long portA = 1, long portB =2, long moduleNumber = 1);
Interface	ICalibrator4

DoneCalConfidenceCheckECAL Method

Description	Concludes the Confidence Check and sets the ECal module back into the idle state.
VB Syntax	*cal*.**DoneCalConfidenceCheckECAL**
Variable	**(Type) - Description**
cal	A Calibrator **(object)**
Return Type	None
Default	None
Examples	*cal*.DoneCalConfidenceCheckECAL
C++ Syntax	HRESULT DoneCalConfidenceCheckECAL( );
Interface	ICalibrator

Read-only

Execute Method

Description	This method can be used with SCPI command :SYST:ERR? to convert scpi errors into text.
VB Syntax	Scpi.**Execute**(SCPI_Command As String)
Variable	**(Type) - Description**
scpi	A ScpiStringParser **(Object)**
SCPI_Command	**(String)** - Any valid SCPI command
Return Type	String
Default	Not Applicable

Examples	`Dim scpi As ScpiStringParser`
	`Set scpi = app.ScpiStringParser`
	`scpi.Execute("SYST:PRES");`
	`ErrorString = scpi.Execute("SYST:ERROr?");`

C++ Syntax	Execute([in] BSTR SCPI_Command,[out,retval] BSTR * pQueryResponse);
Interface	IScpiStringParser2

ExecuteShortcut Method

Description	Executes a Macro (shortcut) stored in the analyzer. Use app.getShortcut to list existing macros. Use app.putShortcut to associate the macro number with the file.
VB Syntax	*app.***ExecuteShortcut** *index*
Variable	**(Type) - Description**
app	An Application **(object)**
index	**(long integer)** - Number of the macro stored in the analyzer.
Return Type	Not Applicable
Default	Not Applicable
Examples	`app.ExecuteShortcut 1`
C++ Syntax	HRESULT ExecuteShortcut(long index)
Interface	IApplication

Write-only

GenerateErrorTerms Method

Description	Generates the error terms for the specified calibration type, stores the error terms in a Cal Set, saves the Cal Set, and returns the Cal Set GUID. If ALL the data for the cal type has NOT been acquired an error message is returned.
VB Syntax	*value* = *obj*.**GenerateErrorTerms**
Variable	**(Type) - Description**
value	(String) - Variable to store the returned GUID or error message.
obj	Any of the following: GuidedCalibration (object) SMCType (object) VMCType (object)
Return Type	String
Default	Not Applicable
Examples	`string = SMC.GenerateErrorTerms`
C++ Syntax	HRESULT put_GenerateErrorTerms(BSTR* calsetGUID);
Interface	IGuidedCalibration SMCType VMCType

Write-only

GenerateSteps Method

Description	Returns the number of steps required to complete the calibration type.
VB Syntax	*value* = *obj*.**GenerateSteps**
Variable	**(Type) - Description**
value	(long) - Variable to store the returned number of steps
obj	Any of the following: GuidedCalibration (object) SMCType (object) VMCType (object)
Return Type	Long
Default	Not Applicable
Examples	`value = SMC.GenerateSteps`
C++ Syntax	HRESULT put_GenerateSteps(long* steps);
Interface	IGuidedCalibration SMCType VMCType

About the AuxIO connector

Read-only

GetAuxIO Method

Description	This method returns the IAuxIO interface.
VB Syntax	*app.***GetAuxIO**
Variable	**(Type) - Description**
app	An Application **(object)**
Return Type	IHWAuxIO
Default	Not Applicable

Example	`Dim app As AgilentPNA835x.Application`
	`Dim aux As HWAuxIO`
	`Set aux = app.GetAuxIO`

C++ Syntax	HRESULT GetAuxIO (IHWAuxIO **pAux);
Interface	IApplication

Write-only

GetCalStandard Method

Description	Returns a handle to a calibration standard for modifying its definitions. To select a standard for performing a calibration (use Calibrator.AquireCalStandard).
VB Syntax	*calkit.***GetCalStandard***(index)*
Variable	**(Type) - Description**
calkit	A calKit **(object)**
index	**(long) -** Number of calibration standard. Choose **1** to **30;** (there are 30 cal standards in every kit).
Return Type	calStandard
Default	Not Applicable

Examples	`Dim short As CalStandard`
	`Set short = calKit.getCalStandard(1)`
	`short.label = "myShort"`

C++ Syntax	HRESULT GetCalStandard(long standardNumber, ICalStandard **pCalStd)
Interface	ICalKit

GetCalManager Method

Description	This method returns the ICalManager interface.
VB Syntax	*app.***GetCalManager**
Variable	**(Type) - Description**
app	An Application **(object)**
Return Type	ICalManager*
Default	Not Applicable

Example

```
dim app as AgilentPNA835x.Application
dim mgr as CalManager
set mgr = app.GetCalManager
```

C++ Syntax	HRESULT GetCalManager(ICalManager **mgr);
Interface	IApplication

Get CalSetByGUID Method

Description	Requests a Cal Set by GUID. Returns an ICal Set interface.
VB Syntax	*calMgr.***GetCalSetByGUID** *(GUID)*
Variable	**(Type) - Description**
calMgr	**(object) -** A CalManager object
GUID	**(string)** - GUID of the Cal Set being requested.
Return Type	Interface object
Default	Not Applicable
Example	`calMgr.GetCalSetByGUID (2B893E7A-971A-11d5-8D6C-00108334AE96)`
C++ Syntax	HRESULT GetCalSetByGUID(BSTR* strGUID, ICal Set* pCalSet);
Interface	ICalManager

About Cal Sets

Read-only

GetCalSetCatalog Method

Description	Returns a string containing a list of comma-separated GUIDs in the following format: {FD6F863E-9719-11d5-8D6C-00108334AE96}, {1B03B2CE-971A-11d5-8D6C-00108334AE96}, {2B893E7A-971A-11d5-8D6C-00108334AE96}
VB Syntax	*value* = *calMgr*.**GetCalSetCatalog**
Variable	**(Type) - Description**
value	**(string)** - Variable to store the returned GUID list
calMgr	**(object) -** A CalManager object
Return Type	String
Default	Not Applicable
Example	`value = calMgr.GetCalSetCatalog`
C++ Syntax	HRESULT GetCalSetCatalog(BSTR);
Interface	ICalManager

GetCalSetUsageInfo Method

Description Returns a string identifying the Cal Set currently in use by the specified channel.

This method identifies the Cal Set being used by returning its GUID.

This method also identifies the "Error Term set" within the Cal Set.

Error term sets are identified by integers, with set 0 belonging to the original (non-interpolated) terms. As stimulus values for a channel are changed causing interpolation to be required, a new Error Term set is constructed within the Cal Set to hold the interpolated Error Terms. The sets are sequentially numbered 1, 2, 3, and so forth. These Error Term sets are destroyed when they are no longer being used.

If there is no Cal Set in use for the given channel, the <GUID> argument is set to the empty string.

VB Syntax *calMgr.***GetCalSetUsageInfo** (*chan, GUID, EtermID*)

Variable **(Type) - Description**

calMgr **(object) -** A CalManager object

chan **(long [in]) -** channel of the Cal Set being requested

GUID **(string [out])** - variable to store the GUID of the Cal Set being requested. If there is no Cal Set in use for the given channel, the <GUID> argument is set to the empty string.

EtermID **(long [out]) -** variable to store the error term ID being requested. If the returned argument is greater than 0, the set is being interpolated.

Return Type String , Long Integer

Default Not Applicable

Example

```
calMgr.GetCalSetUsageInfo (1, GUID, EtermID)
```

C++ Syntax HRESULT GetCalSetUsageInfo (long lChannel, BSTR* CalSetGUID, long* etermSetID);

Interface ICalManager

Read only

GetCalTypes

Description Returns a list of available calibration types, both Standard and Custom. The Standard CalTypes are the same on all PNA's, but the Custom CalTypes are not necessarily the same. They are dependent on the custom components that have been installed in the instrument.

VB Syntax v = CalManager.**GetCalTypes**

Variable **(Type) - Description**

v NameGuidPair: contains the calibration type name and associated GUID for each cal type known to the PNA

Return Type **(variant)** of type VT_ARRAY. The array is two dimensional containing elements of type variant.

Default Not Applicable

Examples

```
v =CalManager.GetCalTypes
```

C++ Syntax HRESULT GetCalTypes(VARIANT * NameGuidPair)

Interface ICalManager2

Read-only

GetComplex Method

Description Retrieves complex data from the specified location.
See also getNAComplex , getData , and getPairedData Methods

VB Syntax *measData*.**getComplex** *location, numPts, real(), imag()*

Variable **(Type) - Description**

measData An IArrayTransfer interface which supports the Measurement object

location **(enum NADataStore - IArrayTransfer)** - Where the data you want is residing. Choose from:

0 - naRawData

1 - naCorrectedData

2 - naMeasResult

3 - naRawMemory

4 - naMemoryResult

5 - naDivisor - When reading data from, or writing data to, the normalization divisor, you must first create a divisor trace using DataToDivisor Method.

See the Data Access Map

numPts **(long integer)** - Number of data points requested
[out] - specifies number of data elements returned
[in] - specifies the data being requested or the capacity of the arrays

real **(single)** - Array to store the real values

imag **(single)** - Array to store the imaginary values

Return Type Single

Default Not Applicable

Examples

```
Dim real(201) AS Single
Dim imag(201) AS Single
Dim pts as Integer
Dim measData As IArrayTransfer
Set measData = app.ActiveMeasurement
measData.getComplex naCorrectedData, pts, real(0), imag(0)
```

C++ Syntax IArrayTransfer - HRESULT getComplex(tagNADataStore DataStore, long* pNumValues, float* pReal, float* pImag)

Interface IArrayTransfer

getDataByString Method

Description Retrieves variant data from the specified location in your choice of formats.

VB Syntax *data* = *meas*.**getDataByString** *location, format*

Variable **(Type) - Description**

data **(variant)** - Array to store the data.

meas **(object) -** A Measurement object

location **(string)** – Name of the buffer to be read. Choose from:

"naRawData"

"naCorrectedData"

"naMeasResult"

"naRawMemory"

"naMemoryResult"

"naDivisor" - When reading data from, or writing data to, the normalization divisor, you must first create a divisor trace using DataToDivisor Method.

See the Data Access Map

format **(enum NADataFormat)** - Format in which you would like the data. It does not have to be the displayed format. Choose from:

0 - naDataFormat_LinMag

1 - naDataFormat_LogMag

2 - naDataFormat_Phase

3 - naDataFormat_Polar *

4 - naDataFormat_Smith *

5 - naDataFormat_Delay

6 - naDataFormat_Real

7 - naDataFormat_Imaginary

8 - naDataFormat_SWR

*** Specfiy Smith or Polar** formats to obtain complex data pairs, which require a two-dimensional array **varData (numpts, 2)** to accomodate both real and imaginary data.

All scalar formats return a single dimension **varData(numpts).**

Return Type Variant array

Default Not Applicable

Examples

```
meas.getDataByString “VectorResult0”, naDataFormat_Phase
```

C++ Syntax HRESULT getDataByString(BSTR location, tagDataFormat dataFormat, VARIANT * pData);

Interface IMeasurement

GetData Method

Description Retrieves variant data from the specified location in your choice of formats. To get smoothed data from any of the specified locations, the format must be the same as the displayed format.

Note: This method returns a variant which is less efficient than methods available on the IArrayTransfer interface.

Note: If you plan to **Put** this data back into analyzer, putDataComplex (variant data) method requires complex, two-dimensional data. Therefore, request the data in **Polar** format.

VB Syntax *data = meas.***GetData** *location, format*

Variable **(Type) - Description**

data Variant array to store the data.

meas A Measurement **(object)**

location **(enum NADataStore) -** Where the data you want is residing. Choose from:

0 - naRawData

1 - naCorrectedData

2 - naMeasResult

3 - naRawMemory

4 - naMemoryResult

5 - naDivisor When reading data from, or writing data to, the normalization divisor, you must first create a divisor trace using DataToDivisor Method.

See the Data Access Map

format **(enum NADataFormat) -** Format in which you would like the data. It does not have to be the displayed format. Choose from:

0 - naDataFormat_LinMag

1 - naDataFormat_LogMag

2 - naDataFormat_Phase

3 - naDataFormat_Polar *

4 - naDataFormat_Smith *

5 - naDataFormat_Delay

6 - naDataFormat_Real

7 - naDataFormat_Imaginary

8 - naDataFormat_SWR

*** Specfiy Smith or Polar** formats to obtain complex data pairs, which require a two-dimensional array **varData (numpts, 2)** to accommodate both real and imaginary data.

All scalar formats return a single dimension **varData(numpts).**

naDataFormat_Phase returns degrees. However, putDataScalar method accepts data in radians (not degrees) and displays in degrees.

Return Type Variant array - automatically dimensioned to the size of the data

Default Not Applicable

Examples

```
Dim varData As Variant
varData = meas.GetData(naMeasResult,naDataFormat_Phase)
'Print Data
For i = 0 to chan.NumberOfPoints-1
 Print varData(i)
Next i
```

C++ Syntax HRESULT getData(tagNADataStore DataStore, tagDataFormat DataFormat, VARIANT *pData)

Interface IMeasurement

Read-only

GetECALModuleInfoEx Method

Description This property replaces Get ECALModuleInfo Method.

Returns the following information about the connected ECAL module: model number, serial number, connector type, calibration date, min and max frequency.

The characterization within the ECal module that this information will be read from is specified by **ECALCharacterizationEx**. The default value is 0.

VB Syntax *moduleInfo* = *cal*.**GetECALModuleInfoEx** *(module)*

Variable **(Type) - Description**

moduleInfo **(string)** - variable to store the module information

cal A Calibrator **(object)**

module **(long integer)** ECal module.

Choose from modules **1** through **8**

Use IsECALModuleFoundEx to determine the number of modules connected to the PNA

Return Type String

Default Not Applicable

Examples

```
info = cal.GetECALModuleInfoEx(2)
```

Example return string:

```
ModelNumber: 85092-60007, SerialNumber: 01386, ConnectorType:
N5FN5F, PortAConnector: Type N (50) female, PortBConnector: Type
N (50) female, MinFreq: 30000, MaxFreq: 9100000000,
NumberOfPoints: 250, Calibrated: July 4 2002
```

C++ Syntax HRESULT GetECALModuleInfoEx(long moduleNumber, BSTR* info);

Interface ICalibrator4

Read-only

GetErrorTerm Method - Obsolete

Description

Note: This command is replaced by CalSet.getErrorTerm.

Retrieves error term data that is used for error correction. The data is complex pairs. Memory for the returned Variant is allocated by the server. The server returns a variant containing a two-dimensional safe Array.

This method returns a variant which is less efficient than getErrorTermComplex on the ICalData interface.

Learn about reading and writing Calibration data.

VB Syntax *data* = *cal*.**getErrorTerm** *term, rcv. src*

Variable **(Type) - Description**

data Variant array to store the data.

cal A Calibrator **(object)**

term **(enum As NaErrorTerm).** Choose from:

naErrorTerm_Directivity_Isolation
naErrorTerm_Match
naErrorTerm_Tracking

rcv **(long integer)** - Receiver Port

src **(long integer)** - Source Port

To get this Error Term	Specify these parameters: *term*	 *rcv*	 *src*
Fwd Directivity	naET_Directivity Isolation	1	1
Rev Directivity	naET_Directivity Isolation	2	2
Fwd Isolation	naET_Directivity Isolation	2	1
Rev Isolation	naET_Directivity Isolation	1	2
Fwd Source Match	naErrorTerm_Match	1	1
Rev Source Match	naErrorTerm_Match	2	2
Fwd Load Match	naErrorTerm_Match	2	1
Rev Load Match	naErrorTerm_Match	1	2
Fwd Reflection Tracking	naErrorTerm_Tracking	1	1
Rev Reflection Tracking	naErrorTerm_Tracking	2	2
Fwd Trans Tracking	naErrorTerm_Tracking	2	1
Rev Trans Tracking	naErrorTerm_Tracking	1	2

Return Type Variant

Default Not Applicable

Examples

```
Dim varError As Variant
varError = cal.getErrorTerm(naErrorTerm_Tracking,2,1)
```

C++ Syntax HRESULT getErrorTerm(tagNAErrorTerm ETerm, long ReceivePort, long SourcePort, VARIANT* pData)

Interface ICalibrator

GetErrorTerm Method

Description Queries data from the Cal Set that was acquired for a specific standard.

Learn more about reading and writing Cal Data using COM.

Before calling this method you must open the Cal Set with OpenCalSet.. If the Cal Set is not open, this method returns E_NA_Cal Set_ACCESS_DENIED.

The data is complex pairs. The server returns a variant containing a two-dimensional safe array. Memory for the returned Variant is allocated by the server and must be released by client.

Note: See also getErrorTermComplex on the ICalData2 interface to avoid using the variant data type.

VB Syntax *data* = *CalSet.***getErrorTerm** *setID, term, rcv, src*

Variable **(Type) - Description**

data Variant array to store the data.

CalSet A Cal Set **(object)**

setID (**long integer) –** specifies which error term set to read data from. (0 is the master set of eterms.)

To get data from interpolated error terms, you must first determine if an interpolated set exists by calling the GetCalSetUsageInfo method. If it returns a number greater than 0 for the error term set ID, then the channel is currently using interpolated arrays. In this case, you can read from either the interpolated array (setID > 0) or the master array (setID = 0).

Note::Interpolated error terms are destroyed when no longer being used.

term **(enum As NaErrorTerm2).** Choose from:

0 - naET_Directivity (rcv = src)

1 - naET_SourceMatch (rcv = src)

2 - naET_ReflectionTracking (rcv = src)

3 - naET_TransmissionTracking (rcv != src)

4 - naET_LoadMatch (rcv != src)

5 - naET_Isolation (rcv != src)

rcv **(long integer)** - Receiver Port

src **(long integer)** - Source Port

Return Type Variant

Default Not Applicable

Examples

```
Dim varError As Variant
varError = CalSet.getErrorTerm(0,naET_TransmissionTracking,2,1)
```

C++ Syntax HRESULT getErrorTerm(long setID, tagNAErrorTerm2 ETerm, long ReceivePort, long SourcePort, VARIANT* pData)

Interface ICalSet

Read-only

GetErrorTermByString

Description	Returns the data array for a specific error term from the calset
VB Syntax	*pdata* **=** *Calset***.GetErrorTermByString**(*SetNumber*, *bufferName*)
Variable	**(Type) - Description**
calset	A Calset **(object)**
SetNumber	**(long integer)** SetNumber = 0 is the master set of error terms for this calset. Interpolated error terms reside in sets > 0. To determine what set number is being used by a channel, see **ICalManager::GetCalSetUsage**.
bufferName	**(String)** The string used to identify a particular buffer in the calset. An example string for port 3 directivity in a full 2 port cal might be "Full 2 Port Cal (2,3)::Directivity(3,3)".
pdata	**(Variant)** The pdata array is a variant. This data is usually two dimensional. Each element of the variant array is a 2 array of type single. The two elements represent the real and imaginary parts of a complex datum. **Note:** This structure is compatible with scripting clients who can only use variants. For alternative methods that use typed arrays, see ICalData3.
Return Type	Variant
Default	Not Applicable
Examples	See an example
C++ Syntax	HRESULT GetErrorTermByString (long SetNumber, BSTR bufferName, VARIANT* pdata);
Interface	ICalSet2

Read-only

GetErrorTermComplex Method

Description	Retrieves error term data from the error correction buffer. The data is in complex pairs. Learn more about reading and writing Cal Data using COM. **Note:** This method exists on a non-default interface. If you cannot access this method, use the GetErrorTerm Method on ICalibrator.
VB Syntax	*eData.***GetErrorTermComplex** *term, rcv, src, numPts, real(), imag()*
Variable	**(Type) - Description**
eData	An ICalData pointer to the Calibrator object
term	**(enum NAErrorTerm) -** The error term to be retrieved. Choose from: • **naErrorTerm_Directivity_Isolation** • **naErrorTerm_Match** • **naErrorTerm_Tracking**
rcv	**(long integer)** - Receiver Port
src	**(long integer)** - Source Port
numPts	**(long integer) -** on input, max number of data points to return; on output: indicates the actual number of data points returned.
real()	**(single)** - array to accept the **real** part of the error-term. One-dimensional for the number of data points.
imag()	**(single)** - array to accept the **imaginary** part of the error-term. One-dimensional for the number of data points.

To get this Error Term	**Specify these parameters:** *term*	*rcv*	*src*
Fwd Directivity	naET_Directivity Isolation	1	1
Rev Directivity	naET_Directivity Isolation	2	2
Fwd Isolation	naET_Directivity Isolation	2	1
Rev Isolation	naET_Directivity Isolation	1	2
Fwd Source Match	naErrorTerm_Match	1	1
Rev Source Match	naErrorTerm_Match	2	2
Fwd Load Match	naErrorTerm_Match	2	1
Rev Load Match	naErrorTerm_Match	1	2
Fwd Reflection Tracking	naErrorTerm_Tracking	1	1
Rev Reflection Tracking	naErrorTerm_Tracking	2	2

Fwd Trans Tracking	naErrorTerm_Tracking	2	1
Rev Trans Tracking	naErrorTerm_Tracking	1	2

Return Type Single

Default Not Applicable

Examples

```
ReDim rel(numpts)
ReDim img(numpts)
Dim eData As ICalData
Set eData = chan.Calibrator
eData.getErrorTermComplex naErrorTerm_Directivity_Isolation, 1,
1, 201, rel(0), img(0)
```

C++ Syntax HRESULT raw_getErrorTermComplex(tagNAErrorTerm ETerm, long ReceivePort, long SourcePort, long* pNumValues, float* pReal, float* pImag)

Interface ICalData

GetErrorTermComplex Method

Description Queries error term data from the Cal Set. The data is in complex pairs. Learn more about reading and writing Cal Data using COM.

Note: This method exists on a non-default interface. If you cannot access this method, use the GetErrorTerm Method on ICal Set.

VB Syntax *eData.***GetErrorTermComplex** *setID, term, rcv, src, numPts, real(), imag()*

Variable **(Type) - Description**

eData An ICalData2 pointer to the Cal Set object

setID (**long integer) –** specifies which error term set to read data from. (0 is the master set of eterms.)

To get data from interpolated error terms, you must first determine if an interpolated set exists by calling the GetCalSetUsageInfo method. If it returns a number greater than 0 for the error term set ID, then the channel is currently using interpolated arrays. In this case, you can read from either the interpolated array (setID > 0) or the master array (setID = 0).

Note::Interpolated error terms are destroyed when no longer being used.

term **(enum NAErrorTerm2)** - The error term to be retrieved. Choose from:

0 - naET_Directivity

1 - naET_SourceMatch

2 - naET_ReflectionTracking

3 - naET_TransmissionTracking

4 - naET_LoadMatch

5 - naET_Isolation

rcv **(long integer)** - Receiver Port

src **(long integer)** - Source Port

numPts **(long integer)** - on input, max number of data points to return;
on output: indicates the actual number of data points returned.

real() **(single)** - array to accept the **real** part of the error-term. One-dimensional for the number of data points.

imag() **(single)** - array to accept the **imaginary** part of the error-term. One-dimensional for the number of data points.

Return Type Single

Default Not Applicable

Examples

```
dim numpts as long
numpts = ActiveChannel.NumberOfPoints
ReDim r(numpts)  ' real part
ReDim i(numpts)  ' imaginary part
Dim CalSet as CalSet
set CalSet = pna.GetCalManager.GetCal SetByGUID( txtGUID )
Dim eData As ICalData2
Set eData = CalSet
eData.getErrorTermComplex 0, naET_LoadMatch, 1, 2, numpts,
r(0),i (0)
```

C++ Syntax HRESULT getErrorTermComplex(long setID, tagNAErrorTerm2 ETerm, long ReceivePort, long SourcePort, long* pNumValues, float* pReal, float* pImag)

Interface ICalData2

Read-only

GetErrorTermComplexByString

Description	Returns data for a specific error term.
VB Syntax	*ICalData3.***GetErrorTermComplexByString**(*etermSetID, bufferName, lnumPoints, rea(0)l, imag(0)*)
Variable	**(Type) - Description**
ICalData3	An ICalData3 pointer to a CalSet (Object)
etermSetID	(Long) SetNumber = 0 is the master set of error terms for this Cal Set. Interpolated error terms reside in sets > 0. To determine what set number is being used by a channel, see ICalManager.GetCalSetUsage.
bufferName	**(String)** The string used to identify a particular buffer in the Cal Set. An example string for port 3 directivity in a full 2 port cal might be "Full 2 Port Cal (2,3)::Directivity(3,3)".
lnumpoints	**[In] (long)** The size of the float arrays in arguments"real" & "imag". **[Out] (long)** The number of actual data buckets returned in the real and imag arrays. see note
real	**(Single)** The real component of the complex data.
imag	**(Single)** The imaginary component of the complex data.
Return Type	Single
Default	Not Applicable
Examples	See example
C++ Syntax	HRESULT PutErrorTermComplexByString(BSTR bufferName, long lnumPoints, float* real, float* imag);
Interface	ICalData3

Note: *lnumpoints* is an In/Out parameter.

On the way **in**, it indicates the **max** number of values the user is requesting. This is critical because the capacity of the "read" & "imag" arrays must be allocated by the user to hold at least this number of elements. On the way **out,** it indicates the values actually returned.

GetErrorTermList Method

Description Returns the list of Error Terms contained in this Cal Set for the CalType specified in the OpenCal Set method. Learn more about reading and writing Cal Data using COM.

The list is a comma separated, textual representation of the error terms with the term name followed by the port path in parentheses:

Term (n, n),
Term (m,n)

Before calling this method you must open the Cal Set with OpenCal Set.. If the Cal set is not open, this method returns E_NA_Cal Set_ACCESS_DENIED.

Use StringToNAErrorTerm2 to convert the list entrees to values that can be used with GetErrorTerm and PutErrorTerm.

Note: The port path designation (m n) indicates the ports that contribute to the error being compensated. Directivity, source match and reflection tracking are single port characteristics, designated in this list by (n n) where n equals the port being characterized.
Other terms characterize the interaction between ports. For example, the load match term is describing the match at port (m) while looking into port (n). Thus the notation (m n) indicates the two ports that contribute to the loadmatch error.

VB Syntax *CalSet*.**GetErrorTermList** (SetID, count, strList)

Variable **(Type) - Description**

CalSet **(object) -** A Cal Set object

SetID (long) - specifies the error term set to query. Use 0 for the master set.

count (long) - the number of error terms in the returned list

strList **(string) -** comma separated list of error terms found in Cal Set

Return Type Not Applicable

Default Not Applicable

Examples

```
dim count as Integer
dim list as string
OpenCalSet (naCalType_TwoPortSOLT 1, 2)
GetErrorTermList( 0, count, list)
CloseCalSet( )
```

Assuming the cal set contained the full set of error terms for this two-port Cal, the returned list would be:

```
"Directivity(1 1),SourceMatch(1 1),ReflectionTracking(1
1),TransmissionTracking(2 1),LoadMatch(2 1),Isolation(2
1),Directivity(2 2),SourceMatch(2 2),ReflectionTracking(2
2),TransmissionTracking(1 2),LoadMatch(1 2),Isolation(1 2)"
```

C++ Syntax HRESULT GetErrorTermList (long etermSetID, long* count, BSTR* strList);

Interface ICalSet

Read-only

GetErrorTermList2

Description	Returns a list of error terms names found in the calset, containing the specified prefix.
VB Syntax	**GetErrorTermList2**(*SetNumber*, *calTypePrefix*)
Variable	**(Type) - Description**
SetNumber	**(Long)** SetNumber = 0 is the master set of error terms for this caltype, in this calset. Interpolated error terms reside in sets > 0. To determine what set number is being used, see **ICalManager::GetCalSetUsage.**
caltypePrefix	**(String)** The string used to identify calset data as belonging to a specific caltype. (By convention, calset buffers are named "CalType:Item".) An example prefix for a two port cal on ports 2 and 3 might be: "Full 2 Port Cal (2,3)". If the prefix is empty, all terms are returned. This string is used as a filter so that only the buffer names of interest are returned.
list	**(Variant)** Variant containing a string array of error term names
Return Type	Variant
Default	Not Applicable
Examples	See example
C++ Syntax	HRESULT GetErrorTermList2 (long SetNumber, BSTR caltypePrefix, VARIANT* list**)**
Interface	ICalSet2

Get ExternalTestSetIO Method

Description This method returns the IExternalTestSetIO interface.

VB Syntax *app.***GetExternalTestSetIO**

Variable **(Type) - Description**

app An Application **(object)**

Return Type IHWExternalTestSetIO

Default Not Applicable

Example

```
Dim app As AgilentPNA835x.Application
Dim ets As HWExternalTestSetIO
Set ets = app.GetExternalTestSetIO
```

C++ Syntax HRESULT GetExternalTestSetIO (IHWExternalTestSetIO **ptestset);

Interface IApplication

About Marker Search

Read-only

GetFilterStatistics Method

Description	Returns all four Filter Statistics resulting from a SearchFilterBandwidth. To retrieve individual filter statistics, use meas.FilterCF, meas.FilterBW, meas.FilterLoss, meas.FilterQ properties.
VB Syntax	*meas.***GetFilterStatistics** *cf,bw,loss,q*
Variable	**(Type) - Description**
meas	A Measurement **(object)**
cf,bw,loss,q	Dimensioned variables to store the returned values
Return Type	**(double)** *cf* **(single)** *bw,loss,q*
Default	Not Applicable

Examples

```
'Dimension variables
Dim cf as Double
Dim bw as Single
Dim loss as Single
Dim q as Single

meas.GetFilterStatistics cf,bw,loss,q
```

C++ Syntax	HRESULT GetFilterStatistics(double* centerFreq, float* bw, float* loss, float* quality)
Interface	IMeasurement

GetGuid Method

Description	Returns a string containing the GUID identifying this Cal Set. Each Cal Set is assigned a GUID (global unique ID). GUIDs are used to retrieve and select Cal Sets on the PNA. Learn more about reading and writing Cal Data using COM.
VB Syntax	*value* = *CalSet*.**GetGuid**
Variable	**(Type) - Description**
value	**(string)** - Variable to store the returned GUID
CalSet	**(object) -** A Cal Set object
Return Type	String
Default	Not Applicable
Examples	`guid = CalSet.GetGuid 'Read`
C++ Syntax	HRESULT GetGUID(BSTR* pGUIDString);
Interface	ICalSet

get_InputVoltage Method

Description Reads the ADC input voltage from Analog IN (pin 14) of the AUX IO connector

VB Syntax *volts* = *AuxIO*.**get_InputVoltage**

Variable **(Type) - Description**

volts **(double) -** variable to store the return value

AuxIO **(object) -** A Hardware Auxiliary Input / Output object

Return Type Double

Default 0

Examples

```
Dim aux as HWAuxIO
Set aux = PNA.getAuxIO
volts = aux.get_InputVoltage 'read voltage on Analog In
(pin 14)
```

C++ Syntax HRESULT get_InputVoltage (double* Voltage);

Interface HWAuxIO

get_Input1 Method

Description Reads a hardware latch that captures high to low transitions on Input1 of the Material Handler IO. Reading the latch causes it to reset and is ready for the next transition. The hardware latch is only capable of capturing one transition per query. Additional transitions are ignored until after the next query.

Momentarily driving Input1 high, then low, causes a transition to be detected and latched.

VB Syntax *inp1* = *handlerIo*.**get_Input1**

Variable **(Type) - Description**

inp1 **(variant) -** A variable to store the return value

handlerIo **(object)** - A HandlerIO object

Return Type Variant -

0 - a high to low transition occurred at Input1 since the last time it was queried.

1 - no high to low transition occurred.

Default Not Applicable

Examples

```
input1 = handlerIo.get_Input1 'Read
```

C++ Syntax HRESULT get_Input1 (VARIANT* Data);

Interface IHWMaterialHandlerIO

About the MaterialHandler connector

Read-only

Get MaterialHandlerIO Method

Description This method returns the MaterialHandlerIO interface.

VB Syntax *app.***GetMaterialHandlerIO**

Variable **(Type) - Description**

app An Application **(object)**

Return Type IHWMaterialHandlerIO

Default Not Applicable

Example

```
Dim app As AgilentPNA835x.Application
Dim hand As HWMaterialHandlerIO
Set hand = app.GetMaterialHandlerIO
```

C++ Syntax HRESULT GetMaterialHandlerIO (IHWMaterialHandlerIO **phand);

Interface IApplication

GetNAComplex Method

Description Retrieves complex data from the specified location.
See also getComplex and getData Method.

VB Syntax *measData*.**getNAComplex** *location, numPts, data*

Variable **(Type) - Description**

measData An IArrayTransfer interface which supports the Measurement object

location **(enum NADataStore)** - Where the data you want is residing. Choose from:

0 - naRawData

1 - naCorrectedData

2 - naMeasResult

3 - naRawMemory

4 - naMemoryResult

5 - naDivisor - When reading data from, or writing data to, the normalization divisor, you must first create a divisor trace using DataToDivisor Method.

See the Data Access Map

numPts **(long integer)** - Number of data points requested
[out] - specifies number of data elements returned
[in] - specifies the data being requested or the capacity of the *dComplex* array

data **(NAComplex)** - A one-dimensional array of NaComplex to store the data.

Return Type NAComplex

Default Not Applicable

Examples

```
Dim dComplex(201) AS NaComplex
Dim measData As IArrayTransfer
Dim pts as Long
Set measData = app.ActiveMeasurement
measData.getNAComplex naCorrectedData, pts, dComplex(0)
```

Notes The data is stored as Real and Imaginary (**Re** and **Im**) members of the NaComplex user defined type. You can access each number individually by iterating through the array.

```
For i = 0 to NumPts-1
 dReal (i) = dcomplex (i).Re
 dImag (i) = dcomplex (i).Im
Next i
```

C++ Syntax HRESULT getNAComplex(tagNADataStore DataStore, long* pNumValues, TsComplex* pComplex)

Interface IArrayTransfer

GetNumberOfGroups Method

Description	Returns the number of groups a channel has yet to acquire. To set the number of groups for a channel, use Number Of Groups Method
VB Syntax	*value* = *chan*.**GetNumberOfGroups**
Variable	**(Type) - Description**
value	**(Long Integer)** - Number of groups
chan	Channel **(object)**
Return Type	**(Long Integer)**
Default	Not Applicable
Examples	`groups = chan.GetNumberOfGroups 'Read`
C++ Syntax	HRESULT GetNumberOfGroups(long* numberOfGroups);
Interface	IChannel3

get_Output Method

Description Type 1 and Type2 configurations: Returns the last value written to the selected output pin.

Type3 configuration: Returns the current state of the selected output pin. If an Input1 trigger occurs, the state may not be the same value as was written.

All configurations: Data is written using put_Output Method.

VB Syntax *data* = *handlerIo*.**get_Output** *(pin)*

Variable **(Type) - Description**

data **(variant) -** A variable to store the return value. The returned value will be one of the following:

0 - TTL Low

1 - TTL High

handlerIo **(object)** - A HandlerIO object

pin **(enum as NAMatHandlerOutput)** - output to read. Choose from:

naOutput1 (0)
naOutput1User (1)
naOutput2 (2)
naOutput2User (3)

Learn about User Output

Return Type Variant

Default Not Applicable

Examples

```
data = handlerIo.get_Output(naOutput1)
```

C++ Syntax HRESULT get_Output (tagNAMatHandlerOutput Output, VARIANT* Data);

Interface IHWMaterialHandlerIO

About the Aux I/O Connector

Read-only

get_OutputVoltage Method

Description Reads voltages on the DAC/Analog Output 1 and Output 2 (pins 2 and 3 of the Aux I/O connector)

VB Syntax *volts* = *AuxIO*.**get_OutputVoltage** *(output)*

Variable **(Type) - Description**

volts **(double) -** variable to store the return value

AuxIO **(object) -** A Hardware Auxiliary Input / Output object

output **(variant)** Number of the output DAC to read voltage from. Choose from:

1 - Output DAC 1 -(pin 3)

2 - Output DAC 2 -(pin 2)

Return Type Double

Default Not Applicable

Examples

```
Dim aux as HWAuxIO
Set aux = PNA.getAuxIO
volts = aux.get_OutputVoltage(1) 'read voltage from Analog Out 1
(pin3)
```

C++ Syntax HRESULT get_OutputVoltage(VARIANT Output, double* Voltage);

Interface IHWAuxIO

Read-only

get OutputVoltageMode Method

Description This command returns the mode of the selected "Analog Out" line on the Auxiliary IO connector. The modes give the user the option to have the requested voltage applied immediately or not until the sweep is done. To set the mode, use put_OutputVoltageMode Method.

VB Syntax *mode* = *auxIo*.**get_OutputVoltageMode** *(output)*

Variable **(Type) - Description**

mode **(enum NAOutputVoltageMode) -** variable to store the returned mode.

naWaitEOS - While in this mode any voltage changes sent to the selected analog out will only get applied to the output between sweeps.

naNoWait - While in this mode any voltage changes sent to the selected analog out will occur right away without waiting until the end of a sweep, the voltage gets applied immediately.

auxIo **(object) -** A Hardware Auxiliary Input / Output object

output (double) Analog Output. Choose from **1** or **2**

Return Type enum as NAOutputVoltageMode

Default naWaitEOS

Examples

```
vOutMode = auxIo.get_OutputVoltageMode (1)
```

C++ Syntax HRESULT get_OutputVoltageMode(VARIANT Output, tagNAOutputVoltageMode* pMode);

Interface IHWAuxIO

GetPairedData Method

Description Retrieves pairs of data from the specified location.

Note: This method exists on a non-default interface. If you cannot access this method, use the Get Data Method on IMeasurement.

VB Syntax *measData*.**getPairedData** *location, format, numPts, d1, d2*

Variable **(Type) - Description**

measData An IArrayTransfer interface which supports the Measurement object

location **(enum NADataStore)** - Where the data you want is residing. Choose from:

0 - naRawData

1 - naCorrectedData

2 - naMeasResult

3 - naRawMemory

4 - naMemoryResult

5 - naDivisor - When reading data from, or writing data to, the normalization divisor, you must first create a divisor trace using DataToDivisor Method.

See the Data Access Map

format **(enum NAPairedDataFormat)** - Format in which you would like the Paired data. Choose from:

- **naLinMagPhase** - Linear magnitude and phase
- **naLogMagPhase** - Log magnitude and phase
- **naRealImaginary** - Real and Imaginary

Note: Selecting **naRealImaginary** format is the same as using the getComplex method

numPts **(long integer)** - Number of data points requested
[out] - specifies number of data elements returned
[in] - specifies the data being requested or the capacity of the *dPaired* array

d1 **(single)** - Array to store the magnitude / real values

d2 **(single)** - Array to store the phase / imaginary values

Return Type Two Single arrays

Default Not Applicable

Examples

```
Dim logm() As Single
Dim phase() As Single
Public measData As IArrayTransfer
Set measData = app.ActiveMeasurement
Dim numpts As Long
numPoints = app.ActiveChannel.NumberOfPoints
ReDim logm(numPoints)
ReDim phase(numPoints)

measData.getPairedData naCorrectedData, naLogMagPhase,
numPoints, logm(0), phase(0)

Print values(0), values(1)
```

C++ Syntax HRESULT getPairedData(tagNADataStore DataStore, tagNAPairedDataFormat PairFormat, long* pNumValues, float* pReal, float* pImag)

Interface IArrayTransfer

Read-only

get_Port Method

Description Returns the value from the specified "readable" port.

VB Syntax *data* = *handlerIo*.**get_Port (port)**

Variable **(Type) - Description**

data **(variant) -** A variable to store the return value. The following table shows what the returned data represents:

Port	MSB...LSB 8...0
C	C3...C0
D	D3...D0
E	D3...D0 + C3...C0

handlerIo **(object)** - A HandlerIO object

port **(enum as NAMatHandlerPort)** - port to get data from. Choose from:

naPortC - (2)

naPortD - (3)

naPortE - (4)

Note: Reading data from the Write-only ports (A,B,F,G,H) will return an error. Ports C and D must be put in Read mode before reading from C, D, or E using PortMode Property.

Return Type Variant

Default 0

Examples

```
data = handlerIo.get_Port(naPortC)
```

C++ Syntax HRESULT get_Port (tagNAMatHandlerPort Port, VARIANT* Data);

Interface IHWMaterialHandlerIO

get_PortCData Method

Description Reads a 4-bit value from Port C of the Aux I/O connector (pins 22-25) and the Material Handler IO (pins 21-24 Anritsu) - (pins 22-25 Avantest).

Note: These lines are connected to both the Handler IO and Aux IO in the PNA.

VB Syntax *value* = *AuxIO*.**get_PortCData**

Variable **(Type) - Description**

value **(variant)** - Variable to store the returned data

AuxIO **(object) -** A Hardware Auxiliary Input / Output object

Return Type Integer

Default None

Examples

```
value = auxIo.get_PortCData 'Reading a value of 15 when in Positive Logic indicates Port C lines C0, C1, C2, C3 are High. If in Negative Logic they are Low.
```

C++ Syntax HRESULT get_PortCData(VARIANT* Data);

Interface IHWAuxIO

GetReferenceMarker Method

Description	Returns a handle to the reference marker.
VB Syntax	*meas*.**GetReferenceMarker**
Variable	**(Type) - Description**
meas	A Measurement **(object)**
Return Type	Object
Default	Not Applicable
Examples	`meas.GetReferenceMarker`
C++ Syntax	HRESULT GetReferenceMarker(IMarker** refMarker)
Interface	IMeasurement

Write-only

GetRequiredEtermNames

Description Returns an array of strings specifying the error terms required by the caltype's correction algorithm in order to correct the specified parameter.

This function interrogates a specific caltype (caltypeGUID) for the list of error terms it would need in order to correct the specified parameter. All the standard S Parameter calibration types embed port specifiers in the error term name. The specific port information is gleaned from the passed parameter. For example, to query the error term requirements specific to a two port cal on ports 1 and 3, issue this with a parameter of S13 or S31. The buffer names returned will be formatted in this way:

Full 1 Port SOLT(1,3):TransmissionTracking(3,1)

VB Syntax **EtermNames = GetRequiredEtermNames**(CalTypeGUID As String, Parameter As String)

Variable **(Type) - Description**

caltypeGUID: [in] the GUID of the desired calibration type

parameter [in] string specifying the parameter to be corrected

EtermNames [out] array of strings containing the error term names.

Note: In C++ Allocated by server. Must be freed by caller using SysFreeString.

Return Type Not Applicable

Default Not Applicable

Examples enames = GetRequiredEtermNames(ctGUID, Parm)

C++ Syntax HRESULT GetRequiredEtermNames(BSTR caltypeGUID, BSTR parameter, VARIANT* EtermNames)

Interface ICalManager2

GetScalar Method

Description Retrieves scalar data from the specified locations.

Note: This method exists on a non-default interface. If you cannot access this method, use the Get Data Method on IMeasurement.

Note: You can **NOT** use this command to get complex data.

VB Syntax *measData*.**getScalar** *location, format, numPts, data*

Variable **(Type) - Description**

measData An IArrayTransfer interface which supports the Measurement object

location **(enum NADataStore)** - Where the data you want is residing. Choose from:

0 - naRawData

1 - naCorrectedData

2 - naMeasResult

3 - naRawMemory

4 - naMemoryResult

5 - naDivisor - When reading data from, or writing data to, the normalization divisor, you must first create a divisor trace using DataToDivisor Method.

See the Data Access Map

format **(enum DataFormat)** - Scalar format in which you would like the data. Choose from:

- naDataFormat_Delay
- naDataFormat_Imaginary
- naDataFormat_LinMag
- naDataFormat_LogMag
- naDataFormat_Phase
- naDataFormat_Real
- naDataFormat_SWR

numPts **(long integer)** - Number of data points requested
[out] - specifies number of data elements returned
[in] - specifies the data being requested or the capacity of the *dScalar* array

data **(single)** - Array to store the scalar data.

Return Type Single

Default Not Applicable

Examples

```
Dim dScalar() As Single
Dim measData As IArrayTransfer
Set measData = app.ActiveMeasurement
Dim numpts as Long
numpts = app.ActiveChannel.NumberOfPoints
ReDim dScalar(numPoints)

measData.getScalar naCorrectedData, naDataFormat_LogMag, numpts,
dScalar(0)
Print dScalar(0), dScalar(1)
```

C++ Syntax HRESULT getScalar(tagNADataStore DataStore, tagDataFormat DataFormat, long* pNumValues, float* pVals)

Interface IArrayTransfer

Read-only

GetShortcut Method

Description Returns the Title, Path, and optional argument strings, of the specified Macro (shortcut). Use this method to list the titles and paths of macros in the analyzer.

VB Syntax *app.***GetShortcut** *index, title, path, arguments*

Variable **(Type) - Description**

app An Application **(object)**

index **(long) -** Number of the macro. Use a number between **1** and **12.**

title **(string)** - **Title** of the specified macro. (Appears in the softkey label)

path **(string)** - **Pathname** of the specified macro.

arguments **(string) -** Arguments for the specified macro

Return Type String

Default Not Applicable

Example

```
Dim t As String
Dim p As String
Dim arg As String
Dim i As Integer
For i = 1 to 12
 app.GetShortcut i,t,p,arg
 Print t,p
Next
```

C++ Syntax HRESULT GetShortcut(long Number, BSTR* title, BSTR* pathname, BSTR* arguments)

Interface IApplication

Remarks Shortcuts can also be defined and accessed using the macro key on the front panel. However, the benefit of this feature is primarily for the interactive user

GetSnPData Method

Description Reads SnP data from the selected measurement. Learn more about SnP that is returned from the PNA.

VB Syntax *data* = *meas*.**GetSnPData** *type*

Variable (Type) - **Description**

data Variant array to store the data.

meas A Measurement **(object)**

type **(string)** - Type of SnP data to return. If unspecified, <n> is set to 2. Choose from:

"S1P" returns data for the active measurement.

"S2P" returns data for the current 2-port measurement (4 S-parameters).

"S3P" returns data for the current 3 port measurement (9 S-parameters). Valid only on instruments with more than 2 ports.

SnP data can be output using several data formatting options. See SnPFormat Property

Return Type Variant array - automatically dimensioned to the size of the data

Default Not Applicable

Examples

```
Dim pna As AgilentPNA835x.Application
Dim meas As AgilentPNA835x.IMeasurement3
Dim snp
Set pna = CreateObject("AgilentPnA835x.application")
Set meas = pna.ActiveMeasurement
snp = meas.GetSnPData("s2p")

Dim param
Dim point

' snp returns a 3 dimensional array
' dimension 1 is the parameter, s11, s21, s12, s22
' snp(param,point,pair)
'--------------------------------------
For param = LBound(snp, 1) To UBound(snp, 1)
  Debug.Print "Parameter: " & (param + 1)
  For point = LBound(snp, 2) To UBound(snp, 2)
  Debug.Print "  Point: " & (point + 1)
    Debug.Print "   Real", snp(param, point, 0)    ' real
element
    Debug.Print "   Imag", snp(param, point, 1) 'imaginary
element
  Next
Next
```

C++ Syntax HRESULT GetSnPData(BSTR snptype, VARIANT * response)

Interface IMeasurement3

Read-only

getSourcePowerCalDataEx Method

Description

Note: This method replaces getSourcePowerCalData Method

Retrieves (as variant data type) source power calibration data, if it exists, from the channel.

If the channel is sweeping the source backwards, then the first data point is the highest frequency value; the last data point is the lowest. Use the Get X-Axis Values command to return the X-axis values in the displayed order.

Note: This method returns a variant which is less efficient than methods available on the ISourcePowerCalData interface

VB Syntax *data* = *chan*.**getSourcePowerCalData** (*buffer, sourcePort)*

Variable **(Type) - Description**

data **(variant)** – Array to store the data.

chan **(object)** – A Channel object

buffer **(enum NASourcePowerCalBuffer) -** The requested source power cal data buffer.

0 - naCorrectionValues This is the only buffer currently available

sourcePort **(long integer)** – The source port for which calibration data is being requested.

Return Type Variant array – automatically dimensioned to the size of the data.

Default Not Applicable

Examples

```
Dim varData As Variant
Const port1 As Long = 1
varData = chan.getSourcePowerCalDataEx (naCorrectionValues,
port1)
'Print the data
For i = 0 to chan.NumberOfPoints - 1
Print varData(i)
Next i
```

C++ Syntax HRESULT getSourcePowerCalDataEx(tagNASourcePowerCalBuffer bufSelect, long sourcePort, VARIANT *pData);

Interface IChannel4

Read-only

getSourcePowerCalDataScalarEx Method

Description

Note: This method replaces getSourcePowerCalDataScalar Method

Retrieves (as scalar values) source power calibration data, if it exists, from this channel.

If the channel is sweeping the source backwards, then the first data point is the highest frequency value; the last data point is the lowest. Use the Get X-Axis Values2 command to return the X-axis values in the displayed order.

Note: This method exists on a non-default interface. If you cannot access this method, use the getSourcePowerCalDataEx Method on IChannel4.

VB Syntax *chanData*.**getSourcePowerCalDataScalar** *buffer, sourcePort, numValues, data*

Variable **(Type) - Description**

chanData **(interface)** – An ISourcePowerCalData2 interface on the Channel object.

buffer **(enum NASourcePowerCalBuffer)** - The requested source power cal data buffer.

0 - naCorrectionValues Currently this is the only buffer available.

sourcePort **(long integer)** – The source port for which calibration data is being requested.

numValues **(long integer)** – Number of data values.

[out] – specifies number of data values returned.

[in] – specifies number of values being requested (this must not be larger than the capacity of the data array).

data **(single)** – Array to store the data.

Return Type Single

Default Not Applicable

Examples

```
Dim numValues As Long
Dim scalarCalValues() As Single
Dim chanData As ISourcePowerCalData2
Const port1 As Long = 1
numValues = app.ActiveChannel.NumberOfPoints
ReDim scalarCalValues(numValues)
Set chanData = app.ActiveChannel

chanData.getSourcePowerCalDataScalarEx naCorrectionValues,
port1, numValues, scalarCalValues(0)

'Print the data
For i = 0 to numValues - 1
Print scalarCalValues(i)
Next I
```

C++ Syntax HRESULT getSourcePowerCalDataScalarEx(tagNASourcePowerCalBuffer bufSelect, long sourcePort, long *pNumValues, float *pData);

Interface ISourcePowerCalData2

Read-only

GetStandard Method

Description Queries data from the Cal Set that was acquired for a specific standard. Learn more about reading and writing Cal Data using COM.

Before calling this method you must open the Cal Set with OpenCal Set.. If the Cal Set is not open, this method returns E_NA_Cal Set_ACCESS_DENIED.

The data is complex pairs. The server returns a variant containing a two-dimensional safe array. Memory for the returned Variant is allocated by the server and must be released by client.

Note: See also getStandardComplex on the ICalData2 interface to avoid using the variant data type.

VB Syntax *data = CalSet.***getStandard** *standard, rcv , src*

Variable **(Type) - Description**

data Variant array to store the data.

CalSet A **Cal Set (object)**

standard **(enum NACalClass)** Standard to be measured. Choose from:

1 - naClassA

2 - naClassB

3 - naClassC

4 - naClassD

5 - naClassE

6 - naReferenceRatioLine

7 - naReferenceRatioThru

SOLT Standards

1 - naSOLT_Open

2 - naSOLT_Short

3 - naSOLT_Load

4 - naSOLT_Thru

5 - naSOLT_Isolation

TRL Standards

1 - naTRL_Reflection

2 - naTRL_Line_Reflection

3 - naTRL_Line_Tracking

4 - naTRL_Thru

5 - naTRL_Isolation

rcv **(long integer)** - Receiver Port

src **(long integer)** - Source Port

Return Type **(variant)** - two-dimensional array (0:1, 0:NumberOfPoints-1)

Default Not Applicable

Examples

```
Dim varStd As Variant
Dim varStd2 As Variant

Cal Set.OpenCalSet( naCalType_TwoPortSOLT, 1, 2)
varStd = CalSet.getStandard(naSOLT_Thru,2,1)
varStd2 = Cal Set.getStandard(naSOLT_Thru,1,2)
Cal Set.CloseCalSet( )
```

C++ Syntax HRESULT getStandard(tagNACalClass stdclass, long ReceivePort, long SourcePort, VARIANT* pData)

Interface ICalSet

Read-only

GetStandardByString

Description Returns the data for a specific standard.

VB Syntax **GetStandardByString**(*bufferName, pdata*)

Variable	**(Type) - Description**
bufferName	**(String)** The string used to identify a particular buffer in the calset. An example string requesting the data for the Load standard in a full 2 port cal might be "Full 2 Port Cal (2,3)::S11C(3,3)".
pdata	**(Variant)** The pdata array is a variant. This data is usually two dimensional. Each element of the variant array is a 2 array of type single. The two elements represent the real and imaginary parts of a complex datum.

Note: This structure is compatible with scripting clients who can only use variants. For alternative methods that use typed arrays, see ICalData3.

Return Type Variant

Default Not Applicable

Examples See the following example

C++ Syntax **HRESULT GetStandardByString(BSTR** bufferName**, VARIANT*** pdata**)**

Interface ICalSet2

Write-only

GetStandardComplex Method

Description Queries standards acquisition data from the Cal Set. The data is in complex pairs. Learn more about reading and writing Cal Data using COM.

Before calling this method from the ICalData2 interface you must open the Cal Set with OpenCal Set. If the Cal Set is not open, this method returns E_NA_Cal Set_ACCESS_DENIED.

Note: This method exists on a non-default interface. If you cannot access this method, use the GetStandard Method on ICal Set

VB Syntax *interface.***getStandardComplex** *class, rcv, src, numPts, real(), imag()*

Variable **(Type) - Description**

interface An ICalData pointer to the Calibrator object **or**
An ICalData2 pointer to the Cal Set object(preferrred)

class **(enum NACalClass)** Standard to be measured. Choose from:

1 - naClassA

2 - naClassB

3 - naClassC

4 - naClassD

5 - naClassE

6 - naReferenceRatioLine

7 - naReferenceRatioThru

SOLT Standards

1 - naSOLT_Open

2 - naSOLT_Short

3 - naSOLT_Load

4 - naSOLT_Thru

5 - naSOLT_Isolation

TRL Standards

1 - naTRL_Reflection

2 - naTRL_Line_Reflection

3 - naTRL_Line_Tracking

4 - naTRL_Thru

5 - naTRL_Isolation

rcv **(long integer)** - Receiver Port *src* **(long integer)** - Source Port

numPts **(long integer)** - on input, max number of data points to return; on output: indicates the actual number of data points returned.

real() **(single)** - array to accept the real part of the calibration data. One-dimensional for the number of data points.

imag() **(single)** - array to accept the imaginary part of the calibration data. One-dimensional for the number of data points.

Return Type **(single)**

Default Not Applicable

Examples

```
Dim numpts as long
numpts = ActiveChannel.NumberOfPoints
ReDim r(numpts)  ' real part
ReDim i(numpts)  ' imaginary part
Dim Cal Set as Cal Set
set Cal Set = pna.GetCalManager.GetCal SetByGUID( txtGUID )
Dim sData As ICalData2
Set sData = Cal Set
sdata.getStandardComplex naSOLT_Open, 1, 1, numpts, r(0), i(0)
```

C++ Syntax HRESULT getStandardComplex(tagNACalClass stdclass, long ReceivePort, long SourcePort, long* pNumValues, float* pReal, float* pImag)

Interface ICalData2

Read-only

GetStandardComplexByString

Description	Queries the Cal Set for specific standard data.
VB Syntax	*ICalData3*.**GetStandardComplexByString**(*bufferName, lnumPoints, real(0), imag(0)*)
Variable	**(Type) - Description**
ICalData3	An **I**CalData3 pointer to a CalSet (**Object**)
bufferName	**(String)** The string used to identify a particular buffer in the Cal Set. An example string requesting the data for the Load standard in a full 2 port cal might be "Full 2 Port Cal (2,3)::S11C(3,3)".
lnumpoints	**in (Long)** The maximum number of values in the real and imaginary arrays being requested. **out (Long)** The actual number of values returned in the real and imaginary arrays.
real	**(Single)** The real component of the complex data.
imag	**(Single)** The imaginary component of the complex data.
Return Value	Single
Default	Not Applicable
Examples	See example
C++ Syntax	HRESULT GetStandardComplexByString(BSTR bufferName, long* lnumPoints, float* real, float* imag);
Interface	ICalData3

GetStandardsList Method

Description Returns the list of Standards contained in this Cal Set for the CalType specified in the OpenCal Set method. Learn more about reading and writing Cal Data using COM.

The list is a comma separated, textual representation of the error terms with the term name followed by the port path in parentheses.

Standard (n, n),
Standard (m, n)

Before calling this method you must open the Cal Set with OpenCal Set. If the Cal Set is not open, this method returns E_NA_Cal Set_ACCESS_DENIED.

Use StringToNACalClass to convert the list entrees to values that can be used with GetStandard and PutStandard.

Note: The port path designation (m n) indicates the receive and source ports for the measurement. Shorts, opens and loads are single port devices, designated in this list by (n n) where n equals the port to which the device is connected. These devices are all characterized by reflection measurements.
The dual port thru device is characterized by both transmission and reflection measurements in order to compensate for load match and tracking terms.
The notation (n n) indicates the reflection measurement for this device.
The notation (m n) indicates the transmission measurement, where the source and receive ports are different.

VB Syntax *CalSet*.**GetStandardsList** *(count, list)*

Variable **(Type) - Description**

CalSet **(object) -** A Cal Set object

count **(long [out])** - indicates the number of items returned in the list

list **(string)** - Variable to store the returned Comma separated list of items.

Return Type String

Default Not Applicable

Examples

```
dim count as Integer
dim list as string
OpenCalSet (naCalType_TwoPortSOLT, 1, 2)
GetStandardsList( count, list)
CloseCalSet( )
```

Assuming the Cal Set contained the full set of standards for this two port cal, the returned list would be:

```
"Open(1 1),
Short(1 1),
Load(1 1),
Thru(1 1),
Isolation(2 1),
Open(2 2),
Short(2 2),
Load(2 2),
Thru(2 2),
Isolation(1 2)
Thru(2 1),
Thru(1 2)"
```

C++ Syntax HRESULT GetStandardsList(long* count, BSTR* list);

Interface ICalSet

Read-only

GetStandardList2

Description	Returns a list of standards contained by this calset for the specified caltype.
VB Syntax	**GetStandardList2**(*calType, list*)
Variable	**(Type) - Description**
caltype	**(String)** The string used to identify calset data as belonging to a specific caltype. (By convention, calset buffers are named "CalType:Item"). An example string for a two port cal on ports 2 and 3 might be: "Full 2 Port Cal (2,3)". If the string is empty, all standards are returned. This string is used as a filter so that only the buffer names of interest are returned.
list	**(Variant)** Variant containing a string array of standards for the specified caltype.
Return Type	Variant
Default	Not Applicable
Examples	See the following example
C++ Syntax	HRESULT GetErrorTermList2(BSTR caltype, VARIANT* list)
Interface	ICalSet2

Read-only

GetStandardsForClass Method

Description	Get the calibration standard numbers for a specified calibration class. To set the calibration number use SetStandardsForClass Method
VB Syntax	*calkit*.**GetStandardsForClass** (*calclassorder, std1, std2, std3, std4, std5, std6, std7)*
Variable	**(Type) - Description**
calKit	A CalKit **(object)**
calclassorder	**(enum NACalClassOrder)** Choose from: 0 - naRefl_1_S11 1 - naRefl_2_S11 2 - naRefl_3_S11 3 - naTran_1_S21 4 - naRefl_1_S22 5 - naRefl_2_S22 6 - naRefl_3_S22 7 - naTran_1_S12 8 - naRefl_1_S33 9 - naRefl_2_S33 10 - naRefl_3_S33 11 - naTran_1_S32 12 - naTran_1_S23 13 - naTran_1_S31 14 - naTran_1_S13 15 - naTRL_T 16 - naTRL_R 17 - naTRL_L
std1…std7	**(long)** Calibration Standard Number. Nominal values from **1** through **30**. **0** indicates that a standard number has not been selected.

Return Type Not applicable

Default Not applicable

Examples

```
calkit.GetStandardsForClass naRefl_3_S11, std1, std2, std3, std4,
std5, std6, std7
```

C++ Syntax HRESULT GetStandardsForClass(NACalClassOrder calclassorder, long std1, long std2, long std3, long std4, long std5, long std6, long std7)

Interface ICalKit

Read-only

GetStepDescription Method

Description Returns the description of the specified step in the calibration process.

VB Syntax *value* = *obj*.**GetStepDescription** *(n)*

Variable **(Type) - Description**

value (string) - Variable to store the returned number of steps.

obj Any of the following:

GuidedCalibration (object)

SMCType (object)

VMCType (object)

n (Long) Step in the calibration process.

Use GenerateSteps to determine the total number of steps.

Return Type String

Default Not Applicable

Examples

```
value = SMC.GetStepDescription(5)
```

C++ Syntax HRESULT get_GetStepDescription(long step, BSTR* str);

Interface IGuidedCalibration

SMCType

VMCType

Read-only

GetTestResult Method

Description Returns the result of limit line testing. There are three ways to use this command:

- If neither optional parameter is specified, limit results for ALL data is returned.
- If one parameter is specified (*start*), the limit result for that data point is returned.
- If both parameters are specified, limit results are returned beginning with *start*, and ending with (*start*+*size*)-1

VB Syntax *testRes* = *limts*.**GetTestResult [***start,size***]**

Variable **(Type) - Description**

testRes **(enum NALimitTestResult)** - A dimensioned variable to store test results. If a limit line is not tested, a PASS is returned.

0 - naLimitTestResult_None
1 - naLimitTestResult_Fail
2 - naLimitTestResult_Pass

limts A LimitTest **(object)**

start **(long)** - Optional argument. A start data point number to return limit test results.

size **(long)** - Optional argument. Number of data points from *start* to return limit test results.

Return Type Long Integer

Default Not Applicable

Examples

```
Dim testRes As NALimitTestResult
testRes = limts.GetTestResult
Select Case testRes

Case 1
Print "Fails"

Case 2
Print "Pass"

End Select
```

C++ Syntax HRESULT GetTestResult(long lStart, long lSize, tagNALimitTestResult *pVal)

Interface ILimitTest

About Trace Statistics

Read-only

GetTraceStatistics Method

Description Returns all four Trace Statistics. To retreive individual Trace statistics, use Mean, PeakToPeak, StandardDeviation properties. Use ShowStatistics to display the statistics of the screen.

VB Syntax *meas*.**GetTraceStatistics** *pp,mean,stdev*

Variable **(Type) - Description**

meas A Measurement **(object)**

pp,mean,stdev **(double)** - Dimensioned variables to store the returned values

Return Type Double

Default Not Applicable

Examples

```
'Dimension variables
Dim pp As Double
Dim mean As Double
Dim stdv As Double
meas.GetTraceStatistics pp, mean, stdv
```

C++ Syntax HRESULT GetTraceStatistics(double* pp, double* mean, double* stdDeviation)

Interface IMeasurement

GetXAxisValues2 Method

Description Returns the channel's X-axis values into a dimensioned Typed array. GetXAxisValues2 is a convenient method for determining the frequency of each point when the points are not linearly spaced - as in segment sweep.

Note: This method will fail if called using a scripting client such as VBScript or Agilent Vee, (see remarks)

Note: In Segment Sweep, chan.NumberofPoints will return the total number of data points for the combined segments.

VB Syntax *chan*.**GetXAxisValues2** *numPts,data*

Variable **(Type) - Description**

chan **(object)** - A Channel object

numPts **(long integer)** - Number of data points in the channel

data **(double)** Single dimensioned array of data matching the number of points in the channel.

Return Type double

Default Not applicable

Examples

```
Dim App As Application
Set App = New Application
Dim numPoints As Long
Dim values() As Double
numPoints = App.ActiveChannel.NumberOfPoints
ReDim values(numPoints)
App.ActiveChannel.GetXAxisValues2 numPoints, values(0)
Print values(0), values(1)
```

C++ Syntax HRESULT GetXAxisValues2(long* pNumValues, double* stimulus)

Interface IChannel

Remarks:

This method will fail if called using a scripting client such as VBScript or Agilent Vee. Use the GetXAxisValues method as a replacement for these COM environments.

This method also cannot be called using late-bound typing in Visual Basic. For instance, if, in the example above, the first line were replaced with "Dim App as Object", then this method would fail.

Read-only

GetXAxisValues Method

Description	Returns the stimulus values for the measurement. To understand how this property is useful, see IMeasurement2 Interface.
VB Syntax	*data* = *meas*.**GetXAxisValues**
Variable	**(Type) - Description**
data	**(Variant)** Array to store the data.
meas	A Measurement **(object)**
Return Type	Variant
Default	Not Applicable

Examples

```
Dim varData As Variant
Dim i As Integer
varData = meas.GetXAxisValues
'Print Data
For i = 0 To meas.NumberOfPoints - 1
Print varData(i)
Next i
```

C++ Syntax	HRESULT GetXAxisValues(VARIANT* xData);
Interface	IMeasurement2

GetXAxisValues Method

Description Returns the channel's X-axis values. GetXAxisValues is a convenient method for determining the frequency of each point when the points are not linearly spaced - as in segment sweep.

Note: This method returns a variant which is less efficient than GetXAxisValues2.

Note: In Segment Sweep, chan.NumberofPoints will return the total number of data points for the combined segments.

VB Syntax *data* = *chan*.**GetXAxisValues**

Variable **(Type) - Description**

data Variant array to store the data.

chan A Channel **(object)**

Return Type Variant

Default Not Applicable

Examples

```
Dim varData As Variant
Dim i As Integer
varData = chan.GetXAxisValues
'Print Data
For i = 0 To chan.NumberOfPoints - 1
 Print varData(i)
Next i
```

C++ Syntax HRESULT GetXAxisValues (VARIANT* xData)

Interface IChannel

HasCalType Method

Description Verifies that the Cal Set object contains the error terms required to perform the specified correction (CalType) to an appropriate measurement.

The argument list includes specifiers for up to 3 ports. The number of arguments required depends on the CalType specified. The value for each port is set to 0 if not specified.

* order of port arguments is significant for these caltypes

Caltype	Port arguments required
naCalType_Response_Open	Port1
naCalType_Response_Short	Port1
*naCalType_Response_Thru	Port1 (rcv), Port2 (src)
*naCalType_Response_Thru_And_Isol	Port1 (rcv), Port2 (src)
naCalType_OnePort	Port1
naCalType_TwoPort_SOLT	Port1, Port2
naCalType_TwoPort_TRL	Port1, Port2
naCalType_ThreePort_SOLT	Port1, Port2, Port3

VB Syntax *check* = *CalSet*.**HasCalType** *calType, port1, port2, port3*

Variable **(Type) - Description**

check **(boolean) -** variable to store the returned value

TRUE (nonzero) - Cal Set has all of the error terms necessary to apply the specified correction (CalType)

FALSE(0) - Cal Set DOES NOT have all of the error terms necessary to apply the specified CalType

CalSet **(object) -** A Cal Set object

calType (enum as naCalType) - type of correction to be applied. Choose from

0 - naCalType_Response_Open
1 - naCalType_Response_Short
2 - naCalType_Response_Thru
3 - naCalType_Response_Thru_And_Isol
4 - naCalType_OnePort
5 - naCalType_TwoPort_SOLT
6 - naCalType_TwoPort_TRL
7 - naCalType_None
8 - naCalType_ThreePort_SOLT

port1 **(long)** - required. This argument must be specified.

This specifies either:

- the one significant port for an open/short response cal or a 1 port cal.

- or one of the ports involved in a 2 or 3 port cal

- or the ***receive*** port for a thru response / thru-isolation cal.

port2 **(long) -** required for any caltype involving more than one port

This specifies either:

- one of the ports involved in a 2 or 3 port cal (order independent)

- or the **source** port for a thru response / thru-isolation cal

port3 **(long) -** required only for 3 port cal

This specifies either:

- one of the ports involved in a 3 port cal (order independent)

Return Type **VARIANT_BOOL**

Default Not Applicable

Examples

```
value = CalSet.HasCalType(naCalType_TwoPort_TRL, 1, 2)
```

C++ Syntax HRESULT HasCalType(tagNACalType, long port1, long port2, long port3, BOOL *pVal);

Interface ICalSet

Write-only

Hold Method

Description	Puts the Channel in Hold - not sweeping.
VB Syntax	*chan.***Hold** *[sync]*
Variable	**(Type) - Description**
chan	A Channel **(object)**
[sync]	**(boolean)** - Optional argument. A variable set to either True or False. **True** - program control waits until the channel is in the Hold state. **False** - program control continues immediately. You are not guaranteed the channel is in Hold yet.
Return Type	Not Applicable
Default	Not Applicable
Examples	`wate = True` `chan.Hold wate`
C++ Syntax	HRESULT Hold(VARIANT_BOOL bWait)
Interface	IChannel

Hold (channels) Method

Description	Places all channels in hold mode. To place a single channel in hold mode, use channel.Hold Method.
VB Syntax	*chans.***Hold**
Variable	**(Type) - Description**
chans	A Channel collection **(object)**
Return Type	Not Applicable
Default	Not Applicable
Examples	`chans.Hold`
C++ Syntax	HRESULT Hold();
Interface	IChannels2

Write only

Initialize Method

Description	Begins a calibration.
	Note: *chan* must be the active channel.
VB Syntax	*obj*.**Initialize** (*chan, useCalReg)*
Variable	**(Type) - Description**
obj	Any of the following: GuidedCalibration (object) SMCType (object) VMCType (object)
chan	(Long) Channel number to calibrate.
useCalReg	(boolean) **True** or **1** - Cal will be stored in Cal Register ONLY **False** or **0** - Cal will be stored in Cal Register AND User Cal Set. The active channel must be pointing to a User Cal Set.
Return Type	Not Applicable
Default	Not Applicable
Examples	`smc.Initialize(2,1)`
C++ Syntax	HRESULT put_Initialize(long channelnumber, VARIANT_BOOL bNewCalSet);
Interface	IGuidedCalibration SMCType VMCType

Write-only

Item Method

Description Returns an object from the collection of objects.

Note: The order of objects within a collection cannot be assumed.

VB Syntax *Object*[**.Item**]**(***n***)**

Variable **(Type) - Description**

Object Any of the following **(objects):**
CalFactorSegments collection
Cal Sets collection
Channels collection
LimitTest collection
Measurements collection
NaWindows collection
PowerLossSegments collection
PowerSensors collection
Segments collection
Traces collection

.Item Optional - Item is the default property of a collections object and therefore can be called implicitly. For example, the following two commands are equivalent:

```
Channels.Item(3).Averaging = 1
Channels(3).Averaging = 1
```

n **(variant)** - number of the item in the collection.

Note: the Measurements and Traces collections allow you to specify the name of the measurement as a string. For example:
`measCollection("CH_S11_1").InterpolateMarkers`

Return Type **(Object)**

Default Not Applicable

Examples

```
For i = 1 to Traces.Count -1
 Traces.Item(i).YScale = .5dB
Next i
```

C++ Syntax HRESULT Item(VARIANT index, <interface>** pItem)

Interfaces ICalFactorSegments
ICal Sets
IChannels
ILimitTest
IMeasurements
INaWindows
IPowerLossSegments
IPowerSensors
ISegments
ITraces

About the Cal Wizard

Read-Write

LaunchCalWizard Method

Description Launches the Cal Wizard on the PNA and does not return until the Cal Wizard is dismissed.

Note: The Cal Wizard operates on the active measurement. Therefore, activate the measurement to be calibrated before launching the Cal Wizard.

VB Syntax *success* = *app.***LaunchCalWizard** *(newCS)*

Variable **(Type) - Description**

success **(boolean) -** variable to store the returned value
True - The Cal was completed
False - The Cal was canceled without completing the calibration.

app An Application **(object)**

newCS **(boolean)**

True - Cal will be performed on a new Cal Set.
False - Cal will be performed using the existing Cal Set assigned to the channel. If no Cal Set is found, a new Cal Set will be created.

Return Type Boolean

Default Not Applicable

Example

```
dim bSuccess as boolean
dim bNewCalset as boolean
bNewCalSet = false
bSuccess = app.LaunchCalWizard( bNewCalSet)
```

C++ Syntax HRESULT LaunchCalWizard(VARIANT_BOOL bCalsuccess)

Interface IApplication

LaunchPowerMeterSettingsDialog Method

Description	Launches the Power Meter Settings dialog on the PNA. Changing certain values from that dialog will change values of the corresponding properties on this COM object.
VB Syntax	*pwrCal*.**LaunchPowerMeterSettingsDialog**
Variable	**(Type) - Description**
pwrCal	A SourcePowerCalibrator **(object)**
Return Type	None
Default	Not Applicable
Examples	`powerCalibrator.LaunchPowerMeterSettingsDialog`
C++ Syntax	HRESULT put_LaunchPowerMeterSettingsDialog();
Interface	ISourcePowerCalibrator2

Write only

LoadFile Method

Description	Loads a previously-configured mixer attributes file (.mxr)
VB Syntax	*mixer*.**LoadFile** *(filename)*
Variable	**(Type) - Description**
mixer	A IMixer Interface pointer to the Meas **(object)**
filename	(String) Path and file name of the .mxr file
Return Type	String
Default	Not Applicable
Examples	`mixer.LoadFile (File_Name)`
C++ Syntax	HRESULT put_LoadFile(BSTR newVal)
Interface	IMixer

Write-only

ManualTrigger Method

Description Triggers the analyzer when `TriggerSignal = naTriggerManual`.

Note: An **SMC Fixed Output** measurement cannot be triggered using this command. For more information, see the example program.

VB Syntax *app.***ManualTrigger** *[sync],[timeout]*

Variable **(Type) - Description**

app An Application **(object)**

sync **(boolean)** - Optional argument.
A variable set to either True or False.

True - The analyzer waits until the trigger is completed to process subsequent commands.
False - Subsequent commands are processed immediately (the default setting)

timeout **(long) -** Optional argument.
If *sync* is true, *timeout* sets the amount of time the PNA will wait until continuing program execution. Units are milliseconds. A value of -1 (the default setting) causes the PNA to wait indefinitely.

If sync is False, the timeout setting is ignored.

Return Type Not Applicable

Default Not Applicable

Examples

```
' After Manual trigger is executed, the PNA will wait 1 second
to continue program execution
Dim wait as Boolean
wait = True
app.ManualTrigger wait, 1000
```

C++ Syntax HRESULT ManualTrigger(VARIANT_BOOL bSynchronize, long timeout)

Interface IApplication

MessageText Property

Description	Returns text for the specified eventID
VB Syntax	*app*.**MessageText,***eventID,message*
Variable	**(Type) - Description**
app	An Application **(object)**
enentID	**(enum naEventID)** Choose from the list in Working with the Analyzer's Events
message	**(string) -** variable to store the returned message
Return Type	String
Default	Not Applicable
Examples	`RFNA.MessageText naEventID_ARRANGE_WINDOW_EXCEED_CAPACITY, message`
C++ Syntax	HRESULT MessageText(tagNAEventID msgID, BSTR* message)
Interface	IApplication

About Dynamic Range

Write-only

NextIFBandwidth Method

Description A function that returns the Next higher IF Bandwidth value. Use to retrieve the list of available IFBandwidth settings.

VB Syntax *chan.***Next_IFBandwidth** *bw*

Variable **(Type) - Description**

chan A Channel **(object)**

bw **(double)** - The argument that you use to send an IFBandwidth. The function uses this argument to return the Next higher IFbandwidth.

Return Type Double

Default Not Applicable

Examples

```
Public pnbw As Double 'declare variable outside of procedure
```

```
pnBW = chan.IFBandwidth 'put the current IFBW in pnBW
chan.Next_IFBandwidth pnBW 'function returns the Next higher
IFBandwidth.
chan.IFBandwidth = pnBW 'set IFBW to the Next value
```

C++ Syntax HRESULT Next_IFBandwidth (double *pVal)

Interface IChannel

Write-only

NumberOfGroups Method

Description	Sets the Number of trigger signals the channel will receive. After the channels has received that number of trigger signals, the channel switches to Hold mode. To begin sweeping the number of groups, send app.Continuous
VB Syntax	*chan*.**NumberOfGroups** *num, sync*
Variable	**(Type) - Description**
chan	A Channel **(object)**
num	**(long integer)** Number of trigger signals the channel will receive. Choose any number between 1 and 2 million
sync	**(boolean)** Variable set to either: **True -** subsequent commands are not processed until the groups are complete. **Do not use with manual trigger.** **False -** subsequent commands are processed immediately
Return Type	Not Applicable
Default	Not Applicable
Examples	`chan.NumberOfGroups`
C++ Syntax	HRESULT NumberOfGroups(long count, VARIANT_BOOL bWait)
Interface	IChannel

OpenCalSet Method

Description Open the Cal Set to read/write a particular **CalType.** Learn more about reading and writing Cal Data using COM.

This method is a prerequisite to several other Cal Set methods.

A Cal Set can contain more than one **caltype.** This method opens the Cal Set and restrict access to a particular set of terms. Subsequent commands like PutErrorTerm and GetErrorTerm use this information to access the correct error terms in the Cal Set. For example:

```
OpenCal Set( naCalType_TwoPortSOLT, 3, 2, 0)
```

```
PutErrorTerm( naDirectivity, 1, 1, Buffer)
```

The directivity error term for port 1 could belong to any number of caltypes: Full1Port (S11), Full2Port (12), Full2Port (13) or Full3Port (123). The **CalType and port** specifiers in the OpenCalSet call direct the uploaded directivity term to the correct set of error terms.

To close the CalType, see CloseCalSet.

The argument list includes three port specifiers. The following table shows which of these arguments are significant, given the **CalType** specified.

Caltype	**Port arguments required**
naCalType_Response_Open	Port1
naCalType_Response_Short	Port1
*naCalType_Response_Thru	Port1 (rcv), Port2(src)
*naCalType_Response_Thru_And_Isol	Port1 (rcv), Port2(src)
naCalType_OnePort	Port1
naCalType_TwoPort_SOLT	Port1, Port2
naCalType_TwoPort_TRL	Port1, Port2
naCalType_ThreePort_SOLT	Port1, Port2, Port3

* order of port arguments is significant for these caltypes

VB Syntax *CalSet*.**OpenCalSet** (CalType, port1, port2, port3)

Variable **(Type) - Description**

CalSet **(object) -** A Cal Set object

CalType (enum as naCalType) - type of correction to be applied. Choose from

0 - naCalType_Response_Open
1 - naCalType_Response_Short
2 - naCalType_Response_Thru
3 - naCalType_Response_Thru_And_Isol
4 - naCalType_OnePort
5 - naCalType_TwoPort_SOLT
6 - naCalType_TwoPort_TRL

7 - naCalType_None
8 - naCalType_ThreePort_SOLT

port1 **(long)** - required. This argument must be specified.

This specifies either:

- the one significant port for an open/short response cal or a 1 port cal.

- or one of the ports involved in a 2 or 3 port cal

- or the ***receive*** port for a thru response / thru-isolation cal.

port2 **(long) -** required for any caltype involving more than one port

This specifies either:

- one of the ports involved in a 2 or 3 port cal (order independent)

- or the ***source*** port for a thru response / thru-isolation cal

port3 **(long) -** required only for 3 port cal

This specifies either:

- one of the ports involved in a 3 port cal (order independent)

Return Type None

Default Not Applicable

Examples

```
CalSet.OpenCalSet naCalType_ThreePort_SOLT, 3,2,1
```

C++ Syntax HRESULT OpenCalSet (naCalType, port1, [optional] port2, [optional] port3);

Interface ICalSet

Write-Read

Parse Method

Description Executes a SCPI command.

VB Syntax *scpi.***Parse (***"SCPI command"***)**

Variable **(Type) - Description**

scpi A ScpiStringParser **(object)**

SCPI command **(string)** - Any valid SCPI command

Return Type String

Default Not Applicable

Examples

```
Dim scpi As ScpiStringParser
Set scpi = app.ScpiStringParser
Dim startfreq As Double
startfreq = 100e6
'
scpi.Parse ("Sens:Freq:Start " & startfreq)'Write
```

```
Dim str As String
str = scpi.Parse ("Sens:Freq:Start?")'Read
```

C++ Syntax HRESULT Parse(BSTR SCPI_Command, BSTR *pQueryResponse)

Interface IScpiStringParser

Preset Method

Description	**Application Object**: Deletes all traces and windows. In addition, resets the analyzer to factory defined default settings and creates an S11 measurement named "CH1_S11_1" in window 1. **Channel Object:** Resets the channel (object) to factory defined default settings. Does NOT delete the current measurements or add a new measurement. Regardless of the state of the User Preset Enable checkbox, the app.Preset command will always preset the PNA to the factory preset settings, and app.UserPreset will always perform a User Preset.
VB Syntax	*app.***Preset** *chan***.Preset**
Variable	**(Type) - Description**
app	An Application **(object)**
chan	A Channel **(object)**
Return Type	Not Applicable
Default	Not Applicable
Examples	`app.Preset`
C++ Syntax	HRESULT Preset()
Interface	IApplication IChannel

PreviousIFBandwidth Method

Description A function that returns the previous IF Bandwidth value. Use to retrieve the list of available IFBandwidth settings.

VB Syntax *chan.***Previous_IFBandwidth** *bw*

Variable **(Type) - Description**

chan A Channel **(object)**

bw **(double)** - The argument that you use to send an IFBandwidth. The function uses this argument to return the previous IFbandwidth.

Return Type Double

Default Not Applicable

Examples

```
Public pnbw As Double 'declare variable outside of procedure
```

```
PreBW = chan.IFBandwidth 'put the current IFBW in PreBW
chan.Previous_IFBandwidth PreBW 'function returns the Previous
IFBandwidth of the current one.
chan.IFBandwidth = PreBW 'set IFBW to the previous value
```

C++ Syntax HRESULT Previous_IFBandwidth (double *pVal)

Interface IChannel

About Saving to File

Write-only

PrintToFile Method

Description Saves the screen image to a bitmap (.bmp) file.

VB Syntax *app.***PrintToFile** *filename*

Variable **(Type) - Description**

app An Application **(object)**

filename **(string)** Name of the file to save the screen to. The file is saved to the current working directory unless a valid full path name is specified.

Use one of the following suffixes:

- .bmp - not recommended due to large file size
- .jpg - not recommended due to poor quality
- .png - recommended

Return Type Not Applicable

Default Not Applicable

Examples

```
app.PrintToFile "myfile.png"
app.PrintToFile "c:\data\myfile.png"
```

C++ Syntax HRESULT PrintToFile(BSTR bstrFile)

Interface IApplication

PutComplex Method

Description Puts real and imaginary data into the specified location. This method forces the channel into Hold mode to prevent the input data from being overwritten. Learn more about reading and writing Cal Data using COM.

Data put in the raw data store will be **re-processed** whenever a change is made to the measurement attributes such as format or correction.

Data put in the measurement results store will be **overwritten** by any measurement attribute changes.

See also putNAComplex

VB Syntax *measData*.**putComplex** *location, numPts, real(), imag(),* [*format*]

Variable **(Type) - Description**

measData An IArrayTransfer interface which supports the Measurement object

location **(enum NADataStore)** Where the Data will be put. Choose from:

0 - naRawData

1 - naCorrectedData

2 - naMeasResult

3 - naRawMemory

4 - naMemoryResult

5 - naDivisor - When reading data from, or writing data to, the normalization divisor, you must first create a divisor trace using DataToDivisor Method.

See the Data Access Map

numPts **(long integer)** - Number of data points in the channel

real() **(single) -** Array containing real data values

imag() **(single)** -Array containing imaginary data values *format* **(enum NADataFormat)** optional argument - display format of the real and imaginary data. Only used if destination is naMeasResult or naMemoryResult buffer. If unspecified, data is assumed to be in naDataFormat_Polar

0 - naDataFormat_LinMag

1 - naDataFormat_LogMag

2 - naDataFormat_Phase

3 - naDataFormat_Polar

4 - naDataFormat_Smith

5 - naDataFormat_Delay

6 - naDataFormat_Real

7 - naDataFormat_Imaginary

8 - naDataFormat_SWR

Return Type Not Applicable

Default Not Applicable

Examples

```
Dim measData As IArrayTransfer
Set measData = app.ActiveMeasurement

measData.putComplex naMemoryResult, 201,
real(0),imag(0),naDataFormat_SWR
```

C++ Syntax HRESULT putComplex(tagNADataStore DataStore, long lNumValues, float* pReal, float* pImag, tagDataFormat displayFormat)

Interface IArrayTransfer

PutDataComplex Method

Description Puts complex data into the specified location. This method forces the channel into Hold mode to prevent the input data from being overwritten.

VB Syntax *meas*.**putDataComplex** *location, data*

Variable **(Type) - Description**

meas A measurement **(object)**

location **(enum NADataStore)** Where the Data will be put. Choose from:

0 - naRawData

1 - naCorrectedData

2 - naMeasResult

3 - naRawMemory

4 - naMemoryResult

5 - naDivisor - When reading data from, or writing data to, the normalization divisor, you must first create a divisor trace using DataToDivisor Method.

- Data put in 0 - naRawData will be re-processed whenever a change is made to the measurement attributes such as format or correction.
- Data put in 2 - naMeasResult will be overwritten by any measurement attribute changes.
- When putting data into 3 - naRawMemory:

1. Put the analyzer in hold mode
2. Call DataToMemory to initialize a memory buffer
3. Call putDataComplex(naRawMemory, data)

This ensures that the memory buffer is appropriately initialized before receiving new data.

See the Data Access Map

data **(variant)** - A **two-dimensional** variant array.

Note: All buffers except naMeasResult and naMemoryResult require Complex data

Return Type Not Applicable

Default Not Applicable

Examples

```
' Put 201 points worth of raw (complex) data into the
measurement
' Note that an array of complex numbers is represented by a 2-D
array where the first rank is the number of points, and the 2nd
rank is always size 2 (max index 1) representing the Real and
Imag parts of the complex number.

' complex array of data (2nd dimension of size 2 represents
Re/Im
Dim data(200,1) )
For i = 0 to 200
' Set Real part of data point i
data(i,0) = i/200;
' Set Imag part of data point i
data(i,1) = i/200;
Next
app.ActiveMeasurement.putDataComplex naRawData, data
```

C++ Syntax HRESULT putDataComplex(tagNADataStore DataStore, VARIANT complexData)

Interface IMeasurement

Write-only

PutErrorTerm Method - Obsolete

Description	**Note**: This command is replaced by CalSet.putErrorTerm. Puts variant error term data into the error-correction buffer. See Accessing data. Learn about reading and writing Calibration data.
VB Syntax	*cal.***putErrorTerm**(*term,rcv, src, data*)
Variable	**(Type) - Description**
cal	A Calibrator **(object)**
term	**(enum As NaErrorTerm)** naErrorTerm_Directivity_Isolation naErrorTerm_Match naErrorTerm_Tracking
rcv	**(long integer)** - Receiver Port
src	**(long integer)** - Source Port
data	**(variant)** Error term data in a two-dimensional array (0:1, 0:numpts-1).

To get this	**Specify these parameters:**		
Error Term	*term*	*rcv*	*src*
Fwd Directivity	naET_Directivity Isolation	1	1
Rev Directivity	naET_Directivity Isolation	2	2
Fwd Isolation	naET_Directivity Isolation	2	1
Rev Isolation	naET_Directivity Isolation	1	2
Fwd Source Match	naErrorTerm_Match	1	1
Rev Source Match	naErrorTerm_Match	2	2
Fwd Load Match	naErrorTerm_Match	2	1
Rev Load Match	naErrorTerm_Match	1	2
Fwd Reflection Tracking	naErrorTerm_Tracking	1	1
Rev Reflection Tracking	naErrorTerm_Tracking	2	2
Fwd Trans Tracking	naErrorTerm_Tracking	2	1
Rev Trans Tracking	naErrorTerm_Tracking	1	2
Fwd Trans Tracking	naErrorTerm_Tracking	2	1

Return Type Not Applicable

Default Not Applicable

Examples

```
Dim varError As Variant
varError = cal.putErrorTerm (naErrorTerm_Tracking,2,1,VarData)
```

C++ Syntax HRESULT putErrorTerm(tagNAErrorTerm ETerm, long ReceivePort, long SourcePort, VARIANT varData)

Interface ICalibrator

Write-only

PutErrorTerm Method

Description	Puts error term data into the Cal Set. Learn more about reading and writing Cal data using COM Before calling this method you must open the Cal Set with OpenCal Set.. If the Cal Set is not open, this method returns E_NA_Cal Set_ACCESS_DENIED. The data must be complex pairs, contained in a two-dimensional VARIANT array. **Note:** See also PutErrorTermComplex on the ICalData2 interface to avoid using the variant data type.
VB Syntax	*CalSet.***putErrorTerm** (*term, rcv, src, data*)
Variable	**(Type) - Description**
CalSet	**(object) -** A Cal Set object
term	**(enum As NaErrorTerm2)** Error Term. Choose from: 0 - naET_Directivity (src = rcv) 1 - naET_SourceMatch (src = rcv) 2 - naET_ReflectionTracking (src = rcv) 3 - naET_TransmissionTracking (src != rcv) 4 - naET_LoadMatch (src != rcv) 5 - naET_Isolation (src != rcv)
rcv	**(long integer)** - Receiver Port
src	**(long integer)** - Source Port
data	**(variant)** Error term data in a two-dimensional array (0:1, 0:numpts-1).
Return Type	Not Applicable
Default	Not Applicable

Examples

```
Private Sub Form_Load()
Set pna=CreateObject("AgilentPNA835x.Application")
InitPhonyData
PutPhonyData
End Sub

Private Sub InitPhonyData()
Dim i
Dim numpts
numpts = ActiveChannel.NumberOfPoints
ReDim v(numpts - 1, 1)

For i = 0 To numpts - 1
v(i, 0) = i
v(i, 1) = 0
Next

End Sub

Private Sub PutPhonyData()
Dim cset As CalSet
Set cmgr = pna.GetCalManager
Set cset = cmgr.CreateCalSet(1)
cset.OpenCalSet naCalType_OnePort, 1
cset.putErrorTerm naET_Directivity, 1, 1, v
cset.putErrorTerm naET_ReflectionTracking, 1, 1, v
cset.putErrorTerm naET_SourceMatch, 1, 1, v
cset.CloseCalSet
cset.Description = "Phony One Port"
guid = cset.GetGUID

End Sub
```

C++ Syntax HRESULT putErrorTerm(tagNAErrorTerm2 ETerm, long ReceivePort, long SourcePort, VARIANT varData)

Interface ICalSet

Write-only

PutErrorTermByString

Description	Writes a name/array pair to the calset
VB Syntax	**PutErrorTermByString**(*bufferName, vdata*)
Variable	**(Type) - Description**
bufferName	**(String)** The string used to specify a particular buffer in the calset. An example string for port 3 directivity in a full 2 port cal might be "Full 2 Port Cal (2,3)::Directivity(3,3)".
vdata	**(Variant)** The vdata array is a variant. This data is usually two dimensional. Each element of the variant array is a 2 array of type single. The two elements represent the real and imaginary parts of a complex datum. **Note:** This structure is compatible with scripting clients who can only use variants. For alternative methods that use typed arrays, see ICalData3.
Return Type	Not Applicable
Default	Not Applicable
Examples	See example
C++ Syntax	HRESULT PutStandardByString(BSTR bufferName, VARIANT vardata)
Interface	ICalSet2

Write-only

PutErrorTermComplex Method

Description Puts error term data into the error-correction data buffer. Learn more about reading and writing Cal data using COM

VB Syntax *data.***putErrorTermComplex** *term, rcv, src, numPts, real(), imag()*

Variable **(Type) - Description**

data An ICalData pointer to the Calibrator object

term **(enum NAErrorTerm) -** The error term to be retrieved. Choose from:

- **naErrorTerm_Directivity_Isolation**
- **naErrorTerm_Match**
- **naErrorTerm_Tracking**

rcv **(long integer)** - Receiver Port

src **(long integer)** - Source Port

numPts **(long integer) -** number of data points in the array

real() **(single)** - array containing the **real** part of the calibration data. One-dimensional: the number of data points.

imag() **(single)** - array containing the **imaginary** part of the calibration data. One-dimensional: the number of data points.

To get this Error Term	Specify these parameters: *term*	*rcv*	*src*
Fwd Directivity	naET_Directivity Isolation	1	1
Rev Directivity	naET_Directivity Isolation	2	2
Fwd Isolation	naET_Directivity Isolation	2	1
Rev Isolation	naET_Directivity Isolation	1	2
Fwd Source Match	naErrorTerm_Match	1	1
Rev Source Match	naErrorTerm_Match	2	2
Fwd Load Match	naErrorTerm_Match	2	1
Rev Load Match	naErrorTerm_Match	1	2
Fwd Reflection Tracking	naErrorTerm_Tracking	1	1
Rev Reflection Tracking	naErrorTerm_Tracking	2	2
Fwd Trans Tracking	naErrorTerm_Tracking	2	1
Rev Trans Tracking	naErrorTerm_Tracking	1	2

Fwd Trans Tracking	naErrorTerm_Tracking	2	1

Return Type Not Applicable

Default Not Applicable

Examples

```
Dim eData As ICalData
Set eData = chan.Calibrator
eData.putErrorTermComplex naErrorTerm_Directivity_Isolation, 1,
1, 201, rel(0), img(0)
```

C++ Syntax HRESULT putErrorTermComplex(tagNAErrorTerm ETerm, long ReceivePort, long SourcePort, long* pNumValues, float* pReal, float* pImag)

Interface ICalData

PutErrorTermComplex Method

Description Puts error term data into the Cal Set. Learn more about reading and writing Cal data using COM

Before calling this method you must open the Cal Set with OpenCal Set. If the Cal Set is not open, this method returns E_NA_Cal Set_ACCESS_DENIED.

VB Syntax *data.***putErrorTermComplex** *term, rcv, src, numPts, real(), imag()*

Variable **(Type) - Description**

data An ICalData2 pointer to the Cal Set object

term **(enum NAErrorTerm2)** - The error term to be written. Choose from:

0 - naET_Directivity

1 - naET_SourceMatch

2 - naET_ReflectionTracking

3 - naET_TransmissionTracking

4 - naET_LoadMatch

5 - naET_Isolation

rcv **(long integer)** - Receiver Port

src **(long integer)** - Source Port

numPts **(long integer) -** number of data points in the array

real() **(single)** - array containing the **real** part of the calibration data. One-dimensional: the number of data points.

imag() **(single)** - array containing the **imaginary** part of the calibration data. One-dimensional: the number of data points.

Return Type Not Applicable

Default Not Applicable

Examples

```
Dim eData As ICalData2
Set eData = app.GetCalManager.Cal Sets.Item(1)
eData.putErrorTermComplex naET_LoadMatch, 1, 2, numpts, rel(0),
img(0)
```

C++ Syntax HRESULT putErrorTermComplex(tagNAErrorTerm2 ETerm, long ReceivePort, long SourcePort, long* pNumValues, float* pReal, float* pImag)

Interface ICalData2

Write-only

PutErrorTermComplexByString

Description	Writes a name/array pair to the calset
VB Syntax	**PutErrorTermComplexByString**(bufferName, lnumPoints, real(0), imag(0))
Variable	**(Type) - Description**
bufferName	**(String)** The string used to identify a particular buffer in the calset. An example string for port 3 directivity in a full 2 port cal might be "Full 2 Port Cal (2,3)::Directivity(3,3)".
lnumpoints	**(Long)** The capacity of the float arrays in arguments "real" & "imag".
real	**(Single)** The real component of the complex data.
imag	**(Single)** The imaginary component of the complex data. **Note**: The size of the real and imaginary arrays should be the same.
Return Value	Single
Default	Not Applicable
Examples	See example
C++ Syntax	HRESULT PutErrorTermComplexByString(BSTR bufferName, long lnumPoints, float* real, float* imag);
Interface	ICalData3

PutScalar Method

Description Puts Scalar data in the Measurement Result buffer. The putScalar array is not processed by the analyzer; it is just displayed. Any change to the measurement state (changing the format, for example) will cause the putScalar data to be overwritten with the data processed from the raw data buffer.

VB Syntax *measData*.**putScalar***, format, numPts, data*

Variable **(Type) - Description**

measData An IArrayTransfer interface which supports the Measurement object.

format **(enum NADataFormat)** Format of the data. Choose from:

0 - naDataFormat_LinMag

1 - naDataFormat_LogMag

2 - naDataFormat_Phase

5 - naDataFormat_Delay

6 - naDataFormat_Real

7 - naDataFormat_Imaginary

8 - naDataFormat_SWR

Note: Smith and Polar formats are not allowed.

See the Data Access Map

numPts **(integer)** - Number of values. Usually the number of points in the trace (chan.NumberOfPoints).

data **(single)** - A one-dimensional array of Scalar data matching the number of points in the current measurement.

Return Type Not Applicable

Default Not Applicable

Examples

```
Dim measData As IArrayTransfer
Set measData = app.ActiveMeasurement

measData.putScalar naDataFormat_LogMag, 201, dScalar(0)
```

C++ Syntax HRESULT putScalar\(tagDataFormat eFormat, long lNumValues, float* pArrayOfScalar)

Interface IArrayTransfer

PutNAComplex Method

Description Puts complex data into the specified location. This method forces the channel into Hold mode to prevent the input data from being overwritten. The data is processed and displayed.

Data put in the naRawData store will be **re-processed** whenever a change is made to the measurement attributes such as format or correction.

Data put in the naMeasResult store will be **overwritten** by any measurement attribute changes (such as moving a marker).

Note: This method uses NAComplex which is a user-defined data type. If you cannot or prefer not to use this data type, use the putComplex method.

VB Syntax *measData*.**putNAComplex** *location, numPts, data,* [*format*]

Variable **(Type) - Description**

measData An IArrayTransfer interface which supports the Measurement object

location **(enum NADataStore)** Where the Data will be put. Choose from:

0 - naRawData

1 - naCorrectedData

2 - naMeasResult

3 - naRawMemory

4 - naMemoryResult

5 - naDivisor - When reading data from, or writing data to, the normalization divisor, you must first create a divisor trace using DataToDivisor Method.

See the Data Access Map

numPts **(long integer)** - Number of data points in the channel

data **(NAComplex)** - A one-dimensional array of Complex data matching the number of points in the current measurement.

format **(enum NADisplayFormat) -** Optional argument. Format of the data. If unspecified, naDataFormat_Polar is assumed. Only used when the destination store is naMeasResult or naMemoryResult.

Return Type Not Applicable

Default Not Applicable

Examples

```
Dim measData As IArrayTransfer
Set measData = app.ActiveMeasurement

measData.putNAComplex naMemoryResult, 201, dRawComplex(0)
```

C++ Syntax HRESULT putNAComplex(tagNADataStore DataStore, long lNumValues, TsComplex* pArrayOfComplex, tagDataFormat displayFormat)

Interface IArrayTransfer

Write-only

put_Output Method

Description	Writes a TTL HI or TTL Low to output pins 3 or 4 of the Material Handler IO connector. Each pin also has a latched output which is written to with USER. With the latched (USER) outputs, the value is not applied to the associated pin until a positive edge is detected at INPUT1 (pin 2).
VB Syntax	*handlerIo*.**put_Output** *(pin*) = *value*
Variable	**(Type) - Description**
handlerIo	**(object)** - A HandlerIO object
pin	**(enum as NAMatHandlerOutput)** - pin to write data to. Choose from: **naOutput1 - (0) -** pin3 **naOutput1User (1) -** pin3 latched (applied to pin 3 on positive edge of Input1-pin2) **naOutput2 (2) -** pin4 **naOutput2User (3) -** pin4 latched (applied to pin 4 on positive edge of Input1-pin2)
value	**(Variant)** Value to write to the selected pin. Choose from **0** - TTL LOW **1** - TTL HIGH
Return Type	Not Applicable
Default	0
Examples	`handlerIo.put Output(naOutput1)= 1`
C++ Syntax	HRESULT put_Output (tagNAMatHandlerOutput Output, VARIANT Data);
Interface	IHWMaterialHandlerIO

put_OutputVoltage Method

Description Writes voltages on the DAC/Analog Output 1 and Output 2 (pins 2 and 3 of the Aux I/O connector)

VB Syntax *AuxIO*.**put_OutputVoltage** *output, voltage*

Variable **(Type) - Description**

AuxIO **(object) -** A Hardware Auxiliary Input / Output object

output **(variant)** Number of the output DAC to write voltage to. Choose from:

1 Output DAC 1 - (pin 2)

2 Output DAC 2 - (pin 3)

voltage **(double)** Voltage to write to the output DAC. Choose a voltage from -10 to 10

Return Type None

Default None

Examples

```
HWAuxIO.put_OutputVoltage 1,9 'set Analog Out1 to +9v
```

C++ Syntax HRESULT put_OutputVoltage (VARIANT Output, double Voltage);

Interface IHWAuxIO

Write-only

put_OutputVoltageMode Method

Description	This command sets the mode of the selected "Analog Out" line on the Auxiliary IO connector. The modes give the user the option to have the requested voltage applied immediately or not until the sweep is done. To read the mode on each output use get_OutputVoltageMode Method.
VB Syntax	*auxIo*.**put_OutputVoltageMode** *(output, mode)*
Variable	**(Type) - Description**
auxIo	**(Object)** An AuxIO object
output	Analog Output to receive mode setting. Choose from **1** or **2**
mode	**(enum NAOutputVoltageMode)** **naWaitEOS** - While in this mode any voltage changes sent to the selected analog out will only get applied to the output between sweeps. **naNoWait** - While in this mode any voltage changes sent to the selected analog out will occur right away without waiting until the end of a sweep, the voltage gets applied immediately.
Return Type	NAOutputVoltageMode
Default	naWaitEOS
Examples	`auxIo.put_OutputVoltageMode 1, naWaitEOS   'Write`
C++ Syntax	HRESULT put_OutputVoltageMode(VARIANT Output, tagNAOutputVoltageMode dNewMode);
Interface	IHWAuxIO

put Port Method

Description Writes a value to the specified port. Use the get_Port Method to read the settings from the "readable" ports (C, D, E).

VB Syntax *handlerIo*.**put_Port** *(port*) = *value*

Variable **(Type) - Description**

handlerIo **(object)** - A HandlerIO object

port **(enum as NAMatHandlerPort)** - port to put data into. Choose from:

naPortA - (0)

naPortB - (1)

naPortC - (2)

naPortD - (3)

naPortE - (4)

naPortF - (5)

naPortG - (6)

naPortH - (7)

value The number of the data bits to set. The following table shows what the *value* represents:

Note: When writing to port G, port C must be set to output mode
When writing to port H, both port C and port D must be set to output mode. Use Port Mode Property

Port	Max allowable <num>	MSB..LSB 23..0	
A	255	A7...A0	Write-only
B	255	B7...B0	Write-only
C	15	C3...C0	Read-Write
D	15	D3...D0	Read-Write
E	255	D3...D0 + C3...C0	Read-Write
F	65535	B7...B0 + A7...A0	Write-only
G	1048575	C3...C0 + B7...B0 + A7...A0	Write-only
H	16777215	D3...D0 + C3...C0 + B7...B0 + A7...A0	Write-only

Return Type Not Applicable

Default Not Applicable

Examples

```
handlerIo.put Port(naPortB)= 15
```

C++ Syntax HRESULT put_Port (tagNAMatHandlerPort Port, VARIANT Data);

Interface IHWMaterialHandlerIO

put_PortCData Method

Description	Writes a 4-bit value to Port C on the Aux I/O connector (pins 22-25) and the Material Handler IO (pins 21-24 Anritsu) - (pins 22-25 Avantest). **Note**: These lines are connected to both the Handler IO and Aux IO in the PNA. Therefore, this command will affect both of these connectors in the same way.
VB Syntax	*AuxIO*.**put_PortCData** *num*
Variable	**(Type) - Description**
AuxIO	**(object) -** A Hardware Auxiliary Input / Output object
num	**(variant)** - 4 bit binary value. Choose from 0-15
Return Type	None
Default	None
Examples	`HWAuxIO.put_PortCData 15  'If Positive Logic, Port C lines C0, C1, C2, C3 go High. If Negative Logic, they go Low.`
C++ Syntax	HRESULT put_PortCData(VARIANT Data);
Interface	IHWAuxIO

PutDataScalar Method

Description Puts formatted variant scalar data into the measurement result buffer. The data will be immediately processed and displayed. Subsequent changes to the measurement state will be reflected on the display.

Always precede this command by setting the format on the measurement to be consistent with the format of the data being sent to the analyzer. In this way, the display annotation will be correct.

Execution of this command does not change the display format.

VB Syntax *meas*.**putDataScalar** *format, data*

Variable **(Type) - Description**

meas A measurement **(object)**

format **(enum NADataFormat)** Format of the data. **This value is presently ignored by the PNA.** Data is always presented in the current format.

Choose from:

0 - naDataFormat_LinMag

1 - naDataFormat_LogMag

2 - naDataFormat_Phase

3 - naDataFormat_Polar

4 - naDataFormat_Smith

5 - naDataFormat_Delay

6 - naDataFormat_Real

7 - naDataFormat_Imaginary

8 - naDataFormat_SWR

Notes:

- The getData (variant) method includes a "format" argument, which allows scalar (one-dimensional) data. To put data back into the "raw" data buffer using this (putDataComplex) method, specify **Polar** format when using the getData method.
- **Phase** format accepts data in radians (not degrees) and displays in degrees. To convert to degrees: radians * (57.29577951308233) = degrees. The getData method returns degrees if the request is for phase data.

data **(variant)** - A 1-dimension array of single precision floating point numbers.

Return Type Not Applicable

Default Not Applicable

Examples

```
' Put 201 points worth of scalar data into the measurement
' 200 is max index, so 0 to 200 is 201 points
Dim data(200) ' array of 201 (scalar) data points
' Fill the array
For i = 0 to 200
data(i) = i/200;
Next
app.ActiveMeasurement.putDataScalar 0, data
```

C++ Syntax HRESULT putDataScalar(tagNADataStore DataStore, VARIANT scalarArray)

Interface IMeasurement

Write-only

PutShortcut Method

Description Defines a Macro (shortcut) file in the analyzer. This command links a file name and path to the Macro file. You still need to put the macro file in the analyzer at the location indicated by this command.

VB Syntax *app.***PutShortcut** *index,title,path*

Variable **(Type) - Description**

app An Application **(object)**

index **(long)** - Number of the macro to be stored in the analyzer. If the index number already exists, the existing macro is replaced with the new macro.

title **(string)** - The name to be assigned to the macro

path **(string)** - Full pathname to the existing macro "executable" file.

Return Type Not Applicable

Default Not Applicable

Examples

```
app.PutShortcut 1,"Test","C:\Automation\MyTest.vbs"
```

C++ Syntax HRESULT PutShortcut(long Number, BSTR title, BSTR pathname)

Interface IApplication

putSourcePowerCalDataEx Method

Description

Note: This method replaces putSourcePowerCalData Method

Inputs source power calibration data (as variant data type) to this channel for a specific source port.

If the channel is sweeping the source backwards, then the first data point is the highest frequency value; the last data point is the lowest. Use the Get X-Axis Values command to return the X-axis values in the displayed order.

The calibration is not valid if the current number of points on the channel is not equal to the number of values that were input.

Note: This method sends variant data which is less efficient than methods available on the ISourcePowerCalData interface.

VB Syntax *chan*.**putSourcePowerCalData** *buffer, sourcePort, data*

Variable **(Type) - Description**

chan **(object)** – A Channel object

buffer **(enum NASourcePowerCalBuffer) -** The source power cal data buffer to write to.

0 - naCorrectionValues This is the only data buffer currently available.

sourcePort **(long integer)** – The source port for which calibration data is being requested.

data **(variant)** – Array of source power cal data being input.

Return Type None

Default Not Applicable

Examples

```
chan.putSourcePowerCalDataEx naCorrectionValues, 1, varData
```

C++ Syntax HRESULT putSourcePowerCalDataEx(tagNASourcePowerCalBuffer bufSelect, long sourcePort, VARIANT varData);

Interface IChannel4

putSourcePowerCalDataScalarEx Method

Description

Note: This method replaces putSourcePowerCalDataScalar Method

Inputs source power calibration data (as scalar values) to this channel for a specific source port.

If the channel is sweeping the source backwards, then the first data point is the highest frequency value; the last data point is the lowest. Use the Get X-Axis Values2 command to return the X-axis values in the displayed order.

VB Syntax *chanData*.**putSourcePowerCalDataScalarEx** *buffer, sourcePort, numValues, data*

Variable **(Type) - Description**

chanData **(interface)** – An ISourcePowerCalData2 interface on the Channel (object)

buffer **(enum NASourcePowerCalBuffer) -** The source power cal data buffer to write to.

0 - naCorrectionValues This is the only buffer currently available.

sourcePort **(long integer)** – The source port for which calibration data is being input.

numValues **(long integer)** – Number of data values being input.

Note: If this does not equal the current number of points on the channel, the calibration will not be valid.

data **(single)** – Array of source power cal data being input.

Return Type None

Default Not Applicable

Examples

```
Dim chanData As ISourcePowerCalData2
Set chanData = app.ActiveChannel
chanData.putSourcePowerCalDataScalarEx naCorrectionValues, 1,
201, scalarCalValues(0)
```

C++ Syntax HRESULT putSourcePowerCalDataScalarEx(tagNASourcePowerCalBuffer bufSelect, long sourcePort, long numValues, float *pData);

Interface ISourcePowerCalData2

Write-only

PutStandard Method

Description Puts data into the CalSet. Learn more about reading and writing Cal data using COM

Before calling this method you must open the Cal Set with OpenCal Set. If the Cal Set is not open, this method returns E_NA_Cal Set_ACCESS_DENIED.

The data is complex pairs. The server returns a variant containing a two-dimensional safe array. Memory for the returned Variant is allocated by the server and must be released by client.

Note: See also PutStandardComplex on the ICalData2 interface to avoid using the variant data type.

VB Syntax *obj.***putStandard** *class, rcv, src*, *data*

Variable **(Type) - Description**

obj **(object) -** A Calibrator or Cal Set object

class **(enum NACalClass)** Standard. Choose from:

1 - naClassA

2 - naClassB

3 - naClassC

4 - naClassD

5 - naClassE

6 - naReferenceRatioLine

7 - naReferenceRatioThru

SOLT Standards

1 - naSOLT_Open

2 - naSOLT_Short

3 - naSOLT_Load

4 - naSOLT_Thru

5 - naSOLT_Isolation

TRL Standards

1 - naTRL_Reflection

2 - naTRL_Line_Reflection

3 - naTRL_Line_Tracking

4 - naTRL_Thru

5 - naTRL_Isolation

rcv **(long)** - Receiver Port

src **(long)** - Source Port

data **(variant)** Two dimensional array (NUMPTS, 2)

Return Type Not Applicable

Default Not Applicable

Examples

```
Dim cmgr as CalManager
Dim cset As CalSet
Set cmgr = pna.GetCalManager
Set cset = cmgr.CreateCalSet(1)
cset.OpenCalSet naCalType_OnePort, 1
cset.putStandard naSOLT_Open, 1, 1, varOpen
cset.putStandard naSOLT_Short, 1, 1, varShort
cset.putStandard naSOLT_Load, 1, 1, varLoad
cset.ComputeErrorTerms
cset.CloseCalSet
cset.Description = "Uploaded one port cal"
guid = cset.GetGUID

End Sub
```

C++ Syntax HRESULT putStandard(tagNACalClass stdclass, long ReceivePort, long SourcePort, VARIANT varData)

Interface ICalibrator
ICalSet

Write-only

PutStandardByString

Description Writes a name array/pair to the calset.

VB Syntax **PutStandardByString**(*bufferName*, *vdata*)

Variable **(Type) - Description**

bufferName **(String)** The string used to specify a particular buffer in the calset. An example string requesting the data for the Load standard in a full 2 port cal might be "Full 2 Port Cal (2,3)::S11C(3,3)".

vdata **(Variant)** The variant containing a safearray of variants. This data is usually two dimensional.

Note: The vardata array is a safearray of variants wrapped in a variant. This structure is compatible with scripting clients who can only use variants. For alternative methods that used typed arrays, see ICalData3.

Return Type Not Applicable

Default Not Applicable

Examples See an example

C++ Syntax HRESULT PutStandardByString(BSTR bufferName, VARIANT vardata);

Interface ICalSet2

PutStandardComplex Method

Description Puts standards acquisition data into the Cal Set. Learn more about reading and writing Cal data using COM

Before calling this method you must open the Cal Set with OpenCal Set. If the Cal Set is not open, this method returns E_NA_Cal Set_ACCESS_DENIED.

VB Syntax *interface*.**putStandardComplex** *class, rcv, src, numPts,real(),imag()*

Variable **(Type) - Description**

interface A ICalData pointer to the Calibrator object **or**
A ICalData2 pointer to the Cal Set object

class **(enum NACalClass)** Standard. Choose from:

1 - naClassA

2 - naClassB

3 - naClassC

4 - naClassD

5 - naClassE

6 - naReferenceRatioLine

7 - naReferenceRatioThru

SOLT Standards

1 - naSOLT_Open

2 - naSOLT_Short

3 - naSOLT_Load

4 - naSOLT_Thru

5 - naSOLT_Isolation

TRL Standards

1 - naTRL_Reflection

2 - naTRL_Line_Reflection

3 - naTRL_Line_Tracking

4 - naTRL_Thru

5 - naTRL_Isolation

rcv **(long integer)** - Receiver Port

src **(long integer)** - Source Port

numPts **(long integer)** - number of data points in the arrays being sent.

real()	**(single)** - one-dimensional array containing the **real** part of the acquisition data. (0:points-1)
imag()	**(single)** - one-dimensional array containing the **imaginary** part of the acquisition data. (0:points-1)

Return Type Not Applicable

Default Not Applicable

Examples

```
Dim sdata As ICalData2
Set sdata = calmanager.CreateCal Set( 1 )
sdata.putStandardComplex naSOLT_Open, 1, 1, numpts, rel(0),
img(0)
```

C++ Syntax HRESULT putStandardComplex(tagNACalClass stdclass, long ReceivePort, long SourcePort, long lNumValues, float* pReal, float* pImag)

Interface ICalData
ICal Set

Write-only

PutStandardComplexByString

Description	Writes a name/array pair to the calset.
VB Syntax	**PutStandardComplexByString**(*bufferName, lnumPoints , real(o) , imag(0)*)
Variable	**(Type) - Description**
bufferName	**(String)** The string used to identify a particular buffer in the calset. An example string for port 3 directivity in a full 2 port cal might be "Full 2 Port Cal (2,3)::Directivity(3,3)".
lnumpoints	**(Long)** The size of the arrays in arguments 4 and 5.
real	**(Single)** The real component of the complex data.
imag	**(Single)** The imaginary component of the complex data. **Note:** The size of the real and imaginary arrays should be the same.
Return Value	Single
Default	Not Applicable
Examples	See example
C++ Syntax	HRESULT PutStandardComplexByString(BSTR bufferName, long lnumPoints, float* real, float* imag);
Interface	ICalData3

Write-only

Quit Method

Description	Terminates the Network Analyzer application.
VB Syntax	*app*.**Quit**
Variable	**(Type) - Description**
app	An Application **(object)**
Return Type	Not Applicable
Default	Not Applicable
Examples	`app.Quit`
C++ Syntax	HRESULT Quit()
Interface	IApplication
Remarks	Under the rules of COM, the server should not exit until all references to it have been released. This method is a brute force way of terminating the application. Be sure to release all references (or terminate the client program) before attempting to restart the Network Analyzer application. An alternate approach to terminating the application is to make the application invisible (app.Visible = False) and release all references. The server will shutdown.

ReadData Method

Description Reads a 13-bit data word from the specified address. Data is read using the AD0 through AD12 lines of the external test set connector. The instrument generates the appropriate timing signals. It automatically controls timing signals LDS, LAS and RLW to strobe the address, and then read the data, from the external test set. See the timing diagram for Address and Data I/O read.

VB Syntax *value* = *ExtIO*.**ReadData** *(address)*

Variable **(Type) - Description**

value **(variant)** - Variable to store the returned data

ExtIO **(object) -** An ExternalTestSetIO object

address **(variant)** - address to read data from.

Return Type Variant

Default Not Applicable

Examples

```
value = ExtIO.ReadData (15)
```

C++ Syntax HRESULT ReadData (VARIANT Address, VARIANT* Data);

Interface IHWExternaTestSetIO

ReadRaw Method

Description Reads a 16-bit value from the external test set. The 16-bit value is comprised of lines AD0 - AD12, Sweep Holdoff In and Interrupt In (inverted).

When this command is used the analyzer does NOT generate the appropriate timing signals; it simply reads the lines. The user needs to first use the WriteRaw method to do the initial setup. The RLW line (pin25) must be set to the appropriate level in order to read the test set connected.

Below is the format of data that is read with ReadRaw:

Pin	Bit	Signal name
22	0	AD0*
23	1	AD1*
11	2	AD2*
10	3	AD3*
9	4	AD4*
21	5	AD5*
20	6	AD6*
19	7	AD7*
6	8	AD8*
5	9	AD9*
4	10	AD10*
17	11	AD11*
3	12	AD12*
2	13	Sweep Holdoff In
13	14	Interrupt In (inverted internally)
na	15	Always Zero, grounded internally

*These lines are dependent on the state of RLW (pin25).
Writing a 0(low) to RLW will set lines AD0-AD12 to write mode.
Writing a 1(high) to RLW will set lines AD0-AD12 to read mode.

VB Syntax *value* = *ExtIO*.**ReadRaw** *(address)*

Variable **(Type) - Description**

value **(variant)** - Variable to store the returned data

ExtIO **(object) -** An External IO object

address **(variant)** - Address to read data from

Return Type Real

Default Not Applicable

Examples

```
value = ExtIO.ReadRaw (address)
```

C++ Syntax HRESULT ReadRaw(VARIANT* Input);

Interface IHWExternalTestSetIO

Recall Method

Description Recalls a measurement state, calibration state, or both from the hard drive into the analyzer.

Use app.Save to save the measurement and calibration state.

VB Syntax *app.***Recall** (*filename.ext*)

Variable **(Type) - Description**

app An Application **(object)**

filename.ext **(string)** - Filename and extension of the file to be recalled.
Valid extensions:

- .sta - Instrument State
- .cal - Calibration file
- .cst - Both Instrument State and Calibration file
- .cti - Citifile (data will always be formatted. See Recalling Citifiles Using the PNA)

Files are stored in the default folder "C:\Program Files\Agilent\Network Analyzer\Documents
To recall from a different folder, specify the pathname in the *filename.ext*. argument.

Return Type Not Applicable

Default Not Applicable

Examples

```
app.Recall (mystate.sta) 'Recalls "mystate.sta" from the default folder
app.Recall ("C:\Program Files\Agilent\Network Analyzer\Documents\Newfolder\MyState.cst) 'Recalls "mystate.cst" from the specified folder
```

C++ Syntax HRESULT Recall(BSTR bstrFile)

Interface IApplication

Recall Kits Method

Description	Recalls the calibration kits definitions that were stored with the SaveKits command.
VB Syntax	*app.***RecallKits**
Variable	**(Type) - Description**
app	An Application **(object)**
Return Type	Not Applicable
Default	Not Applicable
Examples	`app.RecallKits`
C++ Syntax	HRESULT RecallKits()
Interface	**I**Application

Write-only

Remove Method

Description	Removes an item from a collection of objects.
VB Syntax	*Object.***Remove** *item*
Variable	**(Type) - Description**
Object	Any of the following **(objects)** CalFactorSegments collection Cal Sets collection Measurements collection NAWindows collection PowerLossSegments collection Segments collection **Note:** Segments, CalFactorSegments, and PowerLossSegments have an OPTIONAL argument [size] referring to the number of segments to remove, starting with the *item* parameter.
item	**(variant)** - Item number to be removed
Return Type	Not Applicable
Default	Not Applicable

Examples

```
Measurements.Remove 3 'Removes measurement 3
segments.Remove 2,20 'Removes 20 segments (2 - 21)
```

C++ Syntax

HRESULT Remove(VARIANT index); //Measurements

HRESULT Remove(VARIANT index); //Cal Sets

HRESULT Remove(long windowNumber); //NAWindows

HRESULT Remove(VARIANT index, long size); //Segments

HRESULT Remove(VARIANT index, long size); //CalFactorSegments

HRESULT Remove(VARIANT index, long size); //PowerLossSegments

Interface

IMeasurements

INAWindows

ISegments

ICalFactorSegments
ICal Sets

IPowerLossSegments

About Presetting the Analyzer

Write-only

Reset Method

Description	Removes all existing windows and measurements from the application. (Unlike Preset, does not create a new measurement.)
VB Syntax	*app.***Reset**
Variable	**(Type) - Description**
app	An Application **(object)**
Return Type	Not Applicable
Default	Not Applicable
Examples	`app.Reset`
C++ Syntax	HRESULT Reset()
Interface	IApplication

Write-only

RestoreCalKitDefaults Method

Description Restores the original properties of the specified Cal Kit, overwritting the last definition with the factory defaults.

VB Syntax *app.***RestoreCalKitDefaults** (*calKit*)

Variable **(Type) - Description**

app An Application **(object)**

calKit **(enum NACalKit) -** Calibration Kit to restore. Choose from:

1 - naCalKit_85032F_N50
2 - naCalKit_85033E_3_5
3 - naCalKit_85032B_N50
4 - naCalKit_85033D_3_5
5 - naCalKit_85038A_7_16
6 - naCalKit_85052C_3_5_TRL
7 - naCalKit_User7
8 - naCalKit_User8
9 - naCalKit_User9
10 - naCalKit_User10

Return Type Not Applicable

Default Not Applicable

Examples

```
app.RestoreCalKitDefaults naCalKit_MechKit10
```

C++ Syntax HRESULT RestoreCalKitDefaults\(tagNACalKit kit)

Interface IApplication

RestoreCalKitDefaultsAll Method

Description	Restores the original properties of ALL of the Cal Kits, overwritting the last definitions with the factory defaults.
VB Syntax	*app.***RestoreCalKitDefaultsAll**
Variable	**(Type) - Description**
app	An Application **(object)**
Return Type	Not Applicable
Default	Not Applicable
Examples	`app.RestoreCalKitDefaultsAll`
C++ Syntax	HRESULT RestoreCalKitDefaultsAll()
Interface	IApplication

Resume Method

Description	Resumes the trigger mode of all channels that was in effect before sending the channels.Hold method. Channels.Hold must be sent before channels.Resume, using the same instance of the Channels object.
VB Syntax	*chans.***Resume**
Variable	**(Type) - Description**
chans	A Channel collection **(object)**
Return Type	Not Applicable
Default	Not Applicable
Examples	`chans.Resume`
C++ Syntax	HRESULT Resume();
Interface	IChannels2

Write-only

Save Method

Description Saves the appropriate content to the hard drive depending on the file suffix provided. See the table below. Some saved files can be recalled using app.Recall. depending on the content.

VB Syntax *app.***Save**(*filename.ext*)

Variable **(Type) - Description**

app An Application **(object)**

filename.ext **(string)** - Filename and extension of the file to be saved.
Extensions:

- **.cst** - Saves both Instrument State and Cal Set reference - Recalls a calibrated measurement. (Recallable)
- **.sta** - Saves Instrument State only - recalls the instrument state without calibration. (Recallable)
- **.cal** - Calibration file – saves the active Cal Sets currently in use by any channel. Use this mode for archival purposes only. All Cal Sets are saved to a Cal Set data file. This mode provides a method of safeguarding calibration data. This data can be restored to the list of Cal Sets available in the instrument. (Recallable)
- **.prn** - Saves active trace in comma-separated format (not recallable)
- **.bmp** - Saves a Bitmap of the screen (not recallable)
- **.s1p** - Saves 1-port measurement data (not recallable)
- **.s2p** - Saves 2-port measurement data (not recallable)

Files are saved to the default folder "C:\Program Files\Agilent\Network Analyzer\Documents. To save to a different folder, specify the pathname in the *filename.ext*. argument.

Return Type Not Applicable

Default Not Applicable

Examples

```
app.Save(mystate.sta) 'Saves "mystate.sta" to the default folder
app.Save("C:\Program Files\Agilent\Network
Analyzer\Documents\Newfolder\MyState.cst) 'Saves "mystate.cst"
to the specified folder
```

C++ Syntax HRESULT Save(BSTR bstrFile)

Interface IApplication

Save Method

Description	Saves the current Cal Set to disk (recommended method for saving a Cal Set). **Note:** There is also a Save method on the ICalManager and Calibrator interface which saves every Cal Set that currently exists in the instrument to the disk. Learn more about reading and writing Cal data using COM
VB Syntax	*CalSet*.**Save**
Variable	**(Type) - Description**
CalSet	**(object) -** A Cal Set object
Return Type	Not Applicable
Default	Not Applicable
Examples	`myCalSet.Save` See Copy Method for an example application of this command.
C++ Syntax	HRESULT Save();
Interface	ICalSet

Write-only

SaveCalSets Method

Description Writes new or changed Cal Sets out to disk. All Cal Sets are saved in a single file. This file is updated at the following times:

- When a Cal Set has been deleted.
- When a calibration has been performed through the front panel interface.
- When this method is called.
- When ICalSet::Save is called.

Note: The recommended method to save a Cal Set is ICalSet::Save which saves the data for **only** the current Cal Set to the disk.
This method (ICalManager/Calibrator::SaveCalSets) saves **every** Cal Set that currently exists in the instrument to the disk.

Learn more about reading and writing Cal data using COM

VB Syntax *object.***SaveCalSets**

Variable **(Type) - Description**

object **(object) -** A CalManager object or a Calibrator object

Return Type None

Default Not Applicable

Example

```
calMgr.SaveCalSets
```

C++ Syntax HRESULT SaveCalSets( );

Interface ICalManager
ICalibrator

SaveCitiDataData Method

Description	Saves UNFORMATTED trace data to .cti file. Learn more about citifiles.
VB Syntax	*app.***SaveCitiDataData**(*filename.cti*)
Variable	**(Type) - Description**
app	An Application **(object)**
filename.cti	**(string)** - Filename and .cti extension of the file to be saved. Files are saved to the default folder "C:\Program Files\Agilent\Network Analyzer\Documents. To save to a different folder, specify the pathname in the *filename.cti*. argument.
Return Type	Not Applicable
Default	Not Applicable
Examples	`app.SaveCitiDataData ("myDDCitifile.cti") 'Saves "myDDCitifile.cti" to the default folder` `app.SaveCitiDataData ("C:\Program Files\Agilent\Network Analyzer\Documents\Newfolder\myDDCitifile.cti") 'Saves "myDDCitifile.cti" to the specified folder`
C++ Syntax	HRESULT SaveCitiDataData (BSTR bstrFile)
Interface	IApplication5

About Save/Recall

SaveCitiFormattedData Method

Description Saves FORMATTED trace data to .cti file. Learn more about citifiles.

VB Syntax *app.***SaveCitiFormattedData**(*filename.cti*)

Variable **(Type) - Description**

app An Application **(object)**

filename.cti **(string)** - Filename and .cti extension of the file to be saved.

Files are saved to the default folder "C:\Program Files\Agilent\Network Analyzer\Documents. To save to a different folder, specify the pathname in the *filename.cti*. argument.

Return Type Not Applicable

Default Not Applicable

Examples

```
app.SaveCitiFormattedData ("myFDCitifile.cti") 'Saves
"myFDCitifile.cti" to the default folder
```

```
app.SaveCitiFormattedData ("C:\Program Files\Agilent\Network
Analyzer\Documents\Newfolder\myFDCitifile.cti") 'Saves
"myFDCitifile.cti" to the specified folder
```

C++ Syntax HRESULT SaveCitiFormattedData (BSTR bstrFile)

Interface IApplication5

Write only

SaveFile Method

Description	Saves the mixer/converter test setup to a mixer attributes (.mxr) file.
VB Syntax	*mixer*.**SaveFile** *(filename)*
Variable	**(Type) - Description**
mixer	An IMixer Interface pointer to a measurement **(object)**
filename	(String) Path and file name of the .mxr file
Return Type	String
Default	Not Applicable
Examples	`mixer.SaveFile (File_Name)`
C++ Syntax	HRESULT put_SaveFile(BSTR newVal)
Interface	IMixer

Write-only

SaveKits Method

Description	Saves the cal kits, typically after modifying a calibration kit. To load a cal kit into the analyzer from the hard drive, use app.RecallKits.
VB Syntax	*app*.**SaveKits**
Variable	**(Type) - Description**
app	An Application **(object)**
Return Type	Not Applicable
Default	Not Applicable
Examples	`app.SaveKits`
C++ Syntax	HRESULT SaveKits()
Interface	IApplication

About Marker Search

Write-only

SearchFilterBandwidth Method

Description Searches the measurement data with the current BandwidthTarget (default is -3). To continually track the filter bandwidth, use BandwidthTracking.

This feature uses markers 1-4. If not already, they are activated. To turn off these markers, either turn them off individually or DeleteAllMarkers.

The bandwidth statistics are displayed on the analyzer screen. To get the bandwidth statistics, use either GetFilterStatistics or FilterBW, FilterCF , FilterLoss ,or FilterQ.

The analyzer screen will show either Bandwidth statistics OR Trace statistics; not both.

To search a UserRange with the bandwidth search, first activate marker 1 and set the desired UserRange. Then send the SearchFilterBandwidth command. The user range used with bandwidth search only applies to marker 1 searching for the max value. The other markers may fall outside the user range.

VB Syntax *meas.***SearchFilterBandwidth**

Variable **(Type) - Description**

meas A Measurement **(object)**

Return Type Not Applicable

Default Not Applicable

Examples

```
meas.SearchFilterBandwidth
```

C++ Syntax HRESULT SearchFilterBandwidth()

Interface IMeasurement

About Marker Search

Write-only

SearchMax Method

Description	Searches the marker domain for the maximum value.
VB Syntax	*mark.***SearchMax**
Variable	**(Type) - Description**
mark	A Marker **(object)**
Return Type	Not Applicable
Default	Not Applicable
Examples	`mark.SearchMax`
C++ Syntax	HRESULT SearchMax()
Interface	IMarker

SearchMin Method

Description	Searches the marker domain for the minimum value.
VB Syntax	*mark.***SearchMin**
Variable	**(Type) - Description**
mark	A Marker **(object)**
Return Type	Not Applicable
Default	Not Applicable
Examples	`mark.SearchMin`
C++ Syntax	HRESULT SearchMin()
Interface	IMarker

About Marker Search

Write-only

SearchNextPeak Method

Description	Searches the marker's domain for the next peak value.
VB Syntax	*mark*.**SearchNextPeak**
Variable	**(Type) - Description**
mark	A Marker **(object)**
Return Type	Not Applicable
Default	Not Applicable
Examples	`mark.SearchNextPeak`
C++ Syntax	HRESULT SearchNextPeak()
Interface	IMarker

Write-only

SearchPeakLeft Method

Description	Searches the marker's domain for the next **VALID** peak to the left of the marker.
VB Syntax	*mark.***SearchPeakLeft**
Variable	**(Type) - Description**
mark	A Marker **(object)**
Return Type	Not Applicable
Default	Not Applicable
Examples	`mark.SearchPeakLeft`
C++ Syntax	HRESULT SearchPeakLeft()
Interface	IMarker

About Marker Search

Write-only

SearchPeakRight Method

Description	Searches the marker's domain for the next **VALID** peak to the right of the marker.
VB Syntax	*mark.***SearchPeakRight**
Variable	**(Type) - Description**
mark	A Marker **(object)**
Return Type	Not Applicable
Default	Not Applicable
Examples	`mark.SearchPeakRight`
C++ Syntax	HRESULT SearchPeakRight()
Interface	IMarker

Write-only

SearchTarget Method

Description	Searches the marker's domain for the target value (specified with mark.TargetValue). Searches to the right; then at the end of the search domain, begins again at the start of the search domain.
VB Syntax	*mark.***SearchTarget**
Variable	**(Type) - Description**
mark	A Marker **(object)**
Return Type	Not Applicable
Default	Not Applicable
Examples	`mark.SearchTarget`
C++ Syntax	HRESULT SearchTarget()
Interface	IMarker

SearchTargetLeft Method

Description	Moving to the left of the marker position, searches the marker's domain for the target value (specified with mark.TargetValue).
VB Syntax	*mark.***SearchTargetLeft**
Variable	**(Type) - Description**
mark	A Marker **(object)**
Return Type	Not Applicable
Default	Not Applicable
Examples	`mark.SearchTargetLeft`
C++ Syntax	HRESULT SearchTargetLeft()
Interface	IMarker

SearchTargetRight Method

Description	Moving to the right of the marker position, searches the marker's domain for the target value (specified with mark.TargetValue).
VB Syntax	*mark.***SearchTargetRight**
Variable	**(Type) - Description**
mark	A Marker **(object)**
Return Type	Not Applicable
Default	Not Applicable
Examples	`mark.SearchTargetRight`
C++ Syntax	HRESULT SearchTargetRight()
Interface	IMarker

Write-only

SelectCalSet Method

Description	Selects a Cal Set to apply to the measurements on the calling channel. If the cal set's GUID is not found, this method returns E_NA_Cal Set_NOT_FOUND. **Note:** Error Correction is not automatically applied as a result of this command being issued. If there is more than one Cal Type in the Cal Set, you must explicitly choose the Cal Type you want to apply. (See meas.Caltype)I
VB Syntax	*channel.***SelectCalSet** *GUID, restore*
Variable	**(Type) - Description**
channel	**(object) -** A Channel object
GUID	**(string)** - GUID number of the Cal Set to select
restore	**(boolean)** - **True (1)** - The stimulus stored with the cal set will be applied to the channel. **False (0)** - If a conflict is detected between the existing channel settings and the Cal Set stimulus settings, then the following will occur: If interpolation is ON, then interpolation will be attempted. This may fail if the channel frequency is outside the range of the Cal Set. If interpolation is OFF, the selection will be abandoned and an error is returned: E_NA_CAL_STIMULUS_VALUES_EXCEEDED
Return Type	Not Applicable
Default	Not Applicable
Example	`channel.SelectCalSet GUID, 1`
C++ Syntax	HRESULT SelectCalSet (BSTR strGUID, bool bRestore);
Interface	IChannel

SetAllSegments Method

Description Uploads a segment table to the PNA replacing any existing segment table.

- Segments must be ascending in frequency and non-overlapping. If they are not, the segments are 'adjusted' as they are from the front panel control.
- The total number of points for all segments cannot exceed the PNA maximum for a sweep.
- Can set independent IFBW and port power.

Note: For a 4-port PNA, send either 1 port power value or 4 port power values. Sending 2 or 3 values will result in an error.

See an example that uses this command.

VB Syntax *Segs*.**SetAllSegments** (*segdata*)

Variable **(Type) - Description**

segs A Segments **(Collections)**

segdata Variant or Double Array - Segment data

For VARIANT, the underlying type must be appropriate for the element:

- Boolean - segment on/off
- Integer - number of points
- Double - all other elements.

Return Type Not Applicable

Default Not Applicable

Examples See an example using this command

C++ Syntax SetAllSegments (VARIANT Segments);

Interface ISegments2

Write-only

SetCalInfo Method

Description Specifies the type of calibration. This method should be the first method called on the calibrator object. It prepares the internal state for the rest of the calibration. Learn more about reading and writing Cal data using COM

VB Syntax *cal*.**SetCalInfo(***type,rcvPort,srcPort***)**

Variable **(Type) - Description**

cal A Calibrator **(object)**

type **(enum NACalType) -** Calibration type. Choose from:

0 - naCalType_Response_Open
1 - naCalType_Response_Short
2 - naCalType_Response_Thru
3 - naCalType_Response_Thru_And_Isol
4 - naCalType_OnePort
5 - naCalType_TwoPort_SOLT
6 - naCalType_TwoPort_TRL
7 - naCalType_None
8 - naCalType_ThreePort_SOLT

Note: The analyzer can measure both ports simultaneously, assuming you have two of each standard type. For a 2-port cal, See cal.Simultaneous2PortAcquisition

Note: For 1-port cals, the source port = receiver port. For 2-port SOLT and TRL, it doesn't matter which port is specified as source and receiver

rcvPort **(long integer)** - Receiver Port

srcPort **(long integer)** - Source Port

Return Type NACalType

Default **7- naCalType_None**

Examples

```
cal.setCalInfo(naCalType_Response_Open,1,1)
```

C++ Syntax HRESULT SetCalInfo(tagNACalType calType,long portA, long portB)

Interface ICalibrator

SetCalInfo2 Method (for source power cals)

Description Specifies the technique, the channel, and the source port to be used for the source power calibration about to be performed.

Note: This method replaces SetCalInfo Method.

VB Syntax *powerCalibrator*.**SetCalInfo2** *calMethod, channel, sourcePort, [powerOffset,] [display]*

Variable **(Type) - Description**

powerCalibrator **(object)** - A SourcePowerCalibrator3 object

calMethod **(enum NASourcePowerCalMethod)** Selects the calibration method to be used for the source power cal acquisition.

0 – naPowerMeter Power Meter is used for all readings. (same as "fast iteration" box not checked on dialog box)

1 - naPowerMeterAndReceiver Power meter for the first iteration; then use the reference receiver for remaining readings if necessary (same as "fast iteration" box checked on dialog box)

channel **(long integer) -** Number of the PNA channel (not power meter channel) on which the source power cal will be performed. If the channel does not already exist, it will be created.

sourcePort **(long integer) -** Port number on which the source power cal will be performed.

[powerOffset] **(double) -** Optional argument. Sets or returns a power level offset from the PNA test port power. This can be a gain or loss value (in dB) to account for components you connect between the source and the reference plane of your measurement. For example, specify 10 dB to account for a 10 dB amplifier at the input of your DUT. Following the calibration, the PNA power readouts are adjusted by this value. This argument performs the same function as chan.SourcePowerCalPowerOffset Property

[display] **(boolean)** Optional argument. Enables and disables the display of power readings on the PNA screen. After the source power cal data is acquired, this setting is reset to ON. If unspecified, value is set to ON.

True - Display of power readings is ON

False - Display of power readings is OFF

Return Type None

Default Not Applicable

Examples

```
powerCalibrator.SetCalInfo2 naPowerMeter, 1, 1, -10, True
```

C++ Syntax HRESULT SetCalInfo2(tagNASourcePowerCalMethod enumCalMethod, long Channel, long SourcePort, double PowerOffset = 0., VARIANT_BOOL bDisplay = VARIANT_TRUE);

Interface ISourcePowerCalibrator3

SetCenter Method

Description	Changes the center stimulus to the stimulus value of the marker. The start stimulus stays the same and the stop is adjusted. **Note**: This command does not work with channels that are in CW or Segment Sweep mode.
VB Syntax	*mark*.**SetCenter**
Variable	**(Type) - Description**
mark	A Marker **(object)**
Return Type	Not Applicable
Default	Not Applicable
Examples	`mark.SetCenter`
C++ Syntax	HRESULT SetCenter()
Interface	IMarker

SetCW Method

Description	Changes the analyzer to sweep type CW mode and sets the CW frequency to the marker's frequency. Does not change anything if current sweep type is other than a frequency sweep.
VB Syntax	*mark.***SetCW**
Variable	**(Type) - Description**
mark	A Marker **(object)**
Return Type	Not Applicable
Default	Not Applicable
Examples	`mark.SetCW`
C++ Syntax	HRESULT SetCW()
Interface	IMarker

SetElectricalDelay Method

Description	Changes the measurement's electrical delay to the marker's delay value.
VB Syntax	*mark.***SetElectricalDelay**
Variable	**(Type) - Description**
mark	A Marker **(object)**
Return Type	Not Applicable
Default	Not Applicable
Examples	`mark.SetElectricalDelay`
C++ Syntax	HRESULT SetElectricalDelay()
Interface	IMarker

Write/Read

SetFailOnOverRange Method

Description When set TRUE, configures the analyzer to report outOfRange conditions with an **error** code. Any overrange error will return **E_NA_LIMIT_OUTOFRANGE_ERROR**.

Note: This method is for the benefit of VB clients. The analyzer automatically adjusts overrange conditions to the closest acceptable setting. The VB user will not See that an overrange occurred because the HRESULT is not returned if it has a success code. For more information, See Events/OverRange.

VB Syntax *app*.**SetFailOnOverRange** *state*

Variable **(Type) - Description**

app An Application **(object)**

state **(boolean)** -
True (1) - Overrange conditions report an error code
False (0) - Overrange conditions report a success code

Return Type Not Applicable

Default False (0)

VB Example

```
app.SetFailOnOverRange TRUE
On Error Goto ERRHANDLER

'the following overrange will cause ERRHANDLER to be invoked

channel.StartFrequency = 9.9 GHZ
exit
```

```
ERRHANDLER:
  print "something failed"
```

C++ Syntax HRESULT put_SetFailOnOverRange(VARIANT_BOOL mode)

Interface IApplication

SetPowerAcquisitionDevice Method

Description	Sets the power sensor channel (A or B) to be used. This performs the same function as the **Use this sensor only** checkbox in the Power Sensor Settings dialog. **Note:** This method is only necessary when performing an SMC calibration.
VB Syntax	*pwrCal*.**SetPowerAcquisitionDevice** *sensor*
Variable	**(Type) - Description**
pwrCal	(Object) A SourcePowerCalibrator object
sensor	(**enum NAPowerAcquisitionDevice**) The power sensor channel. Choose from: 0 – naPowerSensor_A 1 – naPowerSensor_B
Default	Not Applicable
Examples	`pwrCal.PowerAcquisitionDevice naPowerSensor_A`
C++ Syntax	HRESULT SetPowerAcquisitionDevice(tagNAPowerAcquisitionDevice enumAcqDevice);
Interface	ISourcePowerCalibrator3

Write-only

SetFrequencyLowPass Method

Description	Set the start frequencies when **trans.Mode = LowPass**.
VB Syntax	*trans.***SetFrequencyLowPass**
Variable	**(Type) - Description**
trans	A Transform **(object)**
Return Type	Not Applicable
Default	Not Applicable
Examples	`trans.SetFrequencyLowPass`
C++ Syntax	HRESULT SetFrequencyLowPass(void)
Interface	ITransform

SetReferenceLevel Method

Description	Changes the measurement's reference level to the marker's Y-axis value.
VB Syntax	*mark*.**SetReferenceLevel**
Variable	**(Type) - Description**
mark	A Marker **(object)**
Return Type	Not Applicable
Default	Not Applicable
Examples	`mark.SetReferenceLevel`
C++ Syntax	HRESULT SetReferenceLevel()
Interface	IMarker

Write-only

SetStandardsForClass Method

Description	Set the calibration standard numbers for a specified calibration class. To read the cal standard numbers use GetStandardsForClass Method
VB Syntax	*calKit*.**SetStandardsForClass** *(calclassorder, std1, std2, std3, std4, std5, std6, std7)*
Variable	**(Type) - Description**
calKit	A CalKit **(object)**
calclassorder	**(enum NACalClassOrder)** Cal Class. Choose from: 0 - naRefl_1_S11 1 - naRefl_2_S11 2 - naRefl_3_S11 3 - naTran_1_S21 4 - naRefl_1_S22 5 - naRefl_2_S22 6 - naRefl_3_S22 7 - naTran_1_S12 8 - naRefl_1_S33 9 - naRefl_2_S33 10 - naRefl_3_S33 11 - naTran_1_S32 12 - naTran_1_S23 13 - naTran_1_S31 14 - naTran_1_S13 15 - naTRL_T 16 - naTRL_R 17 - naTRL_L
std1…std7	**(long)** Calibration Standard Number. Choose from **1** through **30**. Std2 through Std7 are optional
Return Type	Not applicable
Default	Not applicable
Examples	**calkit.SetStandardsForClass** naRefl_3_S11, 3, 5, 6 **calkit.SetStandardsForClass** naTran_1_S21, 4

C++ Syntax HRESULT SetStandardsForClass(NACalClassOrder calclassorder, long std1, long std2, long std3, long std4, long std5, long std6, long std7)

Interface ICalKit

Write-only

SetStart Method

Description Changes the start stimulus to the stimulus value of the marker. The stop stimulus stays the same and the span is adjusted.

Note: This command does not work with channels that are in CW or Segment Sweep mode.

VB Syntax *mark.***SetStart**

Variable **(Type) - Description**

mark A Marker **(object)**

Return Type Not Applicable

Default Not Applicable

Examples

```
mark.SetStart
```

C++ Syntax HRESULT SetStart()

Interface IMarker

SetStop Method

Description Changes the stop stimulus to the stimulus value of the marker. The start stimulus stays the same and the span is adjusted.

Note: This command does not work with channels that are in CW or Segment Sweep mode.

VB Syntax *mark.***SetStop**

Variable **(Type) - Description**

mark A Marker **(object)**

Return Type Not Applicable

Default Not Applicable

Examples

```
mark.SetStop
```

C++ Syntax HRESULT SetStop()

Interface IMarker

ShowMarkerReadout Method

Description	Shows and Hides the Marker readout for the active marker in the upper-right corner of the window.
VB Syntax	*win.***ShowMarkerReadout** *state*
Variable	**(Type) - Description**
win	A NAWindow **(object)**
state	**(boolean) -** **True (1)** - Show the Marker readout **False (0)** - Hide the Marker readout
Return Type	Not Applicable
Default	Not Applicable
Examples	`win.ShowMarkerReadout True`
C++ Syntax	HRESULT ShowMarkerReadout(VARIANT_BOOL bState)
Interface	INAWindow

Write-only

ShowStatusBar Method

Description Shows and Hides the Status Bar. The Status Bar is located across the bottom of the display. The following information is shown for the active measurement:

- Channel number
- Parameter
- Correction On or Off
- Remote or Local operation

VB Syntax *app.***ShowStatusBar** *state*

Variable **(Type) - Description**

app An Application **(object)**

state **(boolean)** -
True (1) - Show the Status Bar
False (0) - Hide the Status Bar

Return Type Not Applicable

Default Not Applicable

Examples

```
app.ShowStatusBar True
```

C++ Syntax HRESULT ShowStatusBar (VARIANT_BOOL bState)

Interface IApplication

ShowStimulus Method

Description	Shows and Hides the Stimulus (X-axis) information located at the bottom of the display. The start and stop stimulus values are shown for the active measurement.
VB Syntax	*app.***ShowStimulus** *state*
Variable	**(Type) - Description**
app	An Application **(object)**
state	**(boolean) -** **True (1)** - Show the Stimulus information **False (0)** - Hide the Stimulus information
Return Type	Not Applicable
Default	Not Applicable
Examples	`app.ShowStimulus True`
C++ Syntax	HRESULT ShowStimulus(VARIANT_BOOL bState)
Interface	IApplication

Write-only

ShowTable Method

Description Shows or Hides the specified table for the window's active measurement in the lower part of the window.

VB Syntax *win.***ShowTable** *value*

Variable **(Type) - Description**

win A NAWindow **(object)**

value **(enum naTable) -** The table to show or hide. Choose from:

0 - naTable_None
1 - naTable_Marker
2 - naTable_Segment
3 - naTable_Limit

Return Type Not Applicable

Default Not Applicable

Examples

```
win.ShowTable naTable_limit
```

C++ Syntax HRESULT ShowTable (tagNATableType table)

Interface INAWindow

ShowTitleBars Method

Description Shows and Hides the Title Bars. The Title Bars are across the top of the Network Analyzer Window and each of the measurement windows. The Window name is shown in the Title Bar.

VB Syntax *app.***ShowTitleBars** *state*

Variable **(Type) - Description**

app An Application **(object)**

state **(boolean)**
True (1) - Show the Title Bars
False (0) - Hide the Title Bars

Return Type Not Applicable

Default Not Applicable

Examples

```
app.ShowTitleBars True
```

C++ Syntax HRESULT ShowTitleBars\(VARIANT_BOOL bState)

Interface IApplication

ShowToolbar Method

Description Shows and Hides the specified Toolbar.

VB Syntax *app.***ShowToolbar** *toolbar,state*

Variable **(Type) - Description**

app An Application **(object)**

toolbar **(enum NAToolbarType) -** The toolbar to show or hide. Choose from:

0 - naToolbar_None
1 - naToolbar_ActiveEntry
2 - naToolbar_Markers
3 - naToolbar_Measurement
4 - naToolbar_Stimulus
5 - naToolbar_SweepControl

state **(boolean)** -
True (1) - Show the specified toolbar
False (0) - Hide the specified toolbar

Return Type Not Applicable

Default 1 - naToolbar_ActiveEntry showing; all others hiding.

Examples

```
app.ShowToolbar 1,1 'shows the active entry toolbar
```

C++ Syntax HRESULT ShowToolbar(tagNAToolbarType toolbar, VARIANT_BOOL bState)

Interface IApplication

Single Method

Description Sets the trigger count to 1, which will cause the channel to respond to exactly one trigger signal from any source (internal, external, or manual).

VB Syntax *chan*.**Single** *[sync]*

Variable **(Type) - Description**

chan A Channel **(object)**

[sync] **(boolean)** -Optional argument. A variable set to either True or False.

True - The analyzer waits until the trigger is completed to process subsequent commands.
False - Subsequent commands are processed immediately.

Return Type Not Applicable

Default Not Applicable

Examples

```
sync = True
chan.Single sync
```

C++ Syntax HRESULT Single(VARIANT_BOOL bWait)

Interface IChannel

StringToNACalClass Method

Description Converts the returned strings from GetStandardsList into the enumeration (NACalClass) and the port numbers required for PutStandard and GetStandard methods that transmit data in and out of the Cal Set.

Learn more about reading and writing Cal data using COM

VB Syntax *CalSet*.**StringToNACalClass** (*list*, *std, rcv, src)*

Variable **(Type) - Description**

CalSet **(object) -** A Cal Set object

list **(string)** - a string containing the textual description of the standard.

std **(enum NACalClass)** Choose from:

1 - naClassA

2 - naClassB

3 - naClassC

4 - naClassD

5 - naClassE

6 - naReferenceRatioLine

7 - naReferenceRatioThru

SOLT Standards

1 - naSOLT_Open

2 - naSOLT_Short

3 - naSOLT_Load

4 - naSOLT_Thru

5 - naSOLT_Isolation

TRL Standards

1 - naTRL_Reflection

2 - naTRL_Line_Reflection

3 - naTRL_Line_Tracking

4 - naTRL_Thru

5 - naTRL_Isolation

rcv **(long)** - port number of the receiver

src **(long)** - port number of the source

Return Type Not Applicable

Default Not Applicable

Examples **guid = CalSet.StringToNACalClass(*list, std, rcv, src)***

C++ Syntax HRESULT StringtoNACalClass (BSTR* str, NACalClass* item, long *rcv, long *src);

Interface ICalSet

StringtoNAErrorTerm2 Method

Description	Converts the returned strings from GetErrorTermList into the enumeration (NAErrorTerm2) and the port numbers required for PutErrorTerm and GetErrorTerm methods that transmit data in and out of the Cal Set. Learn more about reading and writing Cal data using COM
VB Syntax	*Cal Set*.**StringToNAErrorTerm2** (*list, eterm, rcv, src*)
Variable	**(Type) - Description**
Cal Set	**(object) -** A Cal Set object
list	**(string)** - a string containing the textual description of the error term.
eterm	**(enum As NaErrorTerm2).** Choose from: 0 - naET_Directivity (rcv = src) 1 - naET_SourceMatch (rcv = src) 2 - naET_ReflectionTracking (rcv = src) 3 - naET_TransmissionTracking (rcv != src) 4 - naET_LoadMatch (rcv != src) 5 - naET_Isolation (rcv != src)
rcv	**(long) -** port number of the receiver
src	**(long)** - port number of the source
Return Type	Not Applicable
Default	Not Applicable
Examples	`CalSet.StringToNAErrorTerm2 str, term, rcv, src`
C++ Syntax	HRESULT StringToNAErrorTerm2 (BSTR* str, NAErrorTerm2* item, long *rcv, long *src);
Interface	ICalSet

UserPreset Method

Description Performs a User Preset. There must be an active User Preset state file (see UserPresetLoadFile and UserPresetSaveState) or an error will be returned.

Regardless of the state of the User Preset Enable checkbox, the app.Preset command will always preset the PNA to the factory preset settings, and app.UserPreset will always perform a User Preset.

VB Syntax *app.*__UserPreset__

Variable **(Type) - Description**

app An Application **(object)**

Return Type Not Applicable

Default Not Applicable

Examples

```
app.UserPreset
```

C++ Syntax HRESULT UserPreset()

Interface IApplication6

UserPresetLoadFile Method

Description	Loads an existing instrument state file (.sta or .cst) to be used for User Preset. Subsequent execution of app.UserPreset will cause the PNA to assume this instrument state. Regardless of the state of the User Preset Enable checkbox, the app.Preset command will always preset the PNA to the factory preset settings, and app.UserPreset will always perform a User Preset.
VB Syntax	*app.***UserPresetLoadFile** *(file)*
Variable	**(Type) - Description**
app	An Application **(object)**
file	**(String)** Full path, name, and extension of the file to be loaded.
Return Type	Not Applicable
Default	Not Applicable
Examples	`app.UserPresetLoadFile ("C:\Program Files\Agilent\Network Analyzer\Documents\10MHzto20GHz.sta")`
C++ Syntax	HRESULT UserPresetLoadFile (BSTR bstrFile)
Interface	IApplication6

UserPresetSaveState Method

Description	Saves the current instrument settings as UserPreset.sta. Subsequent execution of app.UserPreset will cause the PNA to assume this instrument state. Regardless of the state of the User Preset Enable checkbox, the app.Preset command will always preset the PNA to the factory preset settings, and app.UserPreset will always perform a User Preset.
VB Syntax	*app.***UserPresetSaveState**
Variable	**(Type) - Description**
app	An Application **(object)**
Return Type	Not Applicable
Default	Not Applicable
Examples	`app.UserPresetSaveState`
C++ Syntax	HRESULT UserPresetSaveState()
Interface	IApplication6

WriteData Method

Description Writes a 13-bit value to the specified address using the AD0 through AD12 lines of the external test set connector. The PNA generates the appropriate timing signals. It automatically controls timing signals LDS, LAS and RLW to strobe the address, then the data, to the external test set. See the timing diagram for Address and Data I/O read.

VB Syntax *ExtIO*.**WriteData** *(address) = value*

Variable **(Type) - Description**

ExtIO **(object) -** An External IO object

address **(variant)** - Address to be written to.

value **(variant)** - 13-bit word to write

Return Type Not Applicable

Default Not Applicable

Examples

```
ExtIO.WriteData (15) = 12
```

C++ Syntax HRESULT WriteData(VARIANT Address, VARIANT Data);

Interface IHWExternaTestSetIO

Write-only

WriteRaw Method

Description

Writes a 16-bit value to the external test set connector lines AD0 - AD12, RLW, LAS and LDS. The analyzer does NOT generate the appropriate timing signals. The user has control of all 16 lines using this write method.

Note: When RLW (pin25) is set to 1 (high) it causes lines AD0 - AD12 to float. It disables their output latches and sets the hardware for reading. LDS and LAS are not affected by this behavior.

Below is the format of data that is written with WriteRaw:

* This Output will float if RLW (bit-13) is set high

Pin	Bit	Signal name
22	0	AD0*
23	1	AD1*
11	2	AD2*
10	3	AD3*
9	4	AD4*
21	5	AD5*
20	6	AD6*
19	7	AD7*
6	8	AD8*
5	9	AD9*
4	10	AD10*
17	11	AD11*
3	12	AD12*
25	13	RLW
24	14	LDS
8	15	LAS

VB Syntax	*ExtIO*.**WriteRaw** *value*
Variable	**(Type) - Description**
ExtIO	**(object) -** An External IO object
value	**(variant)** - Data to be written
Return Type	Not Applicable
Default	Not Applicable
Examples	`ExtIO.WriteRaw 12`
C++ Syntax	HRESULT WriteRaw(VARIANT Output);
Interface	IHWExternalTestSetIO

OnCalEvent

Description	Triggered by a calibration event. See a list of CAL Events. **Note:** Some Severe Events are also used as Error Messages
VB Syntax	Sub *app_*__OnCalEvent__(ByVal *eventID* As Variant, ByVal *chanNum* As Variant, ByVal *measNum* As Variant)
Variable	**(Type) - Description**
app	An Application **(object)**
eventID	Code number of the event which occurred
chanNum	Channel Number of the event
measNum	Measurement Number of the event
Return Type	Not Applicable
Default	Not Applicable

Examples

```
Sub pna_OnCalEvent(ByVal eventID As Variant, ByVal
channelNumber As Variant, ByVal measurementNumber As Variant)
'
MsgBox ("A Calibration event has occured")
End Sub
```

C++ Syntax	HRESULT OnCalEvent(VARIANT eventID, VARIANT channelNumber, VARIANT measurementNumber)
Interface	IApplication

OnChannelEvent

Description Triggered by a channel event. See a list of Channel Events

Note: Some Severe Events are also used as Error Messages

VB Syntax Sub *app*_**OnChannelEvent**(ByVal *eventID* As Variant, ByVal *chanNum* As Variant)

Variable	(Type) - Description
app	An Application **(object)**
eventID	Code number of the event which occurred
chanNum	Channel Number of the event
Return Type	Not Applicable
Default	Not Applicable

Examples

```
Sub pna_OnChannelEvent(ByVal eventID As Variant, ByVal
channelNumber As Variant)
  MsgBox "A channel event occured"
End Sub
```

C++ Syntax HRESULT OnChannelEvent(VARIANT eventID, VARIANT channelNumber)

Interface IApplication

OnDisplayEvent

Description Triggered by a display event. See a list of Display Events

Note: Some Severe Events are also used as Error Messages

VB Syntax Sub *app*_**OnDisplayEvent**(ByVal *eventID* As Variant, ByVal *winNum* As Variant, ByVal *traceNum* As Variant)

Variable	(Type) - Description
app	An Application **(object)**
eventID	Code number of the event which occurred
winNum	Window Number of the event
traceNum	Trace Number of the event
Return Type	Not Applicable
Default	Not Applicable

Examples

```
Sub pna_OnDisplayEvent(ByVal eventID As Variant, ByVal
windowNumber As Variant, ByVal traceNumber As Variant)
MsgBox ("A Display event has occured")
End Sub
```

C++ Syntax HRESULT OnDisplayEvent(VARIANT eventID, VARIANT windowNumber, VARIANT traceNumber)

Interface IApplication

OnHardwareEvent

Description Triggered by a hardware event. See a list of Hardware Events

Note: Some Severe Events are also used as Error Messages

VB Syntax Sub *app*_**OnHardwareEvent**(ByVal *eventID* As Variant)

Variable	(Type) - Description
app	An Application **(object)**
eventID	Code number of the event which occurred

Return Type Not Applicable

Default Not Applicable

Examples

```
Private Sub pna_OnHardwareEvent(ByVal eventID As Variant)
MsgBox ("A Hardware event has occured")
End Sub
```

C++ Syntax HRESULT OnHardwareEvent(VARIANT eventID)

Interface IApplication

OnMeasurementEvent

Description Triggered by a measurement event. See a list of Measurement Events.

Note: Some Severe Events are also used as Error Messages

VB Syntax Sub *app*_**OnMeasurementEvent**(ByVal *eventID* As Variant, ByVal *measNum* As Variant)

Variable	(Type) - Description
app	An Application **(object)**
eventID	Code number of the event which occurred
measNum	Measurement Number of the event
Return Type	Not Applicable
Default	Not Applicable

Examples

```
Private Sub pna_OnMeasurementEvent(ByVal eventID As Variant, ByVal
measurementNumber As Variant)

MsgBox ("A Measurement event has occured")

End Sub
```

C++ Syntax HRESULT OnMeasurementEvent(VARIANT eventID, VARIANT measurementNumber)

Interface IApplication

OnSCPIEvent

Description Triggered by a SCPI event. See a list of SCPI Events

Note: Some Severe Events are also used as Error Messages

VB Syntax Sub *app*_**OnSCPIEvent**(ByVal *eventID* As Variant)

Variable	**(Type) - Description**
app	An Application **(object)**
eventID	Code number of the event which occurred
Return Type	Not Applicable
Default	Not Applicable

Examples

```
Private Sub pna_OnSCPIEvent(ByVal eventID As Variant)
MsgBox ("A SCPI event has occured")
End Sub
```

C++ Syntax HRESULT OnSCPIEvent(VARIANT eventID)

Interface IApplication

OnSystemEvent

Description Triggered by a system event. See a list of System Events, also known as general events.

Note: Some Severe Events are also used as Error Messages

VB Syntax Sub *app*_**OnSystemEvent**(ByVal *eventID* As Variant)

Variable	(Type) - Description
app	An Application **(object)**
eventID	Code number of the event which occurred
chanNum	Channel Number of the event

Return Type Not Applicable

Default Not Applicable

Examples

```
Private Sub pna_OnSystemEvent(ByVal eventID As Variant)
MsgBox ("A System event has occured")
End Sub
```

C++ Syntax HRESULT OnSystemEvent(VARIANT eventID)

Interface IApplication

OnUserEvent

Description Reserved for future use.

VB Syntax Sub app_**OnUserEvent**

COM CalSet program examples using C++

- Get example, cycles through the calsets collection, printing values from each error term buffer.
- Put example, creates a calset and a buffer using SafeArrayCreate.
- Put and Get example, creates a calset, writes a buffer to it and reads it back.

Get example

This example cycles through the calsets collection, printing values from each error term buffer.

The example uses the methods **GetErrorTermList2 and GetErrorTermByString**. The methods **GetStandardsList2 and GetStandardByString** are used similarly.

```
#include "stdafx.h"
#include "atlbase.h"
using namespace std;
#import "C:\Program Files\Agilent\Network Analyzer\835x.tlb" raw_interfaces_only, nam
inline void HR( HRESULT hr)
{
 if (FAILED(hr))
  throw hr;
}
int main()
{
 CoInitialize(NULL);
 CComPtr spPNA;
 if (FAILED(CoCreateInstance(CLSID_Application, NULL, CLSCTX_SERVER, IID_IApplication
 {
  MessageBox(NULL, "could not create PNA","",0);
  return 1;
 }
 try {

  long setCount;
  CComBSTR unfiltered("");
  // ** initialize interface handles
  CComPtr spMgr;
  HR(spPNA->GetCalManager(&spMgr ));
  CComPtr spSets;
  HR(spMgr->get_CalSets( &spSets) );   // Get the calset collection
  HR(spSets->get_Count( &setCount));
  // ** loop through the collection
  for (int i = 1; i <= setCount;i++)
  {CComVariant itemNum(i);
   CComPtr spSet;
      CComQIPtr spSet2;
   HR(spSets->Item( itemNum, &spSet  ));
   HR(spSet->QueryInterface( &spSet2 ));

   VARIANT buflist;
   HR(spSet2->GetErrorTermList2(0, unfiltered, &buflist));
```

```
   // ** loop through all the error term buffers in the calset
   VARIANT* pvStrings;
   HR(SafeArrayAccessData( buflist.parray, (void**)&pvStrings));
   for ( int bufNum = 0; bufNum <buflist.parray->rgsabound[0].cElements; bufNum++ )
   {
    VARIANT vOut;
    BSTR bufName = pvStrings[bufNum].bstrVal;
    HR(spSet2->GetErrorTermByString( 0, bufName, &vOut));
    cout <<(LPCTSTR) CString(bufName)
    "\n";
    if
     (vOut.parray->cDims != 2) throw 1;
    long indices[2];
    char formatted[100];
    int maxpts = vOut.parray->rgsabound[1].cElements;
    int maxparts = vOut.parray->rgsabound[0].cElements;
    for (int pt = 0; pt <maxpts; pt++)
     {
     indices[0]="pt;"
     indices[1]="0;"
     VARIANT valReal,valImag;          SafeArrayGetElement(
     vOut.parray,indices,&valReal);
     indices[1]++;
     &valImag);
      sprintf(formatted,"[%d]:%f\t%f\n",pt,valReal.fltVal,valImag.fltVal);
     cout<<formatted;
     }
     HR(SafeArrayUnaccessData(buflist.parray));
     catch(HRESULT hr)CComBSTR bstrMsg;
     spPNA->get_MessageText((enum NAEventID) hr, &bstrMsg );
 }
 spPNA.Release();
 CoUninitialize();
 return 0;
}
```

Put example

This example creates a calset and a buffer using SafeArrayCreate.

The example uses the methods **PutErrorTermByString** method to put the buffer in the calset. The **PutStandardByString** is a similarly used method.

```
// PutErrorTermByString.cpp : Defines the entry point for the console application.
//
#include "stdafx.h"
#include "atlbase.h"

#import "C:\Program Files\Agilent\Network Analyzer\835x.tlb" raw_interfaces_only, nam
inline void HR( HRESULT hr)
{
```

```
 if (FAILED(hr))
  throw hr;
}
using namespace std;
int main()
{
 CoInitialize(NULL);
 CComPtr spPNA;
 if (FAILED(CoCreateInstance(CLSID_Application, NULL, CLSCTX_SERVER, IID_IApplication
 {
  MessageBox(NULL, "could not create PNA","",0);
  return 1;
 }
 try {
  HR(spPNA->Preset());

  // generate a safearray of floats

  SAFEARRAYBOUND bounds[2];
   bounds[0].cElements = 201;
  bounds[0].lLbound = 0;
  bounds[1].cElements = 2;
  bounds[1].lLbound = 0;
  float realPart = 1.0;
  float imagPart = 0.0;
  long indices[2];
  long maxPts = 201;
  SAFEARRAY* psa = SafeArrayCreate( VT_R4, 2, bounds );
  for (int pt = 0 ; pt <maxpts; pt++
  )
  {
  indices[0]="pt;"
  indices[1]="1;"
  realPart +="pt;"
  HR(SafeArrayPutElement(psa,indices,(void*)&realPart));
  (void*)&imagPart));
  }
  //wrap the array in Variant
  for IDispatch complexData;
  complexData.vt="VT_ARRAY;"
  complexData.parray="psa;"
  Create a calset and put buffer.
  CComPtr spMgr;
  spSet;
  CComQIPtr spSet2;
  HR(spPNA->GetCalManager(&spMgr ));
  HR(spMgr->CreateCalSet(1, &spSet));
  spSet2 = spSet;
  CComBSTR bufName("MyPhonyCalSet:MyPhonyBuffer");
  HR(spSet2->PutErrorTermByString( bufName, complexData));
  HR(spSet2->Save());
 }
 catch (HRESULT hr)
 {
  CComBSTR bstrMsg;
  spPNA->get_MessageText((enum NAEventID) hr, &bstrMsg );
  MessageBox( NULL, (LPCTSTR)CString(bstrMsg), "Error",MB_OK);
```

```
 }
 spPNA.Release();
 CoUninitialize();
 return 0;
```

Put and Get example for ICalData3

This example creates a calset, writes a buffer to it and reads it back. The example uses the methods **PutErrorTermComlexByString** and **GetErrorTermComlexByString** methods. The **PutStandardComplexByString** and **GetStandardComplexByString** methods are used similarly.

```
#include "stdafx.h"
#include "atlbase.h"
#include <iostream>
#include <vector>

#import "C:\Program Files\Agilent\Network Analyzer\835x.tlb" raw_interfaces_only, nam
inline void HR( HRESULT hr)
{
 if (FAILED(hr))
  throw hr;
}
using namespace std;
int main()
{
 CoInitialize(NULL);
 CComPtr spPNA;
 if (FAILED(CoCreateInstance(CLSID_Application, NULL, CLSCTX_SERVER, IID_IApplication
 {
  MessageBox(NULL, "could not create PNA","",0);
  return 1;
 }
 try {
  HR(spPNA->Preset());

  // generate some data for our calset buffer
  std::vector<float> real(201,0);
  std::vector<float> imag(201,0);
  for (int i = 0; i <real.size(); i++)
   {
   real[i]="(float)i;"
   }
   //needed interface pointers CComPtr <spMgr>;
  CComPtr <spset>;
  CComQIPtr <spcaldata3>;
  // Create a calset
  HR(spPNA->GetCalManager(&spMgr ));
  HR(spMgr->CreateCalSet(1, &spSet));
  spCalData3 = spSet;
  // insert a buffer
  CComBSTR bufName("Example Cal Set:Bogus Data Buffer");
  HR(spCalData3->PutErrorTermComplexByString( bufName, real.size(), &real[0], &imag[0
  HR(spSet->Save());
  // read the buffer back out
  long pts = real.size();
```

```
  real.assign(pts,0);
  imag.assign(pts,0);
  HR(spCalData3->GetErrorTermComplexByString( 0, bufName, &pts, &real[0], &imag[0]));
 }
 catch (HRESULT hr)
 {

  CComBSTR bstrMsg;
  spPNA->get_MessageText((enum NAEventID) hr, &bstrMsg );
  MessageBox( NULL, (char*)_bstr_t(bstrMsg.m_str), "Error",MB_OK);
 }
 spPNA.Release();
 CoUninitialize();
 return 0;
}
```

Getting Trace Data from the Analyzer

This Visual Basic program:

- Retrieves Scalar Data from the Analyzer and plots it.
- Retrieves Paired Data from the Analyzer and plots it.
- Retrieves Complex Data from the Analyzer and plots it.

To use this code, prepare a form with the following:

- Two MSCharts named **MSChart1** and **MSChart2**
- Three buttons named **GetScalar, GetPaired, GetComplex**

Note: You can get MSChart in Visual Basic by clicking **Project / Components / Microsoft Chart Control**

```
'Put this in a module
Public dlocation As NADataStore
Public numpts As Long
Public fmt As NADataFormat
Public app As Application
Public measData As IArrayTransfer
Public chan As Channel
```

```
Sub Form_Load()
'Change analyzerName to your analyzer's full computer name
Set app = CreateObject("AgilentPNA835x.Application", "analyzerName")
```

```
Set measData = app.ActiveMeasurement
Set chan = app.ActiveChannel

'To pick a location to get the data from remove the comment from one of these
dlocation = naRawData
'dlocation = naCorrectedData
'dlocation = naMeasResult
'dlocation = naRawMemory
'dlocation = naMemoryResult

'setup MSChart1 and MSChart2
'right click on the chart and select:
' - line chart
' - series in rows
End Sub
```

```
Sub GetComplex_Click()
ReDim Data(numpts) As NAComplex
Dim Real(201) AS Single
Dim Imag(201) AS Single
numpts = chan.NumberOfPoints
```

```
'You cannot change the format of Complex Data
Call trigger
'get data
measData.GetNAComplex dlocation, numpts, Data(0)
```

```
'plot data
Dim i As Integer

For i = 0 To numpts - 1
 Real(i) = Data(i).Re
 Imag(i) = Data(i).Im
Next i
MSChart1 = Real()
MSChart2.Visible = True
MSChart2 = Imag()
Call Sweep
End Sub
```

```
Sub GetPaired_Click()
ReDim Real(numpts) As Single
ReDim Imag(numpts) As Single
numpts = chan.NumberOfPoints
```

```
' To pick a format, remove the comment from one of these
fmt = naLogMagPhase
'fmt = naLinMagPhase
Call trigger
'Get data
measData.getPairedData dlocation, fmt, numpts, Real(0), Imag(0)
'Plot Scalar
MSChart1 = Real()
MSChart2.Visible = True
MSChart2 = Imag()
Call Sweep
End Sub
```

```
Sub GetScalar_Click()
ReDim Data(numpts) As Single
```

```
numpts = chan.NumberOfPoints
'To pick a format remove the comment from one of these
fmt = naDataFormat_LogMag
'fmt = naDataFormat_LinMag
'fmt = naDataFormat_Phase
'fmt = naDataFormat_Delay
'fmt = naDataFormat_Real
'fmt = naDataFormat_Imaginary
Call trigger
'Get data
measData.GetScalar dlocation, fmt, numpts, Data(0)
'Plot Data
MSChart1 = Data()
MSChart2.Visible = False
Call Sweep
End Sub
```

```
Sub trigger()
```

```
'The analyzer sends continuous trigger signals
app.TriggerSignal = naTriggerInternal
```

```
'The channel will only accept one, then go into hold
'Sync true will wait for the sweep to complete
```

```
sync=True
```

```
chan.Single sync
End Sub
```

```

Sub Sweep()
'The channel goes back to accepting all triggers
chan.Continuous
End Sub
```

Perform a Source Power Cal using COM

This program can be run in either Visual Basic 6 or as a VBScript program. The PNA can run *.vbs programs as macros.

This program demonstrates:

- Performing a source power calibration of Port 2 for Channel 1.
- Reading the calibration data.

Learn more about Power Calibrations

See an example that Uploads a Source Power Cal

See Other COM Example Programs

To run this program, you need:

- One of the following power meters connected to the PNA through GPIB: E4416A, E4417A, E4418A/B, E4419A/B, 437B, 438A, EPM-441A, EPM-442A

Note: If your power meter is other than these, you can create your own Power Meter Driver using our template.

- Your PC and PNA both connected to a LAN (for communicating with each other).

To make this program work in VBS, save the following code in a text editor file such as Notepad and save as *.vbs.

To make this program work in Visual Basic 6:

1. Create a new project
2. Click **Project**, **Add New Module**, click **Open**.
3. Paste the following code into the code window.
4. Delete the first two lines (comment and Main)
5. Click **Project, Properties**. Under **Startup Object**, select **Sub Main**
6. Click **Project, References**, and select the Agilent PNA Series Type Library.

```
' Run the Main subroutine
Main
Public Sub Main()
Dim PNA, chan, pwrcal                                ' PNA COM objects
Const naPowerMeter = 0, naPowerMeterAndReceiver = 1 ' enum NASourcePowerCalMethod
Const naPowerSensor_A = 0                            ' enum NAPowerAcquisitionDevice
Const naCorrectionValues = 0                         ' enum NASourcePowerCalBuffer
Const port = 2                                       ' PNA port #2 as source port
Const offset = 0                                     ' cal power offset value
Const bDisplay = True                                ' whether to display data during
acquire
```

```
Dim stimulus, calvalues, strResult

' Instantiate our PNA COM objects
Set PNA = CreateObject("AgilentPNA835x.Application")
Set chan = PNA.Channels(1)
Set pwrcal = PNA.SourcePowerCalibrator

' Set the number of sweep points to 21 on Channel 1.
chan.NumberOfPoints = 21

' Specify the GPIB address of the power meter
' that will be used in performing the calibration.
pwrcal.PowerMeterGPIBAddress = 13

' Turn use of the loss table OFF (this assumes there is
' virtually no loss in the RF path to the power sensor
' due to a splitter, coupler or adapter).
pwrcal.UsePowerLossSegments = False

' Turn frequency checking OFF (so one power sensor is used for the entire cal
' acquisition sweep regardless of frequency span).
pwrcal.UsePowerSensorFrequencyLimits = False

' Specify a nominal power accuracy tolerance (IterationsTolerance) in dB for the
' calibration, and the maximum number of iterations to adjust power at each point,
' attempting to achieve within tolerance of the desired power.  If at any stimulus
' point the power fails to reach within the set tolerance of the desired power
' after the maximum number of iterations, the power at that point will be set to the
' value determined by the last iteration (the Source Power Cal dialog box will
' indicate the FAIL, but we can still apply the cal if desired when it's complete).
' Each iteration is based upon a SETTLED power reading (see comments preceding the
' next two properties below).
pwrcal.IterationsTolerance = 0.1
pwrcal.MaximumIterationsPerPoint = 3

' The worst-case window of power uncertainty (for a calibration which meets
' tolerance) is the sum of the iteration tolerance and the power meter settling
' tolerance (which is described below).
' At each stimulus point, the PNA takes power meter readings and determines when
' they have settled by comparing the magnitude difference between consecutive
' readings versus a nominal dB tolerance limit (ReadingsTolerance) on that magnitude
' difference.  When consecutive readings are within tolerance of each other, or
' if they are not within tolerance but we've taken a maximum number of readings
' (ReadingsPerPoint), the PNA does a weighted average of the readings taken at that
' stimulus point and that is considered our settled power reading.
pwrcal.ReadingsTolerance = 0.1
pwrcal.ReadingsPerPoint = 5

' Setup of information pertaining to this specific cal acquisition.  Includes the
' method (type of devices) that will be used to perform the cal -- choose either
' naPowerMeter or naPowerMeterAndReceiver.  naPowerMeterAndReceiver uses the power
' meter for the first iteration of each point and the PNA's reference receiver for
' subsequent iterations, so is much faster than using power meter only
 naPowerMeter).
' But the power meter accounts for compression when calibrating at the output of an
' active device, whereas the reference receiver cannot unless it is coupled to the
' cal reference plane (on a PNA which allows direct access to the receivers).
```

```
' 'offset' specifies if the cal power level is offset (positive value for a gain,
' negative value for a loss) from the PNA port power setting on the channel when
' no source power cal is active.  This is to account for components between the PNA
' test port and cal reference plane.  In this example, we will calibrate at the PNA
' test port, so there is no offset (it is zero).
' 'bDisplay' indicates whether to display the source power cal dialog during the
' source power cal acquisition (the dialog will chart the corrected power readings).
pwrcal.SetCalInfo2 naPowerMeter, chan.channelNumber, port, offset, bDisplay

' Perform synchronous source power cal acquisition sweep using the sensor attached
' to Channel A of the power meter.  This assumes that the power sensor is already
' connected to Port 2 of the PNA.
pwrcal.AcquirePowerReadings naPowerSensor_A, True

' Conclude the calibration.  This applies the cal data to PNA channel memory,
' and turns the correction ON for Port 2 on Channel 1, but does NOT save the
' calibration.
pwrcal.ApplyPowerCorrectionValues

' At this point, if you choose to save the instrument state as a ".CST" file,
' the calibration will be saved with the instrument state in that file.
' Read the stimulus values from Channel 1.
stimulus = chan.GetXAxisValues

' Read the source power correction data.
calvalues = chan.getSourcePowerCalDataEx(naCorrectionValues, port)

' Print the data using a message box (here, Chr returns the ASCII characters
' for Tab (9) and Linefeed (10)).
strResult = "Stimulus" & Chr(9) & Chr(9) & "Cal Value" & Chr(10)
For i = 0 To UBound(stimulus)
strResult = strResult & stimulus(i) & Chr(9) & calvalues(i) & Chr(10)
Next
MsgBox strResult
End Sub
```

Upload a Source Power Cal using COM

This program can be run in either Visual Basic 6 or as a VBScript program. The PNA can run *.vbs programs as macros.

This program demonstrates:

- Uploading a source power calibration of Port 2 for Channel 1.
- Reading the calibration data.

Learn more about Power Calibrations

See Other COM Example Programs

To run this program you need:

- Your PC and PNA both connected to a LAN (for communicating with each other).

To make this program work in VBS, save the following code in a text editor file such as Notepad and save as *.vbs.

To make this program work in Visual Basic 6:

1. Create a new project
2. Click **Project**, **Add New Module**, click **Open**.
3. Paste the following code into the code window.
4. Delete the first two lines (comment and Main)
5. Click **Project, Properties**. Under **Startup Object**, select **Sub Main**
6. Click **Project, References**, and select the Agilent PNA Series Type Library.

```
' Run the Main subroutine
Main
Public Sub Main()
Dim PNA, chan                ' PNA COM objects
Const naCorrectionValues = 0 ' enum NASourcePowerCalBuffer
Const port = 2               ' PNA port #2 as source port
Dim stimulus, calvalues
Dim power, calpower, strResult
' Instantiate our PNA COM objects
Set PNA = CreateObject("AgilentPNA835x.Application")
Set chan = PNA.Channels(1)

' Set the number of sweep points to 2 on Channel 1.
chan.NumberOfPoints = 2

' Ensure there's currently no source power cal on for this channel and port.
chan.SourcePowerCorrection(port) = False
```

```
' Specify if the cal power level is offset (positive value for a gain, negative
' value for a loss) from the PNA port power setting on the channel when
' no source power cal is active.  This is to account for components
' between the PNA test port and cal reference plane.
' In this example, let's set up our calibration
' at the output of an amplifier with 15 dB gain.
chan.SourcePowerCalPowerOffset(port) = 15

' Send our source power correction data to the PNA.  For purpose of simplicity
' in this example, we'll set up for no correction (0) at our start stimulus and
' 0.5 dB at our stop stimulus (recall that our sweep currently has just 2 points).
calvalues = Array(0, 0.5)
chan.putSourcePowerCalDataEx naCorrectionValues, port, calvalues

' Set the number of sweep points to 21 on Channel 1.
chan.NumberOfPoints = 21

' Read the fixed power level for this port on Channel 1.
power = chan.TestPortPower(port)

' Turn the source power cal on.
chan.SourcePowerCorrection(port) = True

' Again read the fixed power level for this port on Channel 1
' (with our calibration turned on, this should now include the 15 dB offset
' we indicated our power amplifier provides).
calpower = chan.TestPortPower(port)

' Read the stimulus values from Channel 1.
stimulus = chan.GetXAxisValues

' Read back the source power correction data, now interpolated for 21 points
calvalues = chan.getSourcePowerCalDataEx(naCorrectionValues, port)

' Print the data using a message box (here, Chr returns the ASCII characters
' for Tab (9) and Linefeed (10)).
strResult = "PNA port power = " & power & Chr(10)
strResult = strResult & "Power at reference plane = " & calpower & Chr(10) & hr(10)
strResult = strResult & "Stimulus" & Chr(9) & Chr(9) & "Cal Value" & Chr(10)
For i = 0 To UBound(stimulus)
strResult = strResult & stimulus(i) & Chr(9) & calvalues(i) & Chr(10)
Next
MsgBox strResult
End Sub
```

Upload Segment Table

This VBScript program does the following:

- Creates an array of segment data
- Fills the array with start and stop frequencies, number of points, IFBW, and port power for each segment.
- Uploads the array to the PNA

This VBScript (*.vbs) program can be run as a macro in the PNA. To do this, copy the following code into a text editor file such as Notepad and save the file on the PNA hard drive as segTable.vbs. Learn how to setup and run the macro.

See Other COM Example Programs

```
' Create the application instance, and preset the application
Set app = CreateObject("AgilentPNA835x.Application")
app.Preset
Dim chan
Set chan = app.ActiveChannel
chan.sweeptype = 4

'To specify uncoupled (separate) port power, set the following to 0
chan.CouplePorts = 1

Dim segs
Set segs = chan.Segments
Dim win
Set win = app.NAWindows(1)
win.ShowTable 2

' Multipliers
kHz = 1000
MHz = kHz*1000
GHz = MHz*1000

' Create segments from 10MHz to 3GHz
StartFreq = 10 * MHz
StopFreq  =  3 * GHz
'
' Number of segments to create
numSegs = 20

'Create a 2-D array of segments.
'  1st dimension is number of settings; 9
'  2nd dimension is number of segments.
'  Each dimension is set to 1 less than desired size to allow for index 0
Dim segdata
ReDim segData(9, numSegs-1)
' Width of frequency segment, used below
SegmentWidth = (StopFreq-StartFreq)/10

' Fill all segments with data
```

```
For i = 0 To numSegs-1
' element 0=segment state (True-on | False-off)
segdata(0, i) = True

' element 1=Num Points in this segment
segdata(1, i) = 100

' element 2=Start Freq in Hz
segdata(2, i) = StartFreq + i * SegmentWidth

' element 3=Stop Freq in Hz
segdata(3, i) = segdata(2, i) + SegmentWidth

' element 4=IFBW in Hz
segdata(4, i) = 35000

' element 5=Dwell Time in sec
segdata(5, i) = 0

' element 6=Port 1 Power to -1 dBm
segdata(6, i) = -1

'The following 3 values are ignored if port power is coupled.
' element 7=Port 2 Power to -2 dBm
segdata(7, i) = -2

'The following 2 values are ignored on a 2-port PNA
' element 8=Port 3 Power to -3 dBm
segdata(8, i) = -3

' element 9=Port 4 Power to -4 dBm
segdata(9, i) = -4

Next

'Configure Independent segment settings
'Independent Dwell time setting is not currently available
segs.IFBandwidthOption = True
segs.SourcePowerOption = True

' Push the segment data into the PNA's Active Channel
segs.SetAllSegments segdata
```

Create and Cal an SMC Measurement

This example uses the creates and calibrates an SMC measurement. To run this example without modification you need the following:

- A Mixer setup file saved on the PNA: C:\Program Files\Agilent\Network Analyzer\Documents\Mixer\MyMixer.mxr.
- If the mixer file uses an external LO source, it must also be attached and configured.
- An ECAL module that covers the frequency range of the measurement.
- A power meter must be connected to the PNA using GPIB.

This VBScript (*.vbs) program can be run as a macro in the PNA. To do this, copy the following code into a text editor file such as Notepad and save on the PNA hard drive as SMC.vbs. Learn how to setup and run the macro.

See Other COM Example Programs

```
Dim App
Set App = CreateObject("AgilentPNA835x.Application")
Dim Meas
Set Meas = App.ActiveMeasurement
Meas.Delete

App.CreateCustomMeasurementEx 1, "SMC_Forward.SMC_ForwardMeas","SC21"
'Other valid strings that can be specified to create a measurement with a parameter
'other than'"SC21" are: "S11", "S22", "IPwr", and "OPwr"

Set Meas = App.ActiveMeasurement
'You can perform mixer setup here or
'recall a previous mixer setup from the PNA Hard drive
'Here are some basic mixer setup commands
Dim mix
Set mix = Meas

'Load your own SMC measurement from the PNA Hard drive.
Meas.LoadFile "C:\Program Files\Agilent\Network
Analyzer\Documents\Mixer\MyMixer.mxr"


Dim CalMgr
Set CalMgr = App.GetCalManager
Dim SMC
Set SMC = CalMgr.CreateCustomCal("SMC")
SMC.Initialize 1, 1
SMC.Do2PortEcal = 1 'specify 0 for mechanical cal, 1 for ecal
'use Factory Characterization
SMC.ECALCharacterization(1) = 0
'only specify the ThruCalMethod if you have a non-insertable DUT
'SMC.ThruCalMethod = "Adapter Removal"
'If you specify Adapter Removal or Unknown Thru calibration
'then you need to specify the connector types of your DUT.
```

```
'Specify a connector that is the same type as your selected
'ECAL characterization. The characterization selected in
'this case is APC 2.4 female APC 2.4 male so the connectors
'specified for my DUT have to be APC 2.4 but can be any sex.
'SMC.ConnectorType(1) = "APC 2.4 female"
'SMC.ConnectorType(2) = "APC 2.4 female"
SMC.OmitIsolation = 1
SMC.AutoOrient = 1
' 1- forward, 2-reverse, or Both
SMC.CalibrationPort = "1"

Dim steps
steps = SMC.GenerateSteps
For i = 1 To steps
MsgBox SMC.GetStepDescription(i)
SMC.AcquireStep i
Next

Dim calset
calset = SMC.GenerateErrorTerms
Msgbox("SMC Cal Complete!")
```

Create and Cal a VMC Measurement

The following VMC calibration options are presented in **VB Script** examples:

- Full 2-port ECAL
- For Mixer Characterization ONLY
- Mechanical Calibration

Note: Each of the following programs load a mixer setup file.
Substitute "Mixer009.mxr" with your own .mxr file.

Full 2-port ECAL

```
Dim App
Set App = CreateObject("AgilentPNA835x.Application")
Dim Meas
Set Meas = App.ActiveMeasurement
Meas.Delete

App.CreateCustomMeasurementEx ( 2, "FCA.VMCMeas", "VC21" )
'Other valid strings that can be specified to create a measurement with a parameter
other than"VC21" are: "S11", and "S22"
Set Meas = App.ActiveMeasurement

Meas.LoadFile "C:\\Program Files\\Agilent\\Network
Analyzer\\Documents\\mixer\\Mixer009.mxr"

Dim CalMgr
Set CalMgr = App.GetCalManager
Dim VMC
Set VMC = CalMgr.CreateCustomCal("VMC")
VMC.Initialize 1, 1
VMC.Do2PortEcal = 1
VMC.Do1PortEcal = 1
VMC.ECALCharacterization(1, 1) = 0 'APC 2.4 male APC 2.4 female
VMC.ECALCharacterization(1, 0) = 0
' could be Default, Flush Thru, Unknown Thru, or Adapter Removal
VMC.ThruCalMethod = "Adapter Removal"
VMC.OmitIsolation = 1
VMC.AutoOrient = 1
VMC.EcalOrientation1Port(1) = "A1"
VMC.CharacterizeMixerOnly = 0
VMC.LoadCharFromFile = 0
' to load mixer characterization from file, specify LoadCharFromFile = 1,
' then VMC.CharFileName = "C:\\Program Files\\Agilent\\Network
' Analyzer\\Documents\\YourFile.s2p" (specify your own .s2P filename)

VMC.ConnectorType(1) = "APC 2.4 female"
VMC.ConnectorType(2) = "APC 2.4 female"

Dim steps
steps = VMC.GenerateSteps
```

```
For i = 1 To steps
MsgBox VMC.GetStepDescription(i)
VMC.AcquireStep i
Next

Dim calset
calset = VMC.GenerateErrorTerms
```

Mixer Characterization ONLY

```
Dim App
Set App = CreateObject("AgilentPNA835x.Application")
Dim Meas
Set Meas = App.ActiveMeasurement
Meas.Delete
app.CreateCustomMeasurementEx ( 2, "FCA.VMCMeas", "VC21" )
'Other valid strings that can be specified to create a measurement with a parameter
other than "VC21" are: "S11", and "S22"

Set Meas = App.ActiveMeasurement
Meas.LoadFile "C:\\Program Files\\Agilent\\Network
Analyzer\\Documents\\mixer\\Mixer009.mxr"

Dim CalMgr
Set CalMgr = App.GetCalManager
Dim VMC
Set VMC = CalMgr.CreateCustomCal("VectorMixerCal.VMCType")
VMC.Initialize 1, 1
VMC.Do2PortEcal = 1
VMC.Do1PortEcal = 1
VMC.ECALCharacterization(1, 1) = 0
VMC.ECALCharacterization(1, 0) = 0
VMC.ThruCalMethod = "Default"
VMC.OmitIsolation = 1
VMC.AutoOrient = 1
VMC.EcalOrientation1Port(1) = "A1"
VMC.CharacterizeMixerOnly = 1
VMC.LoadCharFromFile = 0
VMC.CharFileName = "C:\\Program Files\\Agilent\\Network
Analyzer\\Documents\\MyMixerChar.s2p"

Dim steps
steps = VMC.GenerateSteps
For i = 1 To steps
MsgBox VMC.GetStepDescription(i)
VMC.AcquireStep i
Next

Dim calset
calset = VMC.GenerateErrorTerms
MsgBox VMC.CharFileName
```

Mechanical Calibration

```
Dim App
```

```
Set App = CreateObject("AgilentPNA835x.Application")
Dim Meas
Set Meas = App.ActiveMeasurement
Meas.Delete
app.CreateCustomMeasurementEx ( 2, "FCA.VMCMeas", "VC21" )
'Other valid strings that can be specified to create a measurement with a parameter
other than "VC21" are: "S11", and "S22"

Set Meas = App.ActiveMeasurement
Meas.LoadFile "C:\\Program Files\\Agilent\\Network
Analyzer\\Documents\\mixer\\Mixer009.mxr"
Dim CalMgr
Set CalMgr = App.GetCalManager
Dim VMC
Set VMC = CalMgr.CreateCustomCal("VectorMixerCal.VMCType")
VMC.Initialize 1, 1
VMC.Do2PortEcal = 1
VMC.Do1PortEcal = 0
VMC.ECALCharacterization(1, 1) = 0
VMC.ThruCalMethod = "Default"
VMC.OmitIsolation = 1
VMC.AutoOrient = 1
VMC.CharacterizeMixerOnly = 0
VMC.LoadCharFromFile = 0
VMC.ConnectorType(1) = "APC 3.5 female"
VMC.ConnectorType(2) = "APC 3.5 male"
VMC.CalKitType(1) = "85033D/E"
VMC.CalKitType(2) = "85033D/E"

Dim steps
steps = VMC.GenerateSteps
For i = 1 To steps
MsgBox VMC.GetStepDescription(i)
VMC.AcquireStep i
Next

Dim calset
calset = VMC.GenerateErrorTerms
```

Create an SMC Fixed Output Measurement

This Visual Basic example creates a calibrated SMC fixed output measurement using a controlled LO. Then a single sweep is taken and data is retrieved.

Fixed output measurements are only supported on SMC (not VMC) measurements. Fixed output measurements require that an external LO source be sweeping and synchronized with the PNA source. FCA can perform this synchronization using the external source configuration settings. The fastest, and recommended, method of controlling the LO source is Hardware List (BNC) triggering mode. However, in this mode, FCA channels will not respond to manual triggers. Therefore, the following example uses the "Single" IChannel::Single mechanism to trigger a sweep.

Both VMC and SMC measurements require that mixer parameters be setup before making the measurement. The mixer parameters are not applied until calling IMixer::Calculate.

You can also setup the mixer parameters using IMixer::LoadFile. With this interface, the specified mixer file is loaded and immediately applied.

```
option explicit
dim app
set app = createobject("agilentpna835x.application")
app.preset

' Put the channel in hold (highly recommended)
app.ActiveChannel.Hold 1

' Select calibration. Apply stimulus settings to the channel
app.ActiveChannel.SelectCalSet "{C9080B34-EF1F-4ED5-B07D-250B45962B99}",1

' Delete the standard measurement
app.ActiveMeasurement.Delete

' Create an SC21 measurement
app.CreateCustomMeasurementEx 1, "SMC_Forward.SMC_ForwardMeas","SC21"

' Set the number of points to 11
app.ActiveChannel.NumberOfPoints = 11

' Setup the mixer parameters for a swept LO, fixed output measurement

dim mixer
set mixer = app.ActiveMeasurement
mixer.InputStartFrequency = 200e6
mixer.InputStopFrequency = 700e6
mixer.LORangeMode(1) = 0  ' 0 = Swept mode
mixer.OutputFixedFrequency = 3.4e9
mixer.InputPower = -17
mixer.LOPower(1) = 10

' Specify the LO name, for controlled LO. This name can be setup in the External
Source Config Dialog
mixer.LOName(1) = "8360"

' The CALC method calculates the LO frequency from the other parameters,
' It also applies ALL mixer parameters to the channel.
mixer.Calculate 3 ' Calculate the LO range
```

```
' Create an S11 in the same channel
app.CreateCustomMeasurementEx 1, "SMC_Forward.SMC_ForwardMeas","S11"
dim S11Meas
set S11Meas = app.ActiveMeasurement

' Create an IPwr in the same channel
app.CreateCustomMeasurementEx 1, "SMC_Forward.SMC_ForwardMeas","IPwr"

' Create an OPwr in the same channel
app.CreateCustomMeasurementEx 1, "SMC_Forward.SMC_ForwardMeas","OPwr"

' Perform a single sweep synchronously.
app.ActiveChannel.Single 1
function ToString(complexDataArray)
dim dataAsString
dim point
for point = 0 to UBound(data)
dataAsString = dataAsString & "(" & data(point,0) & "," & data(point,1) & ") "
next
ToString = dataAsString
end function

' Retrieve the SC21 data
dim data
'Get the calibrated real/imaginary values
data = mixer.GetData(1,3)
wscript.echo ToString(data)

' Retrieve the S11 data
'Get the calibrated real/imaginary values
data = S11Meas.GetData(1,3)
wscript.echo ToString(data)
```

Create a Pulsed Measurement

The following example demonstrates how to create a pulsed measurement using the Pulsed Application DLL. It first gets valid configuration settings and then uses those settings to configure the PNA and external pulsed generators.

To run this program, you need:

- Pulsed Application (Option H08)
- IF Access (Option H11)
- External Pulse Generators
- External Pulse Modulator / Pulse Bias

Learn how to install and register the pulsed .dll on your PC

See the ConfigureNarrowBand3 Method for sending and returning parameters to the .dll.

See the documentation for the following COM IF Configuration commands.

See the SCPI IF Configuration commands.

Learn about the Pulsed Application.

```
'Interfaces
Dim OApp As AgilentPNA835x.Application
Dim OIF As AgilentPNA835x.IFConfiguration
```

```
'Pulsed parameters
Dim DPRF As Double
Dim DOffset As Double
Dim DSampleRate As Double
Dim LNumTaps As Long
Dim LBW As Long
Dim IPrecision as Integer
Dim BPG81110 As Boolean
Dim BFixedPRF As Boolean
```

```
'pulsed DLL interface
Dim OPulsed As New AgilentPNAPulsed.Application
DPRF=5123 'Hz
LBW=100 'Hz
BPG81110=True 'Using the Agilent 81110A Pulse Generator
BFixedPRF=True 'Do not change the PRF during filter alignment.  Only adjust the
IFBW.
```

```
'Calculate precision of pulse generators so that the config function returns the
correct precision with the right filter. For example, DPRF=5000 Hz with a pulse
generator that will only take a total of four numeric values
' (5.123 kHz)
'->log10(DPRF)=3.709
'->int(3.709)=3
'->3-3=0
```

```
'The algorithm will use a 10^x value for decrementing the PRF for null computation.
 This means that the first numeric digit from the right should be the one that is
decremented by the pulsed algorithm (i.e. 5.122 kHz) to compute the filter nulls.
This ensures that the pulse generators receive a PRF that is within their precision
with the associated nulling IFBW.
```

```
IPrecision = Int(Flog10(CSng(DPRF))) - 3
```

```
'Send desired pulsed parameters to the pulsed configuration DLL. The DLL will return a new set of pulse parameters that provide the proper filter nulling.
OPulsed.ConfigNarrowBand3 DPRF, LNumTaps, LBW, DOffset, DSampleRate, IPrecision, BFixedPRF,BPG81110
```

```
'Send configuration to PNA
```

```
'Connect to the PNA application
Set OApp = CreateObject("AgilentPNA835x.Application")
Set OIF = OApp.ActiveChannel.IFConfiguration
```

```
'Set PNA IFBW close to that returned by pulsed algorithm.  This ensures that the proper settling time is set on the PNA.
```

```
OApp.ActiveChannel.IFBandwidth = LBW
OIF.IFFilterSamplePeriodMode = naMANUAL
OIF.IFFilterSamplePeriod = DSampleRate
OIF.IFFilterSource = naIFFilterSourceManual
OIF.IFFilterSampleCount = LNumTaps
OIF.IFGateEnable = True
OApp.ActiveChannel.FrequencyOffsetState = naON
OApp.ActiveChannel.FrequencyOffsetFrequency = DOffset
```

```
'Set receivers to medium gain setting
```

```
OIF.IFGainMode("ALL") = naMANUAL
OIF.IFGainLevel("A") = 1
OIF.IFGainLevel("B") = 1
OIF.IFGainLevel("R1") = 1
OIF.IFGainLevel("R2") = 1
```

```
Public Function Flog10(SGNum As Single) As Single
Flog10 = Log(SGNum) / Log(10)
End Function
```

```
'Enter Code here to send configuration to external pulse generators
```

ECAL Confidence Check

This Visual Basic program:

- Initializes the PNA objects.
- Performs a complete ECAL confidence check

Before using this code:

- The active channel must contain an S11 measurement with a 1-port or N-port calibration
- Prepare a form with two buttons named **cmdRun** and **cmdQuit**

```
Private oPNA As AgilentPNA835x.Application
Private oChan As Channel
Private oCal As Calibrator
Private oMeas As Measurement

Private Sub cmdRun_Click()
Dim iMeasIndex As Integer

Set oPNA = CreateObject("AgilentPNA835x.Application", "MachineName")
Set oChan = oPNA.ActiveChannel
Set oCal = oChan.Calibrator

iMeasIndex = 1

' Loop through measurements until an S11 on the active channel
' is found, or the end of the measurement collection is reached.
Do
   Set oMeas = oPNA.Measurements(iMeasIndex)
   If oMeas.Parameter = "S11" And _
      oMeas.channelNumber = oChan.channelNumber Then Exit Do
   iMeasIndex = iMeasIndex + 1
   If iMeasIndex > oPNA.Measurements.Count Then
      MsgBox "No S11 measurement found on the active channel." _
             & " Create an S11 measurement, then try again."
      Exit Sub
   End If
Loop

' Set up trace view so we are viewing only the data trace.
```

```
oMeas.View = naData
' Acquire the S11 confidence check data from ECal Module A
' into the memory buffer.
oCal.AcquireCalConfidenceCheckECALEx "S11", 1
' Turn on trace math so the trace shows data divided by memory.
' You can be confident the S11 calibration is reasonably good if
' the displayed trace varies no more than a few tenths of a dB
' from 0 dB across the entire span.
oMeas.TraceMath = naDataDivMemory
End Sub

Sub cmdQuit_Click()
' Turn off trace math
' in case someone clicks Quit without having clicked Run
If oMeas <> Nothing Then oMeas.TraceMath = naDataNormal
' Conclude the confidence check to set the ECal module
' back to it's idle state.
If oCal <> Nothing Then oCal.DoneCalConfidenceCheckECAL
' End the program
   End
End Sub
```

Limit Line Testing with COM

This Visual Basic program:

- Turns off existing Limit Lines
- Establishes Limit Lines with the following settings:
    - Frequency range - 4 GHz to 8 GHz
    - Maximum value - (10dB)
    - Minimum value - (-30dB)
- Turns on Lines, Testing, and Sound

If using Global Pass/Fail to report limit results, trigger the PNA after configuring and enabling Limit lines.

```
Public limts As LimitTest
Set limts = meas.LimitTest
'All Off
For i = 1 To 20
 limts(i).Type = naLimitSegmentType_OFF
Next i

'Set up Limit Lines
limts(1).Type = naLimitSegmentType_Maximum
limts(1).BeginResponse = 10
limts(1).EndResponse = 10
limts(1).BeginStimulus = 4000000000#
limts(1).EndStimulus = 8000000000#
limts(2).Type = naLimitSegmentType_Minimum
limts(2).BeginResponse = -30
limts(2).EndResponse = -30
limts(2).BeginStimulus = 4000000000#
limts(2).EndStimulus = 8000000000#

'Turn on Lines, Testing, and Sound
limts.LineDisplay = 1
limts.State = 1
limts.SoundOnFail = 1
```

COM Events Example

This Visual Basic program shows how to monitor the end of sweep.
The program will set sweep time to various amounts and BEEPs when sweep is completed. This method allows other processes to continue while waiting for end-of-sweep. This program stops after 10 loops.

Note: To avoid **Permission Denied** problems, this should be run on the PNA and not a PC. To run it from a PC both units must be "trusted" and on the same domain/workgroup.

```
Option Explicit
Dim na As AgilentPNA835x.Application
Dim WithEvents naEvnt As AgilentPNA835x.Application
Dim ch As AgilentPNA835x.Channel
Dim sweepComplete As Boolean

Private Sub Form_Load()

Dim N As Integer
Set na = CreateObject("AgilentPNA835x.application")
na.preset
Set ch = na.ActiveChannel
na.DisallowAllEvents            ' Turn off all events
Set naEvnt = na                 ' Enable event interrupts
Do
N = N + 1                      ' Loop counter
ch.sweepTime = 1 + (Rnd * 9)  ' Set random sweep-time from 1-10 sec
sweepComplete = False
ch.Single False                ' Trigger sweep
naEvnt.AllowEventCategory naEventCategory_CHANNEL, True  ' Enable Channel event
Do
DoEvents                     ' Allows other processes to continue
Loop Until sweepComplete = True
naEvnt.AllowEventCategory naEventCategory_CHANNEL, False  ' Disable event until
ready for next one
Beep                           ' Do end-of-sweep processing here;

Loop Until N > 10
End

End Sub

Private Sub naEvnt_OnChannelEvent(ByVal eventID As Variant, ByVal chNumber As
Variant)
' In this example we don't care about the channel info
If eventID = naEventID_CHANNEL_TRIGGER_COMPLETE Then sweepComplete = True
End Sub
```

Configure for COM-DCOM Programming

Before developing or running a COM program, you should first establish communication between your PC and the analyzer. This process is referred to as gaining **Access** to the analyzer. You should then register the PNA type library on your PC.

DCOM (Distributed Component Object Model) refers to accessing the PNA from a remote PC.

COM refers to accessing the PNA application from the analyzer PC.

- **Access Concepts**
- **Access Procedures**
- **Register the PNA Type Library on Your PC**
- **Problems?**

Note: After performing a Firmware Upgrade you must copy the new type library to your development PC to get access to new COM commands. See Register the analyzer on your PC.

Other Topics about COM Concepts

For detailed information on this subject, see http://na.tm.agilent.com/pna/DCOMSecurity.html

Access Concepts

PNAs are shipped from the factory such that **Everyone** has permission to launch and access the PNA application via COM/DCOM. The term **Everyone** refers to a different range of users depending on whether the PNA is a member of a **Domain** or **Workgroup** (it must be one or the other; not both). By default, the PNA is configured as members of a workgroup. Therefore, **Everyone** includes only those users who have been given logon accounts on the PNA.

Workgroup

A workgroup is established by the **PNA administrator** declaring the workgroup name and declaring the PNA as a member of the workgroup. A workgroup does not require a network administrator to create it or control membership.

Everyone includes only those users who have been given logon accounts on the PNA.

By default, the PNA is configured as members of a workgroup named WORKGROUP.

Note: To setup a logon account for a new user, see Additional Users.
The easiest method of gaining DCOM access, is to make the user's account name and password on the PNA to EXACTLY match their PC logon account name and password.

Domain

A domain is typically a large organizational group of computers. Network administrators maintain the domain and control which machines have membership in it.

Everyone includes those people who have membership in the domain. In addition, those with logon accounts can also access the analyzer.

Summary

- A **Workgroup** requires no maintenance, but allows DCOM access to only those users with a log-on account for the PNA.
- A **Domain** requires an administrator, but all members of the domain and those with logons to the analyzer are allowed DCOM access to the PNA.

The following section "Access Procedures" provides a tighter level of security allowing only **selected** (not **Everyone**) domain and workgroup users DCOM **Access** and **Launch** capability of the PNA.

Access Procedures

Perform this procedure for the following reasons:

- To allow only selected users (not everyone) remote Access and remote Launch capability to the PNA. Launch capability is starting the PNA application if it is not already open.
- To verify that you have DCOM access to the analyzer.

Note: Before doing this procedure, you must first have a logon account on the PNA. See PNA User Accounts

The following procedure grants specific users DCOM access and launch capability of the PNA application:

To perform this procedure, you must first minimize the PNA application.

How do I know which Operating System I have?

Windows 2000	Windows XP
On the PNA, click the Windows **Start** button	On the PNA, click the Windows **Start** button
Click **Run**	Click **Run**
In the **Open:** box, type **dcomcnfg**	In the **Open:** box, type **dcomcnfg**
Click **OK**	Click **OK**
In the Distributed COM Configuration Properties window, Click on **Agilent PNA Series** in the Applications list. Then click **Properties...**	Open the following folder sequence: Component Services Window Component Services Computers My Computer DCOM Config Right click **Agilent PNA Series** Click **Properties**
Click the **Security** tab	Click the **Security** tab
Click **Use custom access permissions** then click **Edit**	Under **Access Permissions**, click **Customize**, then click **Edit**

In **Registry Value Permissions**, select **Everyone,** then click **Remove**	Select **Everyone**, then click **Remove**
Click **Add**	Click **Add**
You could either select one or more of these groups to have access to the PNA, or specific users. To give specific users access, click **Show users** or **Members**, then select the name from the list.	Type a group name or user account name
Click **Add,** then click **OK**	Click **OK**
Launch Permission	
Click **Use custom launch permissions**, then click **Edit**	Under **Launch Permissions**, click **Customize**, then click **Edit**
In Registry Value Permissions, select **Everyone** then click **Remove**	Select **Everyone**, then click **Remove**
Click **Add**	Click **Add**
You could either select one or more of these groups to have launch permission of the PNA, or specific users. • To give groups launch permission, select the group from the list. • To give specific users launch permission, click **Show users** or **Members**, then select the name from the list.	Type a group name or user account name
Click **Add,** then click **OK**	Click **OK**

Register the PNA Type Library on Your PC

The type library contains the PNA object model. On your PC, there is a Registry file that keeps track of where object models are located. Therefore, you must register the type library on the PC that will be used to develop code and run the program. It is much more efficient to have the type library registered at design time (BEFORE running your COM program).

Do the following two items before proceeding:

1. Connect your PC and the PNA to LAN.
2. Either map a drive to the analyzer or copy the type library files on a floppy disk or other media. See Drive Mapping.

Note: To register the type library on your PC, you must be logged on as an administrator of your PC. Learn about User Accounts.

This procedure will do the following:

- Register the Network Analyzer application on your PC.

- Copy and register the proxystub (835xps.DLL) onto the PC.
- Copy and register the PNA type library (835x.tlb) onto the PC.
- Copy and register the FCA type library (fca.tlb) onto the PC.

1. Using Windows Explorer on your PC, find the Analyzer's C: drive. The drive will not be named "C:" on your PC, but a letter you assigned when mapping the drive.
2. Navigate to **Program Files \ Agilent \ Network Analyzer \ Automation**
3. Double-click **pnaproxy.exe** and follow the prompts to Install PNA Proxy. If the installation offers a choice of Modify, Repair, or Remove, then select **Remove**. Then double-click on **pnaproxy.exe** again.
4. When prompted, type the Computer name of the PNA (Learn how to find this).
5. After the install program runs, the PNA and FCA type library should be registered on your PC.
6. Your programming environment may require you to set a reference to the PNA type library now located on your PC. In Visual Basic, click **Project, References**. Then browse to **C:\Program Files\Common Files\Agilent\PNA** Select **835x.tlb**

Problems?

- These procedures will fail if there are any programs using the PNA type library (for example: Visual basic, VEE, Visual Studio, or any other application program that may communicate with the PNA).
- Perform the following procedure if the previous procedure did not return an error, but you cannot connect to the PNA.
- If you received an error, check that both the account name and password used on both the PNA and PC match EXACTLY.
- If you still get errors, see http://na.tm.agilent.com/pna/DCOMSecurity.html.

1. Map a drive from your remote PC to the PNA. Note the drive letter your PC assigns to the PNA. Substitute this drive letter for **PNA** in the following procedure.
2. On your PC, go to a DOS prompt c:>
3. Type **PNA:** (for example **o:**)
4. Type **cd program files\agilent\network analyzer\automation**
5. Type **copy 835xps.dll c:\program files\common files\agilent\pna**
6. Type **copy 835x.tlb c:\program files\common files\agilent\pna**

If you will NOT be using FCA commands, skip steps 6, 7,.8.

6. Type **cd..**
7. Type **cd applications\fca**
8. Type **copy fca.tlb c:\program files\common files\agilent\pna**

9. If it is not already there, copy **regtlib.exe** from **PNA**:\WINNT to your C:\<windows>\system32 directory (<windows> is OS-dependent- it is either windows or WINNT)

0. Type **regtlib C:\program files\common files\agilent\pna\835x.tlb**

1. Type **regsvr32 C:\program files\common files\agilent\pna\835xps.dll**

2. Type **regtlib C:\program files\common files\agilent\pna\fca.tlb**

Perform the Access Procedure after doing these steps.

COM Fundamentals

The following terms are discussed in this topic:

- **Objects**
- **Interfaces**
- **Collections**
- **Methods**
- **Properties**
- **Events**
- **Visual Basic Syntax**

Note: The information contained in this topic is intended to help an experienced SCPI programmer transition to COM programming. This is NOT a comprehensive tutorial on COM programming.

Other Topics about COM Concepts

Objects

The objects of the Network Analyzer (Application) are arranged in a hierarchical order. The Network Analyzer object model lists the objects and their relationship to one another.

In SCPI programming, you must first select a measurement before making settings. With COM, you first get a handle to the object (or collection) and refer to that object in order to change or read settings (properties).

For more information on working with objects, see Getting a Handle to an Object.

Interfaces

A COM Interface is the connection to an object. When you get a handle to an object, you are actually using an interface to an object. This is important if you are developing PNA code that will run on multiple code versions. For more information, see PNA Interfaces.

Collections

A collection is an object that contains several other objects of the same type. For example, the **Channels** collection contains all of the channel objects.

Note: In the following examples, the collections are referred to as a variable. Before using a collection object, you must first get an instance of that object. For more information, see Getting a Handle to an Object

Generally, items in a collection can be identified by **number** or by **name**. The order for objects in a collection cannot be assumed. They are always unordered and begin with 1. For example, in the following procedure, chans(1) is used to set averaging on the **first** channel in the Channels collection (not necessarily channel 1).

```
Sub SetAveraging()
  chans(1).AveragingFactor = 10
End Sub
```

The following procedure uses the measurement string name to set the display format for a measurement in the measurements collection.

```
meass("CH1_S11_1").Format = 1
```

You can also manipulate an entire collection of objects if the objects share common methods. For example, the following procedure sets the dwell time on all of the segments in the collection.

```
Sub setDwell()
For Each seg In segs
  segs.DwellTime = 0.03
Next
End Sub
```

Methods

A method is an action that is performed on an object. For example, **CreateSParameter** is a method on the Application object. The following procedure uses that method to create a new S21 measurement in channel 1 in a new window.

```
Sub CreatMeas
  app.CreateSParameter 1,2,1,1
End Sub
```

Properties

A property is an attribute of an object that defines one of the object's characteristics, such as size, color, or screen location. A property can also change an aspect of the object's behavior, such as whether the object is visible. In either case, to change the characteristics of an object, you change the values of its properties.

For example, the following statement sets the IF Bandwidth of a channel to 1 KHz.

```
Chan.IFBandwidth = 1e3
```

You can also read the current value of a property. The following statement reads the current IF Bandwidth of a channel into the variable **Ifbw.**

```
Ifbw = Chan.IFBandwidth
```

Some properties cannot be set and some cannot be read. The Help topic for each property indicates if you can:

- Set and read the property (Write/Read)
- Only read the property (Read-only)
- Only set the property (Write-only)

Events

An event is an action recognized by an object, such as clicking the mouse or pressing a key. Using events, your program can respond to a user action, program code, or triggered by the analyzer. For example:

```
OnChannelEvent
```

For more information, see Working with the Analyzer's Events.

Visual Basic Syntax

The examples in PNA Help use Visual Basic as the programming environment, which uses 'dot' notation.

To set a property, an object, or reference to an object is followed with:

- a period (**.**)
- property or method
- an equal sign (=)
- the new value

For example:

```
object.property = value 'This Green text following an apostrophe (') is a comment.
```

To read a property, a variable to contain the returned value is followed with:

- an equal sign (=)
- an object, or reference to an object
- a period (**.**)
- property

For example:

```
variable = object.property
```

To execute a method, an object, or reference to an object is followed with:

- a period (**.**)
- the method
- a blank space
- any required parameters

For example:

```
object.method parameters
```

Some methods return values, such as methods that return data. To return data from a method, a variable to contain the returned data is followed with:

- an equal sign (=)
- an object, or reference to an object
- a period (**.**)
- the method
- any required parameters enclosed in parenthesis

```
variable = object.method (parameters)
```

Getting a Handle to an Object

The following are discussed in this topic:

- **What Is a Handle**
- **Declaring an Object Variable**
- **Assigning an Object Variable**
- **Navigating the Object Hierarchy**
- **Getting a Handle to a Collection**

Other Topics about COM Concepts

What Is a Handle

In SCPI programming, you must first select a measurement before changing or reading settings. With COM, you first get a handle to the object (or collection) and refer to that object to change or read its settings. The following analogy illustrates this:

A CAR could be called an object. More precisely, CAR is a class of objects. For example, one of the properties of the CAR class is "**Color**". You can read (by looking) or set (by painting) the color property of A car object. In other words, you can only read or set properties of a specific car object; not the entire car class. Therefore, to read or set a property, you need to get "a handle", or an instance of the object.

This process is also called "accessing an object", "getting an instance of an object", "returning an object". or "referring to an object". You can have handles to many instances of an object at the same time.

Accessing PNA Objects

The PNA Application object is the highest object in the PNA object model hierarchy. Because of that, it is the only object that must be 'created' before it, or any other objects, can be accessed and used. During the creation process, the application object assigned to a variable name, or handle. Throughout your program, that object is used by referring to that variable. All PNA objects can be assigned to a variable, and subsequently referred to, in this same manner.

The following example shows how to create the PNA Application object, as well as illustrate the general steps of get a handle to an object.

There are two steps in the process of getting a handle to analyzer objects:

1. Declaring a Variable
2. Assigning an Object to the Variable

1. Declaring a Variable

Note: The examples in these topics use the Visual Basic Programming Language. See the short section regarding Visual Basic syntax.

Use the Dim statement or one of the other declaration statements (Public, Private, or Static) to declare a variable. The type of variable that refers to an object must be a Variant, an Object, or a specific type of object. Some programming languages, such as VBScript and Agilent VEE, do not allow you to specify variable types.

The following examples ALL declare the variable **pna**. Each subsequent statement is more specific than the previous:

- `Dim pna 'Variant data type.`
- `Dim pna As Object 'Object data type.`
- `Dim pna As AgilentPNA835x.Application ' Specific Application type`
- `Dim pna As AgilentPNA835x.IApplication ' Interface type`

1. If you use a variable without declaring it first, the data type of the variable is Variant. If you don't care about using automatic type checking, and willing to run code less efficiently, this method is very safe and is useable on all programming environments.

2. If you know the specific object type, and your programming environment allows it, you can declare the variable as an object.

3. Declaring a specific object type provides automatic type checking (Intellisense), faster code, and improved readability.

4. Declaring the interface is the most specific way and is beneficial when developing code for multiple firmware revisions. Learn more about Interfaces.

2. Assigning an Object to a Variable

To assign an object instance to a variable, use the **Set** keyword before the object variable that was declared previously. In the following line of code, we SET the current AgilentPNA835x Application to "pna".

```
Set pna = AgilentPNA835x.Application
```

As mentioned earlier, the AgilentPNA835x object is unique because it is the highest level of object in the PNA object model hierarcy. Therefore, we must use the **CreateObject** keyword with the (*classname,server name*) parameters.

- The ***classname*** for the analyzer object is always "AgilentPNA835x.Application".
- To find your analyzer's ***server name***, see View or change full computer name

The following statements create an instance of the Analyzer object.

```
Dim pna AS AgilentPNA835x.Application
Set pna = CreateObject("AgilentPNA835x.Application", "Analyzer46")
```

Note: These statements will start the PNA application if it is not already running on your instrument.

Navigating the Object Hierarchy

Once an instance of the PNA Application is "created", you access all of the PNA objects by navigating the object hierarchy. Navigating the object model hierarchy can be tricky. In addition, you also need to know how to refer to a specific instance of that object. For example, if you have three measurements present on the PNA, how do you refer to the channel 1 measurement? Each object on the PNA Object Model image is linked to an object page. At the top of each object page is a **Description** section and another called "**Accessing the ... Object".** These sections together explain how to navigate the PNA hierarchy to access a specific instance of that object.

From the previous discussion, you may think that you must always declare and assign variables to an object before setting or reading its properties. While this method is best for objects that you will continue to reuse, such as a measurement, it is not always necessary. You can also refer to an object directly.

The TriggerSetup object, which is a child of the Application object. Because we will only need to refer to this object once to set a couple of properties, and it is easy to access, we will refer to it directly. From the previous example, we already have a handle to the Application object in the variable **pna**. The following example uses Visual basic 'dot" notation to refer to the TriggerSetup object, and then the Scope property.

```
pna.TriggerSetup.Scope = naChannelTrigger
```

By referring to the TriggerSetup object directly, we must type the same path whenever we refer to properties on the TriggerSetup object. The following method assigns the pna.TriggerSetup object to a variable that can be reused.

```
Dim trig As Object
Set trig = pna.TriggerSetup
```

Once created, you can treat an object variable exactly the same as the object to which it refers. For example:

```
trig.Scope = naChannelTrigger
trig.Source = naTriggerSourceInternal
```

Getting a Handle to a Collection

The analyzer has several collections of objects which provide a convenient way of setting or reading all of the objects in the collection with a single procedure. Also, there are objects (limit lines for example) that can only be accessed through the collection.

To get a handle to an item in a collection, you can refer to the object by item number or sometimes by name. However, you first have to get a handle to the collection. To assign the collection to a variable, use the same two step process (1. declare the variable, 2. assign the variable using 'Set').

```
Dim meass As Measurements
```

```
Dim meas As Measurement
```

You can then iterate through the entire collection of measurements to read or set properties

```
Sub setFormat()
For Each meas In meass
 meas.Format = naDataFormat_LinMag
Next
End Sub
```

Or you can read or set a property on an individual object in the collection:

```
meass(1).Format = naLinMag
```

Note: Each object and collection has its own unique way of dealing with item names, and numbers. Refer to the Analyzer Object Model for details.

Collections in the Analyzer

Collections are a gathering of similar objects. They are a convenience item used primarily to iterate through the like objects in order to change their settings. Collections generally provide the following generic methods and properties:

```
Item(n)
Count
Add(n)
Remove(n)
```

where **(n)** represents the number of the item in the collection. Some collections may have unique capabilities pertinent to the objects they collect.

Other Topics about COM Concepts

Collections are Dynamic

A collection does not exist until you ask for it. When you request a Channels object (see Getting a Handle to an Object / Collection), handles to each of the channel objects are gathered and placed in an array.

For example, if channels 2 and 4 are the only channels that exist, then the array will contain only 2 items. The command 'channels.Count' will return the number 2, and:

- Channels(1) will contain the channel 2 object.
- Channels(2) will contain the channel 4 object.

The ordering of objects within the collection should not be assumed. If you add a channel to the previous example, as in:

```
Pna.Channels.Add(3)
```

'channels.Count' will now return 3 and:

- Channels(1) will contain the channel 2 object.
- Channels(2) will contain the channel 3 object.
- Channels(3) will contain the channel 4 object.

Primarily, collections are useful for making this type of iteration possible:

```
Dim ch as Channel
For each ch in pna.Channels
  Print ch.Number
  Print ch.StartFrequency
  Print ch.StopFrequency
Next ch
```

As soon as this for-each block has been executed, the Channels object goes out of scope.

COM Data Types

The PNA uses several data types to communicate with the host computer. Before using a variable, it is best to declare the variable as the type of data it will store. It saves memory and is usually faster to access. The following are the most common data types:

- **Long Integer**
- **Single Precision (Real)**
- **Double Precision (Real)**
- **Boolean**
- **String**
- **Object**
- **Enumeration**
- **Variant**

Other Topics about COM Concepts

Long (long integer) variables are stored as signed 32-bit (4-byte) numbers ranging in value from -2,147,483,648 to 2,147,483,647.

Double (double-precision floating-point) variables are stored as IEEE 64-bit (8-byte) floating-point numbers ranging in value from -1.79769313486232E308 to -4.94065645841247E-324 for negative values and from 4.94065645841247E-324 to 1.79769313486232E308 for positive values.

Single (single-precision floating-point) variables are stored as IEEE 32-bit (4-byte) floating-point numbers, ranging in value from -3.402823E38 to -1.401298E-45 for negative values and from 1.401298E-45 to 3.402823E38 for positive values.

Boolean variables are stored as 16-bit (2-byte) numbers, but they can only be True or False. Use the keywords True and False to assign one of the two states to Boolean variables.

When other numeric types are converted to Boolean values, 0 becomes False and all other values become True. When Boolean values are converted to other data types, False becomes 0 and True becomes -1.

String variables hold character information. A String variable can contain approximately 65,535 bytes (64K), is either fixed-length or variable-length, and contains one character per byte. Fixed-length strings are declared to be a specific length. Variable-length strings can be any length up to 64K, less a small amount of storage overhead.

Object variables are stored as 32-bit (4-byte) addresses that refer to objects within the analyzer or within some other application. A variable declared as Object is one that can subsequently be assigned (using the Set statement) to refer to any actual analyzer object.

Enumerations (Enum) are a set of named constant values. They allow the programmer to refer to a constant value by name instead of by number. For example:

```
Enum DaysOfWeek
  Sunday = 0
  Monday = 1
  Tuesday = 2
  Wednesday = 3
  Thursday = 4
  Friday = 5
  Saturday = 6
End Enum
```

Given this set of enumerations, the programmer can then pass a constant value as follows:

```
SetTheDay(Monday)
```

rather than

```
SetTheDay(1)
```

where the reader of the code has no idea what the value 1 refers to.

However, the analyzer RETURNS a long integer, not the text.

```
Day = DaysofWeek(today) 'Day = 1
```

Variant - If you don't declare a data type ("typed" data) the variable is given the Variant data type. The Variant data type is like a chameleon — it can represent many different data types in different situations.

The PNA provides and receives Variant data because there are programming languages that cannot send or receive "typed" data. Variant data transfers at a slower rate than "typed" data.

PNA Interfaces

A COM interface is the connection to an object. When you get a handle to an object, you are actually using an interface to an object. This subtle distinction is relevant to the COM programmer for the following two reasons:

- **Interface Inheritance (Coding for Multiple PNA Versions)**
- **Custom Interfaces.**

Other Topics about COM Concepts

Interface Inheritance (Coding for Multiple PNA Versions)

The PNA continues to evolve and release new firmware / software versions that provide more functionality and features. New commands are added to existing objects, and with them new interfaces are added to support those commands. For example, new commands were added to the Measurement object in PNA release 3.0. These commands are accessible from the new IMeasurement2 interface. This can be important if you develop code using the type library in release 3.0, and run the code on a PNA with an older release, such as 2.0

When you use a command that was new with release 3.0, and you run that code on a PNA with release 2.0 firmware, errors will occur because that PNA does not recognize the new commands. However, even if you do NOT utilize new commands, errors can still occur. The following example shows how this occurs and how to avoid it.

The following Visual Basic statement dimensions the **meas** variable as an object.

```
Dim meas As Measurement
```

When the program compiles, Visual Basic figures out what interface to use to access that object. When dimensioning as an object, VB will use the default interface. As new interfaces are added to an object, they become the default interface. If this program was developed and compiled using the PNA 3.0 type library, the default Interface of the Measurement Object was IMeasurement2. However, if this program is run on an instrument with PNA 2.0 firmware, there was no IMeasurement2 Interface, and an E_NOINTERFACE error will occur.

Therefore, the more robust approach would be to specify the interface instead of the object when declaring a variable.

```
Dim meas As IMeasurement
```

This code will ONLY use the IMeasurement interface; not the default interface.

However, regardless of how you declare a variable, errors will always occur if you use new commands, and run the code on an older instrument.

Custom Interfaces

The PNA object model contains three "custom" interfaces. These interfaces allow the Visual Basic and C++ programmer to transfer data using "typed" variables, which is more efficient than using variant type variables. These interfaces are:

- IArrayTransfer - Measurement object
- ICalData - Calibrator object
- ISourcePowerCalData - Channel object

Working with Events

- What are Events?
- Using the Analyzer's Events
- Event ID's
- Filtering Events
- List of Events
- Out of Range Errors
- Troubleshooting Problems with Events

See Events Example.

Other Topics about COM Concepts

What are Events?

Windows applications work from user-initiated events such as mouse moves and mouse clicks. A mouse-click produces an event that the programmer can either ignore or "handle" by providing an appropriate subroutine like this:

```
Sub DoThis_onClick
 Perform something
End Sub
```

If this subroutine were in your program and the mouse-click event occurs on your PC, it would generate a "Callback" to the client and interupt whatever it was doing and handle the event.

A more practical example of an event in the analyzer is Limit test. If limit test is on and the measurement fails, the analyzer produces a "'Limit-failed" event. If the measurement passed, the analyzer produces a "Limit-succeeded" event.

The Analyzer has a very sophisticated Event structure. Your program **CAN** be notified when one or more events occur. However, it may not be necessary.

For example, the analyzer has an event that will notify your program when a sweep is complete. A simpler alternative is to use a synchronous command which waits for the sweep to complete.

```
sync = True
app.ManualTrigger sync
chan.StartFrequency = 4.5E6
```

This would NOT work if you want the controller to do other things while waiting, like setup a power meter or sort some data. In this case you would like a "callback" from the analyzer to let your program know that the sweep has completed. For an example of this see Events Example.

Another reason to use events is when you want to be notified of several conditions when they occur, such as errors or source unlock conditions. It would not be practical to routinely poll these conditions while executing your program.

Using Events

If you decide to use the COM events to get a callback, your program must do two things:

1. Subscribe to events:

All events in the analyzer are a child of the Application object through the INetworkAnalyzerEvents Interface. You must tell the Application object that you are interested in receiving event callbacks. This process is called subscription.

In Visual Basic, this is done by including "WithEvents" in the declaration statement. The declaration below dimensions an Application object (myPNA) and subscribes to the events produced by the Application.

```
Dim WithEvents myPNA as AgilentPNA835x.Application
```

In C++, this is a bit more involved. You must queryInterface for the IconnectionPointContainer interface, locate the InetworkAnalyzerEvents interface via a call to FindConnectionPoint and call Advise().

2. Implement the Event Handler

When an event occurs, the Application object will "callback" to the client through the InetworkAnalyzerEvents interface.

In VB, click on the object window (upper left pane). Find the Application object and click it. The event interfaces will appear in the upper right pane. As you click on them, VB supplies the first line of code. You fill in the rest of the handler routine to service the event. The following is an example of a event handler subroutine.

Note: In C++, you must type the callback.

```
Private Sub OnChannelEvent( eventID as Variant, channelNumber as Variant)
Select Case (eventID)
 Case naEventID_CHANNEL_TRIGGER_COMPLETE:
  GetData( channelNumber )
 Case naEventID_CHANNEL_TRIGGER_ABORTED:
  MsgBox( "Hey don't touch the front panel!")
End Select
End Sub
```

When the trigger is complete, the application object "fires" the event by making a callback to the event handler `Sub` OnChannelEvent`()`.

Event IDs

3	3	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0
Sev		**C**	**R**	**Facility**											**Code**																

Filtering Events

There are over 140 different events that you subscribe to when you "Dim WithEvents..." (or the equivalent in your programming language). Monitoring all of these conditions slows the speed of the analyzer significantly. The following methods allow you to filter the events so that you only monitor specific conditions.

- AllowEventMessage - monitor a specific event
- AllowAllEvents - monitor ALL events
- DisallowAllEvents - monitor NO events
- AllowEventCategory - monitor specific event categories (discussed later)
- AllowEventSeverity - monitor events having one or more of the following severity levels associated with them.

Code	Severity Enumeration
00	naEventSeveritySUCCESS - the operation completed successfully
01	naEventSeverityINFORMATIONAL - events that occur without impact on the measurement integrity
10	naEventSeverityWARNING - events that occur with potential impact on measurement integrity
11	naEventSeverityERROR - events that occur with serious impact on measurement integrity

List of Events

The following is a list of categories and the general types of events they include. Click the link view the event details.

Category Enumeration	Callback
naEventCategory_PARSER	OnSCPIEvent
naEventCategory_MEASURE	OnMeasurementEvent
naEventCategory_CHANNEL	OnChannelEvent
naEventCategory_HW	OnHardwareEvent
naEventCategory_CAL	OnCalEvent
naEventCategory_USER	OnUserEvent
naEventCategory_DISPLAY	OnDisplayEvent
naEventCategory_GENERAL	OnSystemEvent

Note: Use the MessageText Method to get a text message describing the event.

Out of Range Errors

When you attempt to set a value on an active function that is beyond the range (min or max) of the allowable values, the analyzer limits that value to an appropriate value (min or max) and sets the function to the limited value. From the front panel controls this is visually evident by the limited value in the edit box or by the annotation on the display. An example would be attempting to set the start frequency below 300kHz. The edit control doesn't allow the number to fall below 300kHz.

When the automation user programs a setting (such as start frequency below the allowable limits) the same behavior takes place. The analyzer accepts the limited value. However, in order to learn what setting took place, you have to read the HRESULT.

All automation calls return HRESULTs. By default the HRESULT returned when an overlimit occurs is S_NA_LIMIT_OUTOFRANGE. This value is a success code, meaning that bit 31 in this 32 value is 0. Programmers should check the return code from all automation calls to determine success or failure.

Some C++ macros (like SUCCEEDED(hr) or FAILED(hr)) only check bit 31. So if you are interested in trapping this outOfRange error you will have to check for S_NA_LIMIT_OUTOFRANGE explicity.

Alternatively, you can configure the analyzer to report outOfRange conditions with an error code. Use the method: App.SetFailOnOverRange (true). With this method set TRUE, any overrange error will return E_NA_LIMIT_OUTOFRANGE_ERROR.

This method is provided for the benefit of VB clients. VB users can't detect specific success codes because the VB runtime strips off the HRESULT and only raises a run time error if bit 31 is set, indicating a fail code.

Troubleshooting Problems with Callbacks

When you do callbacks, the client PC becomes the server and the analyzer (server) becomes the client. Callbacks can only take place when both server and client are in the same workgroup or in the same domain. See Configure for COM.

Read and Write Calibration Data using COM

You can read or write two types of Calibration data in the PNA:

- **Standard Measurement data** -raw data resulting from the measurement of a calibration standard.
- **Error Terms** - calculated data using standard measurement data and the algorithms for the specified cal type.

Each of these data are available in the PNA in either variant data or typed data. Learn more about variant and typed data

Other Topics about COM Concepts

Evolution of the Calibration Architecture

PNA 2.0 expanded the use of the Cal Set, which is simply a container for calibration data. In PNA 1.0 the Cal Set was restricted to one cal type and could only be used by the channel that created it. In PNA 2.0, the Cal Set is sized dynamically, can accommodate more than one cal type, and can be used by multiple channels. (Learn more about Cal Sets)

The PNA has two sets of automation interfaces that contain methods for getting and putting Calibration data in a Cal Set:

Set 1 - ICalibrator (variant), ICalData (typed)

The ICalibrator and ICalData interfaces were introduced in PNA 1.0. They contain several methods for putting and getting error terms and standard measurement data.

Set 2 - ICalSet (variant), ICalData2(typed)

The ICalSet interface was introduced with PNA2.0 to support the new Cal Set features. The methods on this interface include, but are not limited to, putting and getting data to and from the Cal Set. In addition, the ICalData2 interface was introduced to work with non-variant data. The following is an example of using ICalSet to read error term data. This example gets a handle to a Cal Set using the GetCalSetByGUID method.

```
dim CMGR as CalManager
dim CSet as CalSet
dim strCalSetGUID as string
dim iEtermSetID as integer
dim caltype as NACalType
dim eTerm as NAErrorTerm2
dim rcvPort as long
dim srcPort as long
CMGR.GetCalSetUsageInfo( channel, strCalsetGUID, iEtermSetID)
set CSet = CMGR.GetCalSetByGUID( strCalSetGUID)
caltype = naResponseOpen
rcvPort = 1
srcPort = 1
eTerm = naET_ReflectionTracking
CSet.Open( caltype, rcvPort, srcPort)
VarData = CSet.GetErrorTerm( ETerm, rcvPort, srcPort)
CSet.Close()
```

Recommendation

For reading and writing calibration data, we strongly recommend using the ICalSet and ICalData2.

Note: The ICalibrator interface is still required for other calibration activities, such as acquiring calibration data.

Using ICalibrator with PNA2.0 Cal Sets

You can still use the ICalibrator interface to read and write calibration data on the 2.0 Cal Sets.

Get data from a Cal Set:

1. Get a handle to the Cal Set using one of the "get" methods on the ICalManager Interface
2. Get a handle to a Calibrator object on the same channel as the Cal Set.
3. Specify the Cal Type and ports with the SetCalInfo method:

The following example reads error term data from a Cal Set

```
WICalibrator.SetCalInfo( caltype, rcvPort, srcPort)
VarData = ICalibrator.GetErrorTerm( ETerm, rcvPort, srcPort)
```

Write data to a Cal Set:

You can either fill an "empty" cal set with data or overwrite an existing Cal Set. The SetCalInfo method will create an empty Cal Set if there is no Active Cal Set on the same channel as the Calibrator object. The following example writes error terms to an empty Cal Set.

```
ICalibrator.SetCalInfo(  caltype, rcvPort, srcPort)
VarData = ICalibrator.putErrorTerm( ETerm, rcvPort, srcPort)
```

Programming the PNA with C++

The programming information contained in this Help system is aimed at the Visual Basic programmer. VB does a lot of work for the programmer when it comes to managing and accessing components. Using a lower level language like C++ requires a more thorough understanding of the underlying tenets of COM. It is not the intent of this section to teach COM programming. The following is intended to acquaint you with some of the basic concepts you need to know in order to program against COM.

- **Initializing COM**
- **Importing the Type Library**
- **Creating the Application Object**
- **Errors**
- **Events**
- **Additional Reading**
- **Example**

Note: The information in this section assumes development on a Windows OS using Microsoft tools.

Other Topics about COM Concepts

Initializing COM

The first thing you must do before performing any COM transactions is to initialize the COM library. You can do this in a number of ways. The most basic of these is a call to **CoInitialize( )** or **CoIntializeEx( )**. Alternatively you can use the MFC (Microsoft Foundation Classes) **AfxOleInit( )**.

Conversely, before your program exits you must uninitialize COM. You can accomplish this with **CoUninitialize( )** or the MFC routine **AfxOleTerm( )**.

Importing the Type Library

To make a component available to the client, the server exports what is called the type library. For the PNA, this file is 835x.tlb. It is located on the PNA's hard drive at **C:\Program Files\ Agilent\ Network Analyzer\ Automation**. See Configure for COM-DCOM Programming.

The type library can be read and deciphered using another COM interface called ITypeLib. VB uses this interface to present, for example, its object browser. Visual C++ can also read type libraries. This is done by importing the type library into your project with a compiler directive:

```
#import "C:\Program Files\Common Files\Agilent\Pna\835x.tlb", named_guids
```

When you compile your program with this statement in it, the compiler creates two other files: **835x.tlh** and **835x.tli**. The first is a header file that contains the type definitions for the PNA's COM interfaces and their methods. The second file contains inline functions that wrap the PNA's interface methods. The wrappers are beneficial in that they contain error reporting for each of the method calls.

The .tlh file defines a smart pointer which you can use to access the PNA's objects. The smart pointer definition looks like this:

```
 _com_smartptr_typedef(Iapplication,  _uuidof(Iapplication))
```

A smart pointer is a term used for a C++ object that encapsulates a pointer used to refer to a COM object. All COM objects derive from the interface IUnknown. This interface has three methods: QueryInterface( ), AddRef( ), and Release( ). The function of the AddRef and Release methods is to maintain a reference count on the object and thus control the object's lifetime. Anytime you copy or create a reference to a COM object, you are responsible for incrementing its reference count. And likewise, when you are finished using that reference, it is your responsibility to Release it. Smart pointers do this work for you, as shown in the example program. In addition, smart pointers will also perform the QueryInterface call when required. QueryInterface is a method that requests a specific interface from an object. In the example program we gain access to the IArrayTransfer interface of the Measurement object. In the ReadMethod routine, we see this:

```
PTransferData = pMeas;
```

The assignment operator is overloaded for the smart pointer and in reality, this simple statement does this:

```
HRESULT hr = pMeas->QueryInterface( IID_IArrayTransfer,(void**)&pTransferData);
```

Using the existing interface pointer (pMeas) to the object, this call asks the object if it supports the IArrayTransfer interface, and if so to return a pointer to it in pTransferData. Smart pointer makes life easier for the C++ programmer. Read more about smart pointers in Microsoft Developer's Network Library *(MSDN).*

Creating the Application Object

The only createable object exported by the PNA is the Application object. Typically this would be done with a call to CoCreateInstance:

```
STDAPI CoCreateInstance(
  CLSID__IApplication, //Class identifier (CLSID) of the object
  NULL, //Pointer to controlling IUnknown
  CLS_CTX_SERVER, //Context for running executable code
  IID_IApplication, //Reference to the IID of the interface
  (void**)&pNA //Address of output variable that receives
  // the interface pointer requested in riid
);
```

With the smart pointer, this is taken care of with the following call:

```
IApplicationPtr pNA; // declare the smart pointer
pNA = IApplicationPtr("AgilentPNA835x.Application.1");
```

Errors

All COM method calls are required to return an HRESULT. This is 32 bit long with a specific format.

- The most significant bit indicates success(0) or failure(1).
- The lower 16 bits indicate the specific failure.

Visual Basic strips off the returned HRESULT and raises an error object for non-successful returns. The C++ programmer must himself be diligent about handling errors. You must check the return value of each COM call to ensure its success.

Events

The Application object sources the INetworkAnalyzerEvents interface. This object is the source for all events. To use events in C++, you must do two things:

1. Implement the INetworkAnalyzerEvents interface - derive an object from INetworkAnalyzerEvents and implement the methods described there.
2. Subscribe to the IconnectionPoint interface of the Application object. - obtain a pointer to the IConnectionPointContainer interface of the Application object and making the following request:

```
FindConnectionPoint( IID_InetworkAnalyzerEvents, &pConnection );
```

A successful call to this interface will return a valid pointer in pConnection. Use this pointer to subscribe to the Application object:

```
pConnect->Advise( IUnknown* punk, DWORD dwCookie);
```

This call provides the server object with a callback address. The Iunkown pointer in this call is the IUnkown pointer of the object that implements the INetworkAnalyzerEvents interface. This is the event sink. The application object needs a pointer to this object in order to call your interface when an event occurs. The **dwCookie** is your subscription key. Use it to unsubscribe (see Unadvise( )).

Additional Reading

"MSDN" - Microsoft Developer's Network Library

"Learning DCOM", by Thuan L. Thai, published by O'Reilly(1999)

"Inside COM", by Dale Rogerson, published by Microsoft Press (1997)

"Understanding ActiveX and OLE", by David Chappell, also published by Microsoft Press (1996)

"Beginning ATL COM Programming", published by Wrox Press (1998)

Example

The example uses the smart pointer created by Microsoft Visual Studio. The calls to CoInitialize and CoUninitialize open and close the COM libraries. In the example, notice that the pointers local to the main routine are explicitly released. When smart pointers go out of scope, they will perform this duty implicitly. However, we are calling CoUninitialize before they have the chance to be destroyed, so we are obliged to release them.

See the example program.

Using COM from .NET

To communicate with the PNA from Microsoft .NET enabled languages such as C# and Visual Basic.NET perform the following steps:

1. Configure your PC and PNA for COM-DCOM Programming.
2. Reference the type library within the development environment (see the following exception for managed C++ projects.) In the process of referencing the type library, a .NET assembly is created that wraps the PNA type library with a .NET friendly interface. This .NET assembly is called an Interop Assembly.

To run a .NET program on the PNA, you will need to install the .NET framework on the PNA. This can be done by running the dotnetfx.exe program, located at: http://www.microsoft.com/downloads/details.aspx?FamilyID=262d25e3-f589-4842-8157-034d1e7cf3a3&DisplayLang=en

If you only intend to run .NET programs on a remote computer, then it is not necessary to install the .NET framework on the PNA."

Exception for managed C++ projects: To generate the Interop Assembly for managed C++ projects, you must use the tlbimp.exe utility. This utility is described in the MSDN documentation. On your PC, click Start then Run then type: tlbimp.exe 835x.tlb and click OK. After doing this you can use the #using directive to include the Interop Assembly on managed C++ projects.

Registering the PNA Primary Interop Assembly (PIA) (OPTIONAL)

The PIA is NOT necessary to communicate with the PNA. The following procedure is useful only when there are two .NET programs that want to share the same PNA interface definitions. Without the PIA, each .NET application would use its own Interop Assembly.

To register the PIA on a machine, you need to have the common language runtime (CLR) installed. This is included with Visual Studio.NET. Then perform the following steps:

Note: In the following steps, replace <local directory> with the full path name of the specified file on your PC.

1. Run the PNAProxy.exe program as described in Configure for COM-DCOM Programming.
2. On the PNA, copy **C:\Program Files\Agilent\Network Analyzer\Automation\AgilentPNA835x.dll** to a local directory on your PC. Make a note of this directory.
3. On your PC, click **Start**, then **Run**, then type: **regasm** <local directory> **\AgilentPNA835x.dll** and click **OK** to register the dll.
4. Again, click **Start**, then **Run**, then type: **gacutil /i** <local directory> **\AgilentPNA835x.dll** and click **OK** to add the assembly to the Global Assembly Cache (GAC).

To **Uninstall the PIA**, perform the following:

1. On your PC, click **Start**, then **Run**, then type: **gacutil /u** <local directory> **\AgilentPNA835x.exe** and click **OK** to remove the assembly from the GAC.
2. On your PC, click **Start**, then **Run**, then type: **regasm /unregsiter** <local directory> **\ agilentpna835x.dll** and click **OK** to unregister the assembly.
3. To uninstall PNA Proxy.exe use the **Add/Remove Programs** utility in the control panel.

List of PNA SCPI Commands

Click on a command for details

Local Lockout - GPIB Message

COMMON COMMANDS

*CLS - Clear Status

*ESE - Event Status Enable

*ESE? - Event Status Enable Query

*ESR - Event Status Enable Register

*IDN? - Identify

*OPC - Operation complete command

*OPC? - Operation complete query

*OPT? - Identify Options Query

*RST - Reset

*SRE - Service Request Enable

*SRE? - Service Request Enable Query

*STB? - Status Byte Query

*TST? - Result of Self-test Query

*WAI - Wait

ABORT Command

ABORt

CALC:CORRECTION Commands

CALCulate<cnum>:CORRection:EDELay:MEDium <num>

CALCulate<cnum>:CORRection:EDELay:TIME <num>

CALCulate<cnum>:CORRection:EDELay:WGCotoff <num>

CALCulate<cnum>:CORRection:[STATe] <on|off>

CALCulate<cnum>:CORRection:TYPE <char>

CALCulate<cnum>:CORRection:OFFSet[:MAGNitude] <num>

CALCulate<cnum>:CORRection:OFFSet:PHASe <num>[<char>] OBSOLETE

CALC:CUSTOM Commands

CALCulate<cnum>:CUST:DEFine <Mname>, <ProgID> [,param]

CALCulate<cnum>:CUSTom:MODify <param>

CALC:DATA Commands

CALCulate<cnum>:DATA <char>

CALCulate<cnum>:DATA:CUSTom <name>,<data>

CALCulate<cnum>:DATA:CUSTom:CATalog?

CALCulate<cnum>:DATA:SnP?

CALC:FILTER Commands

CALCulate<cnum>:FILTer[:GATE]:COUPle:PARameter <char>

CALCulate<cnum>:FILTer[:GATE]:TIME:CENTer <num>

CALCulate<cnum>:FILTer[:GATE]:TIME:SHAPe <char>

CALCulate<cnum>:FILTer[:GATE]:TIME:SPAN <num>

CALCulate<cnum>:FILTer[:GATE]:TIME:STATe <boolean>

CALCulate<cnum>:FILTer[:GATE]:TIME:STARt <num>

CALCulate<cnum>:FILTer[:GATE]:TIME:STOP <num>

CALCulate<cnum>:FILTer[:GATE]:TIME[:TYPE] <char>

CALC:FORMAT Command

CALCulate<cnum>:FORMat <char>

CALC:FUNCTION Commands

CALCulate<cnum>:FUNCtion:DATA?

CALCulate<cnum>:FUNCtion:DOMain:USER[:RANGe] <range>

CALCulate<cnum>:FUNCtion:DOMain:USER:STARt <range>, <start>

CALCulate<cnum>:FUNCtion:DOMain:USER:STOP <range>, <stop>

CALCulate<cnum>:FUNCtion:STATistics[:STATe] <ON|OFF>

CALCulate<cnum>:FUNCtion:TYPE <char>

CALC:LIMIT Commands

CALCulate<cnum>:LIMit:DATA <block>

CALCulate<cnum>:LIMit:DISPlay[:STATe] <ON | OFF>

CALCulate<cnum>:LIMit:SEGMent<snum>AMPLitude:STARt <num>

CALCulate<cnum>:LIMit:SEGMent<snum>AMPLitude:STOP <num>

CALCulate<cnum>:LIMit:SEGMent<snum>STIMulus:STARt <num>

CALCulate<cnum>:LIMit:SEGMent<snum>STIMulus:STOP <num>

CALCulate<cnum>:LIMit:SEGMent<snum>:TYPE <char>

CALCulate<cnum>:LIMit:SOUNd[:STATe] <ON | OFF>

CALCulate<cnum>:LIMit:STATe <ON | OFF>

CALC:MARKER Commands

CALCulate<cnum>:MARKer:AOFF

CALCulate<cnum>:MARKer:BWIDth <num>

CALCulate<cnum>:MARKer<mkr>:COUPling[:STATe]<ON|OFF>

CALCulate<cnum>:MARKer<mkr>:DELTa <ON|OFF>

CALCulate<cnum>:MARKer<mkr>:DISCrete <ON|OFF>

CALCulate<cnum>:MARKer<mkr>:DISTance <num>

CALCulate<cnum>:MARKer<mkr>:FORMat <char>

CALCulate<cnum>:MARKer<mkr>:FUNCtion:APEak:EXCursion <num>

CALCulate<cnum>:MARKer<mkr>:FUNCtion:APEak:THReshold <num>

CALCulate<cnum>:MARKer<mkr>:FUNCtion:DOMain:USER <range>

CALCulate<cnum>:MARKer<mkr>:FUNCtion:DOMain:USER:STARt <start>

CALCulate<cnum>:MARKer<mkr>:FUNCtion:DOMain:USER:STOP <stop>

CALCulate<cnum>:MARKer<mkr>:FUNCtion:EXECute [<func>]

CALCulate<cnum>:MARKer<mkr>:FUNCtion[:SELect] <char>

CALCulate<cnum>:MARKer<mkr>:TARGet <num>

CALCulate<cnum>:MARKer<mkr>:FUNCtion:TRACking <ON | OFF>

CALCulate<cnum>:MARKer:REFerence[:STATe] <ON | OFF>

CALCulate<cnum>:MARKer:REFerence:X <num>

CALCulate<cnum>:MARKer:REFerence:Y?

CALCulate<cnum>:MARKer<mkr>:TYPE <char>

CALCulate<cnum>:MARKer<mkr>:SET <char>

CALCulate<cnum>:MARKer<mkr>[:STATe] <ON|OFF>

CALCulate<cnum>:MARKer<mkr>:X <num>

CALCulate<cnum>:MARKer<mkr>:Y?

CALC:MATH Commands

CALCulate<cnum>:MATH:FUNCtion <char>

CALCulate<cnum>:MATH:MEMorize

CALC:MIXER Command

CALCulate<ch>:MIXer:INPut:POWer:USENominal

CALCulate<ch>:MIXer:XAXis <char>

CALC:NORMALIZE Commands

CALCulate<cnum>:NORMalize[:IMMediate]

CALCulate<cnum>:NORMalize:STATe <ON | OFF>

CALCulate<cnum>:NORMalize:INTerpolation[:STATe] <ON | OFF>

CALC:OFFSet Commands

CALCulate<cnum>:OFFSet:MAGNitude <num>

CALCulate<cnum>:OFFSet:MAGNitude:SLOPe <num>

CALCulate<cnum>:OFFSet:PHASe <num>[<char>]

CALC:PARAMETER Commands

CALCulate<cnum>:PARameter:CATalog?

CALCulate<cnum>:PARameter:DEFine <Mname>,<param>[,load]

CALCulate<cnum>:PARameter:DELete <Mname>

CALCulate:PARameter:DELete:ALL

CALCulate<cnum>:PARameter:MNUMber?

CALCulate<cnum>:PARameter::MODify <param>

CALCulate<cnum>:PARameter:SELect <Mname>

CALC:RDATA Commands

CALCulate<cnum>:RDATA? <char>

CALC:SMOOTHING Commands

CALCulate<cnum>:SMOothing:APERture <num>

CALCulate<cnum>:SMOothing:POINts <num>

CALCulate<cnum>:SMOothing[:STATe] <ON | OFF>

CALC:TRANSFORM Commands

CALCulate<cnum>:TRANsform:COUPle:PARameters <num>

CALCulate<cnum>:TRANsform:TIME:CENTer <num>

CALCulate<cnum>:TRANsform:TIME:IMPulse:WIDTh <num>

CALCulate<cnum>:TRANsform:TIME:KBESsel <num>

CALCulate<cnum>:TRANsform:TIME:LPFREQuency

CALCulate<cnum>:TRANsform:TIME:MARKer:MODE <char>

CALCulate<cnum>:TRANsform:TIME:MARKer:UNIT <char>

CALCulate<cnum>:TRANsform:TIME:SPAN <num>

CALCulate<cnum>:TRANsform:TIME:STARt <num>

CALCulate<cnum>:TRANsform:TIME:STATe <ON | OFF>

CALCulate<cnum>:TRANsform:TIME:STOP <num>

CALCulate<cnum>:TRANsform:TIME:STEP:RTIMe <num>

CALCulate<cnum>:TRANsform:TIME:STIMulus <char>

CALCulate<cnum>:TRANsform:TIME[:TYPE] <char>

CONTROL Commands

CONTrol:ECAL:MODule[num]:STATe <value>

CONTrol:AUXiliary:C[:DATA] <num>

CONTrol:AUXiliary:C:LOGic <char>

CONTrol:AUXiliary:C:MODE <char>

CONTrol:AUXiliary:FOOTswitch?

CONTrol:AUXiliary:FOOTswitch:MODE <char>

CONTrol:AUXiliary:INPut:VOLTage?

CONTrol:AUXiliary:OUTPut<out>:MODE <char>

CONTrol:AUXiliary:OUTPut<out>:VOLTage <num>

CONTrol:AUXiliary:PASSfail:LOGic <char>

CONTrol:AUXiliary:PASSfail:MODe <char>

CONTrol:AUXiliary:PASSfail:SCOPe <char>

CONTrol:AUXiliary:PASSfail:POLicy <char>

CONTrol:AUXiliary:PASSfail:STATus?

CONTrol:AUXiliary:SWEepend <char>

Control External Test Set Commands

CONTrol:EXTernal:TESTset:DATa <addr>,<data>

CONTrol:EXTernal:TESTset:INTerrupt?

CONTrol:EXTernal:TESTset:RAWData <data>

CONTrol:EXTernal:TESTset:SWEepholdoff?

Control Handler Commands

CONTrol:HANDler:C:MODE <char>

CONTrol:HANDler:D:MODE <char>

CONTrol:HANDler:<port>[:DATa] <num>

CONTrol:HANDler:INPut?

CONTrol:HANDler:LOGic <char>

CONTrol:HANDler:OUTPut<num>[:DATa] <num2>

CONTrol:HANDler:OUTPut<num>:USER[:DATa] <num2>

CONTrol:HANDler:PASSfail:LOGic <char>

CONTrol:HANDler:PASSfail:MODe <char>

CONTrol:HANDler:PASSfail:SCOPe <char>

CONTrol:HANDler:PASSfail:POLicy

CONTrol:HANDler:PASSfail:STATus?

CONTrol:HANDler:SWEepend <char>

Control Signal Commands

CONTrol:SIGNal <conn>,<char>

CONTrol:SIGNal:TRIGger:ATBA <bool>

CONTrol:SIGNal:TRIGger:OUTP <bool>

DISPLAY Commands

DISPlay:ANNotation:FREQuency[:STATe] <ON | OFF>

DISPlay:ANNotation:MESSage:STATe <ON | OFF>

DISPlay:ANNotation:STATus <ON|OFF>

DISPlay:ARRange <char>

DISPlay:CATalog?

DISPlay:ENABLE <ON | OFF>

DISPlay:FSIGn <ON | OFF>

DISPlay[:TILE] OBSOLETE
DISPlay:WINDow<wnum>:ANNotation:MARKer:SINGle[:STATe] <bool>
DISPlay:WINDow<wnum>:ANNotation:MARKer:SIZE <char>
DISPlay:WINDow<wnum>:ANNotation:MARKer:STATe <ON|OFF>
DISPlay:WINDow<wnum>:ANNotation:TRACe:STATe <ON|OFF>
DISPlay:WINDow<wnum>:CATalog?
DISPlay:WINDow<wnum>:ENABle <ON | OFF>
DISPlay:WINDow<wnum>:SIZE MIN | MAX | NORM
DISPlay:WINDow<wnum>[:STATe] <ON | OFF>
DISPlay:WINDow<wnum>:TABLe <char>
DISPlay:WINDow<wnum>:TITLe:DATA <string>
DISPlay:WINDow<wnum>:TITLe[:STATe] <ON | OFF>
DISPlay:WINDow<wnum>:TRACe<tnum>:DELete
DISPlay:WINDow<wnum>:TRACe<tnum>:FEED <name>
DISPlay:WINDow<wnum>:TRACe<tnum>MEMory[:STATe] <ON | OFF>
DISPlay:WINDow<wnum>:TRACe<tnum>:SELect
DISPlay:WINDow<wnum>:TRACe<tnum>[:STATe] <ON | OFF>
DISPlay:WINDow<wnum>:TRACe<tnum>:Y[:SCALe]:AUTO
DISPlay:WINDow<wnum>:TRACe<tnum>:Y[:SCALe]:PDIVision <num>
DISPlay:WINDow<wnum>:TRACe<tnum>:Y[:SCALe]:RLEVel <num>
DISPlay:WINDow<wnum>:TRACe<tnum>:Y[:SCALe]:RPOSition <num>

FORMAT Commands

FORMat:BORDer <char>
FORMat[:DATA] <char>

HARDCOPY Command

HCOPy:FILE
HCOPy[:IMMediate]

INITIATE Commands

INITiate:CONTinuous <boolean>
INITiate<cnum>[:IMMediate]

MMEMORY Commands

MMEMory:CATalog[:<char>]? [<folder>]
MMEMory:CDIRectory <folder>
MMEMory:COPY <file1>,<file2>
MMEMory:DELete <file>
MMEMory:LOAD[:<char>] <file>
MMEMory:MDIRectory <folder>
MMEMory:MOVE <file1>,<file2>
MMEMory:RDIRectory <folder>
MMEMory:STORe[:<char>] <file>
MMEMory:STORe:CITI:DATA <filename>
MMEMory:STORe:CITI:FORMat <filename>
MMEMory:STORe:TRACe:CONTents:CITifile <char>
MMEMory:STORe:TRACe:FORMat:CITifile <char>
MMEMory:STORe:TRACe:FORMat:SNP <char>

MMEMory:TRANsfer <fileName>,<dataBlock>

OUTPUT Command

OUTPut[:STATe] <ON | OFF>

SENS:AVERAGE Commands

SENSe<cnum>:AVERage:CLEar

SENSe<cnum>:AVERage:COUNt <num>

SENSe<cnum>:AVERage[:STATe] <ON | OFF>

SENSE:BANDWIDTH Command

SENSe<cnum>:BANDwidth | BWIDth[:RESolution] <num>

SENSe<cnum>:BANDwidth | BWIDth:TRACk <bool>

SENSE:CORRECTION Commands

SENSe<cnum>:CORRection:CCHeck[:ACQuire] <char>

SENSe<cnum>:CORRection:CCHeck:DONE

SENSe<cnum>:CORRection:CCHeck:PARameter <Mname>

SENSE:CORRECTION:CKIT Commands

SENSe:CORRection:CKIT:CLEar[:IMMediate]

SENSe:CORRection:CKIT:COUNt?

SENSe:CORRection:CKIT:ECAL<mod>:CLISt?

SENSe:CORRection:CKIT:ECAL<mod>:INFormation? [<char>]

SENSe:CORRection:CKIT:ECAL:LIST?

SENSe:CORRection:CKIT:IMPort <string>

SENSe:CORRection:CKIT:INITialize[:IMMediate]

SENSe:CORRection:CKIT:LOAD <string>

S**ENSE:CORRECTION:COLLECT**

SENSe:CORRection:COLLect[:ACQuire] <class>[,sub]

SENSe:CORRection:COLLect:APPLy

SENSe<cnum>:CORRection:COLLect:METHod <char>

SENSe<cnum>:CORRection:COLLect:SAVE

SENSE:CORRECTION:COLLECT:CKIT Commands

SENSe:CORRection:COLLect:CKIT:CONNector:ADD <family>,<start>,<stop>,<z0>,<gender>,<media>,<cutoff>

SENSe:CORRection:COLLect:CKIT:CONNector:CATalog?

SENSe:CORRection:COLLect:CKIT:CONNector:DELete

SENSe:CORRection:COLLect:CKIT:CONNector:FNAME <name>

SENSe:CORRection:COLLect:CKIT:CONNector:SNAMe <family>,<gender>,<port>

SENSe:CORRection:COLLect:CKIT:DESCription <string>

SENSe:CORRection:COLLect:CKIT:INFormation? <module>[,char]

SENSe:CORRection:COLLect:CKIT:NAME <name>

SENSe:CORRection:COLLect:CKIT:OLABel<class> <name>

SENSe:CORRection:COLLect:CKIT:OLIST[class]?

SENSe:CORRection:COLLect:CKIT:ORDer<class> <std> [,<std>] [,<std>] [,<std>] [,<std>] [,<std>] [,<std>]

SENSe:CORRection:COLLect:CKIT:RESet <num>

SENSe:CORRection:COLLect:CKIT[:SELect] <num>

SENSe:CORRection:COLLect:CKIT:STANdard:C0 <num>

SENSe:CORRection:COLLect:CKIT:STANdard:C1 <num>

SENSe:CORRection:COLLect:CKIT:STANdard:C2 <num>

SENSe:CORRection:COLLect:CKIT:STANdard:C3 <num>

SENSe:CORRection:COLLect:CKIT:STANdard:CHARacter <char>

SENSe:CORRection:COLLect:CKIT:STANdard:DELay <num>

SENSe:CORRection:COLLect:CKIT:STANdard:FMAX <num>

SENSe:CORRection:COLLect:CKIT:STANdard:FMIN <num>

SENSe:CORRection:COLLect:CKIT:STANdard:IMPedance <num>

SENSe:CORRection:COLLect:CKIT:STANdard:L0 <num>

SENSe:CORRection:COLLect:CKIT:STANdard:L1 <num>

SENSe:CORRection:COLLect:CKIT:STANdard:L2 <num>

SENSe:CORRection:COLLect:CKIT:STANdard:L3 <num>

SENSe:CORRection:COLLect:CKIT:STANdard:LABel <name>

SENSe:CORRection:COLLect:CKIT:STANdard:LOSS <num>

SENSe:CORRection:COLLect:CKIT:STANdard:REMove

SENSe:CORRection:COLLect:CKIT:STANdard:SDEScription <string>

SENSe:CORRection:COLLect:CKIT:STANdard[:SELECT] <num>

SENSe:CORRection:COLLect:CKIT:STANdard:TYPE <char>

SENSe:CORRection:COLLect:CKIT:STANdard:TZImag <num>

SENSe:CORRection:COLLect:CKIT:STANdard:TZReal <num>

SENSe:CORRection:COLLect:CKIT:TRLoption:IMPedance <char>

SENSe:CORRection:COLLect:CKIT:TRLoption:RPLane <char>

SENSE:CORRECTION:COLLECT:GUIDED Commands

SENSe<cnum>:CORRection:COLLect:GUIDed:ACQuire <std>

SENSe<cnum>:CORRection:COLLect:GUIDed:CKIT:PORT<pnum>:CATalog?

SENSe<cnum>:CORRection:COLLect:GUIDed:CKIT:PORT<pnum>[:SELect] <kit>

SENSe<cnum>:CORRection:COLLect:GUIDed:CONNector:CATalog?

SENSe<cnum>:CORRection:COLLect:GUIDed:CONNector:PORT<pnum>[:SELect] <conn>

SENSe<cnum>:CORRection:COLLect:GUIDed:DESCription? <step>

SENSe<cnum>:CORRection:COLLect:GUIDed:INITiate

SENSe<cnum>:CORRection:COLLect:GUIDed:METHod

SENSe<cnum>:CORRection:COLLect:GUIDed:SAVE

SENSe<cnum>:CORRection:COLLect:GUIDed:STEPs?

SENSE:CORRECTION:COLLECT:SESSION Commands

SENSe<ch>:CORRection:COLLect:SESSion<n>:ACQuire <step>

SENSe<ch>:CORRection:COLLect:SESSion<n>:CKIT:PORT<pnum>:CATalog?

SENSe<ch>:CORRection:COLLect:SESSion<n>:CKIT:PORT<pnum>[:SELect] <kit>

SENSe<ch>:CORRection:COLLect:SESSion<n>:CONNector:PORT<pnum>[:SELect] <conn>

SENSe<ch>:CORRection:COLLect:SESSion<n>:DESCription? <step>

SENSe<ch>:CORRection:COLLect:SESSion<n>:INITiate<string>

SENSe<ch>:CORRection:COLLect:SESSion<n>:DONE

SENSe<ch>:CORRection:COLLect:SESSion<n>:SAVE?

SENSe<ch>:CORRection:COLLect:SESSion<n>:STEPs?

SENSE:CORRECTION:COLLECT:SESSION:SMC Commands

SENSe<ch>:CORRection:COLLect:SESSion<n >:SMC:ECAL:CHARacteriza <mod> ,<char>

SENSe<ch>:CORRection:COLLect:SESSion<n>:SMC:PWRCal:SRCPort <char>

SENSe<ch>:CORRection:COLLect:SESSion<n>:SMC:TWOPort:ECAL:ORIentation[:STATe] <bool>

SENSe<ch>:CORRection:COLLect:SESSion<n>:SMC:TWOPort:ECAL:PORTmap <mod>, <string>

SENSe<ch>:CORRection:COLLect:SESSion<n >:SMC:TWOPort:METHod <char>

SENSe<ch>:CORRection:COLLect:SESSion<n >:SMC:TWOPort:OMITisolat <bool>

SENSe<ch>:CORRection:COLLect:SESSion<n >:SMC:TWOPort:OPTion <char>

SENSE:CORRECTION:COLLECT:SESSION:VMC Commands

SENSe<ch>:CORRection:COLLect:SESSion<n>:VMC:CHARacterize:CAL:FILename <string>

SENSe<ch>:CORRection:COLLect:SESSion<n>:VMC:MIXer:CHARacterize:CAL: OPTion <char>

SENSe<ch>:CORRection:COLLect:SESSion<n>:VMC:MIXer:CHARacterize:CAL: REVerse <bool>

SENSe<ch>:CORRection:COLLect:SESSion<n>:VMC: MIXer:ECAL:CHARacteriza <mod> ,<char>

SENSe<ch>:CORRection:COLLect:SESSion<n>:VMC:MIXer:ECAL:PORTmap <mod>, <string>

SENSe<ch>:CORRection:COLLect:SESSion<n>:VMC:OPERation <char>

SENSe<ch>:CORRection:COLLect:SESSion<n >:VMC:TWOPort:ECAL:CHARacteriza <mod> ,<char>

SENSe<ch>:CORRection:COLLect:SESSion<n >:VMC:TWOPort:ECAL:ORIentation[:STATe] <bool>

SENSe<ch>:CORRection:COLLect:SESSion<n >:VMC:TWOPort:ECAL:PORTmap <mod>, <string>

SENSe<ch>:CORRection:COLLect:SESSion<n >:VMC:TWOPort:METHod <char>

SENSe<ch>:CORRection:COLLect:SESSion<n >:VMC:TWOPort:OMITisolat <bool>

SENSe<ch>:CORRection:COLLect:SESSion<n >:VMC:TWOPort:OPTion <char>

SENSE:CORRECTION:CSET Commands

SENSe:CORRection:CSET:CATalog?

SENSe<cnum>:CORRection:CSET:DELete <string>

SENSe<cnum>:CORRection:CSET:DESCription <string>

SENSe<cnum>:CORRection:CSET:GUID <string>

SENSe<cnum>:CORRection:CSET[:SELect] <char>

SENSe<cnum>:CORRection:CSET:SAVE <char>

SENSe<cnum>:CORRection:CSET:TYPE:CATalog?

SENSE:CORRECTION:

SENSe<cnum>:CORRection:EXTension:PORT<pnum>[:TIME] <num>

SENSe<cnum>:CORRection:EXTension:RECeiver<Rnum>[:TIME] <num>

SENSe<cnum>:CORRection:EXTension[:STATe] <ON | OFF>

SENSe:CORRection:IMPedance:INPut:MAGNitude <num>

SENSe<cnum>:CORRection:INTerpolation[:STATe] <ON | OFF>

SENSe<cnum>:CORRection:ISOLation[:STATe] <ON | OFF>

SENSe:CORRection:PREFerences:ECAL:ORIentation[:STATe] <ON|OFF>

SENSe:CORRection:PREFerences:ECAL:PMAP <module>,<string>

SENSe<cnum>:CORRection:RVELocity:COAX <num>

SENSe:CORRection:SFORward[:STATe] <boolean>

SENSe<cnum>:CORRection[:STATe] <ON | OFF>

SENSe:CORRection:TSTandards[:STATe] <boolean>

SENSe:CORRection:TYPE:CATalog? <char>

SENSE:COUPLE Command

SENSe<cnum>:COUPle <ALL | NONE>

SENSe<cnum>:COUPle:PARameter

SENSE:FREQUENCY Commands

SENSe<cnum>:FREQuency:CENTer <num>

SENSe<cnum>:FREQuency[:CW |:FIXed] <num>

SENSe<cnum>:FREQuency:SPAN <num>

SENSe<cnum>:FREQuency:STARt <num>

SENSe<cnum>:FREQuency:STOP <num>

SENSe:IF Commands

SENSe<cnum>:IF:FILTer:SAMPle:COUNt <num>

SENSe<cnum>:IF:FILTer:SAMPle:COUNt:MODE <char>

SENSe<cnum>:IF:FILTer:SAMPle:PERiod <num>

SENSe<cnum>:IF:FILTer:SAMPle:PERiod:CATalog?

SENSe<cnum>:IF:FILTer:SAMPle:PERiod:MODE <char>

SENSe<cnum>:IF:GAIN:ALL[:STATe] <char>

SENSe<cnum>:IF:GAIN:LEVel <id>, <level>

SENSe<cnum>:IF:GAIN[:STATe]?, <id>

SENSe<cnum>:IF:GATE:STATe <boolean>

SENSe<cnum>:IF:SOURce:PATH <id>, <char>

SENSe:MIXer Commands

SENSe<ch>:MIXer:APPLy

SENSe<ch>:MIXer:AVOidspurs <bool>

SENSe<ch>:MIXer:CALCulate <char>

SENSe<ch>:MIXer:IF:FREQuency:DENominator <value>

SENSe<ch>:MIXer:IF:FREQuency:FIXed <value>

SENSe<ch>:MIXer:IF:FREQuency:NUMerator <value>

SENSe<ch>:MIXer:IF:FREQuency:SIDeband <value>

SENSe<ch>:MIXer:IF:FREQuency:STARt <value>

SENSe<ch>:MIXer:IF:FREQuency:STOP <value>

SENSe<ch>:MIXer:INPut:FREQuency:DENominator <value>

SENSe<ch>:MIXer:INPut:FREQuency:FIXed<value>

SENSe<ch>:MIXer:INPut:FREQuency:NUMerator <value>

SENSe<ch>:MIXer:INPut:FREQuency:STARt <value>

SENSe<ch>:MIXer:INPut:FREQuency:STOP <value>

SENSe<ch>:MIXer:INPut:POWer <value>

SENSe<ch>:MIXer:LO<n>:FREQuency:DENominator <value>

SENSe<ch>:MIXer:LO<n>:FREQuency:FIXed <value>

SENSe<ch>:MIXer:LO<n>:FREQuency:ILTI <value>

SENSe<ch>:MIXer:LO<n>:FREQuency:MODE <char>

SENSe<ch>:MIXer:LO<n>:FREQuency:NUMerator <value>

SENSe<ch>:MIXer:LO<n>:NAME <value>

SENSe<ch>:MIXer:LO<n>:POWer <value>

SENSe<ch>:MIXer:LOAD <name>

SENSe<ch>:MIXer:OUTput:FREQuency:SIDeband <value>

SENSe<ch>:MIXer:OUTput:FREQuency:STARt <value>

SENSe<ch>:MIXer:OUTput:FREQuency:STOP <value>

SENSe<ch>:MIXer:SAVE <name>

SENSe<ch>:MIXer:STAGe <n>

SENSe:OFFSet Commands

SENSe<cnum>:OFFSet:CW <bool>

SENSe<cnum>:OFFSet:DIVisor <num>

SENSe<cnum>:OFFSet:MULTiplier <num>

SENSe<cnum>:OFFSet:OFFSet <num>

SENSe<cnum>:OFFSet:STARt?

SENSe<cnum>:OFFSet:[STATe] <bool>

SENSe<cnum>:OFFSet:STOP?

SENSE:POWER Command

SENSe<cnum>:POWer:ATTenuation <recvr>,<num>

SENSE:ROSCILLATOR Command

SENSe:ROSCillator:SOURce?

SENSE:SEGMENT Commands

SENSe<cnum>:SEGMent<snum>:ADD

SENSe<cnum>:SEGMent:ARBitrary

SENSe<cnum>:SEGMent<snum>:BWIDth[:RESolution] <num>

SENSe<cnum>:SEGMent:BWIDth[:RESolution]:CONTrol <ON | OFF>

SENSe<cnum>:SEGMent:COUNt?

SENSe<cnum>:SEGMent<snum>:DELete

SENSe<cnum>:SEGMent:DELete:ALL

SENSe<cnum>:SEGMent<snum>:FREQuency:CENTer <num>

SENSe<cnum>:SEGMent<snum>:FREQuency:SPAN <num>

SENSe<cnum>:SEGMent<snum>:FREQuency:START <num>

SENSe<cnum>:SEGMent<snum>:FREQuency:STOP <num>

SENSe<cnum>:SEGMent<snum>:POWer[<port>][:LEVel] <num>

SENSe<cnum>:SEGMent:POWer[:LEVel]:CONTrol <ON | OFF>

SENSe<cnum>:SEGMent<snum>[:STATe] <ON | OFF>

SENSe<cnum>:SEGMent<snum>:SWEep:POINts <num>

SENSe<cnum>:SEGMent<snum>:SWEep:TIME <num>

SENSe<cnum>:SEGMent:SWEep:TIME:CONTrol <ON | OFF>

SENSe<cnum>:SEGMent<snum>:X:SPACing <char>

SENSE:SWEEP Commands

SENSe<cnum>:SWEep:DWELl <num>

SENSe<cnum>:SWEep:DWELl:AUTO <ON | OFF>

SENSe<cnum>:SWEep:GENeration <char>

SENSe<cnum>:SWEep:GROups:COUNt <num>

SENSe<cnum>:SWEep:MODE <char>

SENSe<cnum>:SWEep:POINts <num>

SENSe<cnum>:SWEep:SRCPort <1 | 2>

SENSe<cnum>:SWEep:TIME <num>

SENSe<cnum>:SWEep:TIME:AUTO <ON | OFF>

SENSe<cnum>:SWEep:TRIGger:DELay <num>

SENSe<cnum>:SWEep:TRIGger:POINt <ON | OFF>

SENSe<cnum>:SWEep:TYPE <char>

SENSE:X: Command

SENSe<cnum>:X:VALues?

SOURCE Commands

SOURce<cnum>:POWer<port>:ATTenuation <num>

SOURce<cnum>:POWer<port>:ATTenuation:AUTO <ON | OFF>

SOURce<cnum>:POWer:CENTer <num>

SOURce<cnum>:POWer:COUPle <ON | OFF>

SOURce<cnum>:POWer:DETector <INTernal | EXTernal>

SOURce<cnum>:POWer<port>[:LEVel][:IMMediate] [:AMPLitude] <num>

SOURce<cnum>:POWer[:LEVel]:SLOPe <int>

SOURce<cnum>:POWer[:LEVel]:SLOPe:STATe <ON|OFF>

SOURce<cnum>:POWer:SPAN <num>

SOURce<cnum>:POWer:STARt <num>

SOURce<cnum>:POWer:STOP <num>

SOURCE:POWER:CORRECTION Commands

SOURce<cnum>:POWer<port>:CORRection:COLLect:ABORt

SOURce<cnum>:POWer<port>:CORRection:COLLect[:ACQuire] <char>

SOURce<cnum>:POWer<port>:CORRection:COLLect:AVERage[:COUNt] <num>

SOURce<cnum>:POWer<port>:CORRection:COLLect:AVERage:NTOLerance <num>

SOURce<cnum>:POWer:CORRection:COLLect:DISPlay[:STATe] <ON | OFF>

SOURce<cnum>:POWer:CORRection:COLLect:FCHeck[:STATe] <ON | OFF>

SOURce<cnum>:POWer<port>:CORRection:COLLect:ITERation[:COUNt] <num>

SOURce<cnum>:POWer<port>:CORRection:COLLect:ITERation:TOLerance <num>

SOURce<cnum>:POWer<port>:CORRection:COLLect:METHod <char>

SOURce<cnum>:POWer<port>:CORRection:COLLect:SAVE

SOURce<cnum>:POWer:CORRection:COLLect:<pmChan>SENsor[:FRANge] <num1>,<num2>

SOURce<cnum>:POWer:CORRection:COLLect:<pmChan>SENsor:RCFactor <num>

SOURce<cnum>:POWer:CORRection:COLLect:<pmChan>SENsor:SELect

SOURce<cnum>:POWer:CORRection:COLLect:TABLe:DATA <data>

SOURce<cnum>:POWer:CORRection:COLLect:TABLe:FREQuency <data>

SOURce<cnum>:POWer:CORRection:COLLect:TABLe:LOSS[:STATe] <ON | OFF>

SOURce<cnum>:POWer:CORRection:COLLect:TABLe:POINts?

SOURce<cnum>:POWer:CORRection:COLLect:TABLe[:SELect] <char>

SOURce<cnum>:POWer<port>:CORRection:DATA <data>

SOURce<cnum>:POWer<port>:CORRection:LEVel <num>

SOURce<cnum>:POWer<port>:CORRection:OFFSet[:MAGNitude] <num>

SOURce<cnum>:POWer<port>:CORRection[:STATe] <ON|OFF>

STATUS REGISTER Commands

Status Byte Register Commandsr

STATus:QUEStionable:<keyword>

STATus:QUEStionable:INTegrity <keyword>

STATus:QUEStionable:INTegrity:HARDware<keyword>

STATus:QUEStionable:INTegrity:MEASurement<n> <keyword>

STATus:QUEStionable:LIMit<n> <keyword>

STATus:QUEStionable:DEFine<keyword>

STATus:QUEStionable:DEFine:USER<1|2|3><keyword>

Standard Event Status Register Commands

STATus:OPERation<keyword>

STATus:OPERation:AVERaging<n> <keyword>

STATus:OPERation:DEFine<keyword>

STATus:OPERation:DEFine:USER<1|2|3><keyword>

SYSTEM Commands

SYSTem:CHANnels:HOLD

SYSTem:CHANnels:RESume

SYSTem:COMMunicate:GPIB:PMETer:ADDRess <num>

SYSTem:COMMunicate:GPIB:RDEVice:CLOSe <ID>

SYSTem:COMMunicate:GPIB:RDEVice:OPEN <bus>, <addr>, <timeout>

SYSTem:COMMunicate:GPIB:RDEVice:READ? <ID>

SYSTem:COMMunicate:GPIB:RDEVice:RESet

SYSTem:COMMunicate:GPIB:RDEVice:WBINary

SYSTem:COMMunicate:GPIB:RDEVice:WBLock

SYSTem:COMMunicate:GPIB:RDEVice:WRITe <ID>,<string>

SYSTem:CORRection:WIZard <char>

SYSTem:ERRor?

SYSTem:ERRor:COUNt?

SYSTem:FPReset

SYSTem:MACRo:COPY:CHANnel

SYSTem:PRESet

SYSTem:SECurity[:LEVel]

SYSTem:UPReset

SYSTem:UPReset:FPANel[:STATe] <bool>

SYSTem:UPReset:LOAD <file>

SYSTem:UPReset:SAVE[:STATe]

TRIGGER Commands

TRIGger:DELay<num>

TRIGger[:SEQuence]:LEVel <char> OBSOLETE

TRIGger[:SEQuence]:SCOPe <char>

TRIGger[:SEQuence]:SOURce <char>

SCPI Command Tree

See a List of ALL SCPI commands

IEEE- 488.2 Common Commands

Local Lockout

ABORt	Stops all sweeps

CALCulate Click to hide and show CALC commands

:CORRection	Sets Electrical Delay and Phase Offset
:CUSTom	Creates custom measurements
:DATA	Sends and queries data.
:FILTer	Sets time domain gating
:FORMat	Sets the display format
:FSIM:BALun	Configures Balanced measurements.
:FSIM:SEND	Controls Fixture Embed/De-embedding.
:FUNCtion	Controls Trace Statistics.
:LIMit	Controls limit lines for pass / fail testing.
:MARKer	Controls the marker settings.
:MATH	Performs math on the memory trace.
:MIXer	Sets the X-axis display for FCA measurements.
:NORMalize	Performs normalization for a receiver power cal.
:OFFSet	Sets Phase and Mag offset.
:PARameter	Creates and deletes measurements.
:RDATa?	Queries receiver data.
:SMOothing	Controls point-to-point smoothing.
:TRANsform	Controls time domain transform settings.

CONTrol	Controls the rear-panel connectors
DISPlay	Controls the display settings
FORMat	Sets the format for data transfer
HCOPy	Controls hardcopy printing
INITiate	Sets continuous or manual triggering
MMEMory	Saves and recalls instrument states
OUTPut	Turns RF power ON and OFF

SENSe Click to hide and show SENSe commands

:AVErage	Sets sweep averaging parameters
:BANDwidth	Specifies the IF filter bandwidth
:CORRection	Provides non-guided calibration capability
:CORR:CKIT	Manage Cal Kits
:CORR:COLL:CKIT	Defines calibration standards
:CORR:CSET	Manages Cal Sets
:CORR:COLL:GUID	Provides Guided calibration capability
:CORR:COLL:SESS	Provides VMC and SMC calibration capability.
:COUPle	Sets sweep as Chopped or Alternate
:FREQuency	Controls frequency sweep functions
:IF	Controls IF Access capability
:MIXer	Sets mixer measurement configuration
:OFFSet	Sets frequency offset functions
:POWer	Sets receiver attenuation and overpower protection
:ROSCillator	Returns the source of the reference oscillator.
:SEGMent	Defines the segment sweep settings.
:SWEep	Specifies the sweep modes of the analyzer.
:X:VALues	Returns X-axis values

SOURce	Controls the power to the DUT
SOURce:POWer	Provides for Source Power Correction
STATus	Reads the analyzer status registers
SYSTem	Controls the analyzer defaults
TRIGger	Starts or ends a measurement

Abort Command

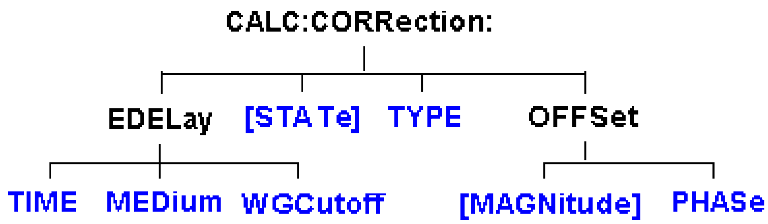
ABORt

(Write-only) Stops all sweeps - then resume per current trigger settings. This command is the same as INITtiate:IMMediate (restart) except if a channel is performing a single sweep, ABORt will stop the sweep, but not initiate another sweep.

Examples	`ABOR` `abort`
Query Syntax	Not applicable
Overlapped?	No
Default	Not applicable

Calc:Correction Commands

Controls error correction functions.



- Click on a blue keyword to view the command details.
- New See Calibrating the PNA Using SCPI
- See a List of all commands in this block.

Note: CALCulate commands act on the selected measurement. You can select one measurement in each channel. To select the measurement use CALC<cnum>:PAR:SEL <MeasName>.

CALCulate<cnum>:CORRection:EDELay:MEDium <char>

(Read-Write) Sets the media used when calculating the electrical delay.

Parameters	
<cnum>	Any existing channel number. If unspecified, value is set to 1.
<num>	Choose from: **COAX** for coaxial medium, **WAVE**guide for waveguide medium.
Examples	`CALC:CORR:EDEL:MED COAX` `calc3:corr:edelay:medium waveguide`
Query Syntax	CALCulate<cnum>:CORRection:EDELay:MEDium?
Return Type	Character
Overlapped?	No
Default	**COAX**

CALCulate<cnum>:CORRection:EDELay:TIME <num>

(Read-Write) Sets the electrical delay for the selected measurement. **Critical Note:**

Parameters

<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<num>	Electrical delay in seconds. Choose any number between**:** **-10.00** and **10.00** Use SENS:CORR:RVEL:COAX <num> to set Velocity factor.
Examples	`CALC1:CORR:EDEL:TIME 1NS` `calculate2:correction:time 0.5e-12`
Query Syntax	CALCulate:CORRection:EDELay:TIME?
Return Type	Numeric
Overlapped?	No
Default	0 seconds

CALCulate<cnum>:CORRection:EDELay:WGCutoff <num>

(Read-Write) Sets the waveguide cutoff frequency used when the electrical delay media is set to WAVEguide. (See CALCulate:CORRection:EDELay:MEDium <char>.)

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1.
<num>	Waveguide cutoff frequency used with the electrical delay calculation.
Examples	`CALC:CORR:EDEL:WGC 18.067 GHz` `calculate3:correction:edelay:wgcutoff 14.047 ghz`
Query Syntax	CALCulate<cnum>:CORRection:EDELay:WGCutoff?
Return Type	Numeric
Overlapped?	No
Default	45 MHz

CALCulate<cnum>: CORRection:[STATe] <bool>

(Read-Write) Turns error correction ON or OFF for the selected measurement on the specified channel. **Critical Note:**

Note: You must set the error correction type (CALC:CORR:TYPE) before turning error correction ON.

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1
<bool>	Correction state. Choose from: **0** - Correction OFF **1** - Correction ON
Examples	**CALC:CORR ON** **calculate:correction:state off**
Query Syntax	CALCulate<cnum>: CORRection: STATe?
Return Type	Boolean
Overlapped?	No
Default	Not Applicable

CALCulate<cnum>:CORRection:TYPE <string>

(Read-Write) Sets the Cal Type for the selected measurement on the specified channel. This is used when a Cal Set is applied.

- Use SENS:CORR:TYPE:CAT? to list the Cal Types in the PNA.
- Use SENS:CORR:CSET:TYPE:CAT? to list the Cal Types contained in the active Cal Set for the channel.
- Use SENS:CORR:COLL:METH to set the Cal type to perform a new calibration,

Critical Note:

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1

<string> Cal Type. Choose from:

- "Full 1 Port"
- "Full 2 Port SOLT"
- "Full 2 Port TRL"
- "Full 3 Port SOLT"
- "Open Response"
- “SMC_2P” or “SMC”
- “SMCRsp+IN”
- “SMCRsp+OUT”
- “SMCRsp”
- "Short Response"
- "Thru Response"
- "Thru Response and Isolation"
- "VMC"

Learn more applying SMC cal sets

Examples

```
CALC:CORR:TYPE "VMC"
```

Query Syntax CALCulate<cnum>:CORRection:TYPE?

Return Type String

Overlapped? No

Default Not Applicable

CALCulate<cnum>:CORRection:OFFSet[:MAGNitude] <num>

(Read-Write) For Receiver Power Calibration, specifies the power level to which the selected (unratioed) measurements data is to be adjusted. This command applies only when the selected measurement is of unratioed power. :**Critical Note:**

Parameters

<cnum> Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.

<num> Cal power level in dBm. No limits are enforced on this value, but the PNA receivers themselves have maximum and minimum power specifications (that may differ between PNA models) which this value must comply with for a valid receiver power cal.

Examples

```
CALC:CORR:OFFS 10DBM
calculate1:correction:offset:magnitude maximum
```

Query Syntax CALCulate<cnum>:CORRection:OFFSet[:MAGNitude]?

Return Type Numeric

Overlapped? No

Default 0dBm

CALCulate<cnum>:CORRection:OFFSet:PHASe <num>[<char>] - OBSOLETE

(Read-Write) Replaced with CALC:OFFSet:PHASe

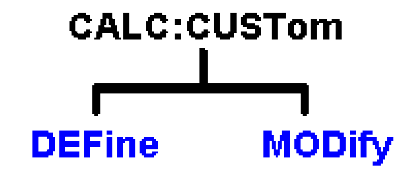
Sets the phase offset for the selected measurement. **Critical Note:**

Parameters

<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<num>	Offset phase value. Choose any number between: **-360** and **360**
<char>	Units for phase. OPTIONAL. Choose either: **DEG** - Degrees (default) **RAD** - Radians
Examples	**CALC:CORR:OFFS:PHAS 10** **calculate:correction:offset:phase 20rad**
Query Syntax	CALCulate:CORRection:OFFSet:PHASe?
Return Type	Numeric, returned value always in degrees
Overlapped?	No
Default	0 degrees

Calculate:Custom Commands

Provides capability to create and modify Frequency Converter Application (opt 083) measurements.



CALCulate<cnum>:CUSTom:DEFine <Mname>, <ProgID> [,param]

(Write-only) Creates a custom measurement. The custom measurement is not automatically displayed. You must also do the following:

- Use DISP:WIND:STATe to create a window if it doesn't already exist.·
- Use DISP:WIND:TRAC:FEED to display the measurement
- Select the measurement (CALC:PAR:SEL) before making additional settings.

See an example using this command to create a VMC and SMC measurement

Parameters

<cnum> Any existing channel number. If unspecified, value is set to 1.

<Mname> Name of the measurement. Any non-empty, unique string, enclosed in quotes.

<ProgID> The type of custom measurement. Choose from:

- FCA.VMCMeas
- SMC_Forward.SMC_ForwardMeas
- SMC_Reverse.SMC_ReverseMeas

[param] Optional parameter specifies the measurement parameter to create.

Program ID	Valid Parameters:
FCA.VMCMeas	"S11" "VC21" "S22"
SMC_Forward.SMC_ForwardMeas	"S11" "SC21" "S22" "IPwr" "OPwr"
SMC_Reverse.SMC_ReverseMeas	"S11" "SC12" "S22" "IPwr" "OPwr"

Examples

```
CALC2:CUST:DEF 'My VC21', 'FCA.VMCMeas','S22'
```

Query Syntax Not applicable

Overlapped? No

Default Not applicable

CALCulate<cnum>:CUSTom:MODify <param>

(Write-only) Changes the selected custom measurement to a different parameter.

See an example using this command for a VMC and SMC measurement

Parameters

<cnum> Channel of the custom measurement to be changed. First, select the measurement using CALC:PAR:SEL.

<param> Parameter to change the custom measurement to. Select a parameter that is valid for the type of measurement.

Type	Valid Parameters
VMC Measurement	"S11" "VC21" "S22"
SMC Forward Measurement	"S11" "SC21" "S22" "IPwr" "OPwr"
SMC Reverse Measurement	"S11" "SC12" "S22" "IPwr" "OPwr"

Examples

```
SYST:PRES
CALC2:CUST:DEF 'My VC21', 'FCA.VMCMeas'
CALC:PAR:SEL 'My VC21'
CALC2:CUST:MOD 'S22'
```

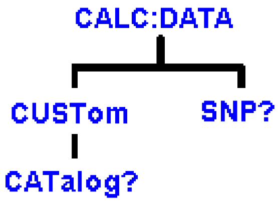
Query Syntax Not applicable

Overlapped? No

Default Not applicable

Calc:Data Commands

Controls sending and receiving data with the PNA



- Click on a blue keyword to view the command details.
- See a List of all commands in this block.
- See Data Access Map

CALCulate<cnum>:DATA <char>,<data>

Write Writes Measurement data, Memory data, Normalization Divisor data, or Error terms.

Data type depends on FORM:DATA command.

CALCulate<cnum>:DATA? <char>

Read Returns Measurement data, Memory data, Normalization Divisor data, or Error terms.

Data type depends on FORM:DATA command.

To write or read

- **Measurement (DATA)**
- **Memory (MEM)**
- **Normalization Divisor (DIV)** (Receiver Power Cal error term)

Parameters

<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.	
<char>	**FDATA**	Formatted trace data from **measResult** location Returns TWO numbers per data point for Polar and Smith Chart format Returns one number per data point for all other formats
	SDATA	Corrected complex trace data from **rawMeas** location Returns TWO numbers per data point
	FMEM	Formatted memory data from **memResult** location Returns TWO numbers per data point for Polar and Smith Chart format Returns one number per data point for all other formats
	SMEM	Corrected complex data from **rawMemory** location Returns TWO numbers per data point

SDIV Receiver Power Cal error term.

Complex data from **Normalization Divisor** location

Returns TWO numbers per data point

Notes:

When reading data from, or writing data to, the normalization divisor, you must first create a divisor trace using CALC:NORMalize[:IMMediate].

If normalization interpolation is ON and the number of points changes after the initial normalization, the divisor data will then be interpolated.

To write or read **Error Terms:**

Note: All Error Terms return two numbers per trace point

For **Response Open and Short** calibrations:

SCORR3 Reflection Tracking Error Term

For **Response Thru** calibrations:

SCORR6 Transmission Tracking Error Term

For **Response Thru and Isolation** calibrations:

Specify this **<char>**	to get this **Term...**
SCORR4	Isolation
SCORR6	Transmission Tracking

For **1-Port** calibrations:

Specify this **<char>**	to get this **Term...**
SCORR1	Directivity
SCORR2	Source Match
SCORR3	Reflection Tracking

For **2-Port SOLT and TRL** calibrations

Specify this **<char>**	to get this **Term...**
SCORR1	Forward Directivity
SCORR2	Forward Source Match
SCORR3	Forward Reflection Tracking
SCORR4	Forward Isolation
SCORR5	Forward Load Match
SCORR6	Forward Transmission Tracking
SCORR7	Reverse Directivity
SCORR8	Reverse Source Match
SCORR9	Reverse Reflection Tracking
SCORR10	Reverse Isolation
SCORR11	Reverse Load Match
SCORR12	Reverse Transmission Tracking

For **FULL 3-Port SOLT** calibrations

Specify this **<char>**	to get this **Term...**	for this **Receiver Port .**
SCORR13	Directivity	**3** (S33)
SCORR14	Source Match	**3** (S33)
SCORR15	Reflection Tracking	**3** (S33)
SCORR16	Isolation	**3** (S31)
SCORR17	Load Match	**3** (S31)
SCORR18	Trans Tracking	**3** (S31)
SCORR19	Isolation	**1** (S13)
SCORR20	Load Match	**1** (S13)
SCORR21	Trans Tracking	**1** (S13)
SCORR22	Isolation	**3** (S32)
SCORR23	Load Match	**3** (S32)
SCORR24	Trans Tracking	**3** (S32)
SCORR25	Isolation	**2** (S23)
SCORR26	Load Match	**2** (S23)
SCORR27	Trans Tracking	**2** (S23)

EXAMPLE

```
CALC:DATA FDATA,Data(x)
calculate2:data sdata,data(r,i)
```

See another example using this command.

Return Type: Block data

Overlapped? - No

Default - Not Applicable

Notes:

- When querying memory, you must first store a trace into memory using CALC:MATH:MEMorize.
- When querying error terms, there must be error terms in the analyzer.
- If interpolation is ON and the number of points changes after the initial calibration, the error terms will then be the interpolated results.
- To get and put receiver data, see CALC:RDATA?
- To get uncorrected ratioed data, turn correction OFF and use Calc:Data SDATA.
- CALCulate commands act on the selected measurement. You can select one measurement in each channel. Therefore, you can have up to four measurements selected at the same time. Select the measurement for each channel using CALC:PAR:SEL.

Learn more about Error Terms

CALCulate<cnum>:DATA:CUSTom <name>,<data>

(Read-Write) Reads or writes data from a custom-named measurement buffer. Specify the measurement using CALCulate:PARameter:SELect. Critical Note:

Parameters

<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<name>	Name of the buffer to be read or written
<data>	Data to be read or written to the custom buffer. Format as one number per data point.
Examples	`CALC:DATA:CUST 'VectorResult0',0,1,2,3,4,5 'Write` `CALC:DATA:CUST? 'VectorResult0' 'Read`
Query Syntax	CALCulate:DATA:CUSTom? <name>
Return Type	Depends on Form:Data

Overlapped? No

Default Not Applicable

CALCulate<cnum>:DATA:CUSTom:CATalog?

(Read-only) Reads the list of buffer names (comma separated list of string values) available from the selected parameter. Specify the measurement using CALCulate:PARameter:SELect. Critical Note:

Parameters

<cnum> Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.

Examples

```
CALC:DATA:CUST:CAT?
calculate:data:custom:catalog?
```

Return Type String

Overlapped? No

Default Not Applicable

CALCulate<cnum>:DATA:SNP? <n>

(Read-only) Reads SnP data from the selected measurement. Learn more about SnP data. Critical Note:

Parameters

<cnum> Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.

<n> Amount of data to return. If unspecified, <n> is set to 2. Choose from:

1 (S1P) returns data for the active measurement.

2 (S2P) returns data for the 2 port parameters associated with the current measurement. Default.

3 (S3P) returns data for the 3 port parameters associated with the current measurement. Valid only on instruments with more than 2 ports.

SnP data can be output using several data formatting options. See the command MMEM:STORe:TRACe:FORMat:SNP.

See also MMEM:STOR <file>.<snp>

Examples

```
CALC:PAR:DEF “MyMeasurement”, S11
CALC:PAR:SEL “MyMeasurement”
CALC:DATA:SNP? 1
```

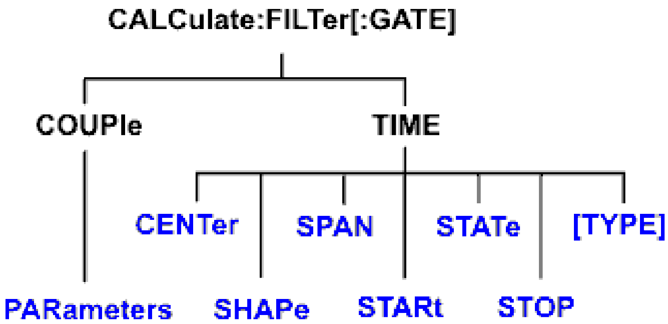
Return Type Depends on FORMat:DATA

Overlapped? No

Default Not Applicable

Calc:Filter Commands

Controls the gating function used in time domain measurements. The gated range is specified with either (start / stop) or (center / span) commands.



- Click on a blue keyword to view the command details.
- See a List of all commands in this block.
- Learn about Gating

Note: CALCulate commands act on the selected measurement. You can select one measurement in each channel. Select the measurement for each channel using CALC:PAR:SEL.

CALCulate<cnum>:FILTer[:GATE]:COUPle:PARameters <num>

(Read-Write) Specifies the time domain gating parameters to be coupled. The settings for those parameters will be copied from the selected measurement to all other measurements on the channel.

- To enable Trace Coupling, use SENS:COUP:PAR
- To specify Transform parameters to couple, use CALC:TRAN:COUP:PAR

arn more about Time Domain Trace Coupling

Critical Note:

Parameters

<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<num>	(Numeric) Parameters to couple. To specify more than one parameter, add the numbers. **1** - Gating Stimulus (Start, Stop, Center, and Span TIME settings.) **2** - Gating State (ON / OFF) **4** - Gating Shape (Minimum, Normal, Wide, and Maximum) **8** - Gating Type (Bandpass and Notch)

Examples	`'To couple all parameters:`
	`CALC:FILT:COUP:PAR 15`
	`'To couple Stimulus and Type:`
	`calculate2:filter:gate:couple:parameters 9`

Query Syntax	CALCulate<cnum>:FILTer:GATE:COUPle:PARameters?
Return Type	Numeric
Overlapped?	No
Default	13 (All parameters except **2 -** Gating State)

CALCulate<cnum>:FILTer[:GATE]:TIME:CENTer <num>

(Read-Write) Sets the gate filter center time. **Critical Note:**

Parameters

<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<num>	Center time in seconds; Choose any number between: **±** (number of points-1) / frequency span **Note**: This command will accept **MIN** or **MAX** instead of a numeric parameter. See SCPI Syntax for more information.
Examples	`CALC:FILT:GATE:TIME:CENT -5 ns` `calculate2:filter:time:center maximum`
Query Syntax	CALCulate<cnum>:FILTer[:GATE]:TIME:CENTer?
Return Type	Numeric
Overlapped?	No
Default	0

CALCulate<cnum>:FILTer[:GATE]:TIME:SHAPe <char>

(Read-Write) Sets the gating filter shape when in time domain. **Critical Note:**

Parameters

<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<char>	Choose from **MAXimum -** the widest gate filter available **WIDE -** **NORMal -** **MINimum -** the narrowest gate filter available
Examples	`CALC:FILT:GATE:TIME:SHAP MAX` `calculate2:filter:time:shape normal`
Query Syntax	CALCulate<cnum>:FILTer[:GATE]:TIME:SHAPe?
Return Type	Character
Overlapped?	No
Default	NORMal

CALCulate<cnum>:FILTer[:GATE]:TIME:SPAN <num>

(Read-Write) Sets the gate filter span time. **Critical Note:**

Parameters

<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<num>	Time span in seconds; Choose any number between: **0** and 2* [(number of points-1) / frequency span] **Note**: This command will accept **MIN** or **MAX** instead of a numeric parameter. See SCPI Syntax for more information.
Examples	`CALC:FILT:GATE:TIME:SPAN 5 ns` `calculate2:filter:time:span maximum`
Query Syntax	CALCulate<cnum>:FILTer[:GATE]:TIME:SPAN?
Return Type	Numeric
Overlapped?	No
Default	20 ns

CALCulate<cnum>:FILTer[:GATE]:TIME:STATe <boolean>

(Read-Write) Turns gating state ON or OFF. **Critical Note:**

Note: Sweep type must be set to LInear Frequency in order to use Transform Gating.

Parameters

<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<boolean>	**ON** (or 1) - turns gating ON. **OFF** (or 0) - turns gating OFF.

Examples

```
CALC:FILT:TIME:STAT ON
calculate2:filter:gate:time:state off
```

Query Syntax	CALCulate<cnum>:FILTer[:GATE]:TIME:STATe?
Return Type	Boolean (1 = ON, 0 = OFF)
Overlapped?	No
Default	OFF

CALCulate<cnum>:FILTer[:GATE]:TIME:STARt <num>

(Read-Write) Sets the gate filter start time. **Critical Note:**

Parameters

<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<num>	Start time in seconds; any number between: **±** (number of points-1) / frequency span **Note**: This command will accept **MIN** or **MAX** instead of a numeric parameter. See SCPI Syntax for more information.

Examples

```
CALC:FILT:TIME:STAR 1e-8
calculate2:filter:gate:time:start minimum
```

Query Syntax	CALCulate<cnum>:FILTer[:GATE]:TIME:STARt?
Return Type	Numeric
Overlapped?	No
Default	10 ns

CALCulate<cnum>:FILTer[:GATE]:TIME:STOP <num>

(Read-Write) Sets the gate filter stop time. **Critical Note:**

Parameters

<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<num>	Stop time in seconds; any number between: **±** (number of points-1) / frequency span **Note**: This command will accept **MIN** or **MAX** instead of a numeric parameter. See SCPI Syntax for more information.

Examples

```
CALC:FILT:TIME:STOP -1 ns
calculate2:filter:gate:time:stop maximum
```

Query Syntax CALCulate<cnum>:FILTer[:GATE]:TIME:STOP?

Return Type Numeric

Overlapped? No

Default 10 ns

CALCulate<cnum>:FILTer[:GATE]:TIME[:TYPE] <char>

(Read-Write) Sets the type of gate filter used. **Critical Note:**

Parameters

<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<char>	Choose from: **BPASs -** Includes (passes) the range between the start and stop times. **NOTCh** - Excludes (attenuates) the range between the start and stop times.

Examples

```
CALC:FILT:TIME BPAS
calculate2:filter:gate:time:type notch
```

Query Syntax CALCulate<cnum>:FILTer[:GATE]:TIME[:TYPE]?

Return Type Character

Overlapped? No

Default BPAS

Calc:Format Command

Note: CALCulate commands act on the selected measurement. You can select one measurement in each channel. Select the measurement for each channel using CALC:PAR:SEL.

- See an example using this command.
- See a List of all commands in this block.
- Learn About Data Format

CALCulate<cnum>:FORMat <char>

(Read-Write) Sets the display format for the measurement. **Critical Note:**

Parameters

<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<char>	Choose from: MLINear, MLOGarithmic, PHASe, IMAGinary, REAL, POLar, SMITh, SWR, GDELay

- MLINear
- MLOGarithmic
- PHASe
- IMAGinary
- REAL
- POLar
- SMITh
- SWR
- GDELay

Examples

```
CALC:FORM MLIN
calculate2:format polar
```

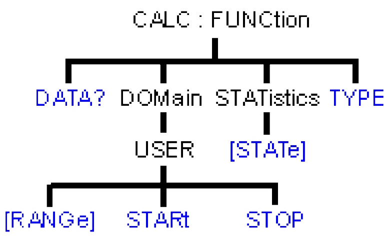
Query Syntax CALCulate<cnum>:FORMat?

Return Type Character

Overlapped? No

Default MLINear

Calc:Function Commands



- Click on a blue keyword to view the command details.
- See a List of all commands in this block.
- Learn about Trace Statistics

Note: CALCulate commands act on the selected measurement. You can select one measurement in each channel. Select the measurement for each channel using CALC:PAR:SEL.

CALCulate<cnum>:FUNCtion:DATA?

(Read-only) Returns the trace statistic data for the selected statistic type for the specified channel. Select the type of statistic with CALC:FUNC:TYPE. **Critical Note:**

Parameters	
<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
Return Type	Depends on FORM:DATA
Example	`CALCulate2:FUNCtion:DATA?`
Overlapped?	No
Default	Not applicable

CALCulate<cnum>:FUNCtion:DOMain:USER[:RANGe] <range>

(Read-Write) Sets the range used to calculate trace statistics. Each channel shares 10 domain ranges. The x-axis range is specified with the CALC:FUNC:DOM:USER:START and STOP commands. **Critical Note:**

Parameters

<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<range>	Range number. Choose from: **0** to **9** **0** is Full Span of the current x-axis range **1 to 9** are user-specified ranges
Examples	`CALC:FUNC:DOM:USER 4` `calculate2:function:domain:user:range 0`
Query Syntax	CALCulate<cnum>:FUNCtion:DOMain:USER[:RANGe]?
Return Type	Numeric
Overlapped?	No
Default	0 - Full Span

CALCulate<cnum>:FUNCtion:DOMain:USER:STARt <range>, <start>

(Read-Write) Sets the start of the specified user-domain range.
To apply this range, use CALC:FUNC:DOM:USER
To set the stop of the range, use CALC:FUNC:DOM:USER:STOP. **Critical Note:**

Note: This command does the same as CALC:MARK:FUNC:DOM:USER:STAR

Parameters

<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<range>	Range number that will receive the start value. Choose an integer between **1** and **9**
<start>	Start value of the specified range. Choose a real number between: the analyzer's **Minimum** and **Maximum** x-axis value.
Examples	`CALC:FUNC:DOM:USER:STAR 1,1e9` `calculate2:function:domain:user:start 2,2e9`
Query Syntax	CALCulate<cnum>:FUNCtion:DOMain:USER:STARt? <range>
Return Type	Numeric
Overlapped?	No
Default	The analyzer's **Minimum** x-axis value

CALCulate<cnum>:FUNCtion:DOMain:USER:STOP <range>, <stop>

(Read-Write) Sets the stop of the specified user-domain range.
To apply this range, use CALC:FUNC:DOM:USER
To set the start of the range, use CALC:FUNC:DOM:USER:START

Critical Note:

Note: This command does the same as CALC:MARK:FUNC:DOM:USER:STOP

Parameters

<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<range>	Range number that will receive the stop value. Choose an integer between **1** and **9**
<stop>	Stop value of the specified range. Choose a real number between: the analyzer's **Minimum** and **Maximum** x-axis value.
Examples	`CALC:FUNC:DOM:USER:STOP 4,5e9` `calculate2:function:domain:user:stop 3,8e9`
Query Syntax	CALCulate<cnum>:FUNCtion:DOMain:USER:STOP? <range>
Return Type	Numeric
Overlapped?	No
Default	The analyzer's **Maximum** x-axis value

CALCulate<cnum>:FUNCtion:STATistics[:STATe] <ON|OFF>

(Read-Write) Displays and hides the measurement (Trace) statistics (peak-to-peak, mean, standard deviation) on the screen.

The analyzer will display either measurement statistics or Filter Bandwidth statistics; not both. **Critical Note:**

Parameters

<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<ON|OFF>	ON - Displays trace statistics OFF - Hides trace statistics
Examples	`CALC:FUNC:STAT ON` `calculate2:function:statistics:state off`
Query Syntax	CALCulate<cnum>:FUNCtion:STATistics[:STATe]?
Return Type	Boolean (1 = ON, 0 = OFF)
Overlapped?	No

Default OFF (0)

CALCulate<cnum>:FUNCtion:TYPE <char>

(Read-Write) Sets statistic TYPE that you can then query using CALC:FUNCtion:DATA?. **Critical Note:**

Parameters

<cnum> Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.

<char> Choose from:
PTPeak - the difference between the max and min data points on the trace.
STDEV - standard deviation of all data points on the trace
MEAN - mean (average) of all data points on the trace

Examples

```
CALC:FUNC:TYPE PTP
calculate2:function:type stdev
```

Query Syntax CALCulate<cnum>:FUNCtion:TYPE?

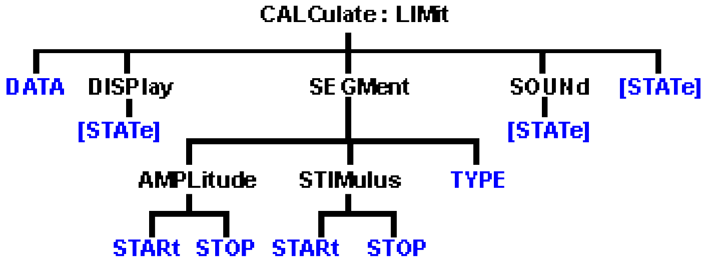
Return Type Character

Overlapped? No

Default PTPeak

Calc:Limit Commands

Controls the limit segments used for pass / fail testing.



- Click on a blue keyword to view the command details.
- See a List of all commands in this block.
- Learn about Limit Lines

Note: CALCulate commands act on the selected measurement. You can select one measurement in each channel. Select the measurement for each channel using CALC:PAR:SEL.

CALCulate<cnum>:LIMit:DATA <block>

(Read-Write) Sets data for limit segments. **Critical Note:**

Parameters

<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<block>	Data for all limit segments in REAL,64 format. The following is the data format for 1 segment: **Type,BegStim, EndStim, BegResp,EndResp**

Type	Type of limit segment. Choose from 0 - Off 1 - Max 2 - Min
BegStim	Start of X-axis value (freq, power, time)
EndStim	End of X-axis value
BegResp	Y-axis value that corresponds with Start of X-axis value
EndResp	Y-axis value that corresponds with End of X-axis value

Examples

The following writes three max limit segments for a bandpass filter.

```
"CALC:LIM:DATA 1,3e5,4e9,-60,0,1,4e9,7.5e9,0,0,1,7.5e9,9e9,0,-
30"
```

Query Syntax CALCulate<cnum>:LIMit:DATA?

Return Type Depends on FORM:DATA - All 100 predefined limit segments are returned.

Overlapped? No

Default 100 limit segments - all values set to 0

CALCulate<cnum>:LIMit:DISPlay[:STATe] <ON | OFF>

(Read-Write) Turns the display of limit segments ON or OFF (if the data trace is turned ON). **Critical Note:**

Parameters

<cnum> Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.

<ON | OFF> **ON** (or 1) - turns the display of limit segments ON.
OFF (or 0) - turns the display of limit segments OFF.

Examples

```
CALC:LIM:DISP:STAT ON
calculate2:limit:display:state off
```

Query Syntax CALCulate<cnum>:LIMit:DISPlay[:STATe]?

Return Type Boolean (1 = ON, 0 = OFF)

Overlapped? No

Default ON

CALCulate<cnum>:LIMit:SEGMent<snum>AMPLitude:STARt <num>

(Read-Write) Sets the start (beginning) of the Y-axis amplitude (response) value. **Critical Note:**

Parameters

<cnum> Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.

<snum> Segment number; if unspecified, value is set to 1.

<num> Choose any number between: **-500** and **500**

Display value is limited to the Maximum and Minimum displayed Y-axis values.

Examples

```
CALC:LIM:SEGM1:AMPL:STAR 10
calculate2:limit:segment2:amplitude:start 10
```

Query Syntax CALCulate<cnum>:LIMit:SEGMent<snum>AMPLitude:STARt?

Return Type Numeric

Overlapped? No

Default 0

CALCulate<cnum>:LIMit:SEGMent<snum>AMPLitude:STOP <num>

(Read-Write) Sets the stop (end) of the Y-axis amplitude (response) value. **Critical Note:**

Parameters

<cnum> Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.

<snum> Segment number; if unspecified, value is set to 1.

<num> Choose any number between: **-500** and **500**

Display value is limited to the Maximum and Minimum displayed Y-axis values.

Examples

```
CALC:LIM:SEGM1:AMPL:STOP 10
calculate2:limit:segment2:amplitude:stop 10
```

Query Syntax CALCulate<cnum>:LIMit:SEGMent<snum>AMPLitude:STOP?

Return Type Numeric

Overlapped? No

Default 0

CALCulate<cnum>:LIMit:SEGMent<snum>STIMulus:STARt <num>

(Read-Write) Sets the start (beginning) of the X-axis stimulus value. **Critical Note:**

Parameters

<cnum> Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.

<snum> Segment number; if unspecified, value is set to 1.

<num> Choose any number within the X-axis span of the analyzer.

Examples

```
CALC:LIM:SEGM1:STIM:STAR 10
calculate2:limit:segment2:stimulus:start 10
```

Query Syntax CALCulate<cnum>:LIMit:SEGMent<snum>STIMulus:STARt?

Return Type Numeric

Overlapped? No

Default 0

CALCulate<cnum>:LIMit:SEGMent<snum>STIMulus:STOP <num>

(Read-Write) Sets the stop (end) of the X-axis stimulus value. **Critical Note:**

Parameters

<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<snum>	Segment number; if unspecified, value is set to 1.
<num>	Choose any number within the X-axis span of the analyzer.
Examples	`CALC:LIM:SEGM1:AMPL:STOP 10` `calculate2:limit:segment2:stimulus:stop 10`
Query Syntax	CALCulate<cnum>:LIMit:SEGMent<snum>STIMulus:STOP?
Return Type	Numeric
Overlapped?	No
Default	0

CALCulate<cnum>:LIMit:SEGMent<snum>:TYPE <char>

(Read-Write) Sets the type of limit segment. **Critical Note:**

Parameters

<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<snum>	Segment number. Choose any number between: **1** and **100** If unspecified, value is set to 1.
<char>	Choose from: **LMAX** - a MAX limit segment. Any response data exceeding the MAX value will fail. **LMIN** - a MIN limit segment. Any response data below the MIN value will fail. **OFF** - the limit segment (display and testing) is turned OFF.
Examples	`CALC:LIM:SEGM:TYPE LMIN` `calculate2:limit:segment3:type lmax`
Query Syntax	CALCulate<cnum>:LIMit:SEGMent<snum>:TYPE?
Return Type	Character
Overlapped?	No
Default	OFF

CALCulate<cnum>:LIMit:SOUNd[:STATe] <ON | OFF>

(Read-Write) Turns limit testing fail sound ON or OFF. **Critical Note:**

Parameters	
<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<ON | OFF>	**ON** (or 1) - turns sound ON. **OFF** (or 0) - turns sound OFF.
Examples	`CALC:LIM:SOUN ON` `calculate2:limit:sound:state off`
Query Syntax	CALCulate<cnum>:LIMit:SOUNd[:STATe]?
Return Type	Boolean (1 = ON, 0 = OFF)
Overlapped?	No
Default	OFF

CALCulate<cnum>:LIMit:STATe <ON | OFF>

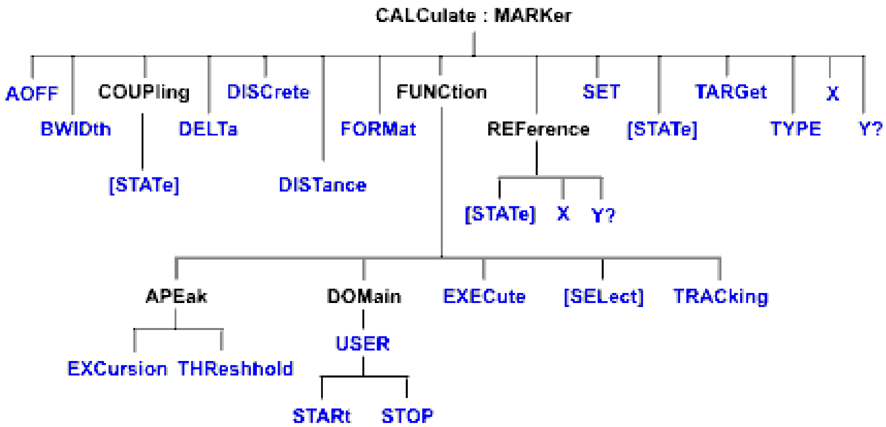
(Read-Write) Turns limit segment **testing** ON or OFF.

- Use CALC:LIM:DISP to turn ON and OFF the **display** of limit segments.
- If using Global Pass/Fail status, trigger the PNA AFTER turning Limit testing ON.
- **Critical Note:**

Parameters	
<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<ON | OFF>	**ON** (or 1) - turns limit testing ON. **OFF** (or 0) - turns limit testing OFF.
Examples	`CALC:LIM:STAT ON` `calculate2:limit:state off`
Query Syntax	CALCulate<cnum>:LIMit:STATe?
Return Type	Boolean (1 = ON, 0 = OFF)
Overlapped?	No
Default	OFF

Calc:Marker Commands

Controls the marker settings used to remotely output specific data to the computer.



- Click on a blue keyword to view the command details.
- See a List of all commands in this block.
- See commands for controlling the marker readout number and size
- Learn about Markers

Note: CALCulate commands act on the selected measurement. You can select one measurement in each channel. Select the measurement for each channel using CALC:PAR:SEL.

Note: The Reference Marker is Marker Number 10

CALCulate<cnum>:MARKer:AOFF

(Write-only) Turns all markers off for selected measurement. **Critical Note:**

Parameters

<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
Examples	`CALC:MARK:AOFF` `calculate2:marker:aoff`
Query Syntax	Not applicable
Overlapped?	No
Default	Not applicable

CALCulate<cnum>:MARKer:BWIDth <num>

(Read-Write) Turns on and sets markers 1 through 4 to calculate filter bandwidth. The <num> parameter sets the value below the maximum bandwidth peak that establishes the bandwidth of a filter. For example, if you want to determine the filter bandwidth 3 db below the bandpass peak value, set <num> to -3.

This feature activates markers 1 through 4. To turn off these markers, either turn them off individually or turn them All Off.

The analyzer screen will show either Bandwidth statistics OR Trace statistics; not both.

To search a User Range with the bandwidth search, first activate marker 1 and set the desired User Range. Then send the CALC:MARK:BWID command. The user range used with bandwidth search only applies to marker 1 searching for the max value. The other markers may fall outside the user range.
Critical Note:

Parameters	
<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<num>	Target value below filter peak. Choose any number between: **-500** and **500**
Examples	`CALC:MARK:BWID -3` `calculate2:marker:bwidth -2.513`
Query Syntax	CALCulate<cnum>:MARKer:BWIDth? Returns the results of bandwith search:
Return Type	Numeric - Four Character values separated by commas: bandwidth, center Frequency, Q, loss.
Overlapped?	No
Default	-3

CALCulate<cnum>:MARKer<mkr>:COUPling[:STATe]<ON|OFF>

(Read-Write) Sets and Reads the state of Coupled Markers (ON and OFF) **Critical Note:**

Parameters	
<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<mkr>	Any existing marker number from 1 to 10; if unspecified, value is set to 1.
<ON|OFF>	**False (0**) - Turns Coupled Markers OFF **True (1)** - Turns Coupled Markers ON
Examples	`CALC:MARK:COUP ON` `calculate2:marker8:coupling off`
Query Syntax	CALCulate<cnum>:MARKer<mkr>:COUPling:[STATe]?

Return Type Boolean (1 = ON, 0 = OFF)

Overlapped? No

Default OFF

CALCulate<cnum>:MARKer<mkr>:DELTa <ON|OFF>

(Read-Write) Specifies whether marker is relative to the Reference marker or absolute.

Note: The reference marker must already be turned ON with CALC:MARK:REF:STATE.

Critical Note:

Parameters

<cnum> Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.

<mkr> Any existing marker number from 1 to 10; if unspecified, value is set to 1.

<ON|OFF> **ON** (or 1) - Specified marker is a Delta marker
OFF (or 0) - Specified marker is an ABSOLUTE marker

Examples

```
CALC:MARK:DELT ON
calculate2:marker8:delta off
```

Query Syntax CALCulate<cnum>:MARKer<mkr>:DELTa?

Return Type Boolean (1 = ON, 0 = OFF)

Overlapped? No

Default OFF

CALCulate<cnum>:MARKer<mkr>:DISCrete <ON|OFF>

(Read-Write) Makes the specified marker display either a calculated value between data points (interpolated data) or the actual data points (discrete data). **Critical Note:**

Parameters

<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<mkr>	Any existing marker number from 1 to 10; if unspecified, value is set to 1.
<ON|OFF>	**ON** (or 1) - Specified marker displays the actual data points **OFF** (or 0) - Specified marker displays calculated data between the actual data points.
Examples	**CALC:MARK:DISC ON** **calculate2:marker8:discrete off**
Query Syntax	CALCulate<cnum>:MARKer<mkr>:DISCrete?
Return Type	Boolean (1 = ON, 0 = OFF)
Overlapped?	No
Default	OFF

CALCulate<cnum>:MARKer<mkr>:DISTance <num>

(Read-Write) Set or query marker distance on a time domain trace.

The Write command moves the marker to the specified distance value. Once moved, you can read the Y axis value or read the X-axis time value. (Distance is calculated from the X-axis time value.)

The Read command reads the distance of the marker.

If the marker is set as delta, the WRITE and READ data is relative to the reference marker.

Critical Note:

Parameters

<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<mkr>	Any existing marker number from 1 to 10; if unspecified, value is set to 1.
<num>	Marker distance in the unit of measure specified with CALC:TRAN:TIME:MARK:UNIT
Examples	**CALC:MARK:DIST .1** **calculate2:marker8:distance 5**
Query Syntax	CALCulate<cnum>:MARKer<mkr>:DISTance?
Return Type	Numeric
Overlapped?	No

Default Not Applicable

CALCulate<cnum>:MARKer<mkr>:FORMat <char>

(Read-Write) Sets the format of the data that will be returned in a marker data query CALC:MARK:Y? and the displayed value of the marker readout. The selection does not have to be the same as the measurement's display format. **Critical Note:**

Parameters

<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<mkr>	Any marker number from 1 to 10; if unspecified, value is set to 1
<char>	Choose from: **DEFault -** The format of the selected measurement **MLINear** - Linear magnitude **MLOGarithmic -** Logarithmic magnitude **IMPedance** - (R+jX) **ADMittance** - (G+jB) **PHASe** - Phase **IMAGinary** - Imaginary part (Im) **REAL** - Real part (Re)l **POLar** - (Re, Im) **GDELay** - Group Delay **LINPhase -** Linear Magnitude and Phase **LOGPhase -** Log Magnitude and Phase
Examples	`CALC:MARK:FORMat MLIN` `calculate2:marker8:format Character`
Query Syntax	CALCulate<cnum>:MARKer<mkr>:FORMat?
Return Type	Character
Overlapped?	No
Default	DEFault

CALCulate<cnum>:MARKer<mkr>:FUNCtion:APEak:EXCursion <num>

(Read-Write) Sets amplitude peak excursion for the specified marker. The Excursion value determines what is considered a "peak". This command applies to marker peak searches (Next peak, Peak Right, Peak Left). **Critical Note:**

Parameters

<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<mkr>	Any existing marker number from 1 to 10; if unspecified, value is set to 1.
<num>	Excursion value. Choose any number between **-500** and **500.**
	Note: This command will accept **MIN** or **MAX** instead of a numeric parameter. See SCPI Syntax for more information.
Examples	**CALC:MARK:FUNC:APE:EXC 10** **calculate2:marker8:function:apeak:excursion maximum**
Query Syntax	CALCulate<cnum>:MARKer<mkr>:FUNCtion:APEak:EXCursion?
Return Type	Numeric
Overlapped?	No
Default	3

CALCulate<cnum>:MARKer<mkr>:FUNCtion:APEak:THReshold <num>

(Read-Write) Sets peak threshold for the specified marker. If a peak (using the criteria set with :EXCursion) is below this reference value, it will not be considered when searching for peaks. This command applies to marker peak searches (Next peak, Peak Right, Peak Left). **Critical Note:**

Parameters

<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<mkr>	Any marker number from 1 to 10; if unspecified, value is set to 1
<num>	Threshold value.Choose any number between **-500** and **500.**
	Note: This command will accept **MIN** or **MAX** instead of a numeric parameter. See SCPI Syntax for more information.
Examples	**CALC:MARK:FUNC:APE:THR -40** **calculate2:marker8:function:apeak:threshold -55**
Query Syntax	CALCulate<cnum>:MARKer<mkr>:FUNCtion:APEak:THReshold?
Return Type	Numeric
Overlapped?	No

Default -100

CALCulate<cnum>:MARKer<mkr>:FUNCtion:DOMain:USER <range>

(Read-Write) Assigns the specified marker to a range number. The x-axis travel of the marker is constrained to the range's span. The span is specified with the CALC:MARK:FUNC:DOM:USER:START and STOP commands, unless range 0 is specified which is the full span of the analyzer.

Each channel shares 10 domain ranges. (Trace statistics use the same ranges.) More than one marker can use a domain range. **Critical Note:**

Parameters

<cnum> Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.

<mkr> Any marker number from 1 to 10; if unspecified, value is set to 1

<span> User span. Choose any Integer from **0 to 9**.

0 is Full Span of the analyzer
1 to 9 are available for user-defined x-axis span

Examples

```
CALC:MARK:FUNC:DOM:USER 1
calculate2:marker8:function:domain:user 1
```

Query Syntax CALCulate<cnum>:MARKer<mkr>:FUNCtion:DOMain:USER?
Returns the user span number that the specified marker is assigned to.

Return Type Numeric

Overlapped? No

Default 0 - Full Span

CALCulate<cnum>:MARKer<mkr>:FUNCtion:DOMain:USER:STARt <start>

(Read-Write) Sets the start of the span that the specified marker's x-axis span will be constrained to.
Use CALC:MARK:FUNC:DOM:USER<range> to set range number
Use CALC:MARK:FUNC:DOM:USER:STOP to set the stop value.

Note: If the marker is assigned to range 0 (full span), the USER:STARt and STOP commands generate an error. You cannot set the STARt and STOP values for "Full Span".

Note: This command does the same as CALC:FUNC:DOM:USER:STAR

Critical Note:

Parameters

<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<mkr>	Any marker number from 1 to 10; if unspecified, value is set to 1
<start>	The analyzer's **Minimum** x-axis value
Examples	**CALC:MARK:FUNC:DOM:USER:START 500E6** **calculate2:marker8:function:domain:user:start 1e12**
Query Syntax	CALCulate<cnum>:MARKer<mkr>:FUNCtion:DOMain:USER:STARt?
Return Type	Numeric
Overlapped?	No
Default	The analyzer's **Minimum** x-axis value

CALCulate<cnum>:MARKer<mkr>:FUNCtion:DOMain:USER:STOP <stop>

(Read-Write) Sets the stop of the span that the marker's x-axis travel will be constrained to. Use CALC:MARK:FUNC:DOM:USER<range> to set range number
Use CALC:MARK:FUNC:DOM:USER:STARt to set the stop value.

Note: If the marker is assigned to range 0 (full span), the USER:STARt and STOP commands generate an error. You cannot set the STARt and STOP values for "Full Span".

Note: This command does the same as CALC:FUNC:DOM:USER:STOP

Critical Note:

Parameters

<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<mkr>	Any marker number from 1 to 10; if unspecified, value is set to 1.
<stop>	Stop value of x-axis span; Choose any number between the analyzer's **MINimum** and **MAXimum** x-axis value.
Examples	**CALC:MARK:FUNC:DOM:USER:STOP 500e6** **calculate2:marker8:function:domain1:user:stop 1e12**

Query Syntax CALCulate<cnum>:MARKer<mkr>:FUNCtion:DOMain:USER:STOP?

Return Type Numeric

Overlapped? No

Default The analyzer's **MAXimum** x-axis value.

CALCulate<cnum>:MARKer<mkr>:FUNCtion:EXECute [<func>]

(Write-only) Immediately executes (performs) the specified search function. If no function is specified, executes the selected function. Select the function with CALC:MARK:FUNCtion:SEL. **Critical Note:**

Parameters

<cnum> Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.

<mkr> Any marker number from 1 to 10; if unspecified, value is set to 1.

<func> Optional argument. The function that is to be performed. Choose from:

MAXimum - finds the highest value
MINimum - finds the lowest value
RPEak - finds the next valid peak to the right
LPEak - finds the next valid peak to the left
NPEak - finds the **next highest** value among the valid peaks
TARGet - finds the target value to the right, wraps around to the left
LTARget - finds the next target value to the left of the marker
RTARget - finds the next target value to the right of the marker

Examples

```
CALC:MARK:FUNC:EXEC
calculate2:marker2:function:execute maximum
```

Query Syntax Not applicable

Overlapped? No

Default Not applicable

CALCulate<cnum>:MARKer<mkr>:FUNCtion[:SELect] <char>

(Read-Write) Sets the search function that the specified marker will perform when executed. To execute (or perform) the function, use:
CALC:MARK:FUNC:EXEC **or**
CALC:MARK:FUNC:TRAC ON to automatically execute the search every sweep. **Critical Note:**

Parameters

<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<mkr>	Any marker number from 1 to 10; if unspecified, value is set to 1.
<char>	Marker function. Choose from: **MAXimum -** finds the highest value **MINimum -** finds the lowest value **RPEak** - finds the next valid peak to the right **LPEak** - finds the next valid peak to the left **NPEak** - finds the **next highest** value among the valid peaks **TARGet -** finds the target value to the right; wraps around to the left **LTARget -** finds the next target value to the left of the marker **RTARget** - finds the next target value to the right of the marker
Examples	`CALC:MARK:FUNC MAX` `calculate2:marker8:function:select ltarget`
Query Syntax	CALCulate<cnum>:MARKer<mkr>:FUNCtion[:SELect]?
Return Type	Character
Overlapped?	No
Default	MAX

CALCulate<cnum>:MARKer<mkr>:TARGet <num>

(Read-Write) Sets the target value for the specified marker when doing Target Searches (CALC:MARK:FUNC:SEL <TARGet | RTARget | LTARget> **Critical Note:**

Parameters

<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<mkr>	Any marker number from 1 to 10; if unspecified, value is set to 1.
<num>	Target value to search for; Units are NOT allowed.
Examples	**CALC:MARK:TARG 2.5** **calculate2:marker8:target -10.3**
Query Syntax	CALCulate<cnum>:MARKer<mkr>:TARGet?
Return Type	Numeric
Overlapped?	No
Default	0

CALCulate<cnum>:MARKer<mkr>:FUNCtion:TRACking <ON | OFF>

(Read-Write) Sets the tracking capability for the specified marker. The tracking function finds the selected search function every sweep. In effect, turning Tracking ON is the same as doing a CALC:MARK:FUNC:EXECute command every sweep. **Critical Note:**

Parameters

<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<mkr>	Any marker number from 1 to 10; if unspecified, value is set to 1.
<ON | OFF>	**ON** (or 1) - The specified marker will "Track" (find) the selected function every sweep. **OFF** (or 0) - The specified marker will find the selected function **only** when the CALC:MARK:FUNC:EXECute command is sent.
Examples	**CALC:MARK:FUNC:TRAC ON** **calculate2:marker8:function:tracking off**
Query Syntax	CALCulate<cnum>:MARKer<mkr>:FUNCtion:TRACking?
Return Type	Boolean (1 = ON, 0 = OFF)
Overlapped?	No
Default	OFF

CALCulate<cnum>:MARKer:REFerence[:STATe] <ON | OFF>

(Read-Write) Turns the reference marker (marker 10) ON or OFF. When turned OFF, existing Delta markers revert to absolute markers. **Critical Note:**

Parameters

<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<ON | OFF>	**ON** (or 1) - turns reference marker ON **OFF** (or 0) - turns reference marker ON
Examples	**`CALC:MARK:REF ON`** **`calculate2:marker:reference:state OFF`**
Query Syntax	CALCulate<cnum>:MARKer:REFerence[:STATe]?
Return Type	Boolean (1 = ON, 0 = OFF)
Overlapped?	No
Default	OFF

CALCulate<cnum>:MARKer:REFerence:X <num>

(Read-Write) Sets and returns the absolute x-axis value of the reference marker (marker 10). **Critical Note:**

<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<num>	X-axis value. Choose any number within the operating domain of the reference marker.
Examples	**`CALC:MARK:REF:X 1e9`** **`calculate2:marker:reference:x 1e6`**
Query Syntax	CALCulate<cnum>:MARKer:REFerence:X?
Return Type	Numeric
Overlapped?	No
Default	If the first Marker, turns ON in the middle of the X-axis span. If not, turns ON at the position of the active marker.

CALCulate<cnum>:MARKer:REFerence:Y?

(Read-only) Returns the absolute Y-axis value of the reference marker. **Critical Note:**

Parameters

<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
Examples	`CALC:MARK:REF:Y?` `calculate2:marker:reference:y?`
Return Type	Character
Overlapped?	No
Default	Not applicable

CALCulate<cnum>:MARKer<mkr>:TYPE <char>

(Read-Write) Sets the type of the specified marker. **Critical Note:**

Parameters

<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<mkr>	Any marker number from 1 to 10; if unspecified, value is set to 1
<char>	Choose from: **NORMal** - a marker that stays on the assigned X-axis position unless moved or searching. **FIXed** - a marker that will not leave the assigned X or current Y-axis position.
Examples	`CALC:MARK:TYPE NORM` `calculate2:marker2:type fixed`
Query Syntax	CALCulate<cnum>:MARKer<mkr>:TYPE?
Return Type	Character
Overlapped?	No
Default	NORMal

CALCulate<cnum>:MARKer<mkr>:SET <char>

(Read-Write) Sets the selected instrument setting to assume the value of the specified marker.

Marker Functions CENT, SPAN, STARt, and STOP do not work with channels that are in CW or Segment Sweep mode.

Critical Note:

Parameters

<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<mkr>	Any marker number from 1 to 10; if unspecified, value is set to 1
<char>	Choose from: **CENTer -** changes center frequency to the value of the marker **SPAN -** changes the sweep span to the span that is defined by the delta marker and the marker that it references. Unavailable if there is no delta marker. **STARt -** changes the start frequency to the value of the marker **STOP -** changes the stop frequency to the value of the marker **RLEVel** - changes the reference level to the value of the marker **DELay** - changes the line length at the receiver input to the phase slope at the active marker stimulus position.
Examples	`CALC:MARK:SET CENT` `calculate2:marker8:set span`
Query Syntax	CALCulate<cnum>:MARKer<mkr>:SET?
Return Type	Character
Overlapped?	No
Default	Not applicable

CALCulate<cnum>:MARKer<mkr>[:STATe] <ON|OFF>

(Read-Write) Turns the specified marker ON or OFF. **Marker 10 is the Reference Marker**. To turn all markers off, use CALC:MARK:AOFF. **Critical Note:**

Parameters

<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<mkr>	Any marker number from 1 to 10; if unspecified, value is set to 1.
<ON|OFF>	**ON** (or 1) - turns marker ON. **OFF** (or 0) - turns marker OFF.

Examples

```
CALC:MARK ON
calculate2:marker8 on
```

Query Syntax CALCulate<cnum>:MARKer<mkr>:STATe?

Return Type Boolean (1 = ON, 0 = OFF)

Overlapped? No

Default Off

CALCulate<cnum>:MARKer<mkr>:X <num>

(Read-Write) Sets the marker's X-axis value (frequency, power, or time). If the marker is set as delta, the SET and QUERY data is relative to the reference marker. **Critical Note:**

Parameters

<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<mkr>	Any marker number from 1 to 10; if unspecified, value is set to 1.
<num>	Any X-axis position within the measurement span of the marker. **Note**: This command will accept **MIN** or **MAX** instead of a numeric parameter. See SCPI Syntax for more information.

Examples

```
CALC:MARK:X 100Mhz
calculate2:marker8:x maximum
```

Query Syntax CALCulate<cnum>:MARKer<mkr>:X?

Return Type Numeric

Overlapped? No

Default First Marker turns ON in the middle of the X-axis span. Subsequent markers turn ON at the position of the active marker.

CALCulate<cnum>:MARKer<mkr>:Y?

(Read-only) Reads the marker's Y-axis value. The format of the value depends on the current CALC:MARKER:FORMAT setting. If the marker is set as delta, the data is relative to the reference marker. The query always returns two numbers:

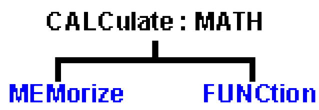
- Smith and Polar formats - (Real, Imaginary)
- LINPhase and LOGPhase - (Real, Imaginary)
- All other formats - (Value,0)

Critical Note:

Parameters	
<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<mkr>	Any marker number from 1 to 10; if unspecified, value is set to 1.
Examples	`CALC:MARK:Y?` `calculate2:marker3:y?`
Query Syntax	CALCulate<cnum>:MARKer<mkr>:Y?
Return Type	Numeric
Overlapped?	No
Default	Not applicable

Calc:Math Command

Controls math operations on the currently selected measurement and memory.



- Click on a blue keyword to view the command details.
- See a List of all commands in this block.
- Learn about Math Operations

Note: CALCulate commands act on the selected measurement. You can select one measurement in each channel. Select the measurement for each channel using CALC:PAR:SEL.

CALCulate<cnum>:MATH:FUNCtion <char>

(Read-Write) Sets math operations on the currently selected measurement and the trace stored in memory. (There MUST be a trace stored in Memory. See CALC:MATH MEM) **Critical Note:**

Parameters

<cnum> Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.

<char> The math operation to be applied. Choose from the following:

NORMal	Trace data only
ADD	Data + Memory
SUBTract	Data - Memory
MULTiply	Data * Memory
DIVide	Data / Memory

Examples

```
CALC:MATH:FUNC NORM
calculate2:math:function subtract
```

Query Syntax CALCulate<cnum>:MATH:FUNCtion?

Return Type Character

Overlapped? No

Default NORMal

CALCulate<cnum>:MATH:MEMorize

(Write-only) Puts the currently selected measurement trace into memory. (Data-> Memory) **Critical Note:**

Parameters	
<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
Examples	**`CALC:MATH:MEM`** **`calculate2:math:memorize`**
Query Syntax	Not applicable
Overlapped?	No
Default	Not applicable

Calc:Mixer Commands

CALCulate<ch>:MIXer:INPut:POWer:USENominal <bool>

(Read-Write) Toggles the Nominal Incident Power setting ON and OFF. This setting is ONLY to be used with SMC measurements, not VMC. Learn more about Nominal Incident Power.

Parameters

<ch>	Any existing channel number. If unspecified, value is set to 1.
<bool>	Nominal Incident Power State. Choose from: 1 -(ON) Turn nominal incident power ON 0 -(OFF) Turn nominal incident power OFF
Examples	**CALC:MIX:INP:POW:USEN 1** **CALCulate2:MIXer:INPut:POWer:USENominal 0**
Query Syntax	CALCulate<ch>:MIXer:INPut:POWer:USENominal?
Return Type	Boolean
Overlapped?	No
Default	0 - OFF

CALCulate<ch>:MIXer:XAXis <char>

(Read-Write) Sets or returns the swept parameter to display on the X-axis for the selected FCA measurement. Learn more about X-axis display.

Note: CALCulate commands act on the selected measurement. You can select one measurement in each channel. Select the measurement for each channel using CALC:PAR:SEL.

Parameters

<ch>	Any existing channel number. If unspecified, value is set to 1.
<char>	Parameter to display on the X-axis. Choose from: **INPUT** - Input frequency span **OUTPUT** - Output frequency span **LO_1** - First LO frequency span **LO_2** - Second LO frequency span
Examples	**CALC:MIX:XAX INPUT** **calc2:mixer:xaxis output** **See an example that creates, selects, and calibrates an SMC and VMC measurement using SCPI.**

Query Syntax CALCulate<ch>:MIXer:XAXis?

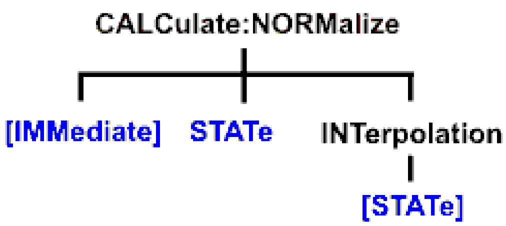
Return Type Character

Overlapped? No

Default OUTPUT

Calc:Normalize Commands

Specifies the normalization features used for a receiver power calibration.



- Click on a blue keyword to view the command details.
- See a List of all commands in this block.
- Learn about Receiver Cal

Save and recall your receiver power calibration (which use .CST file commands):

- SENS:CORR:CSET:SAVE
- SENS:CORR:CSET[:SEL]

Or use these two commands and specify either .STA or .CST file extensions:

- MMEM:LOAD
- MMEM:STOR

Note: CALCulate commands act on the selected measurement. You can select one measurement in each channel. Select the measurement for each channel using CALC:PAR:SEL.

CALCulate<cnum>:NORMalize[:IMMediate]

(Write only) Stores the selected measurement's data to that measurement's "divisor" buffer for use by the Normalization data processing algorithm. This command is not compatible with ratioed measurements such as S-parameters. It is intended for receiver power calibration when the selected measurement is of an unratioed power type. Critical Note:

Parameters	
<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
Examples	`CALC:NORM` `calculate1:normalize:immediate`
Query Syntax	Not Applicable
Overlapped?	No

Default Not Applicable

CALCulate<cnum>:NORMalize:STATe <ON | OFF>

(Read-Write) Specifies whether or not normalization is applied to the measurement. Normalization is enabled only for measurements of unratioed power where it serves as a receiver power calibration. Critical Note:

Parameters

<cnum> Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.

<ON | OFF> **ON (or 1)** - normalization is applied to the measurement.

OFF (or 0) – normalization is NOT applied to the measurement.

Examples

```
CALC:NORM:STAT ON
calculate2:normalize:state off
```

Query Syntax CALCulate<cnum>:NORMalize:STATe?

Return Type Boolean (1 = ON, 0 = OFF)

Overlapped? No

Default OFF

CALCulate<cnum>:NORMalize:INTerpolate[:STATe] <ON | OFF>

(Read-Write) Turns normalization interpolation ON or OFF. Normalization is enabled only for measurements of unratioed power, where it serves as a receiver power calibration. Critical Note:

Parameters

<cnum> Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.

<ON | OFF> **ON (or 1**) – turns interpolation ON.

OFF (or 0) – turns interpolation OFF.

Examples

```
CALC:NORM:INT ON
calculate2:normalize:interpolate:state off
```

Query Syntax CALCulate<cnum>:NORMalize:INTerpolate[:STATe]?

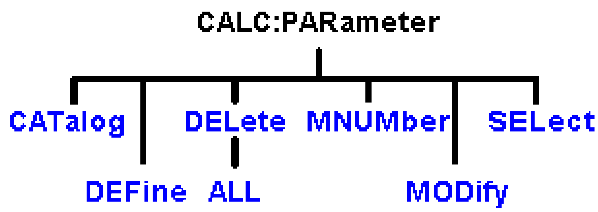
Return Type Boolean (1 = ON, 0 = OFF)

Overlapped? No

Default ON

Calc:Parameter Commands

Lists, creates, selects and deletes measurements.



- Click on a blue keyword to view the command details.
- See a List of all commands in this block.
- Learn about Measurement Parameters

Note: CALCulate commands act on the selected measurement. You can select one measurement in each channel. Select the measurement for each channel using CALC:PAR:SEL.

CALCulate<cnum>:PARameter:CATalog?

(Read-only) Returns the names and parameters of existing measurements for the specified channel. **Critical Note:**

Parameters	
<cnum>	Channel number of the measurements to be listed. If unspecified, <cnum> is set to 1.
Examples	`CALC:PAR:CAT?` `calculate2:parameter:catalog?`
Return Type	String - "<measurement name>,<parameter>,[<measurement name>,<parameter>...]"
Overlapped?	No
Default	"CH1_S11_1,S11"

CALCulate<cnum>:PARameter:DEFine <Mname>,<param>[,port]

(Write-only) Creates a measurement but does NOT display it.

There is no limit to the number of measurements that can be created. However, there is a limit to the number of measurements that can be displayed. See Traces, Channels, and Windows on the PNA.

- Use DISP:WIND:STATe to create a window if it doesn't already exist.
- Use DISP:WIND<wnum>:TRAC<tnum>:FEED <Mname> to display the measurement.

You must select the measurement (CALC<cnum>:PAR:SEL <mname>) before making additional settings.
Critical Note:

Parameters

<cnum>	Channel number of the new measurement. Choose any number between: **1** and **4** If unspecified, value is set to 1.
<Mname>	Name of the measurement. Any non-empty, unique string, enclosed in quotes.
<param>	Parameter

Choose from the following for S-Parameter measurements

S11 | S22 | S12 | S21

For 3-port analyzers only:

S33 | S13 | S31 | S23 | S32

Note: For all non S-Parameter measurements, specify the source port with the optional <port> argument:

Choose from the following for non-ratioed measurements:
A | B | C | R1 | R2

Choose from the following for ratioed measurements:

<param>	Description
AB	A/B
AC	A/C - 3 port analyzers only
BA	B/A
BC	B/C - 3 port analyzers only
CA	C/A - 3 port analyzers only
CB	C/B - 3 port analyzers only
AR1	A/R1
BR1	B/R1

CR1	C/R1 - 3 port analyzers only
AR2	A/R2
BR2	B/R2
R1A	R1/A
R2A	R2/A
R1B	R1/B
R2B	R2/B
R1C	R1/C - 3 port analyzers only
R2R1	R2/R1
R1R2	R1/R2

[port] Optional argument;

For multi-port reflection S-parameter measurements: specifies the PNA port which will provide the load for the calibration. This argument is ignored if a transmission S-parameter is specified.

For all non S-parameter measurements: specifies the source port for the measurement.

Examples

```
CALC4:PAR:DEF 'ch4_S33',S33,2 'Defines an S33 measurement with a load on port2 of the analyzer.
calculate2:parameter:define 'ch1_a', a, 1
```

Query Syntax Not Applicable; see Calc:Par:Cat?

Overlapped? No

Default Not Applicable

CALCulate<cnum>:PARameter:DELete [:NAME]<Mname>

(Write-only) Deletes the specified measurement. **Critical Note:**

Parameters

<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<Mname>	String - Name of the measurement
Examples	`CALC:PAR:DEL 'TEST'` `calculate2:parameter:delete 'test'`
Query Syntax	Not Applicable
Overlapped?	No
Default	Not Applicable

CALCulate:PARameter:DELete:ALL

(Write-only) Deletes all measurements on the PNA. **Critical Note:**

Parameters

Examples	`CALC:PAR:DEL:ALL`
Query Syntax	Not Applicable
Overlapped?	No
Default	Not Applicable

CALCulate<cnum>:PARameter:MNUMber?

(Read-only) Returns the measurement number of the selected measurement. This is useful when needing to identify a measurement by number, such as with Status:Ques:Lim or Status:Oper:Aver commands. **Critical Note:**

Parameters

<cnum>	Channel number of the measurement. If unspecified, <cnum> is set to 1.
Examples	`CALC:PAR:MNUM?` `calculate2:parameter:mnumber?`
Return Type	Numeric
Overlapped?	No
Default	Not Applicable

CALCulate<cnum>:PARameter:MODify <param>

(Write-only) Modifies a standard measurement using the same arguments as CALC:PAR:DEF. To modify an FCA measurement, use CALC:CUST:MOD. **Critical Note:**

Parameters

<cnum>	Channel number of the measurement. The selected measurement on that channel will be changed. If unspecified, <cnum> is set to 1.
<param>	Measurement parameter to change to. Use the same <param> arguments as CALC:PAR:DEF.

Examples

```
SYST:PRESET
CALC:PAR:DEF "MyMeas", S11
CALC:PAR:SEL "MyMeas"
CALC:PAR:MOD AR1 'changes the selected S11 measurement to an
A/R1 measurement
```

Query Syntax	Not Applicable
Overlapped?	No
Default	Not Applicable

CALCulate<cnum>:PARameter:SELect <Mname>

(Read-Write) Sets the selected measurement. Most CALC: commands require that this command be sent before a setting change is made. One measurement on each channel can be selected at the same time. To obtain a list of currently named measurements, use CALC:PAR:CAT? **Critical Note:**

Parameters

<cnum>	Channel number of the measurement to be selected. If unspecified, <cnum> is set to 1.
<Mname>	String - Name of the measurement. (Do NOT include the parameter name.)

Examples

```
CALC:PAR:SEL 'TEST'
calculate2:parameter:select 'test'
```

Query Syntax	CALCulate:PARameter:SELect?
Return Type	String
Overlapped?	No
Default	No Selection

Calc:RData? Command

Note: CALCulate commands act on the selected measurement. You can select one measurement in each channel. Select the measurement for each channel using CALC:PAR:SEL.

CALCulate<cnum>:RDATA? <char>

(Read-only) Returns receiver data for the selected measurement. To query measurement data, see CALC:DATA?

Parameters

<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<char>	Choose from receivers: **A** **B** **R1** **R2** **REF -** returns either R1 or R2 data depending on the source port of the CALC-selected measurement.

Example	`GPIB.Write "INITiate:CONTinuous OFF"`
	`GPIB.Write "INITiate:IMMediate;*wai"`
	`GPIB.Write "CALCulate:RDATA? A"`

Return Type	Depends on FORM:DATA - Two numbers per data point
Overlapped?	No
Default	Not Applicable

Notes:

Generally when you query the analyzer for data, you expect that the number of data values returned will be consistent with the number of points in the sweep.

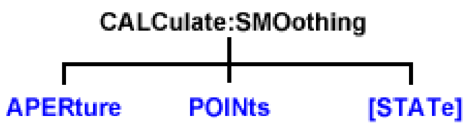
However, if you query **receiver** data while the instrument is sweeping, the returned values may contain zeros. For example, if your request for receiver data is handled on the 45th point of a 201 point sweep, the first 45 values will be valid data, and the remainder will contain complex zero.

This can be avoided by synchronizing this request with the end of a sweep or putting the channel in hold mode.

Learn about Unratioed Measurements

Calc:Smoothing Commands

Controls point-to-point smoothing. Smoothing is a noise reduction technique that averages adjacent data points in a measurement trace. Choose the amount of smoothing by specifying either the number of points or the aperture. Smoothing is not the same as CALC:AVERage which averages each data point over a number of sweeps.



- Click on a blue keyword to view the command details.
- See a List of all commands in this block.
- See an example using some of these commands.
- Learn about Smoothing

Note: CALCulate commands act on the selected measurement. You can select one measurement in each channel. Select the measurement for each channel using CALC:PAR:SEL.

CALCulate<cnum>:SMOothing:APERture <num>

(Read-Write) Sets the amount of smoothing as a percentage of the number of data points in the channel. **Critical Note:**

Parameters	
<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<num>	Percentage value. Choose any number between: **1** and **25**
Examples	**`CALC:SMO:APER 2`** **`calculate2:smoothing:aperture 20.7`**
Query Syntax	CALCulate<cnum>:SMOothing:APERture?
Return Type	Numeric
Overlapped?	No
Default	1.5

CALCulate<cnum>:SMOothing:POINts <num>

(Read-Write) Sets the number of adjacent data points to average. **Critical Note:**

Parameters	
<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<num>	Number of points from 1 point to maximum of 25% of data points in the channel. For example: if number of points in a data trace = 401, the maximum value for points = 100. The points value is always rounded to the closest odd number.
Examples	**CALC:SMO:POIN 50** **calculate2:smoothing:points 21**
Query Syntax	CALCulate<cnum>:SMOothing:POINts?
Return Type	Numeric
Overlapped?	No
Default	3

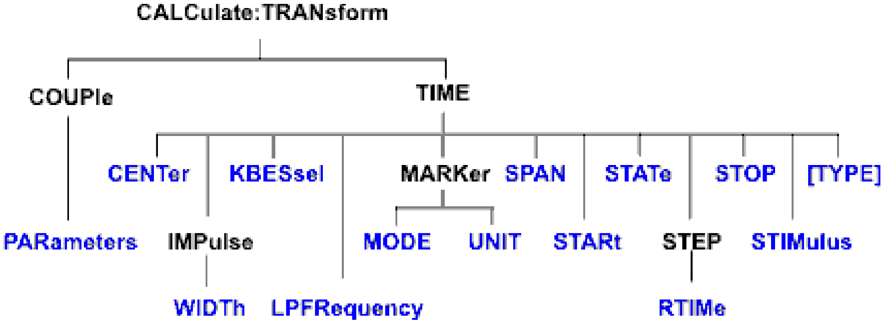
CALCulate<cnum>:SMOothing[:STATe] <ON | OFF>

(Read-Write) Turns data smoothing ON or OFF. **Critical Note:**

Parameters	
<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<ON | OFF>	**ON** (or 1) - turns smoothing ON. **OFF** (or 0) - turns smoothing OFF.
Examples	**CALC:SMO ON** **calculate2:smoothing:state off**
Query Syntax	CALCulate<cnum>:SMOothing[:STATe]?
Return Type	Boolean (1 = ON, 0 = OFF)
Overlapped?	No
Default	OFF

Calc:Transform Commands

Specifies the settings for time domain transform.



- Click on a blue keyword to view the command details.
- See a List of all commands in this block.
- Learn about Time Domain

Note: CALCulate commands act on the selected measurement. You can select one measurement in each channel. Select the measurement for each channel using CALC:PAR:SEL.

CALCulate<cnum>:TRANSform:COUPle:PARameters <num>

(Read-Write) Specifies the time domain transform parameters to be coupled. The settings for those parameters will be copied from the selected measurement to all other measurements on the channel.

- To turn coupling ON and OFF, use SENS:COUP:PAR
- To specify Gating parameters to couple, use CALC:FILT:COUP:PAR

Learn more about Time Domain Trace Coupling

Critical Note:

Parameters

<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<num>	(Numeric) Parameters to couple. To specify more than one parameter, add the numbers. **1** - Transform Stimulus (Start, Stop, Center, and Span TIME settings.) **2** - Transform State (ON / OFF) **4** - Transform Window (Kaiser Beta / Impulse Width) **8** - Transform Mode (Low Pass Impulse, Low Pass Step, Band Pass) **16** - Transform Distance Marker Units

Examples	`'To couple all parameters:`
	`CALC:TRAN:COUP:PAR 31`
	`'To couple Stimulus and Mode:`
	`calculate2:transform:couple:parameters 9`

Query Syntax	CALCulate<cnum>:TRANsform:COUPle:PARameters?
Return Type	Numeric
Overlapped?	No
Default	29 (All parameters except **2 -** Transform State)

CALCulate<cnum>:TRANsform:TIME:CENTer <num>

(Read-Write) Sets the center time for time domain measurements. **Critical Note:**

Parameters	
<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<num>	Center time in seconds; any number between: **±** (number of points-1) / frequency span **Note**: This command will accept **MIN** or **MAX** instead of a numeric parameter. See SCPI Syntax for more information.
Examples	`CALC:TRAN:TIME:CENT 1e-8` `calculate2:transform:time:center 15 ps`
Query Syntax	CALCulate<cnum>:TRANsform:TIME:CENTer?
Return Type	Numeric
Overlapped?	No
Default	0

CALCulate<cnum>:TRANsform:TIME:IMPulse:WIDTh <num>

(Read-Write) Sets the impulse width for the transform window. **Critical Note:**

Parameters

<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<num>	Impulse width in seconds; Choose any number between: **.6 / frequency span** and **1.39 / frequency span**
Examples	`CALC:TRAN:TIME:IMP:WIDTh 10` `calculate2:transform:time:impulse:width 13`
Query Syntax	CALCulate<cnum>:TRANsform:TIME:IMPulse:WIDTh?
Return Type	Numeric
Overlapped?	No
Default	.98 / Default Span

CALCulate<cnum>:TRANsform:TIME:KBESsel <num>

(Read-Write) Sets the parametric window for the Kaiser Bessel window. **Critical Note:**

Parameters

<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<num>	Window width for Kaiser Bessel in seconds; Choose any number between: **0.0** and **13.0**
Examples	`CALC:TRAN:TIME:KBES 10` `calculate2:transform:time:kbessel 13`
Query Syntax	CALCulate<cnum>:TRANsform:TIME:KBESsel?
Return Type	Numeric
Overlapped?	No
Default	6

CALCulate<cnum>:TRANsform:TIME:LPFREQuency

(Write-only) Sets the start frequencies in LowPass Mode. **Critical Note:**

Parameters

<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
Examples	`CALC:TRAN:TIME:LPFR` `calculate2:transform:time:lpfrequency`
Query Syntax	Not applicable
Overlapped?	No
Default	Not applicable

CALCulate<cnum>:TRANsform:TIME:MARKer:MODE <char>

(Read-Write) Specifies the measurement type in order to determine the correct marker distance.

- Select Auto for S-Parameter measurements.
- Select Reflection or Transmission for arbitrary ratio or unratioed measurements.

This setting affects the display of ALL markers for only the ACTIVE measurement.

Learn more about Distance Markers.

Critical Note:

Parameters

<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<char>	Choose from: **AUTO** If the active measurement is an S-Parameter, automatically chooses reflection or transmission. If non S-Parameter measurements, reflection is chosen. **REFLection** Displays the distance from the source to the receiver divided by two (to compensate for the return trip.) **TRANsmission** Displays the distance from the source to the receiver.
Examples	`CALC:TRAN:TIME:MARK:MODE REFL` `calculate2:transform:time:marker:mode auto`
Query Syntax	CALCulate<cnum>:TRANsform:TIME:MARKer:MODE?
Return Type	Character
Overlapped?	No
Default	Auto

CALCulate<cnum>:TRANsform:TIME:MARKer:UNIT <char>

(Read-Write) Specifies the unit of measure for the display of marker distance values. This settings affects the display of ALL markers for only the ACTIVE measurement (unless Distance Maker Units are coupled using CALC:TRAN:COUP:PAR.

Learn more about Distance Markers.

Critical Note:

Parameters

<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<char>	Choose from: **METRs** **FEET** **INCHes**
Examples	`CALC:TRAN:TIME:MARK:UNIT INCH` `calculate2:transform:time:marker:unit feet`
Query Syntax	CALCulate<cnum>:TRANsform:TIME:MARKer:UNIT?
Return Type	Character
Overlapped?	No
Default	METRs

CALCulate<cnum>:TRANsform:TIME:SPAN <num>

(Read-Write) Sets the span time for time domain measurements. **Critical Note:**

Parameters

<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<num>	Span time in seconds; any number between: **0** and 2* [(number of points-1) / frequency span] **Note**: This command will accept **MIN** or **MAX** instead of a numeric parameter. See SCPI Syntax for more information.
Examples	`CALC:TRAN:TIME:SPAN 1e-8` `calculate2:transform:time:span maximum`
Query Syntax	CALCulate<cnum>:TRANsform:TIME:SPAN?
Return Type	Numeric
Overlapped?	No

Default 20 ns

CALCulate<cnum>:TRANsform:TIME:STARt <num>

(Read-Write) Sets the start time for time domain measurements. **Critical Note:**

Parameters

<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<num>	Start time in seconds; any number between: ± (number of points-1) / frequency span **Note**: This command will accept **MIN** or **MAX** instead of a numeric parameter. See SCPI Syntax for more information.

Examples

```
CALC:TRAN:TIME:STAR 1e-8
calculate2:transform:time:start minimum
```

Query Syntax CALCulate<cnum>:TRANsform:TIME:STARt?

Return Type Numeric

Overlapped? No

Default -10 ns

CALCulate<cnum>:TRANsform:TIME:STATe <ON | OFF>

(Read-Write) Turns the time domain transform capability ON or OFF. **Critical Note:**

Note: Sweep type must be set to Linear Frequency in order to use Time Domain Transform.

Parameters

<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<ON|OFF>	**ON** (or 1) - turns time domain ON. **OFF** (or 0) - turns time domain OFF.

Examples

```
CALC:TRAN:TIME:STAT ON
calculate2:transform:time:state off
```

Query Syntax CALCulate<cnum>:TRANsform:TIME:STATe?

Return Type Boolean (1 = ON, 0 = OFF)

Overlapped? No

Default OFF

CALCulate<cnum>:TRANsform:TIME:STOP <num>

(Read-Write) Sets the stop time for time domain measurements. **Critical Note:**

Parameters

<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<num>	Stop time in seconds; any number between: **±** (number of points-1) / frequency span **Note**: This command will accept **MIN** or **MAX** instead of a numeric parameter. See SCPI Syntax for more information.
Examples	`CALC:TRAN:TIME:STOP 1e-8` `calculate2:transform:time:stop maximum`
Query Syntax	CALCulate<cnum>:TRANsform:TIME:STOP?
Return Type	Numeric
Overlapped?	No
Default	10 ns

CALCulate<cnum>:TRANsform:TIME:STEP:RTIMe <num>

(Read-Write) Sets the step rise time for the transform window. **Critical Note:**

Parameters

<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<num>	Rise time in seconds; Choose any number between: **.45 / frequency span** and **1.48 / frequency span**
Examples	`CALC:TRAN:TIME:STEP:RTIM 1e-8` `calculate2:transform:time:step:rtime 15 ps`
Query Syntax	CALCulate<cnum>:TRANsform:TIME:STEP:RTIMe?
Return Type	Numeric
Overlapped?	No
Default	.99 / Default Span

CALCulate<cnum>:TRANsform:TIME:STIMulus <char>

(Read-Write) Sets the type of simulated stimulus that will be incident on the DUT. **Critical Note:**

Parameters

<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<char>	Choose from: **STEP -** simulates a step DUT stimulus **IMPulse -** simulates a pulse DUT stimulus STEP can ONLY be used when CALC:TRAN:TIME:TYPE is set to LPASs (Lowpass). (STEP **cannot** be used with TYPE = BPASs.) **:STIM STEP** will set :TYPE to **LPASs** **:TYPE BPASs** will set :STIM to **IMPulse**

Examples

```
CALC:TRAN:TIME:STIM STEP
calculate2:transform:time:stimulus impulse
```

Query Syntax CALCulate<cnum>:TRANsform:TIME:STIMulus?

Return Type Character

Overlapped? No

Default IMPulse

CALCulate<cnum>:TRANsform:TIME[:TYPE] <char>

(Read-Write) Sets the type of time domain measurement. **Critical Note:**

Parameters

<cnum>	Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.
<char>	Type of measurement. Choose from: **LPASs** - Lowpass; Must also send CALC:TRAN:TIME:LPFRequency before calibrating. **BPASs** - Bandpass; BPASs can **only** be used when CALC:TRAN:TIME:STIM is set to IMPulse. (BPASs **cannot** be used with :STIM = STEP) :STIM **STEP** will set :TYPE to **LPASs** :TYPE **BPASs** will set :STIM to **IMPulse**

Examples

```
CALC:TRAN:TIME LPAS
calculate2:transform:time:type bpas
```

Query Syntax CALCulate<cnum>:TRANsform:TIME[:TYPE]?

Return Type Character

Overlapped? No

IEEE 488.2 Common Commands

***CLS -** Clear Status

***ESE -** Event Status Enable

***ESE?** - Event Status Enable Query

***ESR?** - Event Status Enable Register

***IDN?** - Identify

***OPC** - Operation complete command

***OPC?** - Operation complete query

***OPT?** - Identify Options Query

***RST** - Reset

***SRE -** Service Request Enable

***SRE?** - Service Request Enable Query

***STB?** - Status Byte Query

***TST? -** Result of Self-test Query

***WAI** - Wait

---

*CLS - Clear Status

Clears the instrument status byte by emptying the error queue and clearing all event registers. Also cancels any preceding *OPC command or query. See Status Commands and Reading the Analyzer's Status Registers.

---

*ESE - Event Status Enable

Sets bits in the standard event status enable register. See Status Commands and Reading the Analyzer's Status Registers.

---

*ESE? - Event Status Enable Query

Returns the results of the standard event enable register. The register is cleared after reading it. See Status Commands and Reading the Analyzer's Status Registers.

---

*ESR - Event Status Enable Register

Reads and clears event status enable register. See Status Commands and Reading the Analyzer's Status Registers.

---

*IDN? - Identify

Returns a string that uniquely identifies the analyzer. The string is of the form "Agilent Technologies",<model number>,<serial "number>,<software revision>" .

---

*OPC - Operation complete command

Generates the OPC message in the standard event status register when all pending overlapped operations have been completed (for example, a sweep, or a Default). See Understanding Command Synchronization.

---

*OPC? - Operation complete query

Returns an ASCII "+1" when all pending overlapped operations have been completed. See Understanding Command Synchronization

---

*OPT? - Identify Options Query

Returns a string identifying the analyzer option configuration.

---

*RST - Reset

Executes a device reset and cancels any pending *OPC command or query, exactly the same as a SYSTem:PRESet. The contents of the analyzer's non-volatile memory are not affected by this command.

---

*SRE - Service Request Enable

Before reading a status register, bits must be enabled. This command enables bits in the service request register. The current setting is saved in non-volatile memory. See Status Commands and Reading the Analyzer's Status Registers.

---

*SRE? - Service Request Enable Query

Reads the current state of the service request enable register. The register is cleared after reading it. The return value can be decoded using the table in Status Commands. See also Reading the Analyzer's Status Registers.

---

*STB? - Status Byte Query

Reads the value of the instrument status byte. The register is cleared only when the registers feeding it are cleared. See Status Commands and Reading the Analyzer's Status Registers.

---

*TST? - Result of Self-test Query

Returns the result of a query of the analyzer hardware status. An **0** indicates no failures found. Any other value indicates one or more of the following conditions exist. The value returned is the Weight (or sum of the Weights) of the existing conditions. For example:

- If **4** is returned from *TST?, an **Overpower** condition exists.
- If **6** is returned, both **Unleveled** and **Overpower** conditions exists.

Bit	Weight	Description	Bit is set to 1 when the following conditions exist:
0	1	Phase Unlock	The source has lost phaselock. This could be caused by a reference channel open or a hardware failure.
1	2	Unleveled	The source power is unleveled. This could be a source is set for more power than it can deliver at the tuned frequency. Or it could be caused by a hardware failure.
2	4	Not used	
3	8	EE Write Failed	An attempted write to the EEPROM has failed. This is possibly caused by a hardware failure.
4	16	YIG Cal Failed	The analyzer was unable to calibrate the YIG. Either the phaselock has been lost or there has been a hardware failure.

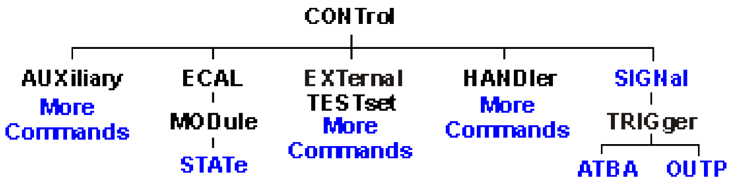
5	32	Ramp Cal Failed	The analyzer was unable to calibrate the analog ramp generator due to a possible hardware failure.
6	64	Not used	

*WAI - Wait

Prohibits the instrument from executing any new commands until all pending overlapped commands have been completed. See Understanding Command Synchronization

Control Commands

Specifies the settings to remotely control the rear panel connectors and ECAL Module state.



- Click on a **blue** keyword to view the command details.
- See a List of all SCPI commands.
- See a pinout and detailed description of the rear panel connectors:
    - Auxilliary IO connector
    - External Test Set IO connector
    - Material Handler IO connector

CONTrol:ECAL:MODule[num]:STATe <value>

(Write-only) Sets the internal state of the selected ECAL module.

Parameters

[num] Optional argument. USB number of the ECal module. If unspecified (only one ECal module is connected to the USB), <num> is set to 1. If two or more modules are connected, use SENS:CORR:COLL:CKIT:INF? to verify their identity.

<value> Integer code for switching the module. The following are codes for Agilent ECal modules.

Agilent 8509x Modules		
State	**Port A**	**Port B**
Open	0	0
Short	43	43
Load	33	33
Mismatch	4	16
Thru	84	
Confidence	20	

Agilent N469x Modules		
State	**Port A**	**Port B**
Open	36	33
Short	39	45
Load	37	37
Mismatch (Offset short)	53	53
Impedance 5 (Offset open)	5	5
Impedance 6 (Offset short)	21	21
Impedance 7 (Offset short)	38	41
Thru	42	
Confidence	40	

Agilent N4431 Modules				
State	**Port A**	**Port B**	**Port C**	**Port D**
Open	-1398	-1384	-2774	-2654
Short	-1350	-1381	-2582	-2642
Load	26985	-26986	-26986	26985
Mismatch	-26986	26985	26985	-26986

Path	Thru	Confidence
AB Path	-2590	-598
AC Path	-4011	85
AD Path	-2517	16042
BC Path	-1650	-598
BD Path	-4011	85
CD Path	-1352	16042

Examples

```
CONT:ECAL:MOD:STAT 36
control:ecal:module2:state 38
```

Query Syntax Not Applicable

Overlapped? No

Default Not Applicable

CONTrol:SIGNal <conn>,<char>

(Read-Write) Configures external triggering in the PNA.

Trigger:Sequence:Source is automatically set to External when **CONTrol:SIGNal** is sent.

Edge triggering is only available on some Microwave PNA models.

For more information, see External Triggering in the PNA.

Parameters

<conn> Rear Panel connector to send or receive trigger signals. Choose from:

BNC1 Trigger IN from rear-panel Trigger IN BNC connector

AUXT Trigger IN from AUX IO connector Pin 19

Note: Only one of the input connectors is active at a time. When a command is sent to one, the PNA automatically makes the other INACTIVE.

BNC2 Trigger OUT to rear-panel Trigger OUT BNC connector.

MATHtrigger - Trigger IN from rear-panel Material Handler connector Pin 18

<char> **INACTIVE -** Disables the specified connector <conn>.

Choose from ONLY the following when <conn> is set to either **BNC1** or **AUXT**:

TIENEGATIVE - (Trigger In Edge Negative) - Triggers the PNA when receiving a negative going signal

TIEPOSITIVE - (Trigger In Edge Positive) - Triggers the PNA when receiving a positive going signal

TILLOW - (Trigger In Level Low) - Triggers the PNA when receiving a low level signal

TILHIGH - (Trigger In Level High) - Triggers the PNA when receiving a High-level signal

Choose from ONLY the following when <conn> is set to **BNC2**:

Use CONTrol:SIGNal:TRIGger:OUTP to enable the BNC2 output.

TOPPAFTER - (Trigger Out Pulse Positive After) - Sends a POSITIVE going TTL pulse at the END of each point during the sweep.

TOPPBEFORE - (Trigger Out Pulse Positive Before) - Sends a POSITIVE going TTL pulse at the START of each point during the sweep.

TOPNAFTER - (Trigger Out Pulse Negative After) - Sends a NEGATIVE going TTL pulse at the END of each point during the sweep.

TOPNBEFORE - (Trigger Out Pulse Negative Before) - Sends a NEGATIVE going TTL pulse at the START of each point during the sweep.

Examples

```
CONT:SIGN BNC1,TIENEGATIVE
control:signal bnc2,toppbefore
```

Query Syntax CONTrol:SIGNal? <conn>

In addition to the arguments listed above, the following is also a possible returned value:

NAVAILABLE - This feature is not available on this PNA

Return Type Character

Overlapped? No

Default At Preset:

BNC1 = INACTIVE
BNC2 = INACTIVE
AUXT = TILHIGH

When Output is enabled:

BNC1 = INACTIVE
BNC2 = TOPPAFTER
AUXT = TILHIGH

CONTrol:SIGNal:TRIGger:ATBA <bool>

(Read-Write) **Accept Trigger Before Armed** Determines what happens to an EDGE trigger signal if it occurs before the PNA is ready to be triggered. (LEVEL trigger signals are always ignored.) For more information, see External triggering.

Parameters

<bool>	Boolean False (0) - A trigger signal is ignored if it occurs before the PNA is ready to be triggered. True (1) - A trigger signal is remembered and then used when the PNA becomes armed (ready to be triggered). The PNA remembers only one trigger signal.
Examples	**CONT:SIGN:TRIG:ATBA 0** **control:signal:trigger:atba true**
Query Syntax	CONTrol:SIGNal:TRIGger:ATBA?
Return Type	Boolean
Overlapped?	No
Default	False

CONTrol:SIGNal:TRIGger:OUTP <bool>

(Read-Write) **Output Enabled** The PNA can be enabled to send trigger signals out the rear-panel TRIGGER OUT BNC connector. Use CONTrol:SIGNal to configure for output triggers.

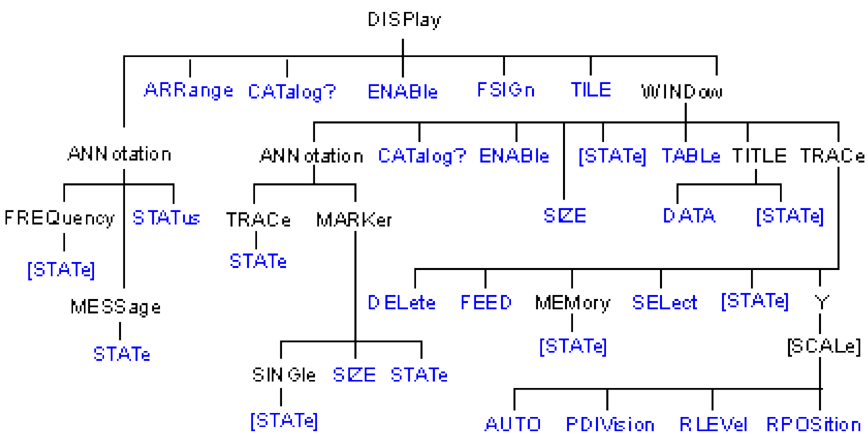
For more information, see External triggering.

Parameters

<bool>	Boolean False (0) - PNA does NOT output trigger signals. True (1) - PNA DOES output trigger signals.
Examples	**CONT:SIGN:TRIG:OUTP 1** **control:signal:trigger:outp false**
Query Syntax	CONTrol:SIGNal:TRIGger:OUTP?
Return Type	Boolean
Overlapped?	No
Default	False

Display Commands

Controls the settings of the front panel screen.



- Click on a blue keyword to view the command details.
- See a List of all commands in this block.
- See an example using some of these commands
- Learn about Screen Setup

DISPlay:ANNotation:FREQuency[:STATe] <ON | OFF>

(Read-Write) Turns frequency information on the display title bar ON or OFF for all windows.

Parameters	
<ON | OFF>	**ON** (or 1) - turns frequency annotation ON. **OFF** (or 0) - turns frequency annotation OFF.
Examples	`DISP:ANN:FREQ ON` `display:annotation:frequency:state off`
Query Syntax	DISPlay:ANNotation:FREQuency[:STATe]?
Return Type	Boolean (1 = ON, 0 = OFF)
Overlapped?	No
Default	**ON (1)**

DISPlay:ANNotation:MESSage:STATe <ON | OFF>

(Read-Write) Enables and disables error pop-up messages on the display.

Parameters	
<ON | OFF>	**ON** (or 1) - enables error pop-up messages **OFF** (or 0) - disables error pop-up messages
Examples	`DISP:ANN:MESS:STAT ON` `display:annotation:message:state off`
Query Syntax	DISPlay:ANNotation:MESSage:STATe?
Return Type	Boolean (1 = ON, 0 = OFF)
Overlapped?	No
Default	**ON (1)**

DISPlay:ANNotation:STATus <ON | OFF>

(Read-Write) Turns the status bar at the bottom of the screen ON or OFF. The status bar displays information for the active window.

Parameters	
<ON | OFF>	**ON** (or 1) - turns status bar ON. **OFF** (or 0) - turns status bar OFF.
Examples	`DISP:ANN:STAT ON` `display:annotation:status off`
Query Syntax	DISPlay:ANNotation:STATus?
Return Type	Boolean (1 = ON, 0 = OFF)
Overlapped?	No
Default	Last state that was set

DISPlay:ARRange <char>

(Read-Write) Places EXISTING measurements into pre-configured window arrangements. Overlay, Stack(2), Split(3), and Quad(4) creates new windows. To learn more, see Arrange Existing Measurements.

Parameters

<char> Window arrangement. Choose from:

- TILE - tiles existing windows
- CASCade - overlaps existing windows
- OVERlay - all traces placed in 1 window
- STACk - 2 windows
- SPLit - 3 windows
- QUAD - 4 windows

Examples

```
DISP:ARR CASC
display:arrange cascade
```

Query Syntax DISPlay:ARRange?

Return Type Character

Overlapped? No

Default TILE

DISPlay:CATalog?

(Read-only) Returns the existing Window numbers.

Return Type String of Character values, separated by commas

Example Two windows with numbers 1 and 2 returns:
"1,2"

Overlapped? No

Default Not applicable

DISPlay:ENABLE <ON | OFF>

(Read-Write) Specifies whether to disable or enable all analyzer display information **in all windows** in the analyzer application. Marker data is not updated. More CPU time is spent making measurements instead of updating the display.

Parameters

<ON | OFF>	**ON** (or 1) - turns the display ON. **OFF** (or 0) - turns the display OFF.
Examples	`DISP:ENAB ON` `display:enable off`
Query Syntax	DISPlay:ENABle?
Return Type	Boolean (1 = ON, 0 = OFF)
Overlapped?	No
Default	ON

DISPlay:FSIGn <ON | OFF>

(Read-Write) Shows or hides the window which displays global pass/fail results.

Parameters

<ON | OFF>	**ON** (or 1) - displays the pass/fail dialog **OFF** (or 0) **-** hides the pass/fail dialog
Examples	`DISP:FSIG ON` `display:fsign off`
Query Syntax	DISPlay:FSIGn?
Return Type	Boolean (1 = ON, 0 = OFF)
Overlapped?	No
Default	OFF

DISPlay[:TILE] - OBSOLETE - Use DISP:ARRange

(Write-only) Tiles the windows on the screen.

Examples	`DISP` `display:tile`
Overlapped?	No
Default	Not Applicable

DISPlay:WINDow<wnum>:ANNotation:MARKer:SINGle[:STATe] <bool>

(Read-Write) Either shows marker readout of only the active trace or all of the traces simultaneously. See other SCPI Marker commands

Parameters	
<wnum>	Any existing window number (**1** to **16**); if unspecified, value is set to 1.
<bool>	**ON** (or 1) - show a single marker per trace **OFF** (or 0) - show up to 4 markers per active trace
Examples	`DISP:WIND:ANN:MARK:SING ON` `display:window:annotation:marker:single off`
Query Syntax	DISPlay:WINDow:ANNotation:MARKer:SINGle?
Return Type	Boolean (1 = ON, 0 = OFF)
Overlapped?	No
Default	OFF

DISPlay:WINDow<wnum>:ANNotation:MARKer:SIZE <char>

(Read-Write) Specifies the size of the marker readout text. See other SCPI Marker commands

Parameters	
<wnum>	Any existing window number (**1** to **16**); if unspecified, value is set to 1.
<char>	Readout text size. Choose from:**NORMal | LARGe**
Examples	`DISP:WIND:ANN:MARK:SIZE LARG` `display:window:annotation:marker:size normal`
Query Syntax	DISPlay:WINDow:ANNotation:MARKer:SIZE?
Return Type	Character
Overlapped?	No
Default	NORMal

DISPlay:WINDow<wnum>:ANNotation:MARKer:STATe <ON | OFF>

(Read-Write) Specifies whether to show or hide the Marker data (when markers are ON) on the selected window. See other SCPI Marker commands

Parameters

<wnum>	Any existing window number (**1** to **16**); if unspecified, value is set to 1.
<ON | OFF>	**ON** (or 1) - turns marker data ON. **OFF** (or 0) - turns marker data OFF.
Examples	**DISP:WIND:ANN:MARK:STAT ON** **display:window:annotation:marker:state off**
Query Syntax	DISPlay:WINDow:ANNotation:MARKer:STATe?
Return Type	Boolean (1 = ON, 0 = OFF)
Overlapped?	No
Default	ON

DISPlay:WINDow<wnum>:ANNotation:TRACe:STATe <ON | OFF>

(Read-Write) Specifies whether to show or hide the Trace Status buttons on the left of the display.

Parameters

<wnum>	Any existing window number (**1** to **16**); if unspecified, value is set to 1.
<ON | OFF>	**ON** (or 1) - turns the buttons ON. **OFF** (or 0) - turns the buttons OFF.
Examples	**DISP:WIND:ANN:TRAC:STAT ON** **display:window:annotation:trace:state off**
Query Syntax	DISPlay:WINDow:ANNotation:TRACe:STATe?
Return Type	Boolean (1 = ON, 0 = OFF)
Overlapped?	No
Default	ON

DISPlay:WINDow<wnum>:CATalog?

(Read-only) Returns the trace numbers for the specified window.

Parameters

<wnum>	Any existing window number (**1** to **16**); if unspecified, value is set to 1.
Example	Window 1 with four traces: **`DISPlay:WINDow1:CATalog?`** `Returns:` **`"1,2,3,4"`**
Return Type	String of Character values separated by commas
Overlapped?	No
Default	Not applicable

DISPlay:WINDow<wnum>:ENABle <ON | OFF>

(Read-Write) Specifies whether to disable or enable all analyzer display information **in the specified window**. Marker data is not updated. More CPU time is spent making measurements instead of updating the display.

Parameters

<wnum>	Any existing window number (**1** to **16**); if unspecified, value is set to 1.
<ON | OFF>	**ON** (or 1) - turns the display ON. **OFF** (or 0) - turns the display OFF.
Examples	**`DISP:WIND:ENABle ON`** **`display:window1:enable off`**
Query Syntax	DISPlay:WINDow<wnum>:ENABle?
Return Type	Boolean (1 = ON, 0 = OFF)
Overlapped?	No
Default	ON

DISPlay:WINDow<wnum>:SIZE <char>

(Read-Write) Sets or returns the window setting of Maximized, Minimized, or Normal. To arrange all of the windows, use DISP:ARR.

Parameters

<wnum>	Any existing window number (**1** to **16**); if unspecified, value is set to 1
<char>	Window size. Choose from: **MIN | MAX | NORM**
Examples	`DISP:WIND:SIZE MAX` `display:window:size norm`
Query Syntax	DISPlay:WINDow:SIZE?
Overlapped?	No
Default	Not Applicable

DISPlay:WINDow<wnum>[:STATe] <ON | OFF>

(Read-Write) Write to create or delete a window on the screen or Read whether a window is present.

Parameters

<wnum>	Window number to create; choose any integer between: **1** and **16**
<ON | OFF>	**ON** (or 1) - The window <wnum> is created. **OFF** (or 0) - The window <wnum> is deleted.
Examples	`DISP:WIND ON` `display:window2:state off`
Query Syntax	DISPlay:WINDow<wnum>[:STATe]?
Return Type	Boolean (1 = ON, 0 = OFF)
Overlapped?	No
Default	Window number "1" **ON**

DISPlay:WINDow<wnum>:TABLe <char>

(Read-Write) Write to show the specified table at the bottom of the analyzer screen or Read to determine what table is visible.

Parameters

<wnum>	Any existing window number (**1** to **16**); if unspecified, value is set to 1
<char>	Table to show. Choose from: **OFF | MARKer | LIMit | SEGMent**
Examples	**`DISP:WIND:TABLe SEGM`** **`display:window:table off`**
Query Syntax	DISPlay:WINDow:TABLe?
Overlapped?	No
Default	OFF

DISPlay:WINDow<wnum>:TITLe:DATA <string>

(Read-Write) Sets data in the window title area. The title is turned ON and OFF with DISP:WIND:TITL:STAT OFF.

Parameters

<wnum>	Any existing window number (**1** to **16**); if unspecified, value is set to 1.
<string>	Title to be displayed. Any characters, enclosed with quotes. If the title string exceeds 50 characters, an error will be generated and the title not accepted. Newer entries replace (not append) older entries.
Examples	**`DISP:WIND:TITL:DATA 'hello'`** **`display:window2:title:data 'hello'`**
Query Syntax	DISPlay:WINDow<wnum>:TITLe:DATA?
Return Type	String
Overlapped?	No
Default	NA

DISPlay:WINDow<wnum>:TITLe[:STATe] <ON | OFF>

(Read-Write) Turns display of the title string ON or OFF. When OFF, the string remains, ready to be redisplayed when turned back ON.

Parameters	
<wnum>	Any existing window number (**1** to **16**); if unspecified, value is set to 1
<ON | OFF>	**ON** (or 1) - turns the title string ON. **OFF** (or 0) - turns the title string OFF.
Examples	**`DISP:WIND:TITL ON`** **`Display:window1:title:state off`**
Query Syntax	DISPlay:WINDow<wnum>:TITLe[:STATe]?
Return Type	Boolean (1 = ON, 0 = OFF)
Overlapped?	No
Default	ON

DISPlay:WINDow<wnum>:TRACe<tnum>:DELete

(Write-only) Deletes the specified trace from the specified window. The measurement parameter associated with the trace is not deleted.

Parameters	
<wnum>	Any existing window number (**1** to **16**); if unspecified, value is set to 1.
<tnum>	The number of the trace to be deleted; if unspecified, value is set to 1
Examples	**`DISP:WIND:TRAC:DEL`** **`display:window2:trace2:delete`**
Query Syntax	Not applicable
Overlapped?	No
Default	NA

DISPlay:WINDow<wnum>:TRACe<tnum>:FEED <name>

(Write-only) Creates a new trace <tnum> and associates (feeds) a measurement <name> to the specified window<wnum>. This command should be executed immediately after creating a new measurement with CALC:PAR:DEF<name>,<parameter>.

To feed the same measurement to multiple traces, create another measurement with the same <name>,<parameter> using the CALC:PAR:DEF command. The analyzer will collect the data only once.

Parameters	
<wnum>	Any existing window number (**1** to **16**); if unspecified, value is set to 1.
<tnum>	Trace number to be created. Choose any Integer between: **1** and **4**
<name>	Name of the measurement that was defined with CALC:PAR:DEF**<name>**,<parameter>
Examples	**DISP:WIND:TRAC:FEED 'test'** **display:window2:trace2:feed 'test'**
Query Syntax	Not applicable
Overlapped?	No
Default	"CH1_S11"

DISPlay:WINDow<wnum>:TRACe<tnum>MEMory[:STATe] <ON | OFF>

(Read-Write) Turns the memory trace ON or OFF.

Parameters	
<wnum>	Any existing window number (**1** to **16**); if unspecified, value is set to 1.
<tnum>	Any existing trace number; if unspecified, value is set to 1
<ON | OFF>	**ON** (or 1) - turns the memory trace ON. **OFF** (or 0) - turns the memory trace OFF.
Examples	**DISP:WIND:TRAC:MEM ON** **display:window2:trace2:memory:state off**
Query Syntax	DISPlay:WIND<wnum>:TRACe<tnum>:MEMory[:STATe]?
Return Type	Boolean (1 = ON, 0 = OFF)
Overlapped?	No
Default	OFF

DISPlay:WINDow<wnum>:TRACe<tnum>:SELect

(Write-only) Activates the specified trace in the specified window for front panel use.

Parameters	
<wnum>	Any existing window number (**1** to **16**); if unspecified, value is set to 1.
<tnum>	Any existing trace number; if unspecified, value is set to 1
Examples	`DISP:WIND:TRAC:SEL` `display:window2:trace2:select`
Query Syntax	Not applicable
Overlapped?	No
Default	NA

DISPlay:WINDow<wnum>:TRACe<tnum>[:STATe] <ON | OFF>

(Read-Write) Turns the display of the specified trace in the specified window ON or OFF. When OFF, the measurement behind the trace is still active.

Parameters	
<wnum>	Any existing window number (**1** to **16**); if unspecified, value is set to 1.
<tnum>	Any existing trace number; if unspecified, value is set to 1
<ON | OFF>	**ON** (or 1) - turns the trace ON. **OFF** (or 0) - turns the trace OFF.
Examples	`DISP:WIND:TRAC ON` `display:window2:trace2:state off`
Query Syntax	DISPlay:WIND<wnum>:TRACe<tnum>[:STATe]?
Return Type	Boolean (1 = ON, 0 = OFF)
Overlapped?	No
Default	ON

DISPlay:WINDow<wnum>:TRACe<tnum>:Y[:SCALe]:AUTO

(Write-only) Performs an **Autoscale** on the specified trace in the specified window, providing the best fit display. Autoscale is performed only when the command is sent; it does NOT keep the trace autoscaled indefinitely.

Parameters	
<wnum>	Any existing window number (**1** to **16**); if unspecified, value is set to 1.
<tnum>	Any existing trace number; if unspecified, value is set to 1
Examples	`DISP:WIND:TRAC:Y:AUTO` `display:window2:trace2:y:scale:auto`
Query Syntax	Not applicable
Overlapped?	No
Default	Not applicable

DISPlay:WINDow<wnum>:TRACe<tnum>:Y[:SCALe]:PDIVision <num>

(Read-Write) Sets the Y axis **Per Division** value of the specified trace in the specified window.

Parameters	
<wnum>	Any existing window number (**1** to **16**); if unspecified, value is set to 1.
<tnum>	Any existing trace number; if unspecified, value is set to 1
<num>	Units / division value. The range of acceptable values is dependent on format and domain. **Note**: This command will accept **MIN** or **MAX** instead of a numeric parameter. See SCPI Syntax for more information.
Examples	`DISP:WIND:TRAC:Y:PDIV 1` `display:window2:trace2:y:scale:pdivision maximum`
Query Syntax	DISPlay:WINDow<wnum>:TRACe<tnum>:Y[:SCALe]:PDIVision?
Return Type	Numeric
Overlapped?	No
Default	10

DISPlay:WINDow<wnum>:TRACe<tnum>:Y[:SCALe]:RLEVel <num>

(Read-Write) Sets the Y axis Reference Level of the specified trace in the specified window.

Parameters

<wnum>	Any existing window number (**1** to **16**); if unspecified, value is set to 1.
<tnum>	Any existing trace number; if unspecified, value is set to 1
<num>	Reference level value. The range of acceptable values is dependent on format and domain.

Note: This command will accept MIN or MAX instead of a numeric parameter. See <u>SCPI Syntax</u> for more information.

Examples

```
DISP:WIND:TRAC:Y:RLEV 0
display:window2:trace2:y:scale:rlevel minimum
```

Query Syntax DISPlay:WINDow<wnum>:TRACe<tnum>:Y[:SCALe]:RLEVel?

Return Type Numeric

<u>Overlapped?</u> No

<u>Default</u> NA

DISPlay:WINDow<wnum>:TRACe<tnum>:Y[:SCALe]:RPOSition <num>

(Read-Write) Sets the **Reference Position** of the specified trace in the specified window

Parameters

<wnum>	Any existing window number (**1** to **16**); if unspecified, value is set to 1.
<tnum>	Any existing trace number; if unspecified, value is set to 1
<num>	Reference position on the screen measured in horizontal graticules from the bottom. The range of acceptable values is dependent on format and domain.

Note: This command will accept **MIN** or **MAX** instead of a numeric parameter. See <u>SCPI Syntax</u> for more information.

Examples

```
DISP:WIND:TRAC:Y:RPOS 0
display:window2:trace2:y:rposition maximum
```

Query Syntax DISPlay:WINDow<wnum>:TRACe<tnum>:Y[:SCALe]:RPOSition?

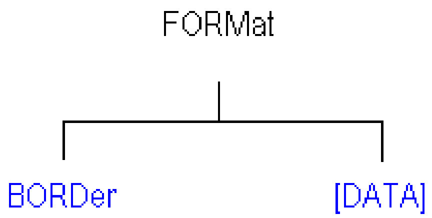
Return Type Numeric

<u>Overlapped?</u> No

<u>Default</u> 5

Format Commands

Specifies the way that data will be transferred when moving large amounts of data. These commands will affect data that is transferred with the CALC:DATA and CALC:RDATA commands.



- Click on a blue keyword to view the command details.
- See a List of all commands in this block.

FORMat:BORDer <char>

(Read-Write) Set the byte order used for GPIB data transfer. Some computers read data from the analyzer in the reverse order. This command is only implemented if FORMAT:DATA is set to :REAL. If FORMAT:DATA is set to :ASCII, the swapped command is ignored.

Parameters

<char> Choose from:

NORMal - Use when your controller is anything other than an IBM compatible computers

SWAPped - for IBM compatible computers

Note: Use **NORMal** if you are using drivers like VISA or development environments like VEE, LabView, or T&M Tool kit.

Examples

```
FORM:BORD SWAP
format:border normal
```

Query Syntax FORMat:BORDer?

Return Type Character

Overlapped? No

Default Normal

FORMat[:DATA] <char>

(Read-Write) Sets the data format for data transfers. To transfer measurement data, use the CALC:DATA command.
To transfer Source Power correction data, use SOURce:POWer:CORRection:COLLect:TABLe:DATA
SOURce:POWer:CORRection:COLLect:TABLe:FREQuency
SOURce:POWer:CORRection:DATA

Parameters	
<char>	Choose from: **REAL,32** - (default value for REAL) Best for transferring large amounts of measurement data. **REAL,64** - Slower but has more significant digits than REAL,32. Use REAL,64 if you have a computer that doesn't support REAL,32. **ASCii,0** - The easiest to implement, but very slow. Use if small amounts of data to transfer. Note: The **REAL,32** and **REAL,64** arguments transfer data in block format as explained in Transferring Measurement Data.
Examples	`FORM REAL,64` `format:data ascii`
Query Syntax	FORMat:DATA?
Return Type	Character,Character
Overlapped?	No
Default	**ASCii,0**

Hardcopy Command

Learn about Printing

HCOPy:FILE <filename>

(Write-only) Saves the screen image to a file.

Parameters

<filename>	Name of the file to save the screen to. The file is saved to the current working directory unless a valid full path name is specified. Use one of the following suffixes: .bmp - not recommended due to large file size .jpg - not recommended due to poor quality .png - recommended
Examples	`HCOPY:FILE "myFile.png"` `hcopy:file "c:\data\myfile.png"`
Query Syntax	Not Applicable
Overlapped?	No
Default	Not Applicable

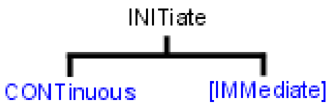
HCOPy[:IMMediate]

(Write-only) Prints the screen to the default printer.

Examples	`HCOP` `hcopy:immediate`
Query Syntax	Not applicable
Overlapped?	No
Default	Not Applicable

Initiate Commands

Controls triggering signals



- Click on a blue keyword to view the command details.
- See a List of all commands in this block.
- Learn about Triggering

INITiate:CONTinuous <boolean>

(Read-Write) Specifies whether the analyzer sends Continuous sweep triggers to triggerable channels or enables Manual triggering.

Parameters

<boolean>	**ON** (or 1) **-** Continuous sweep mode. **OFF** (or 0) - Manual sweep mode.
Examples	`INIT:CONT ON` `initiate:continuous off`
Query Syntax	INITiate:CONTinuous?
Return Type	Boolean (1 = ON, 0 = OFF)
Overlapped?	No
Default	ON

INITiate<cnum>[:IMMediate]

(Write-only) Stops the current sweeps and immediately sends a trigger to the specified channel. (Same as Sweep \ Trigger \ Trigger!)

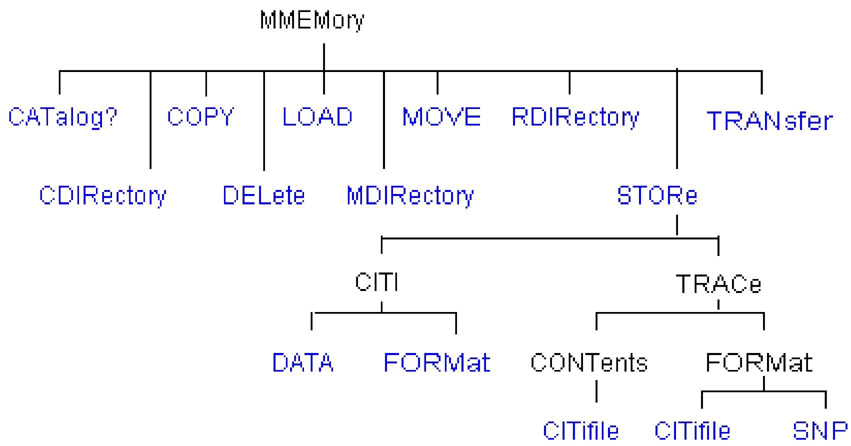
- If the specified channel is in HOLD, it will sweep one time and return to HOLD when complete.
- If Trigger:Scope = Global, all channels will receive a trigger.
- If Trigger:Scope = Channel (only the active channel receives a trigger) and the specified channel is not the active channel, the specified channel will NOT receive a trigger signal.
- If the specified channel is NOT in Manual trigger (INIT:CONT OFF), the analyzer will return an error.
- If channel <cnum> does not exist, the analyzer will return an error.

Note: An **SMC Fixed Output** measurement cannot be triggered using this command. For more information, see the example program.

Parameters	
<cnum>	Any existing channel number. If unspecified, value is set to 1
Examples	**INIT** **initiate2:immediate**
Query Syntax	Not applicable
Overlapped?	Yes
Default	Not applicable

Memory Commands

The memory commands control saving and loading instrument states and measurement trace data to the hard drive. To read and write trace data in GPIB format, see CALC:DATA.



- Click on a blue keyword to view the command details.
- See a List of all commands in this block.
- Learn about Save / Recall and File Types

Specifying Path Names

The MMEM commands use the following rules to specify path names:

- The default folder is "C:\Program Files\Agilent\Network Analyzer\Documents"
- You can change the active directory using MMEMory:CDIRectory.
- Specify only the file name if using the active directory.
- You can also use an absolute path name to specify the folder and file.

MMEMory:CATalog[:<char>]? [<folder>]

(Read-only) Returns a comma-separated string of file names that are in the specified folder. If there are no files of the specified type, "NO CATALOG" is returned.

Parameters

<char> The type of files to list. Choose from:
STATe - Instrument states (.sta)
CORRection - Calibration Data (.cal)
CSTate - Instrument state and Calibration data (.cst)

If unspecified then ALL file types (even unknown types) are listed.

<folder> String - Any existing folder name. See Specifying Path Names

Examples

```
MMEM:CAT? 'lists all files from the current folder
mmemory:catalog:correction? 'C:\Program Files\Agilent\Network
Analyzer\Documents' 'lists .cal files from the specified folder
```

Overlapped? No

Default Not applicable

MMEMory:CDIRectory <folder>

(Read-Write) Changes the folder name.

Parameters

<folder> Any drive and folder name that already exists.
If the same level as "C:\Program Files\Agilent\Network Analyzer\Documents", then no punctuation is required

```
MMEM:CDIR Service
```

If the new folder is at a different level than "C:\Program Files\Agilent\Network Analyzer\Documents", use a slash (\) before the folder name and enclose in quotes.

```
mmemory:cdirectory '\automation' 'changes default directory up
one level.
```

You can use an absolute path to specify the new folder.

```
mmemory:cdirectory 'c:\automation\service'
```

Query Syntax MMEMory:CDIRectory? 'Returns the current folder name

Return Type String

Overlapped? No

Default 'C:\Program Files\Agilent\Network Analyzer\Documents'

MMEMory:COPY <file1>,<file2>

(Write-only) Copies file1 to file2. Extensions must be specified.

Parameters

<file1>	String - Name of the file to be copied. See Specifying Path Names
<file2>	String - Name of the file to be created from file1.
Examples	`MMEM:COPY 'MyFile.cst','YourFile.cst'`
Query Syntax	Not applicable
Overlapped?	No
Default	Not applicable

MMEMory:DELete <file>

(Write-only) Deletes file. Extensions must be specified.

Parameters

<file>	String - Name of the file to be deleted. See Specifying Path Names
Examples	`MMEM:DEL 'MyFile.cst'`
Query Syntax	Not applicable
Overlapped?	No
Default	Not applicable

MMEMory:LOAD[:<char>] <file>

(Write-only) Loads the specified file.

Parameters

<char>	The type of file to load. Choose from: **STATe** - Instrument states (.sta) **CORRection** - Calibration Data (.cal) **CSTate** - Instrument state and Calibration data (.cst) If <char> is unspecified, the extension must be included in the filename. If an extension is specified in <file> that does not agree with <char> then no action is taken.
<file>	String - Name of the file to be loaded. See Specifying Path Names
Examples	`MMEM:LOAD 'MyFile.cst'` `mmemory:load:state 'MyInstState'`

Query Syntax	Not applicable
Overlapped?	No
Default	Not applicable

MMEMory:MDIRectory <folder>

(Write-only) Makes a folder.

Parameters

<folder>	String - Name of the folder to make. See Specifying Path Names
Examples	**MMEM:MDIR 'MyFolder'** **mmemory:mdirectory 'c:\NewFolder'**
Query Syntax	Not applicable
Overlapped?	No
Default	Not applicable

MMEMory:MOVE <file1>,<file2>

(Write-only) Renames <file1> to <file2>. File extensions must be specified.

Parameters

<file1>	String - Name of the file to be renamed. See Specifying Path Names
<file2>	String - Name of the new file.
Examples	**MMEM:MOVE 'MyFile.cst','YourFile.cst'**
Query Syntax	Not applicable
Overlapped?	No
Default	Not applicable

MMEMory:RDIRectory <folder>

(Write-only) Removes the specified folder.

Parameters	
<folder>	String - Name of the folder to remove. See Specifying Path Names
Examples	`MMEM:RDIR 'MyFolder'`
Query Syntax	Not applicable
Overlapped?	No
Default	Not applicable

MMEMory:STORe[:<char>] <file>

(Write-only) Stores the specified file (.sta, .cal, .cst, .s1p, .s2p, and .s3p.).

Learn about saving SnP files on the PNA.

To save *.cti files, use MMEM:STOR:CITI:DATA or MMEM:STOR:CITI:FORM.

Parameters	
<char>	The type of file to store. Choose from: **STATe** - Instrument states (.sta) **CORRection** - Calibration Data (.cal) **CSTate** - Instrument state and Calibration data (.cst) No <char> is specified for s1p, s2p and s3p If unspecified, then the extension must be included in the filename. If an extension is specified in <file> that does not agree with <char> then no action is taken.
<file>	String - Name of any valid file that is not already in existence. See Specifying Path Names

Examples	`MMEM:STOR:STAT 'myState'`
	`mmemory:store 'c:\bin\myState.sta'`
	`MMEM:STOR 'MyData.S2P'`

Query Syntax	Not applicable
Overlapped?	No
Default	Not applicable

MMEMory:STORe:CITI:DATA <filename>

(Write only) Saves UNFORMATTED trace data to .cti file. Learn more.

Parameters

<filename> Any drive and folder name that already exists.
If the same level as "C:\Program Files\Agilent\Network Analyzer\Documents", then no punctuation is required

```
MMEM:STOR:CITI:DATA MYCITIFile
```

If the new folder is at a different level than "C:\Program Files\Agilent\Network Analyzer\Documents", use a slash (\) before the folder name and enclose in quotes.

```
mmemory:cdirectory '\automation' 'changes default directory up one level.
```

You can use an absolute path to specify the new folder.

```
mmemory:cdirectory 'c:\automation\service'
```

Query Syntax Not Applicable

Overlapped? No

Default C:\Program Files\Agilent\Network Analyzer\Documents'

MMEMory:STORe:CITI:FORMat <filename>

(Write only) Saves FORMATTED trace data to .cti file. Learn more.

Parameters

<filename> Any drive and folder name that already exists.
If the same level as "C:\Program Files\Agilent\Network Analyzer\Documents", then no punctuation is required

```
MMEM:STOR:CITI:DATA MYFile
```

If the new folder is at a different level than "C:\Program Files\Agilent\Network Analyzer\Documents", use a slash (\) before the folder name and enclose in quotes.

```
mmemory:cdirectory '\automation' 'changes default directory up one level.
```

You can use an absolute path to specify the new folder.

```
mmemory:cdirectory 'c:\automation\service'
```

Query Syntax Not Applicable

Overlapped? No

Default C:\Program Files\Agilent\Network Analyzer\Documents'

MMEMory:STORe:TRACe:CONTents:CITifile <char>

(Read-Write) Specifies the contents of subsequent citifile save statements. (See Data Define Saves)

Parameters

<char>	Choose from: **SING** - Single trace **DISP** - All displayed traces **AUTO** - All displayed traces
Examples	`MMEM:STOR:TRAC:CONT:CIT SING`
Query Syntax	MMEMory:STORe:TRACe:CONTents?
Return Type	Character
Overlapped?	No
Default	Auto

MMEMory:STORe:TRACe:FORMat:CITifile <char>

(Read-Write) Specifies the format of subsequent citifile save statements. (See Data Define Saves)

Parameters

<char>	Format in which the citifile will be saved with subsequent MMEMory:STORe:CITI:FORMat statements. Choose from: **MA** - Linear Magnitude / degrees **DB** - Log Magnitude / degrees **RI** - Real / Imaginary **AUTO** - Format in which the trace is already displayed. If other than Log Mag, Linear Magnitude, or Real/Imag, then the format will be in Real/Imag.
Examples	`MMEM:STOR:TRAC:FORM:CIT MA`
Query Syntax	MMEMory:STORe:TRACe:FORMat:CITifile?
Return Type	Character
Overlapped?	No
Default	Auto'

MMEMory:STORe:TRACe:FORMat:SNP <char>

(Read-Write) Specifies the format of subsequent .s1p, .s2p, .s3p save statements. (See Data Define Saves).

Parameters

<char>	Choose from: **MA** - Linear Magnitude / degrees **DB** - Log Magnitude / degrees **RI** - Real / Imaginary **AUTO** - data is output in currently selected trace format. If other than LogMag, LinMag, or Real/Imag, then output is in Real/Imag.

Examples `MMEM:STOR:TRAC:FORM:SNP MA`

Query Syntax MMEMory:STORe:TRACe:FORMat:SNP?

Return Type Character

Overlapped? No

Default Auto'

MMEMory:TRANsfer <fileName>,<dataBlock>

(Read-Write) Transfers data between the PNA and an external controller. Other MMEM commands transfer data between the PNA application and the PNA hard drive. If <fileName> already exists, it will be overwritten. The file must be no larger than 20MB.

To read **trace data** from the PNA in block format, use CALC:DATA.

Parameters

<fileName>	String - File name. See Specifying Path Names
<dataBlock>	Block Data - The contents of the file. The data block is a block of binary data. Use the following syntax: #**<num digits><byte count>**<data bytes><NL><END> where: **<num_digits>** specifies how many digits are contained in <byte_count> **<byte_count>** specifies how many data bytes will follow in <data bytes>

Example

```
!Write the data
10 OUTPUT 717;":MMEM:TRAN ""Trace01.csv"",#6012345";Dat$
!# - always sent before definite block data
!6 - specifies that the byte count is six digits (012345)
!12345 - specifies the number of data bytes that will follow,
not counting <NL><END>
!Dat$ - Data

!Read the data back
10 OUTPUT 717;":MMEM:TRAN? ""Trace01.csv"""
20 ENTER 717 USING "#,A";A$
30 ENTER 717 USING "#,A";Digit$
40 Img$="#,"&Digit$&"A"
50 ENTER 717 USING Img$;Byte$
60 Img$=Byte$&"A"
70 ALLOCATE Dat$[VAL(Byte$)]
80 ENTER 717 USING Img$;Dat$
```

Query Syntax MMEMory:TRANsfer? <fileName>

Reads block data from the specified file location.

Overlapped? No

Default Not applicable

Output Command

Learn about Power

OUTPut[:STATe] <ON | OFF>

(Read-Write) Turns RF power from the source ON or OFF.

See note about source power state with instrument state save and recall.

Parameters	
<ON | OFF>	**ON** (or 1) - turns RF power ON **OFF** (or 0) - turns RF power OFF
Examples	`OUTP ON` `output:state off`
Query Syntax	OUTPut[:STATe]?
Return Type	Boolean (1 = ON, 0 = OFF)
Overlapped?	No
Default	ON

Route Command

Learn about Frequency Offset

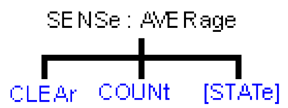
ROUTe:PATH:LOOP:R1 <char>

(Read-Write) Throws internal switch to reference receiver. This feature is only available on PNA models with Option 081 - external reference switch. See block diagram of the reference switch.

Parameters	
<char>	Position of the switch. Choose from: **INTernal** - bypass R1 Loop. Connects the port 1 source directly to the R1 receiver. **EXTernal -** flow through R1 Loop. Allows direct access to the R1 receiver through the Reference 1 front-panel connectors.
Examples	`ROUT:PATH:LOOP:R1 INT` `route:path:loop:r1 external`
Query Syntax	ROUTe:PATH:LOOP:R1?
Return Type	Character
Overlapped?	No
Default	INTernal

Sens:Average Commands

Sets sweep-to-sweep averaging parameters. Averaging is a noise reduction technique that averages each data point over a user-specified number of sweeps. Averaging affects all of the measurements in the channel.



- Click on a blue keyword to view the command details.
- See a List of all commands in this block.
- See an example using some of these commands.
- Learn about Averaging

SENSe<cnum>:AVERage:CLEar

(Write-only) Clears and restarts averaging of the measurement data. Must also set SENS:AVER[:STATe] ON

Parameters	
<cnum>	Any existing channel number; if unspecified, value is set to 1.
Examples	**SENS:AVER:CLE** **sense2:average:clear**
Query Syntax	Not applicable
Overlapped?	No
Default	Not applicable

SENSe<cnum>:AVERage:COUNt <num>

(Read-Write) Sets the number of measurement sweeps to combine for an average. Must also set SENS:AVER[:STATe] ON

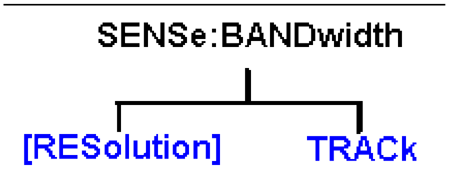
Parameters	
<cnum>	Any existing channel number; if unspecified, value is set to 1.
<num>	Number of measurement sweeps to average. Choose any number between: **1** and **1024**
Examples	**`SENS:AVER:COUN 999`** **`sense2:average:count 73`**
Query Syntax	SENSe<cnum>:AVERage:COUNt?
Return Type	Numeric
Overlapped?	No
Default	1

SENSe<cnum>:AVERage[:STATe] <ON | OFF>

(Read-Write) Turns trace averaging ON or OFF.

Parameters	
<cnum>	Any existing channel number; if unspecified, value is set to 1.
<ON | OFF>	**ON** (or 1) - turns averaging ON. **OFF** (or 0) - turns averaging OFF.
Examples	**`SENS:AVER ON`** **`sense2:average:state off`**
Query Syntax	SENSe<cnum>:AVERage[:STATe]?
Return Type	Boolean (1 = ON, 0 = OFF)
Overlapped?	No
Default	Off

Sense:Bandwidth Commands



Learn about IF Bandwidth

See a list of SCPI commands

SENSe<cnum>:BANDwidth | BWIDth[:RESolution] <num>

(Read-Write) Sets the bandwidth of the digital IF filter to be used in the measurement. (Use either **Sense:Bandwidth** or **Sense:Bwidth**)

Parameters	
<cnum>	Any existing channel number. If unspecified, value is set to 1
<num>	IF Bandwidth in Hz. The list of valid IF Bandwidths is different depending on the PNA model. (Click to see the lists.) If an invalid number is specified, the analyzer will round up to the closest valid number.
Examples	`SENS:BWID 1KHZ` `sense2:bandwidth:resolution 1000`
Query Syntax	SENSe<cnum>:BANDwidth | BWIDth[:RESolution]?
Return Type	Numeric
Overlapped?	No
Default	See Preset IFBW for your PNA model.

SENSe<cnum>:BANDwidth | BWIDth:TRACk <bool>

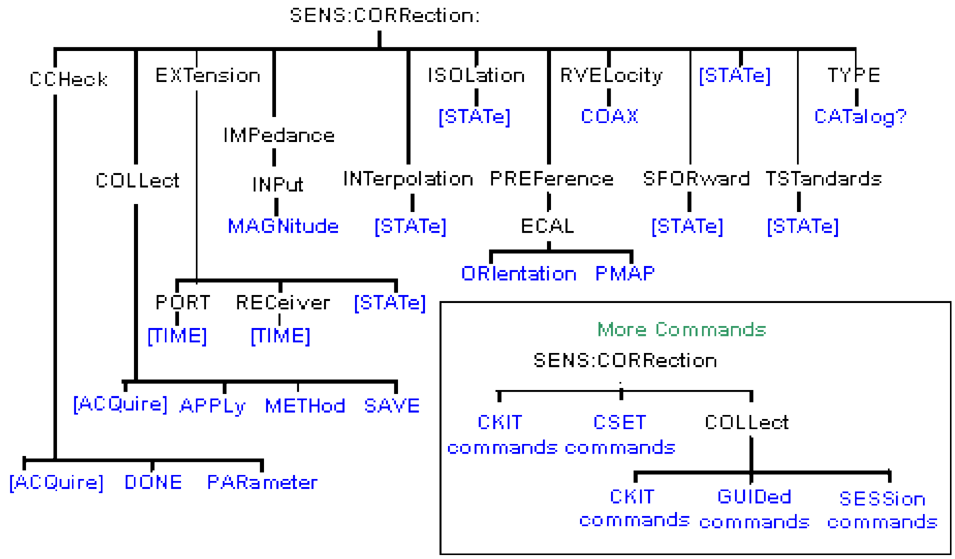
(Read-Write) Sets and returns the state of the Reduce IF BW at Low Frequencies feature.

(Use either **Sense:Bandwidth:Track** or **Sense:Bwidth:Track**).

Parameters	
<cnum>	Any existing channel number. If unspecified, value is set to 1
<bool>	Choose from: **ON** or **1 -** Reduce IF BW at Low Frequencies is set ON **OFF** or **0 -** Reduce IF BW at Low Frequencies is set OFF
Examples	`SENS:BWID:TRAC OFF` `sense2:bandwidth:track 1`
Query Syntax	SENSe<cnum>:BANDwidth | BWIDth:TRACk?
Return Type	Boolean
Overlapped?	No
Default	ON

Sense:Correction Commands

Performs and applies measurement calibration and other error correction features.



- Click on a blue keyword to view the command details.
- New See Calibrating the PNA Using SCPI
- See a List of all commands in this block.
- See a examples using some of these commands.
- Learn about Measurement Calibration

SENSe<cnum>:CORRection:CCHeck[:ACQuire] <mod>[,char]

(Write-only) Reads the 'confidence data' associated with the specified ECal module and puts it into memory. The measurement is selected using SENS:CORR:CCH:PAR. This command is compatible with *OPC.

Parameters

<cnum> Any existing channel number. If unspecified, value is set to 1.

<mod> ECAL Module that contains the confidence data. Choose from:

ECAL1

..through..

ECAL8

[char] Optional argument. Specifies which characterization within the ECal module that the confidence data will be read from.

CHAR0 Factory characterization (data that was stored in the ECal module by Agilent). Default if not specified.

CHAR1 User characterization #1

CHAR2 User characterization #2

CHAR3 User characterization #3

CHAR4 User characterization #4

CHAR5 User characterization #5

Examples

```
SENS:CORR:CCHeck ECAL2
sense2:correction:ccheck:acquire ecal1,char1
```

Query Syntax Not applicable

Overlapped? No

Default Not applicable

SENSe<cnum>:CORRection:CCHeck:DONE

(Write-only) Concludes the Confidence Check and sets the ECal module back into the idle state.

Parameters

<cnum> Any existing channel number. If unspecified, value is set to 1

Examples

```
SENS:CORR:CCH:DONE
sense2:correction:ccheck:done
```

Query Syntax Not applicable

Overlapped? No

Default Not applicable

SENSe<cnum>:CORRection:CCHeck:PARameter <Mname>

(Read-Write) Specifies an existing measurement to be used for the Confidence Check.

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1
<Mname>	Name of the measurement you are selecting for the confidence check. The measurement must already exist.

Examples

```
SENS:CORR:CCH:PAR 'TEST'
'selects the measurement "test" on channel 1 for the confidence check

sense2:correction:ccheck:parameter 'test'
'selects the measurement "test" on channel 2 for the confidence check
```

Query Syntax SENSe<cnum>:CORRection:CCHeck:PARameter?

Returns the name of the selected measurement on channel <cnum>.

Return Type String

Overlapped? No

Default Not applicable

SENSe<cnum>:CORRection:COLLect[:ACQuire] <class>[,subclass]

(Write-only) Measures the specified standards from the selected calibration kit. The calibration kit is selected using the Sense:Correction:Collect:CKIT command.

For using two sets of standards, see SENS:CORR:TST.

Note: Before using this command you must select two items:
1. Select a calibration method using SENS:CORR:COLL:METH
2. Select a measurement using CALC:PAR:SEL. You can select one measurement for each channel.

Parameters

<cnum> Any existing channel number. If unspecified, value is set to 1

<class> **Measures the standards associated with these class labels:**
Choose from:

STAN1 S11A and S22A

STAN2 S11B and S22B

STAN3 S11C and S22C

STAN4 S21T and S12T - usually the THRU standard

STAN5 Generic Isolation; not associated with calibration kit definition

ECAL1 through **ECAL8** ECAL modules

SLSET Sets 'sliding load type', and increments the "number of slides" count. The total number of slides is critical to the correct calculation of the sliding load algorithm. See a sliding load cal example.

SLDONE Computes the sliding load using a circle fit algorithm.

[subclass] Optional argument. For mechanical calibration kits, choose from the following to specifying the standard identified in the SENS:CORR:COLL:CKIT:ORDer list to be acquired. If this argument is not used, the default is **SST1**.

SST1 First standard in the order list

SST2 Second standard in the order list

SST3 Third standard in the order list

SST4 Fourth standard in the order list

SST5 Fifth standard in the order list

SST6 Sixth standard in the order list

SST7 Seventh standard in the order list

If an ECAL module (1 through 8) is specified for <class>, choose one of the following for specifying which characterization within the ECal module will be used for the acquire. If not specified, the default is **CHAR0**.

CHAR0 Factory characterization (data that was stored in the ECal module by Agilent)

CHAR1 User characterization #1

CHAR2 User characterization #2

CHAR3 User characterization #3

CHAR4 User characterization #4

CHAR5 User characterization #5

Examples

```
SENS:CORR:COLL STAN1

'If SENS:CORR:COLL:CKIT:ORDer2 5,3,7
was specified, the following command measures standard 3 (the
second in the order list)
sense1:correction:collect:acquire stan3,sst2
SENS:CORR:COLL ECAL4
sense2:correction:collect:acquire ecal2,char1
```

Query Syntax Not applicable

Overlapped? No

Default Not applicable

SENSe<cnum>:CORRection:COLLect:APPLy

(Write-only) Applies error terms to the measurement that is selected using Calc:Par:Select.

Note: Before using this command you must select a measurement using CALC:PAR:SEL. You can select one measurement for each channel.

Note: This command is only necessary if you need to modify error terms. If you do not need to modify error terms, SENSe<cnum>:CORRection:COLLect:SAVE calculates and then automatically applies error terms after you use SENS:CORR:COLL:ACQuire to measure cal standards.

Parameters

<cnum> Any existing channel number. If unspecified, value is set to 1

Example

```
1. CALCulate2:PARameter:SELect S21_2 'select the measurement to
   apply terms to

2. SENSe2:CORRection:COLLect:METHod SPARSOLT 'set type of cal
   method.

3. CALCulate2:DATA? SCORR1 'download the error term of interest

4. 'Modify the error term here

5. CALCulate2:DATA SCORR1 'upload the error term of interest

6. SENSe2:CORRection:COLLect:APPLy 'applies the error terms to
   the measurement
```

Query Syntax Not applicable

Overlapped? No

Default Not applicable

SENSe<cnum>:CORRection:COLLect:METHod <char>

(Read-Write) Sets the calibration method (also known as 'Calibration Type' on calibration dialog box.) To set the Cal Type for a Cal Set, use CALC:CORR:TYPE.

Note: Before using this command you must select the measurement to be calibrated using CALC:PAR:SEL. You can select one measurement for each channel.

Parameters

<cnum> Any existing channel number. If unspecified, value is set to 1

<char> Choose from:

Method	Description
NONE	No Cal method
REFL1OPEN	Response Open
REFL1SHORT or **REFL1**	Response Short
REFL3	Full 1 port
TRAN1	Response Thru
TRAN2	Response Thru and Isolation
SPARSOLT	Full SOLT 2 port
SPARTRL	TRL Cal (not supported on all PNAs.)

Examples

```
SENS:CORR:COLL:METH REFL1
sense2:correction:collect:method sparsolt
```

Query Syntax SENSe<cnum>:CORRection:COLLect:METHod?

Return Type Character

Overlapped? No

Default Not Applicable

SENSe<cnum>:CORRection:COLLect:SAVE

(Write-only) Calculates the error terms using the selected :METHod and applies the error terms to the selected measurement (turns error correction ON.) Does NOT save the calibration error-terms.

Do NOT use this command during an ECAL.

Note: Before using this command you must select a measurement using CALC:PAR:SEL. You can select one measurement for each channel.

Parameters	
<cnum>	Any existing channel number. If unspecified, value is set to 1
Examples	**`SENS:CORR:COLL:SAVE`** **`sense2:correction:collect:save`**
Query Syntax	Not applicable
Overlapped?	No
Default	Not applicable

SENSe<cnum>:CORRection:EXTension:PORT<pnum>[:TIME] <num>

(Read-Write) Sets the extension value at the specified port. Must also set SENS:CORR:EXT ON.

Note: Before using this command you must select a measurement using CALC:PAR:SEL. You can select one measurement for each channel.

Parameters	
<cnum>	Any existing channel number. If unspecified, value is set to 1
<pnum>	Number of the port that will receive the extension. If unspecified, value is set to 1. Choose from: **1** for Port 1 **2** for Port 2
<num>	The port extension in seconds; may include suffix. Choose a number between: **-10** and **10**
Examples	**`SENS:CORR:EXT:PORT 2MS`** **`sense2:correction:extension:port2 .00025`**
Query Syntax	SENSe<cnum>:CORRection:EXTension:PORT<pnum> [:TIME]?
Return Type	Numeric
Overlapped?	No
Default	0

SENSe<cnum>:CORRection:EXTension:RECeiver<Rnum>[:TIME] <num>

(Read-Write) Sets the extension value at the specified receiver. Must also set SENS:CORR:EXT ON.

Note: Before using this command you must select a measurement using CALC:PAR:SEL. You can select one measurement for each channel.

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1
<Rnum>	Number of the receiver that will receive the extension. If unspecified, value is set to 1 Choose from: **1** for Receiver A **2** for Receiver B
<num>	The electrical length in seconds; may include suffix. Choose a number between: **-10** and **10**
Examples	`SENS:CORR:EXT:REC 2MS` `sense2:correction:extension:receiver2:time .00025`
Query Syntax	SENSe<cnum>:CORRection:EXTension:RECeiver<Rnum> [:TIME]?
Return Type	Numeric
Overlapped?	No
Default	0

SENSe<cnum>:CORRection:EXTension[:STATe] <ON | OFF>

(Read-Write) Turns port extensions ON or OFF.

Note: Before using this command you must select a measurement using CALC:PAR:SEL. You can select one measurement for each channel.

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1
<ON | OFF>	**ON** (or 1) - turns port entensions ON. **OFF** (or 0) - turns port extensions is OFF.
Examples	`SENS:CORR:EXT ON` `sense2:correction:extension:state off`
Query Syntax	SENSe<cnum>:CORRection:EXTension[:STATe]?
Return Type	Boolean (1 = ON, 0 = OFF)
Overlapped?	No

Default OFF

SENSe:CORRection:IMPedance:INPut:MAGNitude <num>

(Read-Write) Sets and returns the system impedance value for the analyzer.

Parameters

<num> System Impedance value in ohms. Choose any number between 0 and 1000 ohms.

Examples

```
SENS:CORR:IMP:INP:MAGN 75
sense:correction:impedance:input:magnitude 50.5
```

Query Syntax SENSe:CORRection:IMPedance:
INPut:MAGNitude?

Return Type Numeric

Overlapped? No

Default 50

SENSe<cnum>:CORRection:INTerpolate[:STATe] <ON | OFF>

(Read-Write) Turns correction interpolation ON or OFF.

Note: Before using this command you must select a measurement using CALC:PAR:SEL. You can select one measurement for each channel.

Parameters

<cnum> Any existing channel number. If unspecified, value is set to 1

<ON | OFF> **ON** (or 1) - turns interpolation ON.
OFF (or 0) - turns interpolation OFF.

Examples

```
SENS:CORR:INT ON
sense2:correction:interpolate:state off
```

Query Syntax SENSe<cnum>:CORRection:INTerpolate[:STATe]?

Return Type Boolean (1 = ON, 0 = OFF)

Overlapped? No

Default ON

SENSe<cnum>:CORRection:ISOLation[:STATe] <ON | OFF>

(Read-Write) Turns isolation cal ON or OFF during Full 2-port calibration. If this command is not sent, the default state is to **disable** Isolation.

Note: This command is ignored for ECal on PNA release 4.26 and greater. Isolation is never performed with ECal as the inherent isolation of the PNA is better than that achieved with ECal.

Parameters	
<cnum>	Any existing channel number. If unspecified, value is set to 1
<ON | OFF>	**ON** (or 1) - turns isolation ON. **OFF** (or 0) - turns isolation OFF.
Examples	**`SENS:CORR:ISOL ON`** **`sense2:correction:isolation:state off`**
Query Syntax	SENSe<cnum>:CORRection:ISOLation[:STATe]?
Return Type	Boolean (1 = ON, 0 = OFF)
Overlapped?	No
Default	OFF - (Isolation disabled)

SENSe:CORRection:PREFerence:ECAL:ORIentation[:STATe] <ON|OFF>

(Read-Write) Specifies whether or not the PNA should perform orientation of the ECal module during calibration. Orientation is a technique by which the PNA automatically determines which ports of the module are connected to which ports of the PNA. Orientation begins to fail at very low power levels or if there is much attenuation in the path between the PNA and the ECal module. If orientation is turned OFF, the SENS:CORR:PREF:ECAL:PMAP command must be used to specify the port connections before performing a cal.

Note: 3-port calibration with a 2-port ECal module does not fully support the mode of orientation = OFF.

Examples	**`SENS:CORR:PREF:ECAL:ORI OFF`** **`sense:correction:preference:ecal:orientation:state on`**
Query Syntax	SENSe:CORRection:PREFerence:ECAL:ORIentation[:STATe]?
Return Type	Boolean (1 = ON, 0 = OFF)
Overlapped?	No
Default	ON (1)

SENSe:CORRection:PREFerence:ECAL:PMAP <module>,<string>

(Read-Write) When ECal module orientation is turned OFF (SENS:CORR:PREF:ECAL:ORI OFF), this command specifies the port mapping (which ports of the module are connected to which ports of the PNA) prior to performing ECal calibrations.

Parameters

<module>	Specifies which ECal module this port map is being applied to. Choose from: **ECAL1** .through. **ECAL8**
<string>	Format this parameter in the following manner: Aw,Bx,Cy,Dz where • A, B, C, and D are literal ports on the ECAL module • w,x,y, and z are substituted for PNA port numbers to which the ECAL module port is connected. Ports of the module which are not used are omitted from the string. For example, on a 4-port ECal module with port A connected to PNA port 2 port B connected to PNA port 3 port C not connected port D connected to PNA port 1 the string would be: A2,B3,D1 If either the receive port or source port (or load port for 2-port cal) of the CALC:PAR:SELected measurement is not in this string and orientation is OFF, an attempt to perform an ECal calibration will fail.
Examples	`SENS:CORR:PREF:ECAL:PMAP ECAL2, 'A1,B2'` `sense:correction:preference:ecal:pmap ecal3, 'a2,b1,c3'`
Query Syntax	SENSe:CORRection:PREFerence:ECAL:PMAP? <module>
Return Type	String
Overlapped?	No
Default	Null string ()

SENSe<cnum>:CORRection:RVELocity:COAX <num>

(Read-Write) Sets the velocity factor to be used with Electrical Delay and Port Extensions.

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1
<num>	Velocity factor. Choose a number between **0** and **10** (.66 polyethylene dielectric; .7 teflon dielectric)
Examples	**`SENS:CORR:RVEL:COAX .66`** **`sense2:correction:rvelocity:coax .70`**
Query Syntax	SENSe<cnum>:CORRection:RVELocity:COAX?
Return Type	Numeric
Overlapped?	No
Default	1

SENSe:CORRection:SFORward[:STATe] <boolean>

(Read-Write) Sets the direction a calibration will be performed when only one set of standards is used.

Use SENSe:CORRection:TSTandards[:STATe] **OFF** to specify that only one set of standards will be used.

Parameters

<boolean>	**ON (1)** - FORWARD direction of a 2-port calibration will be performed. **OFF (0)** - REVERSE direction of a 2-port calibration will be performed.
Examples	**`SENS:CORR:SFOR 1`** **`sense:correction:sforward:state 0`** **See an example using this command**
Query Syntax	SENSe:CORRection:SFORward[:STATe]?
Return Type	Boolean
Overlapped?	No
Default	ON

SENSe<cnum>:CORRection[:STATe] <ON | OFF>

(Read-Write) Specifies whether or not correction data is applied to the measurement.

Note: Before using this command you must select a measurement using CALC:PAR:SEL. You can select one measurement for each channel.

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1
<ON | OFF>	**ON** (or 1) - correction is applied to the measurement. **OFF** (or 0) - correction is NOT applied to the measurement.
Examples	**`SENS:CORR ON`** **`sense2:correction:state off`**
Query Syntax	SENSe<cnum>:CORRection[:STATe]?
Return Type	Boolean (1 = ON, 0 = OFF)
Overlapped?	No
Default	OFF

SENSe:CORRection:TSTandards[:STATe] <boolean>

(Read-Write) Specifies the acquisition of calibration data using TWO set of standards or ONE.

Parameters

<boolean>	**ON (1)** - TWO sets of standards will be used for full 2-port calibration for both Forward and Reverse parameters. **OFF (2)** - ONE set of standards will be used for full 2-port calibration. Set SENSe:CORRection:COLLect:SFORward[:STATe] to **ON** for the forward acquisitions and **OFF** for the reverse acquisitions. When specifying ON (two sets), the SENS:CORR:COLL:ACQuire command uses the same standard index for each calibration class. To specify the calibration standard gender for each port, you must first ensure that the order of calibration class accurately reflects the configuration of your DUT. For example, for a DUT with a male connector on port 1 and a female connector on port 2, order the devices within the S11 classes (A, B, and C) such that the MALE standards are first in the list. Then order the S22 classes specifying the FEMALE standards as the first in the list.
Examples	**`SENS:CORR:TST 1`** **`sense:correction:tstandard:state 0`** **`See an example using this command`**
Query Syntax	SENSe:CORRection:TSTandards[:STATe]?
Return Type	Boolean
Overlapped?	No

Default ON

SENSe:CORRection:TYPE:CATalog? <char>

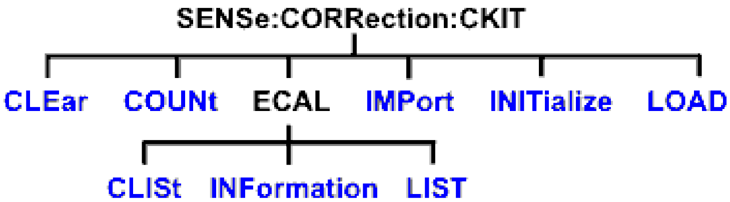
(Read-Write) Lists the Cal Types in the PNA by either GUID or registered name. **Critical Note:**

Parameters

<char>	Specifies the type of list. Choose from: **GUID -** the registered GUID of the Cal Type **NAME -** the registered name of the Cal Type
Examples	`SENS:CORR:TYPE:CAT? GUID`
Query Syntax	SENSe<cnum>:CORRection:TYPE:CATalog? <char>
Return Type	Comma-separated string
Overlapped?	No
Default	Not Applicable

Sense:Correction:CKIT Commands

Manages the list of cal kits that are installed in the PNA.



- Click on a blue keyword to view the command details.
- New See Calibrating the PNA Using SCPI
- See a List of all commands in this block.
- Learn about Modifying Cal Kits

SENSe:CORRection:CKIT:CLEar[:IMMediate]

(Write-only) Deletes ALL installed cal kits.

Parameters	
<cnum>	Any existing channel number. If unspecified, value is set to 1
Examples	**`SENS:CORR:CKIT:CLE`**
Query Syntax	Not Applicable
Overlapped?	No
Default	Not Applicable

SENSe:CORRection:CKIT:COUNt?

(Read-only) Returns the number of installed cal kits.

Parameters	
<cnum>	Any existing channel number. If unspecified, value is set to 1
Examples	**`SENS:CORR:CKIT:COUNt?`**
Query Syntax	SENS:CORR:CKIT:COUNt?
Return Type	Numeric
Overlapped?	No
Default	Not Applicable

SENSe:CORRection:CKIT:ECAL<mod>:CLISt?

(Read-only) Returns a list of characterizations stored in the specified ECal module.

Parameters	
<mod>	ECal module from which to read user characterization numbers. If unspecified, value is set to 1.
Examples	**`Module 1 contains User Characterizations 1 and 3.`** **`SENSe:CORRection:CKIT:ECAL:CLIST?`** **`'Returns the following (0 always indicates the factory characterization):`** **`0,1,3`**
Return Type	Numeric list, separated by commas.
Overlapped?	No
Default	Not Applicable

SENSe:CORRection:CKIT:ECAL<mod>:INFormation? [<char>]

(Read-only) Reads the user-characterization information from the specified ECal module. This command returns the same values as SENS:CORR:COLL:CKIT:INF?

Parameters

<mod> ECal module to read characterizations from. Choose from:

1 through 8. If unspecified, value is set to 1.

<char> Optional argument. Specifies which characterization within the ECal module to read information from. If not specified, value is set to CHAR0.

Choose from:

- CHAR0 Factory characterization (data that was stored in the ECal module by Agilent)
- CHAR1 User characterization #1
- CHAR2 User characterization #2
- CHAR3 User characterization #3
- CHAR4 User characterization #4
- CHAR5 User characterization #5

Examples

```
SENS:CORR:CKIT:ECAL2:INFormation? char5
'Example return string:
ModelNumber: 85092-60007, SerialNumber: 01386, ConnectorType:
N5FN5F, PortAConnector: Type N (50) female, PortBConnector: Type
N (50) female, MinFreq: 30000, MaxFreq: 9100000000,
NumberOfPoints: 250, Calibrated: July 4 2002
```

Return Type Character

Overlapped? No

Default Not Applicable

SENSe:CORRection:CKIT:ECAL:LIST?

(Read-only) Returns a list of index numbers to be used for referring to the ECal modules that are currently attached to the PNA.

Examples

```
SENS:CORR:CKIT:ECAL:LIST?
'If 2 modules are attached to the PNA
'then the returned list will be:
1,2
```

Return Type Numeric list, separated by commas.

Overlapped? No

Default Not Applicable

SENSe:CORRection:CKIT:IMPort <string>

(Write-only) Imports the specified cal kit (.ckt file) and appends the imported kit to the end of the list of kits whenever the file import succeeds.

Parameters

<cnum> Any existing channel number. If unspecified, value is set to 1.

<string> Path and cal kit name.

Examples

```
SENSe2:CORRection:CKIT:IMPort "C:\Program Files\Agilent\Network
Analyzer\Documents\85033D.ckt"
```

Query Syntax Not Applicable

Overlapped? No

Default Not Applicable

SENSe:CORRection:CKIT:INITialize[:IMMediate]

(Write-only) Restores all default factory installed cal kits.

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1
Examples	`SENS:CORR:CKIT:INITialize` `sense2:correction:ckit:initialize`
Query Syntax	Not Applicable
Overlapped?	No
Default	Not Applicable

SENSe:CORRection:CKIT:LOAD <string>

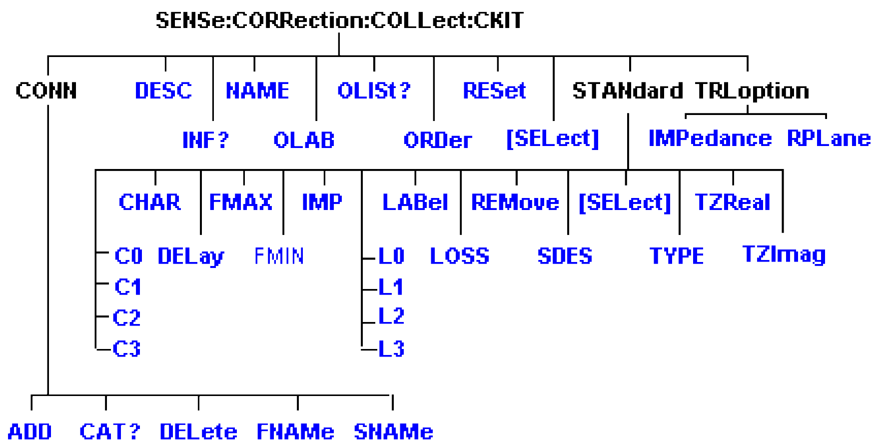
(Write-only) Loads the specified collection of cal kits from a .wks file. You can make your own collection of cal kits from the Advanced Modify Cal Kit menu.

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1
<string>	Path and file name of the cal kit collection.
Examples	`sense2:correction:ckit:load "C:\Program Files\Agilent\Network Analyzer\PnaCalKits\factory\MyCalKits.wks`
Query Syntax	Not Applicable
Overlapped?	No
Default	Not Applicable

Sense:Correction:Collect:CKit Commands

Use to change the definitions of calibration kit standards.



Click on a blue keyword to view the command details.

Most of these commands act on the currently selected standard from the currently selected calibration kit.

- New See Calibrating the PNA Using SCPI
- To select a Calibration kit, use SENS:CORR:COLL:CKIT:SEL.
- To select a Calibration standard, use SENS:CORR:COLL:CKIT:STAN:SEL
- See a List of all commands in this block.
- See an **example** program that CREATES a New Cal Kit
- See an **example** program that MODIFIES an Existing Cal Kit
- Learn about Modifying Cal Kits

Note: You should provide data for every definition field - for every standard in your calibration kit. If a field is not set, the default value may not be what you expect.

SENSe:CORRection:COLLect:CKIT:CONNector:ADD <family>,<start>,<stop>,<z0>,<gender>,<media>,<cutoff>

(Write only) Creates a new connector. The connector is automatically added to the list of available connectors for the currently selected cal kit. If a connector includes both male and female connectors, each connector must be added separately.

Parameters

<family>	(String) Name of connector family. Limited to 50 characters.
<start>	Start frequency
<stop>	Stop frequency
<z0>	Characteristic Impedance of the connector in ohms.
<gender>	Connector gender. Choose from: MALE FEMALE NONE
<media>	Media of the connector. Choose from: **COAX** - coaxial **WAVE** - waveguide
<cutoff>	Cutoff frequency of the connector (waveguide only).
Examples	**SENS:CORR:COLL:CKIT:CONN:ADD "PSC 1.8 mm",0 HZ,999.9 GHZ,50,FEMALE,COAX,0.0** **SENS:CORR:COLL:CKIT:CONN:ADD "PSC 1.8 mm",0 HZ,999.9 GHZ,50,MALE,COAX,0.0**
Query Syntax	Not applicable
Overlapped?	No
Default	Not Applicable

SENSe:CORRection:COLLect:CKIT:CONNector:CATalog?

(Read-only) Returns a comma-separated list of all connectors defined within the currently selected cal kit. The returned string includes the connector family name followed by the connector gender, if any. Kits may include a primary connector family name and additional connector family names.

Connector family names are case sensitive. A connector family named "PSC 2.4" is different from a connector family named "psc 2.4".

Learn more about Connector Family Name

Parameters	None
Examples	**`SENS:CORR:COLL:CKIT:CONN:CAT?`** **`'Returned string`** **`"Type-N (50) male, Type-N (50) female"`**
Overlapped?	No
Default	Not Applicable

SENSe:CORRection:COLLect:CKIT:CONNector:DELete

(Write-only) Deletes the primary connector family name from the selected kit. The PNA allows multiple connector families for each kit. If a kit includes multiple connector families, only the first listed (primary) connector family name is deleted.

Once the connector family is deleted, the connector may not be assigned to any new or existing standard within the kit.

The previously defined standards retain their association to the deleted connector name. To reassign standards to a new connector family name, use SENS:CORR:COLL:CKIT:CONN:SNAMe.

Parameters	None
Examples	**`SENS:CORR:COLL:CKIT:CONN:DEL`**
Query Syntax	Not Applicable
Overlapped?	No
Default	Not Applicable

SENSe:CORRection:COLLect:CKIT:CONNector:FNAME <name>

(Read-Write) Replaces the primary connector family name from the selected kit with a new connector family name. The connector family name is replaced in all standards in the kit that share that name. The PNA allows multiple connector families for each kit. If a kit includes multiple connector families, only the first listed (primary) connector family name is replaced. Use the query form of this command to return the name of the primary connector family.

Parameters	
<name>	New connector family name. Limited to 50 characters.
Examples	`SENS:CORR:COLL:CKIT:CONN:FNAM 'MYPSC35'` `Sense:correction:collect:ckit:connector:name 'My Type N'`
Query Syntax	SENSe:CORRection:COLLect:CKIT:CONNector:FNAMe?
Return Type	String
Overlapped?	No
Default	Not Applicable

SENSe:CORRection:COLLect:CKIT:CONNector:SNAMe <family>,<gender>,<port>

(Read-Write) Assigns a family name to the currently selected standard from the currently selected kit. Specify each port of a 2-port standard individually. Use the query form of this command to read the connector family name assigned to the current standard. The name is not assigned unless the connector family name is previously defined within the selected kit.

Parameters	
<family>	String. Connector family name.
<gender>	Connector gender. Choose from: MALE FEMALE NONE
<port>	Number of the connector port to be assigned the connector family name. 2-port standards such as a thru line must be assigned separately. It is not relevant which connector is port 1 or port 2. **1** Specifies a 1-port standard or the first port of a 2-port standard. **2** Specifies the second port of a 2-port standard.
Examples	`SENS:CORR:COLL:CKIT:CONN:SNAMe "Type-N (50)",MALE,1`
Query Syntax	SENSe:CORRection:COLLect:CKIT:CONNector:SNAMe?
Return Type	String
Overlapped?	No
Default	Not Applicable

SENSe:CORRection:COLLect:CKIT:DESCription <string>

(Read-Write) Modifies the cal kit description field of the selected kit. This description appears in the Edit PNA Cal Kit dialog box.

Parameters

<string>	Description of the cal kit. Limited to 50 characters.
Examples	`SENS:CORR:COLL:CKIT:DESC "My New CalKit"`
Query Syntax	SENSe:CORRection:COLLect:CKIT:DESCription?
Return Type	String
Overlapped?	No
Default	Not Applicable

SENSe:CORRection:COLLect:CKIT:INFormation? <module>[,char]

(Read Only) Reads characterization information from an ECal module.

Parameters

<module> Specifies which ECal module to read from. Choose from:

ECAL1

.through.

ECAL8

[char] Optional argument.

Specifies which characterization within the ECal module to read information from. If this argument is not used, the default is **CHAR0**. **CHAR1** through **CHAR5** are for user characterizations that may have been written to the module by the User Characterization feature on the PNA. Choose from:

CHAR0	Factory characterization (data that was stored in the ECal module by Agilent)
CHAR1	User characterization #1
CHAR2	User characterization #2
CHAR3	User characterization #3
CHAR4	User characterization #4
CHAR5	User characterization #5

Examples

```
SENS:CORR:COLL:CKIT:INF? ECAL4
sense:correction:collect:ckit:information? ecal2,char1
```

Example return string:

```
ModelNumber: 85092-60007, SerialNumber: 01386, ConnectorType:
N5FN5F, PortAConnector: Type N (50) female, PortBConnector: Type
N (50) female, MinFreq: 30000, MaxFreq: 9100000000,
NumberOfPoints: 250, Calibrated: July 4 2002
```

Return Type	Character
Overlapped?	No
Default	Not Applicable

SENSe:CORRection:COLLect:CKIT:NAME <name>

(Read-Write) Sets a name for the selected calibration kit.

Parameters

<name>	Calibration Kit name. Any string name, can include numerics, period, and spaces; any length (although the dialog box display is limited to about 30 characters).

Examples

```
SENS:CORR:COLL:CKIT:NAME 'MYAPC35'
sense:correction:collect:ckit:name 'mytypen'
```

Query Syntax	SENSe:CORRection:COLLect:CKIT:NAME?
Return Type	String
Overlapped?	No
Default	Not Applicable

SENSe:CORRection:COLLect:CKIT:OLABel<class> <name>

(Read-Write) Sets the label for the calibration class designed by <class>. The label is used in the prompts for connecting the calibration standards associated with that <class>.

Parameters

<class>	Number of the calibration class. Choose a number between: 1 and 18. The <class> numbers are associated with the following calibration Classes:

	Class	Description
1	S11A	Reflection standard
2	S11B	Reflection standard
3	S11C	Reflection standard
4	S21T	Thru/Delay standard
5	S22A	Reflection standard
6	S22B	Reflection standard
7	S22C	Reflection standard
8	S12T	Thru/Delay standard
	3-port analyzers only	
9	S33A	Reflection standard
10	S33B	Reflection standard
11	S33C	Reflection standard
12	S32T	Thru/Delay standard
13	S23T	Thru/Delay standard
14	S31T	Thru/Delay standard
15	S13T	Thru/Delay standard
	TRL Calibrations	
16	TRL "T"	Thru standard
17	TRL "R"	Reflect standard
18	TRL "L"	Line standard

<name> Label for the calibration class. Must be enclosed in quotes. Any string between 1 and 12 characters long. Cannot begin with a numeric.

Examples

```
SENS:CORR:COLL:CKIT:OLAB3 'LOADS'
sense:correction:collect:ckit:olabel4 'Thru'
```

Return Type String

Overlapped? No

Default Not Applicable

SENSe:CORRection:COLLect:CKIT:OLIST[class]?

(Read-only) Returns seven values of standards that are assigned to the specified class.

Parameters

<class> Number of the calibration class to be queried. The <class> numbers are associated with the following calibration Classes:

	Class	Description
1	S11A	Reflection standard
2	S11B	Reflection standard
3	S11C	Reflection standard
4	S21T	Thru/Delay standard
5	S22A	Reflection standard
6	S22B	Reflection standard
7	S22C	Reflection standard
8	S12T	Thru/Delay standard
	3-port analyzers only	
9	S33A	Reflection standard
10	S33B	Reflection standard
11	S33C	Reflection standard
12	S32T	Thru/Delay standard
13	S23T	Thru/Delay standard
14	S31T	Thru/Delay standard
15	S13T	Thru/Delay standard
	TRL Calibrations	
16	TRL "T"	Thru standard
17	TRL "R"	Reflect standard
18	TRL "L"	Thru standard

Examples

SENS:CORR:COLL:CKIT:OLIST8?
Always returns 7 standard numbers. Unassigned standards return 0

Return Type Numeric; returns the <class> number of the selected standard.

Overlapped? No

Default Not Applicable

SENSe:CORRection:COLLect:CKIT:ORDer<class> <std> [,<std>] [,<std>] [,<std>] [,<std>]

[,<std>] [,<std>]

(Read-Write) Sets a standard number to a calibration class. Does **NOT** set or dictate the order for measuring the standards. For more information, see Assigning Standards to a Calibration Class

Parameters

<class> Number of the calibration class that is assigned to <standard>. Choose a number between: **1** and **18.** The <class> numbers are associated with the following calibration Classes:

	Class	Description
1	S11A	Reflection standard
2	S11B	Reflection standard
3	S11C	Reflection standard
4	S21T	Thru/Delay standard
5	S22A	Reflection standard
6	S22B	Reflection standard
7	S22C	Reflection standard
8	S12T	Thru/Delay standard
	3-port analyzers only	
9	S33A	Reflection standard
10	S33B	Reflection standard
11	S33C	Reflection standard
12	S32T	Thru/Delay standard
13	S23T	Thru/Delay standard
14	S31T	Thru/Delay standard
15	S13T	Thru/Delay standard
	TRL Calibration	
16	TRL "T"	Thru standard
17	TRL "R"	Reflect standard
18	TRL "L"	Line standard

<std>	Standard number to be assigned to the class; Choose a standard between 1 and 8. One standard is mandatory; up to six additional standards are optional.
Examples	Assigns standard 3 to S11A class: **`SENS:CORR:COLL:CKIT:ORD1 3`** Assigns standard 2 and 5 to S21T class class: **`sense:correction:collect:ckit:order4 2,5`**
Query Syntax	SENSe:CORRection:COLLect:CKIT:ORDer<class>? 'Returns only the first standard assigned to the specified class. To query the remaining standards, use SENSe:CORRection:COLLect:CKIT:OLIST[1-15]?
Return Type	Numeric
Overlapped?	No
Default	Not Applicable

SENSe:CORRection:COLLect:CKIT:RESet <num>

(Write-only) Obsolete in Rev 3.2. Instead use Sens:Corr:Ckit:Init.

Resets the selected calibration kit to factory default definition values.

Parameters	
<num>	The number of the calibration kit to be reset. Choose any integer between: **1** and **8**
Examples	**`SENS:CORR:COLL:CKIT:RESet 1`** **`sense:correction:collect:ckit:reset 4`**
Query Syntax	Not Applicable
Overlapped?	No
Default	Not Applicable

SENSe:CORRection:COLLect:CKIT[:SELect] <num>

(Read-Write) Selects (makes active) a calibration kit for **performing** a calibration or for **modifying** standards. All subsequent "CKIT" commands that are sent apply to this selected calibration kit. Select a calibration standard using SENS:CORR:COLL:CKIT:STAN <num>

Parameters

<num> The number of the calibration kit. Choose from:

Use SENSe:CORRection:COLLect:CKIT:RESet to restore Cal Kits to default values.

	Name
1	User Defined 1
2	User Defined 2
3	User Defined 3
48	User Defined 48
49	User Defined 49
50	User Defined 50
99	ECAL module OBSOLETE - This argument is no longer required. Specify the use of ECal using only Sens:Corr:Coll:Acquire. See example.

Examples

```
SENS:CORR:COLL:CKIT 2
sense2:correction:collect:ckit:select 7
```

Query Syntax SENSe:CORRection:COLLect:CKIT?

Return Type Numeric

Overlapped? No

Default Last kit selected

SENSe:CORRection:COLLect:CKIT:STANdard:C0 <num>

(Read-Write) Sets the C0 value (the first capacitance value) for the selected standard.

Parameters

<num>	Value for C0 in picofarads
Examples	**The following commands set C0=15 picofarads:** **`SENS:CORR:COLL:CKIT:STAN:C0 15`** **`sense:correction:collect:ckit:standard:c0 15`**
Query Syntax	SENSe:CORRection:COLLect:CKIT:STANdard:C0?
Return Type	Numeric
Overlapped?	No
Default	Not Applicable

SENSe:CORRection:COLLect:CKIT:STANdard:C1 <num>

(Read-Write) Sets the C1 value (the second capacitance value) for the selected standard.

Parameters

<num>	Value for C1 in picofarads
Examples	The following two commands set C1=15 picofarads: **`SENS:CORR:COLL:CKIT:STAN:C1 15`** **`sense:correction:collect:ckit:standard:c1 15`**
Query Syntax	SENSe:CORRection:COLLect:CKIT:STANdard:C1?
Return Type	Numeric
Overlapped?	No
Default	Not Applicable

SENSe:CORRection:COLLect:CKIT:STANdard:C2 <num>

(Read-Write) Sets the C2 value (the third capacitance value) for the selected standard.

Parameters

<num>	Value for C2 in picofarads
Examples	The following two commands set C2=(-15) picofarads: **`SENS:CORR:COLL:CKIT:STAN:C2 -15`** **`sense:correction:collect:ckit:standard:c2 -15`**
Query Syntax	SENSe:CORRection:COLLect:CKIT:STANdard:C2?
Return Type	Numeric
Overlapped?	No
Default	Not Applicable

SENSe:CORRection:COLLect:CKIT:STANdard:C3 <num>

(Read-Write) Sets the C3 value (the fourth capacitance value) for the selected standard.

Parameters

<num>	Value for C3 in picofarads
Examples	The following two commands set C3=15 picofarads: **`SENS:CORR:COLL:CKIT:STAN:C3 15`** **`sense:correction:collect:ckit:standard:c3 15`**
Query Syntax	SENSe:CORRection:COLLect:CKIT:STANdard:C3?
Return Type	Numeric
Overlapped?	No
Default	Not Applicable

SENSe:CORRection:COLLect:CKIT:STANdard:CHARacter <char>

(Read-Write) Sets the media type of the selected calibration standard.

Parameters

<char>	Media type of the standard. Choose from: **Coax** - Coaxial Cable **Wave** - Waveguide
Examples	`SENS:CORR:COLL:CKIT:STAN:CHAR COAX` `sense:correction:collect:ckit:standard:character wave`
Query Syntax	SENSe:CORRection:COLLect:CKIT:STANdard:CHARacter?
Return Type	Numeric
Overlapped?	No
Default	Coax

SENSe:CORRection:COLLect:CKIT:STANdard:DELay <num>

(Read-Write) Sets the electrical delay value for the selected standard.

Parameters

<num>	Electrical delay in seconds
Examples	`SENS:CORR:COLL:CKIT:STAN:DEL 50e-12` `sense2:correction:collect:ckit:standard:delay 50ps`
Query Syntax	SENSe:CORRection:COLLect:CKIT:STANdard:DELay?
Return Type	Numeric
Overlapped?	No
Default	Not Applicable

SENSe:CORRection:COLLect:CKIT:STANdard:FMAX <num>

(Read-Write) Sets the maximum frequency for the selected standard.

Parameters

<num>	Maximum frequency in Hertz.
Examples	**SENS:CORR:COLL:CKIT:STAN:FMAX 9e9** **sense:correction:collect:ckit:standard:fmax 9Ghz**
Query Syntax	SENSe:CORRection:COLLect:CKIT:STANdard:FMAX?
Return Type	Numeric
Overlapped?	No
Default	Not Applicable

SENSe:CORRection:COLLect:CKIT:STANdard:FMIN <num>

(Read-Write) Sets the minimum frequency for the selected standard.

Parameters

<num>	Minimum frequency in Hertz.
Examples	**SENS:CORR:COLL:CKIT:STAN:FMIN 1e3** **sense:correction:collect:ckit:standard:fmin 1khz**
Query Syntax	SENSe:CORRection:COLLect:CKIT:STANdard:FMIN?
Return Type	Numeric
Overlapped?	No
Default	Not Applicable

SENSe:CORRection:COLLect:CKIT:STANdard:IMPedance <num>

(Read-Write) Sets the characteristic impedance for the selected standard.

Parameters

<num>	Impedance in Ohms
Examples	**`SENS:CORR:COLL:CKIT:STAN:IMP 75`** **`sense:correction:collect:ckit:standard:impedance 50.3`**
Query Syntax	SENSe:CORRection:COLLect:CKIT:STANdard:IMPedance?
Return Type	Numeric
Overlapped?	No
Default	50

SENSe:CORRection:COLLect:CKIT:STANdard:L0 <num>

(Read-Write) Sets the L0 value (the first inductance value) for the selected standard.

Parameters

<num>	Value for L0 in picohenries
Examples	The following two commands set L0=15 picohenries: **`SENS:CORR:COLL:CKIT:STAN:L0 15`** **`sense:correction:collect:ckit:standard:l0 15`**
Query Syntax	SENSe:CORRection:COLLect:CKIT:STANdard:L0?
Return Type	Numeric
Overlapped?	No
Default	Not Applicable

SENSe:CORRection:COLLect:CKIT:STANdard:L1 <num>

(Read-Write) Sets the L1 value (the second inductance value) for the selected standard.

Parameters

<num>	Value for L1 in picohenries
Examples	The following two commands set L1=15 picohenries: **`SENS:CORR:COLL:CKIT:STAN:L1 15`** **`sense:correction:collect:ckit:standard:l1 15`**
Query Syntax	SENSe:CORRection:COLLect:CKIT:STANdard:L1?
Return Type	Numeric
Overlapped?	No
Default	Not Applicable

SENSe:CORRection:COLLect:CKIT:STANdard:L2 <num>

(Read-Write) Sets the L2 value (the third inductance value) for the selected standard.

Parameters

<num>	Value for L2 in picohenries
Examples	The following two commands set L2=15 picohenries: **`SENS:CORR:COLL:CKIT:STAN:L2 15`** **`sense:correction:collect:ckit:standard:l2 15`**
Query Syntax	SENSe:CORRection:COLLect:CKIT:STANdard:L2?
Return Type	Numeric
Overlapped?	No
Default	Not Applicable

SENSe:CORRection:COLLect:CKIT:STANdard:L3 <num>

(Read-Write) Sets the L3 value (the fourth inductance value) for the selected standard.

Parameters

<num>	Value for L3 in picohenries

Examples	The following two commands set L3=15 picohenries:
	`SENS:CORR:COLL:CKIT:STAN:L3 15`
	`sense:correction:collect:ckit:standard:l3 15`

Query Syntax	SENSe:CORRection:COLLect:CKIT:STANdard:L3?
Return Type	Numeric
Overlapped?	No
Default	Not Applicable

SENSe:CORRection:COLLect:CKIT:STANdard:LABel <name>

(Read-Write) Sets the label for the selected standard. The label is used to prompt the user to connect the specified standard.

Parameters

<name>	Label for the standard; Must be enclosed in quotes. Any string between **1** and **12** characters long. Cannot begin with a numeric.
Examples	`SENS:CORR:COLL:CKIT:STAN:LAB 'OPEN'` `sense:correction:collect:ckit:standard:label 'Short2'`
Query Syntax	SENSe:CORRection:COLLect:CKIT:STANdard:LABel?
Return Type	String
Overlapped?	No
Default	Not Applicable

SENSe:CORRection:COLLect:CKIT:STANdard:LOSS <num>

(Read-Write) Sets the insertion loss for the selected standard.

Parameters	
<num>	Insertion loss in Mohms / sec. (MegaOhms per second of electrical delay)
Examples	`SENS:CORR:COLL:CKIT:STAN:LOSS 3.5e9` `sense:correction:collect:ckit:standard:loss 3`
Query Syntax	SENSe:CORRection:COLLect:CKIT:STANdard:LOSS?
Return Type	Numeric
Overlapped?	No
Default	Not Applicable

SENSe:CORRection:COLLect:CKIT:STANdard:REMove

(Write only) Deletes the selected standard from the selected cal kit.

Examples	`SENS:CORR:COLL:CKIT:STAN:REMove`
Overlapped?	No
Default	Not Applicable

SENSe:CORRection:COLLect:CKIT:STANdard:SDEScription <string>

(Read-Write) Modifies the description of the selected standard of the selected kit. This description appears in the edit kit dialog box.

Parameters	
<string>	Description of the standard. How long can this be?
Examples	`SENS:CORR:COLL:CKIT:STAN:SDES "My New Standard"`
Query Syntax	SENSe:CORRection:COLLect:STANdard:SDEScription?
Return Type	String
Overlapped?	No
Default	Not Applicable

SENSe:CORRection:COLLect:CKIT:STANdard[:SELECT] <num>

(Read-Write) Selects the calibration standard. All subsequent "CKIT" commands to modify a standard will apply to the selected standard. Select a calibration kit using SENS:CORR:COLL:CKIT:SEL

Parameters	
<num>	Number of the standard. Choose any number between: **1** and **30**
Examples	`SENS:CORR:COLL:CKIT:STAN 3` `sense:correction:collect:ckit:standard:select 8`
Query Syntax	SENSe:CORRection:COLLect:CKIT:STANdard[:SELect]?
Return Type	Numeric
Overlapped?	No
Default	1

SENSe:CORRection:COLLect:CKIT:STANdard:TYPE <char>

(Read-Write) Sets the type for the selected standard.

Parameters	
<char>	Choose from: **OPEN** **SHORT** **LOAD** **SLOAD** (sliding load) **THRU** (through) **ARBI**(arbitrary)
Examples	`SENS:CORR:COLL:CKIT:STAN:TYPE LOAD` `sense:correction:collect:ckit:standard:type short`
Query Syntax	SENSe:CORRection:COLLect:CKIT:STANdard:TYPE?
Return Type	Character
Overlapped?	No
Default	Not Applicable

SENSe:CORRection:COLLect:CKIT:STANdard:TZReal <num>

(Read-Write) Sets the TZReal component value of the Terminal Impedance for the selected standard.

Note: Only applicable when the Standard Type is set to **ARBI**

Parameters

<num> Value for TZReal in Ohms

Examples

The following commands set TZReal=15 Ohms:

```
SENS:CORR:COLL:CKIT:STAN:TZReal 15
sense:correction:collect:ckit:standard:TZReal 15
```

Query Syntax SENSe:CORRection:COLLect:CKIT:STANdard:TZReal?

Return Type Numeric

Overlapped? No

Default Not Applicable

SENSe:CORRection:COLLect:CKIT:STANdard:TZImag <num>

(Read-Write) Sets the TZImag component value of the Terminal Impedance for the selected standard.

Note: Only applicable when the Standard Type is set to **ARBI**

Parameters

<num> Value for TZImag in Ohms

Examples

The following two commands set TZImag=15 Ohms:

```
SENS:CORR:COLL:CKIT:STAN:TZImag 15
sense:correction:collect:ckit:standard:TZImag 15
```

Query Syntax SENSe:CORRection:COLLect:CKIT:STANdard:TZImag?

Return Type Numeric

Overlapped? No

Default Not Applicable

SENSe:CORRection:COLLect:CKIT:TRLoption:IMPedance <char>

(Read-Write) Determines how the reference impedance is set for the TRL calibration.

Parameters

<char> **LINE** Specifies that the characteristic impedance of the LINE standard should be used as the system impedance. This ignores any difference between Offset Z0, Offset Loss and system Z0.

SYSTEM Transforms the LINE standard impedance and loss to that of the system impedance for use with the calibration error terms. The TRL calibration will first compute the error terms assuming the "Line" standard impedance is the system's characteristic impedance (same as LINE option), then modify the error terms to include the impedance transformation. This should only be used with coax since the skin effect model used is a coaxial model.

The System Impedance may be changed if necessary using SENS:CORR:IMP:INP:MAGN.

Examples

```
SENS:CORR:COLL:CKIT:TRL:IMP LINE
sense:correction:collect:ckit:trloption:impedance system
```

Query Syntax SENSe:CORRection:COLLect:CKIT:TRLoption:IMPedance?

Return Type Character

Overlapped? No

Default SYST

SENSe:CORRection:COLLect:CKIT:TRLoption:RPLane <char>

(Read-Write) Specifies the standard to be used to establish the position of the measurement reference plane.

Parameters

<char> Standard used to establish the position of the measurement reference plane. Choose from:

THRU Select if the THRU standard is zero-length or very short.

REFLect Select if the THRU standard is not appropriate AND the delay of the REFLECT standard is well defined.

Examples

```
SENS:CORR:COLL:CKIT:TRL:RPL REFL
sense:correction:collect:ckit:trloption:rplane thru
```

Query Syntax SENSe:CORRection:COLLect:CKIT:TRLoption:RPLane?

Return Type Character

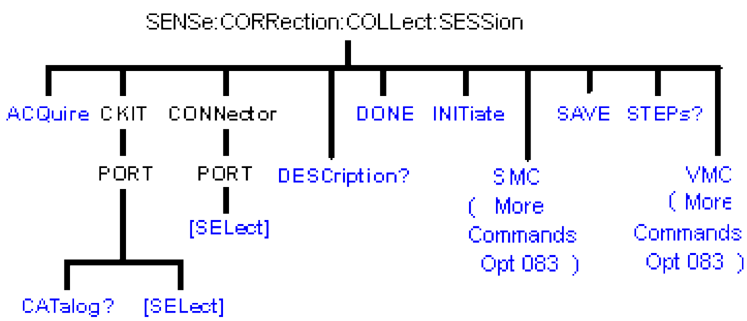
Overlapped? No

Default THRU

SENSe:CORRection:COLLect:SESSion Commands

The commands in this topic are common to perform both SMC and VMC calibrations. A calibration session is a term used to describe an instance of an SMC or VMC calibration. For more commands, see SESS:SMC and SESS:VMC.

Commands to read (STEP?) and describe (DESC?) each step are provided to facilitate a remote user interface.



- Click on a blue keyword to view the command details.
- New See Calibrating the PNA Using SCPI
- See a List of all commands in this block.
- See a SCPI SMC and VMC calibration.
- Learn about SMC and VMC calibrations

SENSe<ch>:CORRection:COLLect:SESSion<n>:ACQuire <step>

(Write only) Acquire a calibration measurement.

Parameters	
<ch>	Any existing channel number. If unspecified, value is set to 1.
<n>	Session number. Choose from 1 to 16.
<step>	Step number to acquire. Use SENS:CORR:COLL:SESS:STEPS? to find the number of steps required for the calibration.
Examples	`SENSe2:CORR:COLL:SESS6:ACQ 5`
Query Syntax	Not Applicable
Overlapped?	No
Default	Not Applicable

SENSe<ch>:CORRection:COLLect:SESSion<n>:CKIT:PORT<p>:CATalog?

(Read only) Returns a list of cal kits that are compatible with the connector on port <p>. The port connector type is set with SENS:CORR:COLL:SESS:CONN:PORT

Parameters

<ch>	Any existing channel number. If unspecified, value is set to 1.
<n>	Session number. Choose from 1 to 16.
<p>	Port number of connector to query for compatible cal kits.
Examples	**SENS2:CORR:COLL:SESS6:CKIT:PORT2:CAT?**
Return Type	Comma separated string values
Overlapped?	No
Default	Not Applicable

SENSe<ch>:CORRection:COLLect:SESSion<n>:CKIT:PORT<p>[:SELect] <calkit>

(Read-Write) Set or return the Cal Kit for the specified port. Use SENS:CORR:COLL:SESS:CKIT:PORT:CAT? to list compatible Cal Kits.

Parameters

<ch>	Any existing channel number. If unspecified, value is set to 1
<n>	Session number. Choose from 1 to 16.
<p>	Port number for which to set cal kit.
<calkit>	Cal Kit Name
Examples	**SENS:CORR:COLL:SESS:CKIT:PORT:SEL 85091A** **SENS2:CORR:COLL:SESS6:CKIT:PORT2:SEL?**
Query Syntax	SENS<ch>:CORR:COLL:SESS<n>:CKIT:PORT<p>[:SEL] ?
Return Type	String
Overlapped?	No
Default	Not Applicable

SENSe<ch>:CORRection:COLLect:SESSion<n>:CONN:PORT<p>[:SEL] <conn>

(Read-Write) Set the connector type and sex for the specified port number. Catalog valid connector types using SENS:CORR:COLL:GUID:CONN:CAT?

Parameters

<ch>	Any existing channel number. If unspecified, value is set to 1
<n>	Session number. Choose from 1 to 16.
<p>	Port number for which to select a connector type.
<conn>	Name of the connector type
Examples	**`SENS2:CORR:COLL:SESS6:CONN:PORT1:SEL "N Type"`**
Query Syntax	SENS<ch>:CORR:COLL:SESS<n>:CONN:PORT<p>[:SEL]?
Return Type	String
Overlapped?	No
Default	Not Applicable

SENSe<ch>:CORRection:COLLect:SESSion<n>:DESC? <step>

(Read-only) Returns the connection prompt for the step. List the number of steps in the calibration using SENS:CORR:COLL:SESS:STEPS?.

Parameters

<ch>	Any existing channel number. If unspecified, value is set to 1
<n>	Session number. Choose from 1 to 16.
<step>	Step number
Examples	**`SENS1:CORR:COLL:SESS6:DESC?3`**
Return Type	Numeric
Overlapped?	No
Default	Not Applicable

SENSe<ch>:CORRection:COLLect:SESSion<n>:DONE

(Write only) Ends the calibration sessions. Use SAVE? to calculate error terms and save the CalSet.

Parameters	
<ch>	Any existing channel number. If unspecified, value is set to 1
<n>	Session number. Choose from 1 to 16.
Examples	`SENS1:CORR:COLL:SESS6:DONE`
Query Syntax	Not Applicable
Overlapped?	No
Default	Not Applicable

SENSe<ch>:CORRection:COLLect:SESSion<n>:INITiate <string>

(Write only) Initiates an SMC or VMC calibration session. Use the session number for subsequent SMC or VMC commands.

Parameters	
<ch>	Any existing channel number. If unspecified, value is set to 1
<n>	Session number. Choose from 1 to 16. If the session number already exists it will be terminated and a new session initiated.
<string>	Name of the calibration. Choose from: **"VMC"** or **"VectorMixerCal.VMCType"** **"SMC"** or **"ScalarMixerCal.SMCType"**
Examples	`SENS1:CORR:COLL:SESS6:INITiate "VectorMixerCal.VMCType"`
Query Syntax	Not Applicable
Overlapped?	No
Default	Not Applicable

SENSe<ch>:CORRection:COLLect:SESSion<n>:SAVE?

(Read only) Finish the SMC or VMC calibration, compute error terms, populate and save the CalSet, and return the GUID of the CalSet.

Parameters

<ch>	Any existing channel number. If unspecified, value is set to 1
<n>	Session number. Choose from 1 to 16.
Examples	**SENS1:CORR:COLL:SESS6:SAVE?**
Return Type	String specifying the GUID of the CalSet produced by this session.
Overlapped?	No
Default	Not Applicable

SENSe<ch>:CORRection:COLLect:SESSion<n>:STEPs?

(Read-only) Returns the number of steps required by the Calibration.

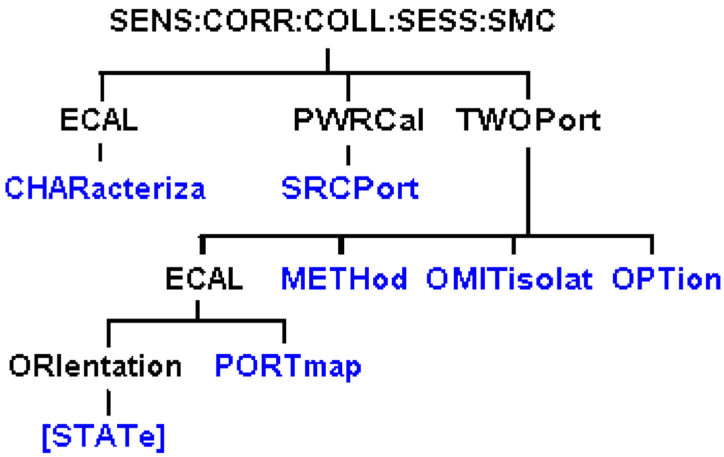
To ensure this query always completes successfully, first send the write command: SENS:CORR:COLL:SESS:STEP, then send the query.

Parameters

<ch>	Any existing channel number. If unspecified, value is set to 1
<n>	Session number. Choose from 1 to 16.
Examples	**SENS1:CORR:COLL:SESS6:STEPs?**
Return Type	Numeric
Overlapped?	No
Default	Not Applicable

SENSe:CORRection:COLLect:SESSion:SMC Commands

Performs scalar (SMC) calibration on a frequency converting device.



- Click on a blue keyword to view the command details.
- New See Calibrating the PNA Using SCPI
- See a List of all commands in this block.
- Learn about FCA Calibrations

NOTE: To configure a power meter and sensor see SOURce:POWer: commands.

SENSe<ch>:CORRection:COLLect:SESSion<n >:SMC:ECAL:CHARacteriza <mod> ,<char>

(Read-Write) Specifies the ECal module and characterization to be used for the SMC calibration.

Parameters

<ch>	Any existing channel number. If unspecified, value is set to 1
<n>	Session number. Choose from 1 to 16.
<mod>	**1** - Electronic Calibration Module
<char>	Specifies which characterization within the ECal module from which to read the confidence data. **0** Factory characterization (data that was stored in the ECal module by Agilent). Default if not specified. **1** User characterization #1 **2** User characterization #2 **3** User characterization #3 **4** User characterization #4 **5** User characterization #5

Examples	**`SENS:CORR:COLL:SESS:SMC:ECAL:CHAR 1,2`**
Query Syntax	SENS:CORR:COLL:SESS:SMC:ECAL:CHAR?
Return Type	Numeric
Overlapped?	No
Default	1,0

SENSe<ch>:CORRection:COLLect:SESSion<n>:SMC:PWRCal:SRCPort <string>

(Read-Write) Specifies which port to calibrate.

Parameters

<ch>	Any existing channel number. If unspecified, value is set to 1
<n>	Session number. Choose from 1 to 16.
<char>	**'1'** Source port 1 (SMC forward direction) **'2'** Source port 2 (SMC reverse direction **'BOTH'** (both forward and reverse directions)
Examples	**`SENS:CORR:COLL:SESS:SMC:PWRC:SRCP 'both'`** **`SENSe2:CORRection:COLLection:SESSion6:SMC:PWRCal:SRCPort '2'`**
Query Syntax	SENS:CORR:COLL:SESS:SMC:PWRC:SRCP?
Return Type	String
Overlapped?	No
Default	1

SENSe<ch>:CORRection:COLLect:SESSion<n>:SMC:TWOPort:ECAL:ORIentation[:STATe] <bool>

(Read-Write) Sets ECAL Auto-Orientation ON or OFF. If setting auto-orientation OFF, you must manually specify the orientation of the ECAL module with SENS:CORR:COLL:SESS:SMC:TWOP:ECAL:PORTmap.

Parameters

<ch>	Any existing channel number. If unspecified, value is set to 1
<n>	Session number. Choose from 1 to 16.
<bool>	**0** = Orientation OFF **1** = Orientation ON
Examples	**`SENS:CORR:COLL:SESS:SMC:TWOP:ECAL:ORI 1`**
Query Syntax	SENS:CORR:COLL:SESS:SMC:TWOP:ECAL:ORI?
Return Type	Boolean
Overlapped?	No
Default	1

SENSe<ch>:CORRection:COLLect:SESSion<n>:SMC:TWOPort:ECAL:PORTmap <mod>, <string>

(Read-Write) Specifies the manual orientation (which ports of the module are connected to which ports of the PNA) when auto-orientation is OFF.

Parameters

<ch>	Any existing channel number. If unspecified, value is set to 1
<n>	Session number. Choose from 1 to 16.
<mod>	**1 -** Electronic Calibration Module

<string>	Format in the following manner: Aw,Bx,Cy,Dz where • A, B, C, and D are literal ports on the ECAL module • w,x,y, and z are substituted for PNA port numbers to which the ECAL module port is connected. Ports of the module which are not used are omitted from the string. **For example**, on a 4-port ECal module with • port A connected to PNA port 2 • port B connected to PNA port 3 • port C not connected • port D connected to PNA port 1 the string would be: A2,B3,D1 If either the receive port or source port (or load port for 2-port cal) of the CALC:PAR:SELected measurement is not in this string and orientation is OFF, an attempt to perform an ECal calibration will fail.
Examples	`SENS:CORR:COLL:SESS:SMC:TWOP:ECAL:PORTmap 1,'A1,B2'`
Query Syntax	SENS:CORR:COLL:SESS:SMC:TWOP:ECAL:PORTmap?
Return Type	String
Overlapped?	No
Default	"A1,B2"

SENSe<ch>:CORRection:COLLect:SESSion<n >:SMC:TWOPort:METHod <string>

(Read-Write) Specifies the guided ECal method for performing the thru portion of the calibration.

Parameters

<ch> Any existing channel number. If unspecified, value is set to 1

<n> Session number. Choose from 1 to 16.

<char> ECAL Method: Choose from:

'DEFAULT' - Default

'ADAP' - Adapter removal

'FLUSH' - Flush Through

'UNKN' - Unknown Thru

Examples `SENS:CORR:COLL:SESS:SMC:TWOPort:METH 'default'`

Query Syntax SENS:CORR:COLL:SESS:SMC:TWOPort:METH?

Return Type String

Overlapped? No

Default DEFAULT

SENSe<ch>:CORRection:COLLect:SESSion<n >:SMC:TWOPort:OMITisolat <bool>

(Read-Write) Select to omit or perform the isolation portion of the ECAL.

Parameters

<ch> Any existing channel number. If unspecified, value is set to 1

<n> Session number. Choose from 1 to 16.

<bool> **1** - Omit isolation

0 - Perform isolation

Examples `SENS:CORR:COLL:SESS:SMC:TWOPort:OMIT 1`

Query Syntax SENS:CORR:COLL:SMC:TWOPort:OMIT?

Return Type Boolean

Overlapped? No

Default 1

SENSe<ch>:CORRection:COLLect:SESSion<n >:SMC:TWOPort:OPTion <string>

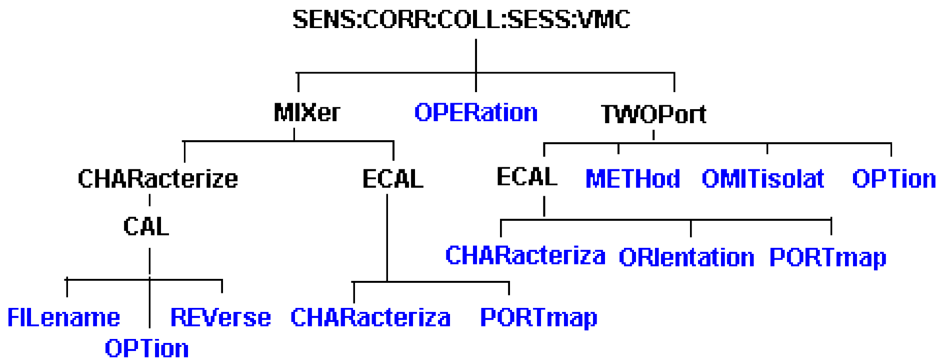
(Read-Write) Sets the SMC calibration to ECAL or MECHanical

Parameters

<ch>	Any existing channel number. If unspecified, value is set to 1
<n>	Session number. Choose from 1 to 16.
<char>	Choose from: **'ECAL'** Electronic Calibration Module **'MECH'** Mechanical Calibration Kit
Examples	**`SENS:CORR:COLL:SESS:SMC:TWOPort:OPTion 'ECAL'`**
Query Syntax	SENS:CORR:COLL:SESS:SMC:TWOPort:OPTion?
Return Type	String
Overlapped?	No
Default	ECAL

SENSe:CORRection:COLLect:SESSion:VMC Commands

Performs a vector (VMC) calibration on a frequency converting device.



- Click on a blue keyword to view the command details.
- New See Calibrating the PNA Using SCPI
- See a List of all commands in this block.
- Learn about VMC Calibration

SENS<ch>:CORRection:COLLect:SESSion<n> :VMC:CHARacterize:CAL:FILename <string>

(Read-Write) Specifies the .S2P filename used for mixer characterization. Use the VMC:MIXer:CHARacterize:CAL: OPTion command to recall the file for mixer characterization. Once recalled, use this command to query the current filename or set a new filename.

Parameters	
<ch>	Any existing channel number. If unspecified, value is set to 1
<n>	Session number. Choose from 1 to 16
<string>	Filename of the S2P used for mixer characterization. Use the full path name, file name, and .S2P suffix
Examples	**SENS2:CORR:COLL:SESS4:VMC:CHAR:CAL:FILE "C:\Program Files\Agilent\Network Analyzer\Documents\MyMixer.S2P"**
Query Syntax	SENS:CORR:COLL:SESS:VMC:CHAR:CAL:FIL?
Return Type	String
Overlapped?	No
Default	C:\Program Files\Agilent\Network Analyzer\Documents\default.s2p

---

SENS<ch>:CORRection:COLLect:SESSion<n> :VMC:MIXer:CHARacterize:CAL: OPTion

<char>

(Read-Write) Sets the mixer characterization method to ECal, Mechanical, or read from a file.

Parameters

<ch>	Any existing channel number. If unspecified, value is set to 1
<n>	Session number. Choose from 1 to 16
<char>	**ECAL -** Electronic Calibration Module **MECH -** Mechanical Calibration Kit **FILE, <filename> -** Retrieve a mixer characterization file. Also specify the filename of the S2P used for mixer characterization. Use the full path name, file name, and .S2P suffix. Use the VMC:CHARacterize:CAL:FILename command to query the filename..
Examples	**SENS:CORR:COLL:SESS:VMC:MIX:CHAR:CAL:OPT ECAL** **SENS2:CORR:COLL:SESS6:VMC:MIXer:CHAR:CAL:OPTion FILE, MyMixerFile.mxr**
Query Syntax	SENS:CORR:COLL:SESS:VMC:MIX:CHAR:CAL:OPT?
Return Type	String
Overlapped?	No
Default	MECH

SENS<ch>:CORRection:COLLect:SESSion<n> :VMC:MIXer:CHARacterize:CAL: REVerse <bool>

(Read-Write) Specifies the direction in which to characterize the calibration mixer. Learn more about the calibration mixer.

Parameters

<ch>	Any existing channel number. If unspecified, value is set to 1
<n>	Session number. Choose from 1 to 16
<bool>	**False (0)** - Characterize the calibration mixer in the SAME direction as that specified in the mixer setup. **True (1)** - Characterize the calibration mixer in the REVERSE direction as that specified in the mixer setup.
Examples	**SENS:CORR:COLL:SESS:VMC:MIX:CHAR:CAL:REV 1**
Query Syntax	SENS:CORR:COLL:SESS:VMC:MIX:CHAR:CAL:REV?
Return Type	Boolean
Overlapped?	No

Default False

SENS<ch>:CORRection:COLLect:SESSion<n > :VMC:MIXer:ECAL:CHARacteriza <mod> ,<char>

(Read-Write) Specifies the ECal module and characterization to be used for the mixer characterization portion of the calibration.

Parameters

<ch> Any existing channel number. If unspecified, value is set to 1

<n> Session number. Choose from 1 to 16

<mod> **1 -** Electronic Calibration Module

<char> Characterization number in the specified ECAL module. Choose from:

0 Factory characterization (data that was stored in the ECal module by Agilent). Default if not specified.

1 User characterization #1

2 User characterization #2

3 User characterization #3

4 User characterization #4

5 User characterization #5

Examples `SENS:CORR:COLL:SESS:VMC:MIX:ECAL:CHAR 1,0`

Query Syntax SENS:CORR:COLL:SESS:VMC:MIX:ECAL:CHAR?

Return Type Numeric

Overlapped? No

Default 1,0

SENS<ch>:CORRection:COLLect:SESSion<n> :VMC:MIXer:ECAL:PORTmap <mod>, <string>

(Read-Write) Sets the port mapping for the mixer characterization with ECal. This command is required if SENS:CORR:COLL:SESS:VMC:MIX:CHAR:CAL:OPT **ECAL** is specified.

Parameters

<ch>	Any existing channel number. If unspecified, value is set to 1
<n>	Session number. Choose from 1 to 16
<mod>	**1** - Electronic Calibration Module
<string>	Choose from: **"A1"** - ECAL module port A is connected to PNA port 1 **"B1"** - ECAL module port B is connected to PNA port 1
Examples	**SENS:CORR:COLL:SESS:VMC:MIX:ECAL:PORT?** **SENS2:CORR:COLL:SESS:VMC:MIXer:ECAL:PORTmap 1,"A1"**
Query Syntax	SENS:CORR:COLL:SESS:VMC:MIX:ECAL:PORTmap?
Return Type	String
Overlapped?	No
Default	"A1"

SENSe<ch>:CORRection:COLLect:SESSion<n>:VMC:OPERation <string>

(Read-Write) Perform either full VMC calibration or mixer characterization only.

Parameters

<ch>	Any existing channel number. If unspecified, value is set to 1
<n>	Session number. Choose from 1 to 16
<char>	**'CAL'** - full calibration and mixer characterization **'CHAR'** - mixer characterization only (no reference mixer required) - Saves an .S2P file with the filename specified in SENS<ch>:CORR:COLL:SESSion<n>:VMC:CHARacterize:CAL:FILename <filename> . If none is specified, a filename is automatically generated and can be queried using the filename command.
Examples	**SENS:CORR:COLL:SESS:VMC:OPER 'CAL'**
Query Syntax	SENS:CORR:COLL:SESS:VMC:OPER?
Return Type	String
Overlapped?	No
Default	CAL

SENS<ch>:CORRection:COLLect:SESSion<n> :VMC:TWOPort:ECAL:CHARacteriza <mod> ,<char>

(Read-Write) Specifies the ECal module and characterization to be used for the VMC calibration.

Parameters

<ch>	Any existing channel number. If unspecified, value is set to 1
<n>	Session number. Choose from 1 to 16
<mod>	**1** - Electronic Calibration Module
<char>	Characterization number in the specified ECAL module. Choose from: **0** Factory characterization (data that was stored in the ECal module by Agilent). Default if not specified. **1** User characterization #1 **2** User characterization #2 **3** User characterization #3 **4** User characterization #4 **5** User characterization #5
Examples	**`SENS:CORR:COLL:SESS:VMC:TWOP:ECAL:CHAR 1,1`**
Query Syntax	SENS:CORR:COLL:SESS:VMC:TWOP:ECAL:CHAR?
Return Type	Integer
Overlapped?	No
Default	1,0

SENS<ch>:CORRection:COLLect:SESSion<n> :VMC:TWOPort:ECAL:ORIentation[:STATe] <bool>

(Read-Write) Sets ECAL orientation for the VMC ECal.

Parameters

<ch>	Any existing channel number. If unspecified, value is set to 1
<n>	Session number. Choose from 1 to 16
<bool>	**1** = ON **0** = OFF
Examples	**SENS:CORR:COLL:SESS:VMC:TWOP:ECAL:ORI 1**
Query Syntax	SENS:CORR:COLL:SESSion:VMC:TWOPort:ECAL:ORIentation[:STATe]?
Return Type	Integer
Overlapped?	No
Default	ON

SENS<ch>:CORRection:COLLect:SESSion<n> :VMC:TWOPort:ECAL:PORTmap <mod>, <string>

(Read-Write) Specifies the manual orientation (which ports of the module are connected to which ports of the PNA) when orientation is turned off.

Parameters

<ch>	Any existing channel number. If unspecified, value is set to 1
<n>	Session number. Choose from 1 to 16
<mod>	**1 -** Electronic Calibration Module

<string> Port Map, formatted in the following manner:

Aw,Bx,Cy,Dz

where:

- A, B, C, and D are literal ports on the ECAL module.
- w,x,y, z are substituted for PNA port numbers to which the ECAL module port is connected.
- Ports of the module which are not used are omitted from the string.

For example, on a 4-port ECal module with:

- port A connected to PNA port 2
- port B connected to PNA port 3
- port C not connected
- port D connected to PNA port 1

the string would be: A2,B3,D1

If either the receive port or source port (or load port for 2-port cal) of the measurement is not in this string and orientation is OFF, an attempt to perform an ECal will fail.

Examples `SENS:CORR:COLL:SESS:VMC:TWOP:ECAL:PORTmap 1,"A2,B1"`

Query Syntax SENS:CORR:COLL:SESS:VMC:TWOP:ECAL:PORTmap?

Return Type string

Overlapped? No

Default "A1,B2"

SENS<ch>:CORRection:COLLect:SESSion<n> :VMC:TWOPort:METHod <string>

(Read-Write) Specifies the guided ECal method for performing the thru portion of the calibration.

Parameters

<ch>	Any existing channel number. If unspecified, value is set to 1
<n>	Session number. Choose from 1 to 16
<char>	**'DEFAULT' -** Default **'ADAP' -** Adapter removal **'FLUSH' -** Flush Through **'UNKN' -** Unknown Thru
Examples	`SENS:CORR:COLL:SESS:VMC:TWOP:METH 'ADAP'` `SENSe2:CORR:COLL:SESSion6:VMC:TWOPort:METHod 'FLUSH'`
Query Syntax	SENS:CORR:COLL:SESS:VMC:TWOP:METH?
Return Type	String
Overlapped?	No
Default	DEFAULT

SENS<ch>:CORRection:COLLect:SESSion<n> :VMC:TWOPort:OMITisolat <bool>

(Read-Write) Select to omit or perform the isolation portion of the ECAL.

Parameters

<ch>	Any existing channel number. If unspecified, value is set to 1
<n>	Session number. Choose from 1 to 16
<bool>	1 - omit isolation 0 - perform isolation
Examples	`SENS:CORR:COLL:SESS:VMC:TWOP:OMIT 1`
Query Syntax	SENS:CORR:COLL:SESS:VMC:TWOP:OMIT?
Return Type	Numeric
Overlapped?	No
Default	1

SENS<ch>:CORRection:COLLect:SESSion<n> :VMC:TWOPort:OPTion <string>

(Read-Write) Sets the 2-port calibration option to ECAL or MECHanical

Parameters

<ch> Any existing channel number. If unspecified, value is set to 1

<n> Session number. Choose from 1 to 16

<char> **Choose from:**

'ECAL' Electronic Calibration Module

'MECH' Mechanical Calibration Kit

Examples

```
SENS:CORR:COLL:SESS:VMC:TWOP:OPT 'MECH'
SENSe2:CORR:COLL:SESSion6:VMC:TWOPort:OPTion 'ECAL'
```

Query Syntax SENS:CORR:COLL:SESS:VMC:TWOP:OPT?

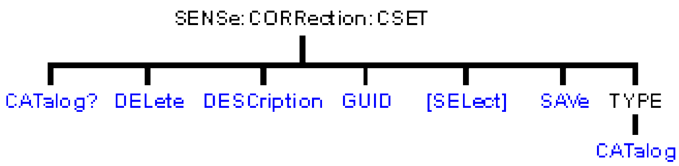
Return Type String

Overlapped? No

Default "MECH"

Sense:Correction:CSET Commands

Performs actions on calibration sets.



- Click on a blue keyword to view the command details.
- New See Calibrating the PNA Using SCPI
- See a List of all commands in this block.
- Learn about Cal Sets

SENSe:CORRection:CSET:CATalog?

(Read-only) Returns a string containing a list of comma-separated GUIDs for Cal Sets in the following format:

{FD6F863E-9719-11d5-8D6C-00108334AE96},

{1B03B2CE-971A-11d5-8D6C-00108334AE96}

Parameters	
Examples	**SENS:CORR:CSET:CAT?** **sense2:correction:cset:catalog?**
Overlapped?	No
Default	Not Applicable

ENSe<cnum>:CORRection:CSET:DELete <string>

(Write-only) Deletes a Cal Set from the set of available Cal Sets. This command immediately updates the Cal Set file on the hard drive. Using the Cal Sets collection is a convenient way to manage Cal Sets.

If the Cal Set identified by the GUID is currently in use, the Cal Set will not be deleted. If you still want to delete a Cal Set that is in use, either turn off correction on the subscribing measurement, turn off subscribed channels, or select a different Cal Set for the subscribed channel.

Parameters	
<cnum>	Any existing channel number. If unspecified, value is set to 1
<string>	The GUID of the Cal Set to be deleted. The curly brackets and hyphens must be included. Not case sensitive.
Examples	**SENS:CORR:CSET:DEL '{2B893E7A-971A-11d5-8D6C-00108334AE96}'** **sense2:correction:cset:delete '{2B893E7A-971A-11d5-8D6C-00108334AE96}'**
Query Syntax	Not Applicable
Overlapped?	No
Default	Not Applicable

SENSe<cnum>:CORRection:CSET:DESCription <string>

(Read-Write) Sets or returns the descriptive string assigned to the selected Cal Set. Change this string so that you can easily identify each Cal Set. Select the Cal Set using SENSe:CORRection:CSET:GUID

Parameters	
<cnum>	Any existing channel number. If unspecified, value is set to 1
<string>	The descriptive string associated with the currently-selected Cal Set
Examples	**SENS:CORR:CSET:DESC 'MyCalSet'** **sense2:correction:cset:description 'thisCalSet'**
Query Syntax	SENSe<cnum>:CORRection:CSET:DESCription?
Return Type	String
Overlapped?	No
Default	Not Applicable

SENSe<cnum>:CORRection:CSET:GUID <string>

(Read-Write) Selects the Cal Set identified by the string parameter (GUID) and applies it to the specified channel.

A Cal Set cannot be selected for a channel which is not On.

If the stimulus settings of the selected Cal Set differ from those of the selected channel, the instrument will automatically change the channel's settings to match the Cal Set.

Parameters	
<cnum>	Any existing channel number. If unspecified, value is set to 1
<string>	GUID of the desired Cal Set. The curly brackets and hyphens must be included.
Examples	**SENS:CORR:CSET:GUID '{2B893E7A-971A-11d5-8D6C-00108334AE96}'** **sense2:correction:cset:guid '{2B893E7A-971A-11d5-8D6C-00108334AE96}'**
Query Syntax	SENSe<cnum>:CORRection:CSET:GUID? Returns the GUID of the currently-selected Cal Set for the specified channel.
Return Type	String
Overlapped?	No
Default	Not Applicable

SENSe<cnum>:CORRection:CSET[:SELect] <char>

(Read-Write) Restores a correction data set from memory. The file name is "**CSET*x*.cst"** where x is the user number assigned to <char>, and .cst specifies a cal set and instrument state. This is not the same syntax as a file saved through the default choices from the front panel, which is "**at00*x*.cst**". For more information on the file naming syntax, see the MMEMory subsystem.

Parameters	
<cnum>	Any existing channel number. If unspecified, value is set to 1
<char>	Choose from: **DEF** - Presets the analyzer **USER01** - Restores User01 calibration data **USER02** - Restores User02 calibration data through... **USER10** - Restores User10 calibration data
Examples	**SENS:CORR:CSET DEF** **sense2:correction:cset:select user02**
Query Syntax	SENSe<cnum>:CORRection:CSET[:SELect]?
Return Type	Character
Overlapped?	No
Default	DEF

SENSe<cnum>:CORRection:CSET:SAVE <char>

(Read Write) Write a correction data set to memory or read the last correction set saved. The file name is saved as "**CSET*x*.cst"** where x is the user number assigned to <char>, and .cst specifies a cal set and instrument state. This is not the same syntax as a file saved through the default choices from the front panel, which is "**at00*x*.cst**". For more information on the filenaming syntax, see the MMEMory subsystem.

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1
<char>	Choose from: USER01 USER02... USER10
Examples	**SENS:CORR:CSET:SAVE USER03** **sense2:correction:cset:save user09**
Query Syntax	SENSe<cnum>:CORRection:CSET:SAVE? Queries the last correction set saved.
Return Type	Character
Overlapped?	No
Default	Not applicable

SENSe<ch>:CORRection:CSET:TYPE:CATalog? [format]

(Read-only) Query the Cal Types available in the selected Cal Set. You can specify the output format: a comma separated list of Guids or a list of names.

Note: CalTypes in earlier versions of PNA (pre 3.0) were indicated by a pre defined enumeration. In 3.0 the identity of the CalType has been expanded to support runtime detectable contribution calibration types. CalTypes are now identified by GUID's (128 bit number) and a name (string).

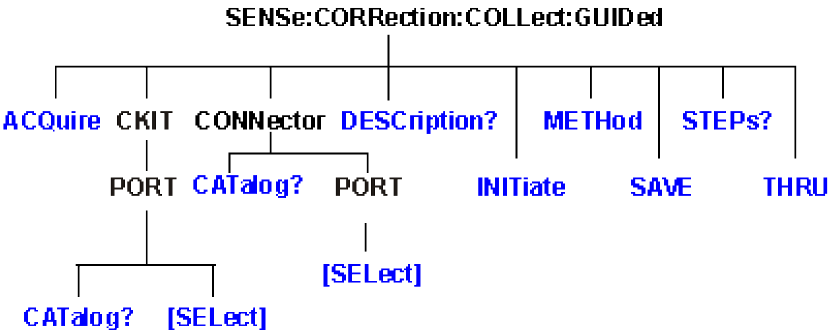
Parameters

<ch>	Any existing channel number. If unspecified, value is set to 1
[format]	(Optional) Format of the output of cal types. choose from: **NAME** - (default) returns a list of cal type string names. **GUID -** returns a list of cal type GUIDs
Examples	**SENS:CORR:CSET:TYPE:CAT? NAME** **SENS2:CORRection:CSET:TYPE:CAT?**
Return Type	**string**
Overlapped?	No

Default Not Applicable

Sense:Correction:Collect:Guided Commands

Performs and applies a GUIDED measurement calibration and other error correction features.



- Click on a blue keyword to view the command details.
- New See Calibrating the PNA Using SCPI
- See a List of all commands in this block.
- See an example using some of these commands.
- Learn about Measurement Calibration

Note: With PNA release 4.8, the following changes occurred to ECal THRU methods. These changes affect the number of calibration steps that are performed, which could cause an error or inaccurate calibration when performed using a SCPI program.

- Adapter Removal Thru method now performs 2-port measurements on both sides of the adapter. It previously performed 2-port measurements on one side and 1-port measurements on the other.
- Unknown Thru and Flush Thru now use two, instead of three, connection steps.

To prevent these problems, use the sequence of commands outlined in the Guided Cal example to query, then perform, the correct number of calibration steps each time the program is run.

SENSe<cnum>:CORRection:COLLect:GUIDed:ACQuire <std>

(Write-only) Initiates the measurement of the specified calibration standard. Executing this command with an unnecessary standard has no affect.

The measured data is stored and used for subsequent calculations of error correction coefficients. All standards must be measured before a calibration can be completed. Any measurement can be repeated until the SENS:CORR:COLL:GUID:SAVE command is executed.

Query the user prompt description using SENS:CORR:COLL:GUID:DESC?

Query the required calibration steps using SENS:CORR:COLL:GUID:STEP?

Parameters	
<cnum>	Any existing channel number. If unspecified, value is set to 1
<std>	Choose from:STAN1, STAN2, STAN3, through STAN40
Examples	**SENS:CORR:COLL:GUID:ACQ STAN1** **sense2:correction:collect:guided:acquire stan1**
Query Syntax	Not Applicable
Overlapped?	No
Default	Not Applicable

SENSe<cnum>:CORRection:COLLect:GUIDed:CKIT:PORT<pnum>:CATalog?

(Read-only) Returns a comma-separated list of valid kits for each port. In addition to mechanical calibration kits, this will include applicable characterizations found within ECal modules currently connected to the PNA. Use items in the list to select the kit to be used with the SENS:CORR:COLL:GUID:CKIT:PORT command.

Parameters	
<cnum>	Any existing channel number. If unspecified, value is set to 1
<pnum>	Any existing port number. If unspecified, value is set to 1
Examples	**SENS:CORR:COLL:GUID:CKIT:PORT1:CAT?** **'When "Type N (50) male" is specified for connector type, returns:** **"85054D, 85032F"**
Return Type	String
Overlapped?	No
Default	Not Applicable

SENSe<cnum>:CORRection:COLLect:GUIDed:CKIT:PORT<pnum>[:SELect] <kit>

(Read-Write) Specifies the calibration kit for each port to be used during a guided calibration. An unused port does NOT need to have a specified Cal Kit.

Note: Sliding loads are not fully supported from Sens:Corr:Coll:Guided... The **Measure** button must be manually pressed.

1. Specify the connector type for the port with SENS:CORR:COLL:GUID:CONN:PORT.
2. Query the valid available kits for the connector on each port with SENS:CORR:COLL:GUID:CKIT:PORT:CAT?
3. Specify the kit using this command.
4. Perform a query of this command. If the <kit> parameter was incorrectly entered, an error will be returned.

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1
<pnum>	Any existing port number. If unspecified, value is set to 1
<kit>	Calibration kit to be used for the specified port.
Examples	`SENS:CORR:COLL:GUID:CKIT:PORT1 '85055A'` `sense2:correction:collect:ckit:port2:select '85092-60010 User 1 ECal'`
Query Syntax	SENSe:CORRection:COLLect:GUIDed:CKIT:PORT<pnum>[:SELect]?
Return Type	String - If the <kit> parameter was incorrectly entered while writing, an error will be returned.
Overlapped?	No
Default	Not Applicable

SENSe<ch>:CORRection:COLLect:GUIDed:CONNector:CATalog?

(Read only) Returns a list of valid connectors based on the connector descriptions of the available cal kits. Use an item from the returned list to specify a connector for SENS:CORR:COLL:GUID:CONN:PORT

Parameters

<ch>	Any existing channel number. If unspecified, value is set to 1
Examples	**SENS:CORR:COLL:GUID:CONN:CAT?** **Returns:** **Type N (50) female, Type N (50) male, APC 7 (50), 3.5 mm (50) male, 3.5 mm (50) female, User Connector A**
Return Type	Comma separated string values
Overlapped?	No
Default	Not Applicable

SENSe<cnum>:CORRection:COLLect:GUIDed:CONNector:PORT<pnum>[:SELect] <conn>

(Read-Write) Specifies a connector type for every port during the Guided Calibration procedure. Valid connector names are stored within calibration kits. Some cal kits may include both male and female connectors. Therefore, specifying connector gender may be required.

Unused ports must be defined as or Not used. If all ports are defined as "Not used", a guided calibration cannot be performed.

- A single port with a valid <conn> name indicates a 1-Port calibration will be performed.
- Two ports with valid <conn> names indicate either a 2-Port or TRL calibration will be performed depending on the standards definition found within the cal kit and the capability of the analyzer. (The analyzer must have 4 receivers for TRL calibrations.).
- Three ports with valid <conn> names indicate a 3-Port calibration will be performed.
- Four ports with valid <conn> names indicate a 4-Port calibration will be performed.

Note:
1. Use SENS:CORR:COLL:GUID:CONN:CAT? to query available connectors before specifying the port connector.
2. Select a connector type using this command.
3. Perform a query of this command. If the <conn> parameter was incorrectly entered, an error will be returned.
4. Specify the cal kit to use for each port with SENS:CORR:COLL:GUID:CKIT:PORT

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1
<pnum>	Any existing port number. If unspecified, value is set to 1

<conn>	DUT connector type to connect with analyzer port <pnum> Some kits may include both male and female connectors so specifying gender may be required. Valid connector names are stored within calibration kits. Query available connectors using SENSe:CORRection:COLLect:GUIDed:CONNector:CATalog?
Examples	**SENS:CORR:COLL:GUID:CONN:PORT1 'Type N (50) female'** **'Indicates the DUT port that connects with the analyzer's Port1 is a TypeN 50 ohm Female connector.**
Query Syntax	SENSe<cnum>:CORRection:COLLect:GUIDed:CONNector:PORT<pnum>[:SELect]?
Return Type	String
Overlapped?	No
Default	Not Applicable

SENSe<cnum>:CORRection:COLLect:GUIDed:DESCription? <step>

(Read-only) Returns the connection description for the specified calibration step.

Parameters	
<cnum>	Any existing channel number. If unspecified, value is set to 1
<step>	A number from 1 to the number of steps required to complete the calibration (Use SENS:CORR:COLL:GUID:STEP? to query the number of steps)
Examples	**SENS:CORR:COLL:GUID:DESC? 10** **'Returns:** **Connect APC 7 Open to port3**
Return Type	String
Overlapped?	No
Default	Not Applicable

SENSe<cnum>:CORRection:COLLect:GUIDed:INITiate [GUID [,bool]]

(Write-only) Initiates a guided calibration. Either create a new cal set or optionally add to / overwrite a specified cal set.

The PNA determines the measurements needed to perform the calibration using the settings specified from the SENS:CORR:COLL:GUID:CONN:PORT and SENS:CORR:COLL:GUID:CKIT:PORT commands.

After this command is executed, subsequent commands can be used to query the number of measurement steps, issue the acquisition commands, query the connection description strings, and subsequently complete a guided calibration.

Parameters	
<cnum>	Any existing channel number. If unspecified, value is set to 1
<GUID>	Optional argument. If not specified a new calset is created. Calset GUID in the form: "{GUID}"; including quotes and curly brackets. The guided cal that is being initiated either supplements the existing cal set, or overwrites the cal set depending on the method, connectors, and ports selected. Learn more about Cal Sets. Must be a valid GUID; an error is reported if the GUID is not found. Query all Cal Set GUIDs with SENS:CORR:CSET:CAT?
<bool>	Optional argument. **False (0)** If cal set stimulus settings differ from the existing channel, do not change channel stimulus settings. Return an error. This is the default setting if not specified. **True (1)** If cal set stimulus settings differ from the existing channel, change the channel stimulus settings to match the cal set settings..
Examples	`SENS:CORR:COLL:GUID:INIT "{2B893E7A-971A-11d5-8D6C-00108334AE96}",1` `sense2:correction:collect:guided:initiate`
Query Syntax	Not Applicable
Overlapped?	No
Default	Not Applicable

SENSe<cnum>:CORRection:COLLect:GUIDed:METHod <char>

(Read-Write) Specifies the method for performing the THRU portion of a guided calibration.

See Note regarding changes to this command.

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1
<char>	**DEFAULT** - allow the PNA to choose the best possible method (from the following) depending on whether the device or ECal module is insertable or non-insertable and given the model number of the PNA. (default selection if omitted.) **ADAP** - Perform Adapter removal calibration **FLUSH** - Perform Flush Thru calibration. **UNKN** - Perform Unknown Thru calibration. **DEFined** - Perform Defined Thru calibration. If performing an ECal, this is the Thru standard in the ECal module. **TRL** - Select TRL Cal Type for 2-Port guided cals. Valid for "TRL ready" Cal Kits with properly assigned TRL cal classes. **SOLT** - Select SOLT Cal Type for 2-Port guided cals. Valid for any kit with properly assigned SOLT cal classes.

Examples

```
SENS:CORR:COLL:GUID:METH ADAP
sense2:correction:collect:guided:method unkn
```

Query Syntax SENSe<cnum>:CORRection:COLLect:GUIDed:METHod

Return Type Character

Overlapped? No

Default Not Applicable

SENSe<cnum>:CORRection:COLLect:GUIDed:SAVE

(Write-only) Completes the guided cal by computing the error correction terms, turning Correction ON, and saving the calibration to a Cal Set.

If all of the required standards have not been measured, the calibration will not complete properly.

Parameters

<cnum> Any existing channel number. If unspecified, value is set to 1

Examples

```
SENS:CORR:COLL:GUID:SAVE
sense2:correction:collect:guided:save
```

Query Syntax Not Applicable

Overlapped? No

Default Not Applicable

SENSe<cnum>:CORRection:COLLect:GUIDed:STEPs?

(Read-only) Returns the number of measurement steps required to complete the current guided calibration. This command is sent after the SENS:CORR:COLL:GUID:INIT, SENS:CORR:COLL:GUID:CONN:PORT and SENS:CORR:COLL:GUID:CKIT:PORT commands.

Parameters

<cnum> Any existing channel number. If unspecified, value is set to 1

Examples

```
SENS:CORR:COLL:GUID:STEP?
sense2:correction:collect:guided:steps?
```

Return Type Numeric

Overlapped? No

Default Not Applicable

SENSe<cnum>:CORRection:COLLect:GUIDed:THRU:PORTs <t1a, t1b, t2a, t2b, t3a, t3b>

(Read-Write) Specifies the port pairs for the thru connections of the calibration.

Note: Only for PNA release 5.0 and greater.

Parameters

<cnum> Any existing channel number. If unspecified, value is set to 1

<t1a,...> Specify Port numbers in pairs:

- For 3-port cals, specify two pairs.
- For 4-port cals, specify three pairs.

t1a, t1b (Thru1 - port A and port B)

t2a, t2b (Thru2 - port A and port B)

t3a, t3b (Thru3 - port A and port B)

Examples

```
SENS:CORR:COLL:GUID:THRU:PORT 1,2,1,3,1,4 '4-port measurement
sense2:correction:collect:guided:thru:ports 1,2,2,3 '3-port
measurement
```

Query Syntax SENSe<cnum>:CORRection:COLLect:GUIDed:THRU:PORTs?

Return Type Numeric

Overlapped? No

Default Port pairings that were used in the previous cal.

Sense:Couple Commands

SENSe:COUPle
|
PARameter
|
[STATe]

- Click on a blue keyword to view the command details.
- See a List of all commands in this block.

SENSe<cnum>:COUPle <ALL | NONE>

(Read-Write) Sets the sweep mode as Chopped or Alternate.

Learn about Alternate Sweep

Parameters	
<cnum>	Any existing channel number; if unspecified, value is set to 1.
<ALL | NONE>	**ALL** - Sweep mode set to Chopped - reflection and transmission measured on the same sweep. **NONE** - Sweep mode set to Alternate - reflection and transmission measured on separate sweeps. Improves Mixer bounce and Isolation measurements. Increases sweep time
Examples	`SENS:COUP ALL` `sense2:couple none`
Query Syntax	SENSe<cnum>:COUPle?
Return Type	Character
Overlapped?	No
Default	ALL

SENSe<cnum>:COUPle:PARameter[:STATe] <bool>

(Read-Write) Turns ON and OFF Time Domain Trace Coupling. All of the measurements in the specified channel are coupled.

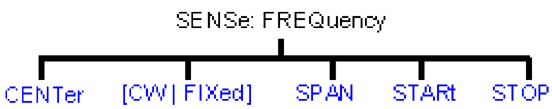
- To select Transform parameters to couple, use CALC:TRAN:COUP:PAR
- To select Gating parameters to couple, use CALC:FILT:COUP:PAR

Learn more about Time Domain Trace Coupling.

Parameters	
<cnum>	Any existing channel number; if unspecified, value is set to 1.
<bool>	**ON (or 1) -** Turns ON Time Domain Trace Coupling. **OFF (or 0) -** Turns OFF Time Domain Trace Coupling.
Examples	**SENS:COUP:PAR 0** **sense2:couple:parameter:state on**
Query Syntax	SENSe<cnum>:COUPle:PARameter[:STATe]?
Return Type	Boolean
Overlapped?	No
Default	ON (or 1)

Sense:Frequency Commands

Sets the frequency sweep functions of the analyzer.



- Click on a blue keyword to view the command details.
- See a List of all commands in this block.
- See an example using some of these commands.
- Learn about Frequency Sweep

SENSe<cnum>:FREQuency:CENTer <num>

(Read-Write) Sets the center frequency of the analyzer.

Parameters	
<cnum>	Any existing channel number. If unspecified, value is set to 1
<num>	Center frequency. Choose any number between the **minimum** and **maximum** frequency limits of the analyzer. Units are Hz **Note**: This command will accept **MIN** or **MAX** instead of a numeric parameter. See SCPI Syntax for more information.
Examples	**SENS:FREQ:CENT 1000000** **sense2:frequency:center 1mhz**
Query Syntax	SENSe<cnum>:FREQuency:CENTer?
Return Type	Numeric
Overlapped?	No
Default	Center of the analyzer's frequency span

SENSe<cnum>:FREQuency[:CW |:FIXed] <num>

(Read-Write) Sets the Continuous Wave (or Fixed) frequency. Must also send SENS:SWEEP:TYPE CW to put the analyzer into CW sweep mode.

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1
<num>	CW frequency. Choose any number between the **minimum** and **maximum** frequency limits of the analyzer. Units are Hz.
	Note: This command will accept **MIN** or **MAX** instead of a numeric parameter. See SCPI Syntax for more information.
Examples	**SENS:FREQ 1000000** **SENS:FREQ:CW MIN** **sense2:frequency:fixed 1mhz**
Query Syntax	SENSe<cnum>:FREQuency[:CW | :FIXed]?
Return Type	Numeric
Overlapped?	No
Default	1 GHz

SENSe<cnum>:FREQuency:SPAN <num>

(Read-Write) Sets the frequency span of the analyzer.

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1
<num>	Frequency span. Choose any number between: **0** (minimum) and the **maximum** frequency span of the analyzer. Units are Hz
	Note: This command will accept **MIN** or **MAX** instead of a numeric parameter. See SCPI Syntax for more information.
Examples	**SENS:FREQ:SPAN 1000000** **sense2:frequency:span max**
Query Syntax	SENSe<cnum>:FREQuency:SPAN?
Return Type	Numeric
Overlapped?	No
Default	Maximum frequency span of the analyzer

SENSe<cnum>:FREQuency:STARt <num>

(Read-Write) Sets the start frequency of the analyzer.

Parameters

<cnum> Any existing channel number. If unspecified, value is set to 1

<num> Start frequency. Choose any number between the **MIN** and **MAX** frequency limits of the analyzer. Units are Hz

If FREQ:START is set greater than FREQ:STOP, then STOP is set equal to START.

Note: This command will accept **MIN** or **MAX** instead of a numeric parameter. See SCPI Syntax for more information.

Examples

```
SENS:FREQ:STAR 1000000
sense2:frequency:start MIN
```

Query Syntax SENSe<cnum>:FREQuency:STARt?

If Sweep type is segment, this query returns the start frequency of the first segment.

Return Type Numeric

Overlapped? No

Default Minimum frequency of the analyzer

SENSe<cnum>:FREQuency:STOP <num>

(Read-Write) Sets the stop frequency of the analyzer.

Parameters

<cnum> Any existing channel number. If unspecified, value is set to 1

<num> Stop frequency. Choose any number between:
the **minimum** and **maximum** frequency limits of the analyzer. Units are Hz

If FREQ:STOP is set less than FREQ:START, then START will be set equal to STOP.

Note: This command will accept **MIN** or **MAX** instead of a numeric parameter. See SCPI Syntax for more information.

Examples

```
SENS:FREQ:STOP 1000000
sense2:frequency:stop max
```

Query Syntax SENSe<cnum>:FREQuency:STOP?

If Sweep type is segment, this query returns the stop frequency of the last segment.

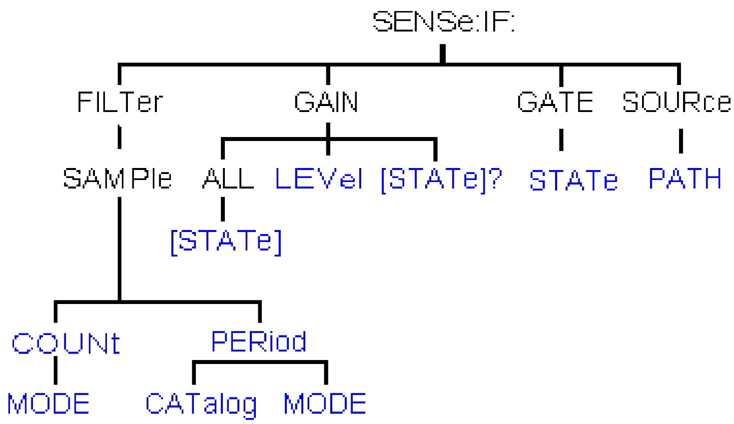
Return Type Numeric

Overlapped? No

Default Maximum frequency of the analyzer

Sense:IF Commands

Controls the IF source and gain settings for use with the PNA H11 Option.



- Click on a blue keyword to view the command details.
- See a List of all commands in this block.
- Learn about the IF Access (H11 option)

SENSe<cnum>:IF:FILTer:SAMPle:COUNt <num>

(Read-Write) Sets and returns the number of taps in the IF filter. The IF filter sample count setting is only used when SENSe:IF:FILTer:SAMPle:COUNt:MODE is set to MANUAL. **Critical Note:**

Parameters	
<cnum>	Existing channel number to manipulate. If unspecified, <cnum> is set to 1.
<num>	An integer value. (MIN and MAX return the minimum and maximum allowed values, respectively.)
Examples	`SENS:IF:FILT:SAMP:COUN 40` `sense2:if:filter:sample:count maximum`
Query Syntax	SENS:IF:FILTer:SAMPle:COUNt?
Return Type	Numeric
Overlapped?	No
Default	Instrument dependent.

SENSe<cnum>:IF:FILTer:SAMPle:COUNt:MODE <char>

(Read-Write) Sets and returns the IF filter sample count mode to the specified value. When in MANUAL mode, the value specified for the IF Filter sample count is used as the number of taps in the IF filter. **Critical Note:**

Parameters

<cnum>	Existing channel number to manipulate. If unspecified, <cnum> is set to 1.
<char>	Choose either AUTO or MANUAL.
Examples	**SENS:IF:FILT:SAMP:COUN:MODE MANUAL** **sense2:if:filter:sample:count:mode AUTO**
Query Syntax	SENS:IF:FILTer:SAMPle:COUNt:MODE?
Return Type	Character
Overlapped?	No
Default	AUTO

SENSe<cnum>:IF:FILTer:SAMPle:PERiod <num>

(Read-Write) Sets and returns the IF filter sample period. The IF filter sample period setting is only used by the instrument when the SENS:IF:FILT:SAMP:PER:MODE is set to MANUAL. **Critical Note:**

Parameters

<cnum>	Existing channel number to manipulate. If unspecified, <cnum> is set to 1.
<num>	Sample period. Choose from values returned from the SENS:IF:FILTer:SAMPle:PERiod:CAT? command.
Examples	**SENS:IF:FILT:SAMP:PER 6 us** **sense2:if:filter:sample:period maximum**
Query Syntax	SENS:IF:FILTer:SAMPle:PERiod?
Return Type	Numeric
Overlapped?	No
Default	Instrument dependent.

SENSe<cnum>:IF:FILTer:SAMPle:PERiod:CATalog?

(Read-Only) Returns the list of allowed IF filter sample periods for this instrument. **Critical Note:**

Parameters

<cnum>	Existing channel number to manipulate. If unspecified, <cnum> is set to 1.
Examples	**SENS:IF:FILT:SAMP:PER:CAT?** **sense2:if:filter:sample:period:catalog?**
Query Syntax	SENS:IF:FILTer:SAMPle:PERiod:CATalog?
Return Type	String
Overlapped?	No
Default	AUTO

SENSe<cnum>:IF:FILTer:SAMPle:PERiod:MODE <char>

(Read-Write) Sets and returns the IF filter sample period mode to the specified value. **Critical Note:**

Parameters

<cnum>	Existing channel number to manipulate. If unspecified, <cnum> is set to 1.
<char>	Sample period mode. Choose from **AUTO** or **MANUAL**.
Examples	`SENS:IF:FILT:SAMP:PER:MODE MANUAL` `sense2:if:filter:sample:period:mode AUTO`
Query Syntax	SENS:IF:FILTer:SAMPle:PERiod:MODE?
Return Type	Character
Overlapped?	No
Default	AUTO

SENSe<cnum>:IF:GAIN:ALL[:STATe] <char>

(Write only) Sets the gain state for ALL receivers to Auto or Manual.

Parameters

<cnum>	Existing channel number to manipulate. If unspecified, <cnum> is set to 1.
<char>	Choose from **AUTO** or **MANUAL**
Examples	`SENS:IF:GAIN:ALL AUTO` `sense:if:gain:all:state manual`
Query Syntax	Not Applicable
Return Type	Character
Overlapped?	No
Default	AUTO

SENSe<cnum>:IF:GAIN:LEVel <id>, <level>

(Read Write) Manually sets the gain level for the specified receiver.

Parameters

<cnum>	Existing channel number to manipulate. If unspecified, <cnum> is set to 1.
<id>	Choose from: 'A', 'B', 'R1', 'R2'. **Note**: The A and R1 receivers are always switched together. B and R2 are also always switched together. For example, if you specify 'A', R1 will also be switched.
<level>	Gain level. Choose from: 0 - Lowest gain setting 1 2 - Highest gain setting
Examples	`SENS:IF:GAIN:LEVel 'A', 1`
Query Syntax	SENSe<cnum>:IF:GAIN:LEVel? <id>
Return Type	Numeric
Overlapped?	No
Default	0

SENSe<cnum>:IF:GAIN[:STATe]?, <id>

(Read only) Returns the gain state for the specified receiver. Use SENS:IF:GAIN:ALL. to set the gain state for all channels.

Parameters

<cnum>	Existing channel number to manipulate. If unspecified, <cnum> is set to 1.
<id>	Choose from: 'A', 'B', 'R1', 'R2'. **Note**: The A and R1 receivers are always switched together. B and R2 are also always switched together. For example, if you specify 'A', R1 will also be switched.
Examples	`SENS:IF:GAIN? 'A'`
Return Type	Boolean
Overlapped?	No
Default	Not Applicable

SENSe<cnum>:IF:GATE:STATe <boolean>

(Read-Write) Sets or returns the IF filter gate state.

Parameters

<cnum>	Existing channel number to manipulate. If unspecified, <cnum> is set to 1.
<boolean>	Choose from ON or OFF
Examples	`SENS:IF:FILT:GATE:STAT ON` `sense2:if:filter:gate:state 0`
Query Syntax	SENS:IF:FILTer:SAMPle:COUNt:MODE?
Return Type	Boolean
Overlapped?	No
Default	AUTO

SENSe<cnum>:IF:SOURce:PATH <id>, <char>

(Read Write) Sets the source path for the specified receiver. An error is returned if <id> is not valid, or if <char> is not valid for the specified <id>.

Parameters

<cnum> Existing channel number to manipulate. If unspecified, <cnum> is set to 1.

<id> Choose from: 'A', 'B', 'R1', 'R2'

Note: The A and R1 receivers are always switched together. B and R2 are also always switched together. For example, if you specify "A", R1 will also be switched.

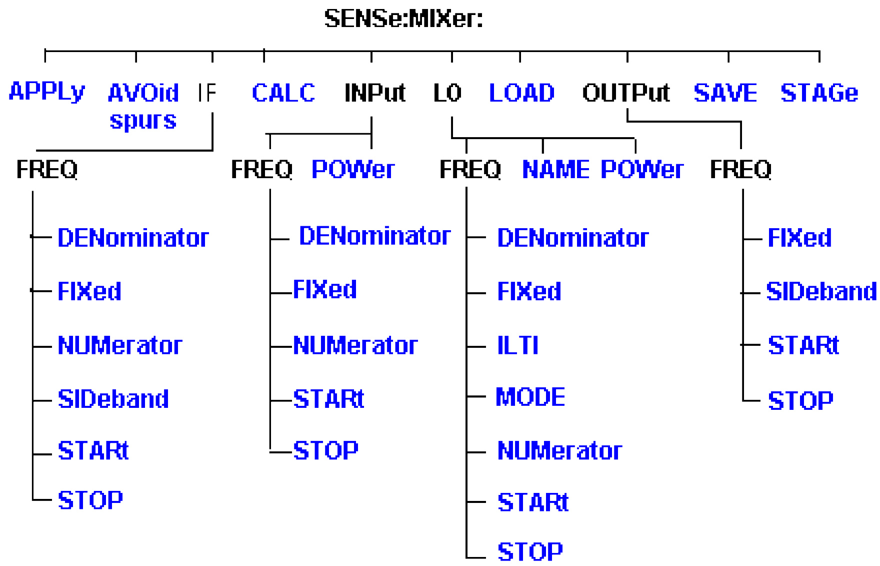
<char> Choose from:

- **NORMal** - the PNA decides the appropriate IF input paths.
- **EXTernal** - always use the rear panel IF inputs.

Examples	`SENS:IF:SOUR:PATH 'A', Ext`
Query Syntax	SENSe<cnum>:IF:SOURce:PATH? <id>
Return Type	Character
Overlapped?	No
Default	Normal

Sense:Mixer Commands

Performs Mixer setup and configuration.



- Click on a blue keyword to view the command details.
- See a List of all commands in this block.
- Learn about the Frequency Converter Application

SENSe<ch>:MIXer:APPLy

(Write only) Applies the mixer setup settings and turns the channel ON. (Performs the same function as the Apply button on the mixer setup dialog box).

Parameters	
<ch>	Any existing channel number. If unspecified, value is set to 1
Examples	`SENS:MIX:APPL`
Query Syntax	Not Applicable
Overlapped?	No
Default	Not Applicable

SENSe<ch>:MIXer:AVOidspurs <bool>

(Read Write) Sets and returns the state of the avoid spurs feature. Learn more about avoid spurs.

Parameters

<ch>	Any existing channel number. If unspecified, value is set to 1
<bool>	Avoid spurs state. Choose from 0 - Avoid spurs OFF 1 - Avoid spurs ON
Examples	`SENS:MIX:AVO` `sense2:mixer:avoidspurs 1`
Query Syntax	SENSe<ch>:MIXer:AVOidspurs?
Return Type	Boolean
Overlapped?	No
Default	0 (OFF)

SENSe<ch>:MIXer:CALCulate <char>

(Write only) Calculates the Input, IF, or Output frequencies of the mixer setup and updates the channel settings.

Parameters

<ch> Any existing channel number. If unspecified, value is set to 1

<char> Mixer port to be calculated. Choose from:

<char>	1st or only stage requires:	In addition, 2nd stage requires:
INPUT	• Output Start and Stop frequencies • LO frequency • Output sideband (High or Low)	• IF Start and Stop frequencies • 2nd LO frequency • IF sideband (High or Low)
Both	NA	• IF Start and Stop frequencies • Both LO frequencies

OUTPUT	• Input Start and Stop frequencies • LO frequency • Output sideband (High or Low)	• IF Start and Stop frequencies • 2nd LO frequency • IF sideband (High or Low)
LO_1	• Input Start and Stop frequencies • Output frequency • Output sideband (High or Low)	• IF Start and Stop frequencies • 2nd LO frequency • IF sideband (High or Low)
LO_2	NA	• Input Start and stop frequencies • 1st LO start and stop frequencies • Output frequency • IF sideband(High or Low • Output sideband(High or Low)

Examples `SENS:MIX:CALC Output`

Query Syntax Not Applicable

Overlapped? No

Default Not Applicable

SENSe<ch>:MIXer:LOAD <name>

(Write-only) Loads a previously-configured mixer attributes file (.mxr)

Parameters

<ch> Any existing channel number. If unspecified, value is set to 1.

<name> Path and file name (including .mxr extension) to load.

Examples

```
SENSe:MIXer:LOAD "C:\Program Files\Agilent\Network
Analyzer\Documents\Mixer\MyMixer.mxr"
```

Overlapped? No

Default Not Applicable

SENSe<ch>:MIXer:SAVE <name>

(Write-only) Saves the settings for the mixer/converter test setup to a mixer attributes file.

Parameters

<ch>	Any existing channel number. If unspecified, value is set to 1.
<name>	Path and file name (including .mxr extension) to save.
Examples	**SENSe:MIXer:SAVE "C:\Program Files\Agilent\Network Analyzer\Documents\Mixer\MyMixer.mxr"**
Overlapped?	No
Default	Not Applicable

SENSe<ch>:MIXer:STAGe <n>

(Read-Write) Number of IF stages of the mixer**.**

Parameters

<ch>	Any existing channel number. If unspecified, value is set to 1.
<n>	Number of stages. Choose either **1** or **2**
Examples	**SENSe:MIX:STAG 2** **SENSe:MIXer:STAGE 1**
Query Syntax	SENSe<ch>:MIXer:STAGe?
Return Type	Numeric
Overlapped?	No
Default	1

SENSe<ch>:MIXer:IF:FREQuency:DENominator <num>

(Read-Write) Sets or returns the denominator value of the IF Fractional Multiplier.

Parameters

<ch>	Any existing channel number. If unspecified, value is set to 1.
<num>	Denominator value.
Examples	**SENSe:MIX:IF:FREQ:DEN 5** **SENSe2:MIXer:IF:FREQ:DENominator 4**
Query Syntax	SENSe<ch>:MIXer:IF:FREQuency:DENominator?
Return Type	Numeric
Overlapped?	No
Default	Not Applicable

SENSe<ch>:MIXer:IF:FREQuency:FIXed <num>

(Read-Write) Sets or returns the fixed frequency of the IF..

Parameters

<ch>	Any existing channel number. If unspecified, value is set to 1.
<num>	Frequency.
Examples	**SENSe:MIXer:IF:FREQ:FIXed 1e9** **SENSe2:MIXer:IF:FREQ:FIXed 1000000000**
Query Syntax	SENSe<ch>:MIXer:IF:FREQuency:FIXed?
Return Type	Numeric
Overlapped?	No
Default	Not Applicable

SENSe<ch>:MIXer:IF:FREQuency:NUMerator <num>

(Read-Write) Sets or returns the numerator value of the IF Fractional Multiplier.

Parameters

<ch>	Any existing channel number. If unspecified, value is set to 1.
<num>	Numerator value of the IF fractional multiplier
Examples	**SENSe:MIX:IF:FREQ:NUM 4** **SENSe2:MIXer:IF2:FREQ:NUMerator 3**
Query Syntax	SENSe<ch>:MIXer:IF:FREQuency:NUMerator?
Return Type	Numeric
Overlapped?	No
Default	Not Applicable

SENSe<ch>:MIXer:IF:FREQuency:SIDeband <char>

(Read-Write) Sets or returns the value of the IF sideband.

Parameters

<ch>	Any existing channel number. If unspecified, value is set to 1.
<char>	Sideband value. Choose from **LOW** - Low or Difference (-) **HIGH** - High or Sum (+)
Examples	**SENS:MIX:IF:FREQ:SIDE LOW** **SENSe2:MIXer:IF:FREQ:SIDEband HIGH**
Query Syntax	SENSe<ch>:MIXer:IF:FREQuency:SIDeband?
Return Type	Character
Overlapped?	No
Default	LOW

SENSe<ch>:MIXer:IF:FREQuency:STARt <num>

(Read-Write) Sets or returns the IF start frequency value of the mixer.

Parameters	
<ch>	Any existing channel number. If unspecified, value is set to 1.
<num>	IF Start Frequency value
Examples	**`SENS:MIX:IF:FREQ:STAR 1e9`** **`SENSe2:MIXer:IF:FREQ:STARt 1000000000`**
Query Syntax	SENSe<ch>:MIXer:IF:FREQuency:STARt?
Return Type	Numeric
<u>Overlapped?</u>	No
<u>Default</u>	Not Applicable

SENSe<ch>:MIXer:IF:FREQuency:STOP <num>

(Read-Write) Sets or returns the stop frequency value of the mixer IF frequency.

Parameters	
<ch>	Any existing channel number. If unspecified, value is set to 1.
<num>	IF Stop Frequency value
Examples	**`SENS:MIX:IF:FREQ:STOP 2e9`** **`SENSe2:MIXer:IF:FREQ:STOP 2000000000`**
Query Syntax	SENSe<ch>:MIXer:IF:FREQuency:STOP?
Return Type	Numeric
<u>Overlapped?</u>	No
<u>Default</u>	Not Applicable

SENSe<ch>:MIXer:INPut:FREQuency:DENominator <value>

(Read-Write) Sets or returns the denominator value of the Input Fractional Multiplier.

Parameters

<ch>	Any existing channel number. If unspecified, value is set to 1.
<value>	Input denominator value.
Examples	**SENS:MIX:INP:FREQ:DEN 5** **SENS2:MIXer:INPut:FREQ:DENominator 4**
Query Syntax	SENSe<ch>:MIXer:INPut:FREQuency:DENominator?
Return Type	Numeric
Overlapped?	No
Default	Not Applicable

SENSe<ch>:MIXer:INPut:FREQuency:FIXed<value>

(Read-Write) Sets or returns the fixed frequency of the input.

Parameters

<ch>	Any existing channel number. If unspecified, value is set to 1.
<value>	Input frequency.
Examples	**SENSe:MIXer:INPut:FREQ:FIXed 1e9** **SENSe2:MIXer:INPut:FREQ:FIXed 1000000000**
Query Syntax	SENSe<ch>:MIXer:INPut:FREQuency:FIXed?
Return Type	Numeric
Overlapped?	No
Default	Not Applicable

SENSe<ch>:MIXer:INPut:FREQuency:NUMerator <value>

(Read-Write) Sets or returns the numerator value of the Input Fractional Multiplier.

Parameters

<ch>	Any existing channel number. If unspecified, value is set to 1.
<value>	Input numerator value.
Examples	**SENS:MIX:INP:FREQ:NUM 3** **SENSe2:MIXer:INPut:FREQ:NUMerator 1**
Query Syntax	SENSe<ch>:MIXer:INPut:FREQ:NUMerator?
Return Type	Numeric
Overlapped?	No
Default	Not Applicable

SENSe<ch>:MIXer:INPut:FREQuency:STARt <value>

(Read-Write) Sets or returns the Input start frequency value of the mixer.

Parameters

<ch>	Any existing channel number. If unspecified, value is set to 1.
<value>	Input Start frequency
Examples	**SENS:MIX:INP:FREQ:STAR 1e9** **SENSe2:MIXer:INPut:FREQ:STARt 1000000000**
Query Syntax	SENSe<ch>:MIXer:INPut:FREQuency:STARt?
Return Type	Numeric
Overlapped?	No
Default	Not Applicable

SENSe<ch>:MIXer:INPut:FREQuency:STOP <value>

(Read-Write) Sets or returns the Input stop frequency value of the mixer.

Parameters

<ch>	Any existing channel number. If unspecified, value is set to 1.
<value>	Input stop frequency

Examples

```
SENS:MIX:INP:FREQ:STOP 2e9
SENSe2:MIXer:INPut:FREQ:STOP 2000000000
```

Query Syntax SENSe<ch>:MIXer:INPut:FREQuency:STOP?

Return Type Numeric

Overlapped? No

Default Not Applicable

SENSe<ch>:MIXer:INPut:POWer <value>

(Read-Write) Sets or returns the value of the Input Power.

Parameters

<ch>	Any existing channel number. If unspecified, value is set to 1.
<value>	Input power in dBm.

Examples

```
SENS:MIX:INP:POW 9
SENSe2:MIXer:INPut:POWer 5
```

Query Syntax SENSe<ch>:MIXer:INPut:POWer?

Return Type Numeric

Overlapped? No

Default Not Applicable

SENSe<ch>:MIXer:LO<n>:FREQuency:DENominator <value>

(Read-Write) Sets or returns the denominator value of the LO Fractional Multiplier.

Parameters

<ch>	Any existing channel number. If unspecified, value is set to 1.
<n>	LO stage number. Choose 1 or 2.
<value>	LO denominator.
Examples	`SENS:MIX:LO:FREQ:DEN 5` `SENSe2:MIXer:LO2:FREQ:DENominator 4`
Query Syntax	SENSe<ch>:MIXer:LO<n>:FREQuency:DENominator?
Return Type	Numeric
Overlapped?	No
Default	1

SENSe<ch>:MIXer:LO<n>:FREQuency:FIXed <value>

(Read-Write) Sets or returns the fixed frequency of the specified mixer LO.

Parameters

<ch>	Any existing channel number. If unspecified, value is set to 1.
<n>	LO stage number. Choose **1** or **2**
<value>	LO frequency.
Examples	`SENS:MIX:LO:FREQ:FIX 1e9` `SENSe2:MIXer:LO2:FREQ:FIXed 1000000000`
Query Syntax	SENSe<ch>:MIXer:LO<n>:FREQuency:FIXed?
Return Type	Numeric
Overlapped?	No
Default	Not Applicable

SENSe<ch>:MIXer:LO<n>:FREQuency:ILTI <bool>

(Read-Write) Specifies whether to use the Input frequency that is **greater than** the LO or **less than** the LO. To learn more, see the mixer setup dialog box help.

Parameters

<ch>	Any existing channel number. If unspecified, value is set to 1.
<n>	LO stage number. Choose **1** or **2**
<bool>	**True (1)** - Use the Input that is Greater than the specified LO. **False (0)** - Use the Input that is Less than the specified LO.
Examples	`SENS:MIX:LO1:FREQ:ILTI 1` `sense2:mixer:lo2:frequency:ilti true`
Query Syntax	SENSe<ch>:MIXer:LO<n>:FREQuency:ILTI?
Return Type	Boolean
Overlapped?	No
Default	False

SENSe<ch>:MIXer:LO<n>:FREQuency:MODE <char>

(Read-Write) Sets or returns the LO sweep mode.

Parameters

<ch>	Any existing channel number. If unspecified, value is set to 1.
<n>	LO stage number. Choose **1** or **2**
<char>	LO sweep mode. Choose either **FIXED** or **SWEPT**
Examples	`SENS:MIX:LO:FREQ:MODE FIXED` `SENSe2:MIXer:LO2:FREQ:MODE swept`
Query Syntax	SENSe<ch>:MIXer:LO<n>:FREQuency:MODE?
Return Type	Character
Overlapped?	No
Default	Fixed

SENSe<ch>:MIXer:LO<n>:FREQuency:NUMerator <value>

(Read-Write) Sets or returns the numerator value of the LO Fractional Multiplier.

Parameters	
<ch>	Any existing channel number. If unspecified, value is set to 1.
<n>	LO stage number. Choose **1** or **2**
<value>	LO Numerator.
Examples	**SENS:MIX:LO:FREQ:NUM 5** **SENSe2:MIXer:LO2:FREQ:NUMerator 4**
Query Syntax	SENSe<ch>:MIXer:LO<n>:FREQuency:NUMerator?
Return Type	Numeric
Overlapped?	No
Default	Not Applicable

SENSe<ch>:MIXer:LO<n>:FREQuency:STARt <value>

(Read-Write) Sets or returns the LO start frequency value. This command can only be used with SMC (not VMC) measurements.

Parameters	
<ch>	Any existing channel number. If unspecified, value is set to 1.
<n>	LO stage number. Choose **1** or **2**
<value>	LO Start Frequency in Hertz.
Examples	**SENS:MIX:LO:FREQ:STAR 5E9**
Query Syntax	SENSe<ch>:MIXer:LO<n>:FREQuency:STARt?
Return Type	Numeric
Overlapped?	No
Default	Not Applicable

SENSe<ch>:MIXer:LO<n>:FREQuency:STOP <value>

(Read-Write) Sets or returns the LO stop frequency value. This command can only be used with SMC (not VMC) measurements.

Parameters

<ch>	Any existing channel number. If unspecified, value is set to 1.
<n>	LO stage number. Choose **1** or **2**
<value>	LO Stop Frequency in Hertz.
Examples	**`SENS:MIX:LO:FREQ:STOP 5E9`**
Query Syntax	SENSe<ch>:MIXer:LO<n>:FREQuency:STOP?
Return Type	Numeric
Overlapped?	No
Default	Not Applicable

SENSe<ch>:MIXer:LO<n>:NAME <value>

(Read-Write) Sets or returns the name of the external LO source.

Parameters

<ch>	Any existing channel number. If unspecified, value is set to 1.
<n>	LO stage number. Choose 1 or 2.
<value>	External LO source name
Examples	**`SENS:MIX:LO:NAME "MySource"`**
Query Syntax	SENSe<ch>:MIXer:LO<n>:NAME?
Return Type	String
Overlapped?	No
Default	Not Applicable

SENSe<ch>:MIXer:LO<n>:POWer <value>

(Read-Write) Sets or returns the value of the LO Power.

Parameters

<ch>	Any existing channel number. If unspecified, value is set to 1.
<n>	LO stage. Choose **1** or **2**
<value>	LO Power in dBm
Examples	**`SENS:MIX:LO:POW 9`**
Query Syntax	SENSe<ch>:MIXer:LO<n>:POWer?
Return Type	Numeric
Overlapped?	No
Default	Not Applicable

SENSe<ch>:MIXer:OUTPut:FREQuency:FIXed <value>

(Read-Write) Sets or returns the output fixed frequency of the mixer. This command can only be used with SMC (not VMC) measurements.

Parameters

<ch>	Any existing channel number. If unspecified, value is set to 1.
<value>	Output fixed frequency in Hertz.
Examples	**`SENS:MIX:OUTP:FREQ:FIX 5e9`**
Query Syntax	SENSe<ch>:MIXer:OUTPut:FREQuency:FIXed?
Return Type	Numeric
Overlapped?	No
Default	Not Applicable

SENSe<ch>:MIXer:OUTPut:FREQuency:SIDeband <value>

(Read-Write) Sets or returns the value of the output sideband.

Parameters

<ch>	Any existing channel number. If unspecified, value is set to 1.
<value>	**Boolean** Sideband value. Choose from **LOW** - Low or Difference (-) **HIGH** - High or Sum (+)
Examples	**`SENS:MIX:OUTP:FREQ:SIDe LOW`** **`SENSe2:MIXer:OUTPut:FREQ:SIDeband HIGH`**
Query Syntax	SENSe<ch>:MIXer:OUTPut:FREQuency:SIDeband?
Return Type	Character
Overlapped?	No
Default	LOW

SENSe<ch>:MIXer:OUTPut:FREQuency:STARt <value>

(Read-Write) Sets or returns the Output start frequency of the mixer.

Parameters

<ch>	Any existing channel number. If unspecified, value is set to 1.
<value>	Output start frequency
Examples	**`SENS:MIX:OUTP:FREQ:STAR 1e9`** **`SENSe2:MIXer:OUTPut:FREQ:STARt 1000000000`**
Query Syntax	SENSe<ch>:MIXer:OUTPut:FREQuency:STARt?
Return Type	Numeric
Overlapped?	No
Default	Not Applicable

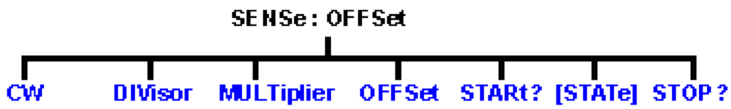
SENSe<ch>:MIXer:OUTPut:FREQuency:STOP <value>

(Read-Write) Sets or returns the Output stop frequency of the mixer.

Parameters	
<ch>	Any existing channel number. If unspecified, value is set to 1.
<value>	Output stop frequency
Examples	**SENS:MIX:OUTP:FREQ:STOP 1e9** **SENSe2:MIXer:OUTPut:FREQ:STOP 1000000000**
Query Syntax	SENSe<ch>:MIXer:OUTPut:FREQuency:STOP?
Return Type	Numeric
Overlapped?	No
Default	Not Applicable

Sense:Offset Commands

Sets the offset frequency functions, causing the stimulus and response frequencies to be different.



- Click on a blue keyword to view the command details.
- See a List of all commands in this block.
- Learn about Frequency Offset

SENSe<cnum>:OFFSet:CW <bool>

(Read-Write) Turns stimulus CW Override mode ON or OFF. Use this setting to establish a fixed (CW) stimulus frequency while measuring the Response over a swept frequency range. Learn more about Frequency Offset.

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1
<bool>	ON (or 1) - turns CW override ON. OFF (or 0) - turns CW overide OFF.
Examples	`SENS:OFFS:CW ON` `sense2:offset:cw off`
Query Syntax	SENSe<cnum>:OFFSet:CW?
Return Type	Boolean
Overlapped?	No
Default	OFF

SENSe<cnum>:OFFSet:DIVisor <num>

(Read-Write) Specifies (along with the multiplier) the value to multiply by the stimulus. Learn more about Frequency Offset.

Parameters	
<cnum>	Any existing channel number. If unspecified, value is set to 1
<num>	Divisor value. Range is 1 to 1000
Examples	`SENS:OFFS:DIV 3` `sense2:offset:divisor 2`
Query Syntax	SENSe<cnum>:OFFSet:DIVisor?
Return Type	Numeric
Overlapped?	No
Default	1

SENSe<cnum>:OFFSet:MULTiplier <num>

(Read-Write) Specifies (along with the divisor) the value to multiply by the stimulus. Learn more about Frequency Offset.

Parameters	
<cnum>	Any existing channel number. If unspecified, value is set to 1
<num>	Multiplier value. Range is +/- 1000. Negative multipliers cause the stimulus to sweep in decreasing direction. For mixer measurements, this would be for setups requiring the RF frequency to be less than LO frequency
Examples	`SENS:OFFS:MULT 2` `sense2:offset:multiplier 4`
Query Syntax	SENSe<cnum>:OFFSet:MULTplier?
Return Type	Numeric
Overlapped?	No
Default	1

SENSe<cnum>:OFFSet:OFFSet <num>

(Read-Write) Specifies an absolute offset frequency in Hz. For mixer measurements, this would be the LO frequency. Learn more about Frequency Offset.

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1
<num>	Offset frequency. Range is +/- 1000 GHz. Offsets can be positive or negative
Examples	**SENS:OFFS:OFFS 1GHz** **sense2:offset:offset 1e9**
Query Syntax	SENSe<cnum>:OFFSet:OFFSet?
Return Type	Numeric
Overlapped?	No
Default	**0 Hz**

SENSe<cnum>:OFFSet:STARt?

(Read-Only) Returns the response start frequency Learn more about Frequency Offset.

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1
Examples	**SENS:OFFS:STAR?** **sense2:offset:start?**
Return Type	Numeric
Overlapped?	No
Default	Not applicable

SENSe<cnum>:OFFSet:[STATe] <bool>

(Read-Write) Enables Frequency Offset Mode on ALL measurements that are present on the active channel. This immediately causes the source and receiver to tune to separate frequencies. The receiver frequencies are specified with the other SENS:OFFSet commands. To make the stimulus settings use the SENS:FREQ commands.

Tip: To avoid unnecessary errors, first make other offset frequency settings, then set Frequency Offset ON. Learn more about Frequency Offset.

Parameters	
<cnum>	Any existing channel number. If unspecified, value is set to 1
<bool>	ON (or 1) - turns Frequency Offset ON. OFF (or 0) - turns Frequency Offset OFF.
Examples	**`SENS:OFFS ON`** **`sense2:offset:state off`**
Query Syntax	SENSe<cnum>:OFFSet:[STATe]?
Return Type	Boolean
Overlapped?	No
Default	OFF (0)

SENSe<cnum>:OFFSet:STOP?

(Read-Only) Returns the response stop frequency. Learn more about Frequency Offset.

Parameters	
<cnum>	Any existing channel number. If unspecified, value is set to 1
Examples	**`SENS:OFFS:STOP`** **`sense2:offset:stop`**
Return Type	Numeric
Overlapped?	No
Default	Not applicable

Sense:Power Command

Learn about Receiver Attenuation

SENSe<cnum>:POWer:ATTenuation <recvr>,<num>

(Read-Write) Sets the attenuation level for the specified receiver.

Note: Attenuation cannot be set with Sweep Type set to Power

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1
<recvr>	Receiver to get attenuation. Choose from: **ARECeiver** - receiver A **BRECeiver** - receiver B
<num>	Choose from: **0** to **35** dB - in 5 dB steps If a number other than these is entered, the analyzer will select the next lower valid value. For example, if 19.9 is entered for <num> the analyzer will switch in 15 dB attenuation.

Examples	`SENS:POW:ATT AREC,10`
	`sense2:power:`
	`  attentuation breceiver,30`

Query Syntax	SENSe<cnum>:POWer:ATTenuation? <rec>
Return Type	Numeric
Overlapped?	No
Default	0

Sense:Roscillator Command

Learn about the Reference Osc.

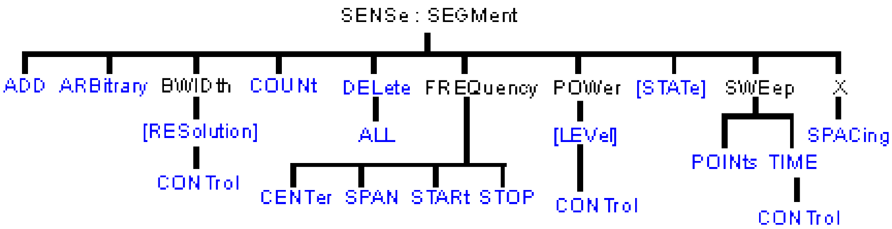
SENSe:ROSCillator:SOURce?

(Read-only) Applying a signal to the Reference Oscillator connector automatically sets the Reference Oscillator to EXTernal. This command allows you to check that it worked.
EXT is returned when a signal is present at the **Reference Oscillator** connector.
INT is returned when **NO** signal is present at the **Reference Oscillator** connector.

Examples	`SENS:ROSC:SOUR?` `sense:roscillator:source?`
Return Type	Character
Overlapped?	No
Default	Not applicable

Sense:Segment Commands

Defines the segment sweep settings. Enable segment sweep with **SENS:SWE:TYPE SEGMent.**



- Click on a blue keyword to view the command details.
- See a List of all commands in this block.
- Learn about Segment Sweep

SENSe<cnum>:SEGMent<snum>:ADD

(Write-only) Adds a segment.

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1
<snum>	Segment number to add. If unspecified, value is set to 1. Segment numbers must be sequential. If a new number is added where one currently exists, the existing segment and those following are incremented by one.
Examples	Two Segments exist (1 and 2). The following command will add a new segment (1). The existing (1 and 2) will become (2 and 3) respectively**.** **`SENS:SEGM1:ADD`** **`sense2:segment1:add`**
Query Syntax	Not applicable. Use Sense:Segment:Count to determine the number of segments in a trace.
Overlapped?	No
Default	Not Applicable

SENSe<cnum>:SEGMent:ARBitrary <ON | OFF>

(Read-Write) Enables you to setup a segment sweep with arbitrary frequencies. The start and stop frequencies of each segment can overlap other segments. Also, each segment can have a start frequency that is greater than its stop frequency which causes a reverse sweep over that segment. Learn more about Arbitrary Segment Sweep.

Parameters	
<cnum>	Any existing channel number. If unspecified, value is set to 1.
<ON | OFF>	**ON** (or 1) - Allows the setup of arbitrary segment sweep. **OFF** (or 0) - Prevents the setup of arbitrary segment sweep.
Examples	`SENS:SEGM:ARB ON` `sense2:segment:arbitrary off`
Query Syntax	SENSe<cnum>:SEGMent:ARBitrary?
Return Type	Boolean (1 = ON, 0 = OFF)
Overlapped?	No
Default	OFF

SENSe<cnum>:SEGMent<snum>:BWIDth[:RESolution] <num>

(Read-Write) Sets the IF Bandwidth for the specified segment. First set SENS:SEGM:BWIDth:CONTrol ON. All subsequent segments that are added assume the new IF Bandwidth value.

Parameters	
<cnum>	Any existing channel number. If unspecified, value is set to 1
<snum>	Segment number to modify. Choose any existing segment number.
<num>	IF Bandwidth in Hz. The list of valid IF Bandwidths is different depending on the PNA model. (Click to see the lists.) If an invalid number is specified, the analyzer will round up to the closest valid number. **Note**: This command will accept **MIN** or **MAX** instead of a numeric parameter. See SCPI Syntax for more information.
Examples	`SENS:SEGM:BWID 1KHZ` `sense2:segment2:bwidth:resolution max`
Query Syntax	SENSe<cnum>:SEGMent<snum>:BWIDth[:RESolution]?
Return Type	Numeric
Overlapped?	No
Default	See Preset IFBW for your PNA model.

SENSe<cnum>:SEGMent:BWIDth[:RESolution]:CONTrol <ON | OFF>

(Read-Write) Specifies whether the IF Bandwidth resolution can be set independently for each segment.

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1
<ON | OFF>	**ON** (or 1) - turns Bandwidth control ON. Bandwidth can be set for each segment **OFF** (or 0) - turns Bandwidth control OFF. Use channel bandwidth setting
Examples	**SENS:SEGM:BWID:CONT ON** **sense2:segment:bwidth:control off**
Query Syntax	SENSe<cnum>:SEGMent:BWIDth[:RESolution]:CONTrol?
Return Type	Boolean (1 = ON, 0 = OFF)
Overlapped?	No
Default	OFF

SENSe<cnum>:SEGMent:COUNt?

(Read-only) Queries the number of segments that exist in the specified channel.

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1
Examples	**SENS:SEGM:COUNt?** **sense2:segment:count?**
Return Type	Numeric
Overlapped?	No
Default	1 segment

SENSe<cnum>:SEGMent<snum>:DELete

(Write-only) Deletes the specified sweep segment.

<cnum>	Any existing channel number. If unspecified, value is set to 1
<snum>	Number of the segment to delete. If unspecified, value is set to 1
Examples	`SENS:SEGM:DEL` `sense2:segment2:delete`
Query Syntax	Not applicable
Overlapped?	No
Default	Not Applicable

SENSe<cnum>:SEGMent:DELete:ALL

(Write-only) Deletes all sweep segments.

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1
Examples	`SENS:SEGM:DEL:ALL` `sense2:segment:delete:all`
Query Syntax	Not applicable
Overlapped?	No
Default	Not Applicable

SENSe<cnum>:SEGMent<snum>:FREQuency:CENTer <num>

(Read-Write) Sets the Center Frequency for the specified segment. The Frequency Span of the segment remains the same. The Start and Stop Frequencies change accordingly.

Note: All previous segment's Start and Stop Frequencies that are larger than the new Start Frequency are changed to the new Start Frequency. All following segment's start and stop frequencies that are smaller than the new Stop Frequency are changed to the new Stop Frequency.

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1
<snum>	Segment number to modify. Choose any existing segment number.
<num>	Center Frequency in Hz. Choose any number between the **minimum** and **maximum** frequency of the analyzer. **Note**: This command will accept **MIN** or **MAX** instead of a numeric parameter. See SCPI Syntax for more information.

Examples

```
SENS:SEGM:FREQ:CENT 1MHZ
sense2:segment2:frequency:center 1e9
```

Query Syntax SENSe<cnum>:SEGMent<snum>:FREQuency:CENTer?

Return Type Numeric

Overlapped? No

Default Stop Frequency of the previous segment. If first segment, start frequency of the analyzer.

SENSe<cnum>:SEGMent<snum>:FREQuency:SPAN <num>

(Read-Write) Sets the Frequency Span for the specified segment. The center frequency of the segment remains the same. The start and stop frequencies change accordingly.

Note: All previous segment's Start and Stop Frequencies that are larger than the new Start Frequency are changed to the new Start Frequency. All following segment's start and stop frequencies that are smaller than the new Stop Frequency are changed to the new Stop Frequency.

Parameters

<cnum> Any existing channel number. If unspecified, value is set to 1

<snum> Segment number to modify. Choose any existing segment number.

<num> Frequency Span in Hz. Choose any number between the **minimum** and **maximum** frequency of the analyzer.

Note: This command will accept **MIN** or **MAX** instead of a numeric parameter. See SCPI Syntax for more information.

Examples

```
SENS:SEGM:FREQ:SPAN 1MHZ
sense2:segment2:frequency:span max
```

Query Syntax SENSe<cnum>:SEGMent<snum>:FREQuency:SPAN?

Return Type Numeric

Overlapped? No

Default If first segment, frequency span of the analyzer. Otherwise 0.

SENSe<cnum>:SEGMent<snum>:FREQuency:START <num>

(Read-Write) Sets the Start Frequency for the specified sweep segment.

Notes

All other segment Start and Stop Frequency values that are larger than this frequency are changed to this frequency.

To return the start and stop frequency of the entire sweep (all segments), Use SENS:FREQ:STARt? and SENS:FREQ:STOP?

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1
<snum>	Segment number to modify. Choose any existing segment number.
<num>	Start Frequency in Hz. Choose any number between the **minimum** and **maximum** frequency of the analyzer. **Note**: This command will accept **MIN** or **MAX** instead of a numeric parameter. See SCPI Syntax for more information.
Examples	`SENS:SEGM:FREQ:STAR 1MHZ` `sense2:segment2:frequency:start minimum`
Query Syntax	SENSe<cnum>:SEGMent<snum>:FREQuency:STARt?
Return Type	Numeric
Overlapped?	No
Default	Stop Frequency of the previous segment. If first segment, start frequency of the analyzer.

SENSe<cnum>:SEGMent<snum>:FREQuency:STOP <num>

(Read-Write) Sets the Stop Frequency for the specified sweep segment.

Notes

All other segment Start and Stop Frequency values that are larger than this frequency are changed to this frequency.

To return the start and stop frequency of the entire sweep (all segments), Use SENS:FREQ:STARt? and SENS:FREQ:STOP?

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1
<snum>	Segment number to modify. Choose any existing segment number.
<num>	Stop Frequency in Hz. Choose any number between the **minimum** and **maximum** frequency of the analyzer. **Note**: This command will accept **MIN** or **MAX** instead of a numeric parameter. See SCPI Syntax for more information.

Examples	`SENS:SEGM:FREQ:STOP 1MHZ` `sense2:segment2:frequency:stop maximum`
Query Syntax	SENSe<cnum>:SEGMent<snum>:FREQuency:STOP?
Return Type	Numeric
Overlapped?	No
Default	If first segment, stop frequency of the analyzer. Otherwise, start frequency of the segment.

SENSe<cnum>:SEGMent<snum>:POWer[<port>][:LEVel] <num>

(Read-Write) Sets the Port Power level for the specified sweep segment. First set SENS:SEGM:POW:CONTrol ON.

All subsequent segments that are added assume the new Power Level value.

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1
<snum>	Segment number to modify. Choose any existing segment number.
<port>	Port number of the source. Choose from 1 or 2. If unspecified, value is set to 1.
<num>	Power level. **Note:** The range of settable power values depends on the PNA model and if source attenuators are installed. To determine the range of values, send SOUR:POW? MAX and SOUR:POW? MIN. (SOUR:POW:ATT:AUTO must be set to ON). Actual achievable leveled power depends on frequency.

Examples	`SENS:SEGM:POW 0` `sense2:segment2:power1:level -10`
Query Syntax	SENSe<cnum>:SEGMent<snum>:POWer[<port>][:LEVel]?
Return Type	Numeric
Overlapped?	No
Default	0

SENSe<cnum>:SEGMent:POWer[:LEVel]:CONTrol <ON | OFF>

(Read-Write) Specifies whether Power Level can be set independently for each segment.

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1
<ON | OFF>	**ON** (or 1) - turns Power Level control ON. Power level can be set for each segment. **OFF** (or 0) - turns Power Level control OFF. Use the channel power level setting.
Examples	`SENS:SEGM:POW:CONT ON` `sense2:segment:power:level:control off`
Query Syntax	SENSe<cnum>:SEGMent:POWer[:LEVel]:CONTrol?
Return Type	Boolean (1 = ON, 0 = OFF)
Overlapped?	No
Default	OFF

SENSe<cnum>:SEGMent<snum>[:STATe] <ON | OFF>

(Read-Write) Turns the specified sweep segment ON or OFF.

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1
<snum>	Segment number to be turned ON or OFF
<ON | OFF>	**ON** (or 1) - turns segment ON. **OFF** (or 0) - turns segment OFF.
Examples	`SENS:SEGM ON` `sense2:segment2:state off`
Query Syntax	SENSe<cnum>:SEGMent[:STATe]? <snum>
Return Type	Boolean (1 = ON, 0 = OFF)
Overlapped?	No
Default	OFF

SENSe<cnum>:SEGMent<snum>:SWEep:POINts <num>

(Read-Write) Sets the number of data points for the specified sweep segment.

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1
<snum>	Any existing segment number. If unspecified, value is set to 1
<num>	Number of points in the segment. The total number of points in all segments cannot exceed **16001.** A segment can have as few as 1 point. **Note**: This command will accept **MIN** or **MAX** instead of a numeric parameter. See SCPI Syntax for more information.

Examples

```
SENS:SEGM:SWE:POIN 51
sense2:segment2:sweep:points maximum
```

Query Syntax SENSe<cnum>:SEGMent<snum>:SWEep:POINts?

Return Type Numeric

Overlapped? No

Default 201

SENSe<cnum>:SEGMent<snum>:SWEep:TIME <num>

(Read-Write) Sets the time the analyzer takes to sweep the specified sweep segment.

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1
<snum>	Any existing segment number.
<num>	Sweep time in seconds. Choose a number between **0** and **100** **Note**: This command will accept **MIN** or **MAX** instead of a numeric parameter. See SCPI Syntax for more information.

Examples

```
SENS:SEGM:SWE:TIME 1ms
sense2:segment2:sweep:time .001
```

Query Syntax SENSe<cnum>:SEGMent<snum>:SWEep:TIME?

Return Type Numeric

Overlapped? No

Default Not Applicable

SENSe<cnum>:SEGMent:SWEep:TIME:CONTrol <ON | OFF>

(Read-Write) Specifies whether Sweep Time can be set independently for each sweep segment.

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1
<ON | OFF>	**ON** (or 1) - turns Sweep Time control ON. Sweep Time can be set for each segment. **OFF** (or 0) - turns Sweep Time control OFF. Uses the channel Sweep Time setting.
Examples	`SENS:SEGM:SWE:TIM:CONT ON` `sense2:segment:sweep:time:control off`
Query Syntax	SENSe<cnum>:SEGMent:SWEep:TIME:CONTrol?
Return Type	Boolean (1 = ON, 0 = OFF)
Overlapped?	No
Default	OFF

SENSe<cnum>:SEGMent<snum>:X:SPACing <char>

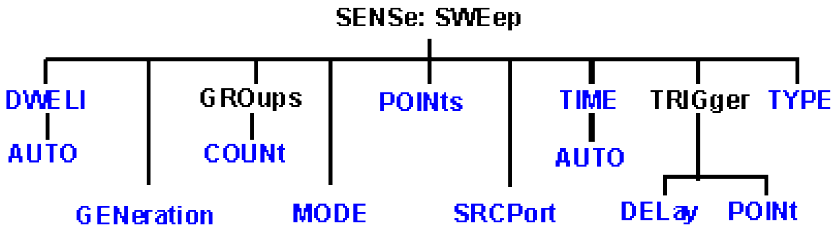
(Read-Write) Sets X-axis spacing ON or OFF

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1
<snum>	Any existing segment number. (This parameter is ignored)
<char>	**LINear** - turns X-axis point spacing OFF **OBASe** - turns X-axis point spacing ON
Examples	`SENS:SEGM:X:SPACing LIN` `sense2:segment1:spacing obase`
Query Syntax	SENSe<cnum>:SEGMent<snum>:X:SPACing?
Return Type	Character
Overlapped?	No
Default	LINear

Sense:Sweep Commands

Specifies the sweep functions of the analyzer.



- Click on a blue keyword to view the command details.
- See a List of all commands in this block.
- Learn about Sweeping

SENSe<cnum>:SWEep:DWELl <num>

(Read-Write) Sets the dwell time between each sweep point.

- Dwell time is **ONLY** available with SENSe:SWEep:GENeration set to **STEPped;** It is **Not** available in **ANALOG**.
- Sending dwell = 0 is the same as setting SENS:SWE:DWEL:AUTO **ON.** Sending a dwell time > 0 sets SENS:SWE:DWEL:AUTO **OFF**.

Parameters	
<cnum>	Any existing channel number. If unspecified, value is set to 1
<num>	Dwell time in seconds.
	Note: This command will accept **MIN** or **MAX** instead of a numeric parameter. See SCPI Syntax for more information.
Examples	**SENS:SWE:DWEL .1** **sense2:sweep:dwell min**
Query Syntax	SENSe<cnum>:SWEep:DWELl?
Return Type	Numeric
Overlapped?	No
Default	0 - (**Note**: dwell time set to 0 is the same as dwell:auto ON)

SENSe<cnum>:SWEep:DWELl:AUTO <ON | OFF>

(Read-Write) Specifies whether or not to automatically calculate and set the minimum possible dwell time. Setting Auto **ON** has the same effect as setting dwell time to **0**.

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1
<ON | OFF>	**ON** (or 1) - turns dwell ON. **OFF** (or 0) - turns dwell OFF.
Examples	`SENS:SWE:DWEL:AUTO ON` `sense2:sweep:dwell:auto off`
Query Syntax	SENSe<cnum>:SWEep:DWELl:AUTO?
Return Type	Boolean (1 = ON, 0 = OFF)
Overlapped?	No
Default	ON

SENSe<cnum>:SWEep:GENeration <char>

(Read-Write) Sets sweep as Stepped or Analog.

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1
<char>	Choose from: **STEPped** - source frequency is CONSTANT during measurement of eah displayed point. More accurate than ANALog. Dwell time can be set in this mode. **ANALog** - source frequency is continuously RAMPING during measurement of each displayed point. Faster than STEPped. Sweep time (not dwell time) can be set in this mode.
Examples	`SENS:SWE:GEN STEP` `sense2:sweep:generation analog`
Query Syntax	SENSe<cnum>:SWEep:GENeration?
Return Type	Character
Overlapped?	No
Default	Analog

SENSe<cnum>:SWEep:GROups:COUNt <num>

(Read-Write) Sets the trigger count (groups) for the specified channel. Set trigger mode to group after setting this count.

Parameters	
<cnum>	Any existing channel number. If unspecified, value is set to 1
<num>	Count (groups) number. Choose any number between: **1** and **2e6** (1 is the same as single trigger)
Examples	**`SENS:SWE:GRO:COUN 10`** **`sense2:sweep:groups:count 50`**
Query Syntax	SENSe<cnum>:SWEep:GROups:COUNt?
Return Type	Numeric
Overlapped?	No
Default	1

SENSe<cnum>:SWEep:MODE <char>

(Read-Write) Sets the trigger mode for the specified channel.

Parameters	
<cnum>	Any existing channel number. If unspecified, value is set to 1
<char>	Trigger mode. Choose from: **HOLD** - channel will not trigger **CONTinuous** - channel triggers indefinitely **GROups** - channel accepts the number of triggers specified with the last SENS:SWE:GRO:COUN <num>
Examples	**`SENS:SWE:MODE CONT`** **`sense2:sweep:mode hold`**
Query Syntax	SENSe<cnum>:SWEep:MODE?
Return Type	Character
Overlapped?	**YES** - SENS:SWE:MODE GROUPS (when INIT:CONT is ON) **NO** - HOLD and CONTinuous
Default	CONTinuous

SENSe<cnum>:SWEep:POINts <num>

(Read-Write) Sets the number of data points for the measurement.

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1
<num>	Choose any number between **1** and **16001**
	Note: This command will accept **MIN** or **MAX** instead of a numeric parameter. See SCPI Syntax for more information.

Examples

```
SENS:SWE:POIN 51
sense2:sweep:points max
```

Query Syntax SENSe<cnum>:SWEep:POINts?

Return Type Numeric

Overlapped? No

Default 201

SENSe<cnum>:SWEep:SRCPort <1 | 2>

(Read-Write) Sets the source port when making non S-parameter measurements. Has no effect on S-parameter measurements.

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1
<1 | 2>	**1** - Source power comes out Port 1 **2** - Source power comes out Port 2

Examples

```
SENS:SWE:SRCP 1
sense2:sweep:srcport 2
```

Query Syntax SENSe<cnum>:SWEep:SRCPort?

Return Type Character

Overlapped? No

Default 1

SENSe<cnum>:SWEep:TIME <num>

(Read-Write) Sets the Sweep time of the analyzer. If sweep time accuracy is critical, use ONLY the values that are attained using the up and down arrows next to the sweep time entry box. See Sweep Time.

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1
<num>	Sweep time in seconds. To select the fastest sweep speed, either send **MIN** as an argument to this command, or send SENS:SWE:TIME:AUTO 1 **Note**: This command will accept **MIN** or **MAX** instead of a numeric parameter. See SCPI Syntax for more information.
Examples	**SENS:SWE:TIME 1ms** **sense2:sweep:time .001**
Query Syntax	SENSe<cnum>:SWEep:TIME?
Return Type	Numeric
Overlapped?	No
Default	NA

SENSe<cnum>:SWEep:TIME:AUTO <ON | OFF>

(Read-Write) Turns the automatic sweep time function ON or OFF.

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1
<ON | OFF>	**ON** (or 1) - turns the automatic sweep time ON. **OFF** (or 0) - turns the automatic sweep time OFF.
Examples	**SENS:SWE:TIME:AUTO 1** **sense2:sweep:time:auto off**
Query Syntax	SENSe<cnum>:SWEep:TIME:AUTO?
Return Type	Boolean (1 = ON, 0 = OFF)
Overlapped?	No
Default	ON

SENSe<cnum>:SWEep:TRIGger:DELay <num>

(Read-Write) Sets and reads the trigger delay for all measurements in the specified CHANNEL. This delay is only applied while TRIG:SOURce EXTernal and TRIG:SCOP CURRent . After an external trigger is applied, the start of the sweep is delayed for the specified delay value plus any inherent latency.

To apply a trigger delay for all channels (Global), use TRIG:DEL.

Parameters	
<cnum>	Any existing channel number. If unspecified, value is set to 1
<num>	Trigger delay value in seconds. Range is from 0 to 107 seconds
Examples	**SENS:SWE:TRIG:DELay .003** **sense2:sweep:trigger:delay 1**
Query Syntax	SENSe<cnum>:SWEep:TRIGger:DELay?
Return Type	Numeric
Overlapped?	No
Default	0

SENSe<cnum>:SWEep:TRIGger:POINt <ON | OFF>

(Read-Write) Specifies whether the specified channel will measure one point for each trigger or all of the measurements in the channel. Setting any channel to POINt mode will automatically set the TRIGger:SCOPe = CURRent.

Parameters	
<cnum>	Any existing channel number. If unspecified, value is set to 1
<ON | OFF>	**ON** (or 1) - Channel measures one data point per trigger. **OFF** (or 0) - All measurements in the channel made per trigger.
Examples	**SENS:SWE:TRIG:POIN ON** **sense2:sweep:trigger:point off**
Query Syntax	SENSe<cnum>:SWEep:TRIGger:POINt?
Return Type	Boolean (1 = Point, 0 = Measurement)
Overlapped?	No
Default	0 - Measurement

SENSe<cnum>:SWEep:TYPE <char>

(Read-Write) Sets the type of analyzer sweep mode.

Parameters	
<cnum>	Any existing channel number. If unspecified, value is set to 1
<char>	Choose from: **LINear | LOGarithmic | POWer | CW | SEGMent** **Note**: SWEep TYPE cannot be set to SEGMent if there are no segments turned ON. A segment is automatically turned ON when the analyzer is started.
Examples	`SENS:SWE:TYPE LIN` `sense2:sweep:type segment`
Query Syntax	SENSe<cnum>:SWEep:TYPE?
Return Type	Character
Overlapped?	No
Default	LINear

X Values Command

SENSe<cnum>:X[:VALues]?

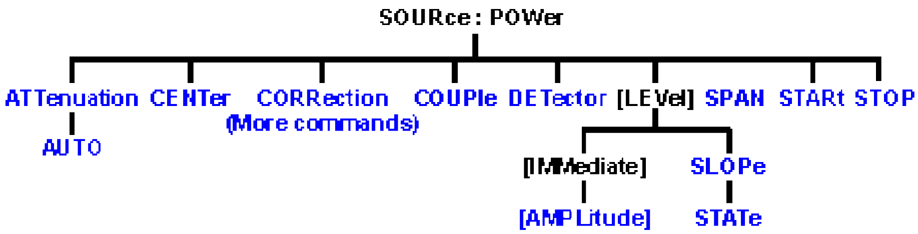
(Read-only) Returns the stimulus values for the specified channel. If the channel is sweeping the source backwards, the values will be in descending order.

Parameters

<cnum>	Any existing channel number; if unspecified, value is set to 1.
Examples	**`SENS:X?`** **`sense2:x:values?`**
Return Type	Depends on FORM:DATA command
Overlapped?	No
Default	Not applicable

Source Commands

Controls the power delivered to the DUT.



- Click on a blue keyword to view the command details.
- See a List of all commands in this block.
- Learn about Power Settings

SOURce<cnum>:POWer<port>:ATTenuation <num>

(Read-Write) Sets the attenuation level for the selected channel. Sending this command turns automatic attenuation control (SOUR:POW:ATT:AUTO) to OFF. If the ports are coupled, changing the attenuation on one port will also change the attenuation on the other port. To turn port coupling OFF use SOURce:POWer:COUPle OFF.

Note: Attenuation cannot be set with **Sweep Type** set to **Power**

Parameters

<cnum> Any existing channel number. If unspecified, value is set to 1

<port> Port number of the attenuator being set. Choose **1** or **2**; If unspecified, value is set to 1.

<num> Attenuation value. The range of settable values depends on the PNA model. To determine the range of values, perform a query using MAX, then MIN, as an argument. Example: SOURce:POWer:ATT? Max

Enter a value between MAX and MIN in 10 dB steps.

If a number other than these is entered, the analyzer will select the next lower valid value. For example, if 19.9 is entered for <num> the analyzer will switch in 10 dB attenuation.

Note: This command will accept **MIN** or **MAX** instead of a numeric parameter. See SCPI Syntax for more information.

Examples

```
SOUR:POW:ATT 10
source2:power:attentuation maximum
```

Query Syntax SOURce<cnum>:POWer<port>:ATTenuation?

Return Type Numeric

Overlapped? No

Default 0

SOURce<cnum>:POWer<port>:ATTenuation:AUTO <ON | OFF>

(Read-Write) Turns automatic attenuation control ON or OFF. Setting an attenuation value (using SOURce:POWer:ATTenuation <num>) sets AUTO **OFF**.

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1.
<port>	Port number of the attenuator being set. Choose **1** or **2**; If unspecified, value is set to 1.
<ON | OFF>	**ON** (or 1) - turns coupling ON. The analyzer automatically selects the appropriate attenuation level to meet the specified power level. **OFF** (or 0) - turns coupling OFF. Attenuation level must be set using SOURce:POWer:ATTenuation <num>.

Examples

```
SOUR:POW2:ATT:Auto On
source2:power:
  attentuation:auto off
```

Query Syntax SOURce<cnum>:POWer:ATTenuation:Auto?

Return Type Boolean (1 = ON, 0 = OFF)

Overlapped? No

Default ON

SOURce<cnum>:POWer:CENTer <num>

(Read-Write) Sets the power sweep center power. Must also set: SENS:SWE:TYPE POWer and SOURce:POWer:SPAN <num>**.**

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1
<num>	Center power. Actual achievable leveled power depends on frequency.

Examples

```
SOUR:POW:CENT -15
source2:power:center -7
```

Query Syntax SOURce<cnum>:POWer:CENTer?

Return Type Numeric

Overlapped? No

Default 0 dBm

SOURce<cnum>:POWer:COUPle <ON | OFF>

(Read-Write) Turns Port Power Coupling ON or OFF.

Parameters	
<cnum>	Any existing channel number. If unspecified, value is set to 1
<ON | OFF>	**ON** (or 1) - turns coupling ON. Power level can be set individually for each source port. **OFF** (or 0) - turns coupling OFF. The same power level is used for both source ports.
Examples	**`SOUR:POW:COUP ON`** **`source2:power:couple off`**
Query Syntax	SOURce<cnum>:POWer:COUPle?
Return Type	Boolean (1 = ON, 0 = OFF)
Overlapped?	No
Default	ON

SOURce<cnum>:POWer:DETector <char>

(Read-Write) Sets the source leveling loop as Internal or External.

Parameters	
<cnum>	Any existing channel number. If unspecified, value is set to 1
<char>	**INTernal** - Internal leveling is applied to the source **EXTernal** - External leveling is applied to the source through a rear-panel connector. ONLY provided on 3 GHz, 6 GHz, and 9 GHz PNA models.
Examples	**`SOUR:POW:DET INT`** **`source2:power:detector external`**
Query Syntax	SOURce<cnum>:POWer:DETector?
Return Type	Character
Overlapped?	No
Default	INTernal

SOURce<cnum>:POWer<port>[:LEVel][:IMMediate][:AMPLitude] <num>

(Read-Write) Sets the RF power output level.

Parameters	
<cnum>	Any existing channel number. If unspecified, value is set to 1
<port>	Port number of the attenuator being set. Choose **1** or **2**; If unspecified, value is set to 1.
<num>	Source power in dBm. **Note:** The range of settable power values depends on the PNA model and if source attenuators are installed. To determine the range of values, perform a query using MAX, then MIN, as an argument. (SOUR:POW:ATT:AUTO must be set to ON) Example: SOURce:POWer? Max Actual achievable leveled power depends on frequency.
Examples	`SOUR:POW1 5DB` `source2:power:level :immediate:amplitude maximum`
Query Syntax	SOURce<cnum>:POWer[:LEVel][:IMMediate][:AMPLitude]?
Return Type	Numeric
Overlapped?	No
Default	0 dBm

SOURce<cnum>:POWer[:LEVel]:SLOPe <num>

(Read-Write) Sets the RF power slope value.

Parameters	
<cnum>	Any existing channel number. If unspecified, value is set to 1
<num>	Slope value in db/GHz. Choose any integer between **-2** and **2** (0 is no slope).
Examples	`SOUR:POW:SLOP 2` `source2:power:slope -2`
Query Syntax	SOURce<cnum>:POWer[:LEVel]:SLOPe?
Return Type	Numeric
Overlapped?	No
Default	0

SOURce<cnum>:POWer[:LEVel]:SLOPe:STATe <ON | OFF>

(Read-Write) Turns Power Slope ON or OFF.

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1
<ON|OFF>	**ON** (or 1) - turns slope ON. **OFF** (or 0) - turns slope OFF.
Examples	`SOUR:POW:SLOP:STAT ON` `source2:power:slope:state off`
Query Syntax	SOURce<cnum>:POWer[:LEVel]:SLOPe:STATe?
Return Type	Boolean (1 = ON, 0 = OFF)
Overlapped?	No
Default	OFF

SOURce<cnum>:POWer:SPAN <num>

(Read-Write) Sets the power sweep span power. Must also set:

SENS:SWE:TYPE POWer and SOURce:POWer:CENTer <num>**.**

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1
<num>	Span power. Actual achievable leveled power depends on frequency.
Examples	`SOUR:POW:SPAN -15` `source2:power:span -7`
Query Syntax	SOURce<cnum>:POWer:SPAN?
Return Type	Numeric
Overlapped?	No
Default	0 dBm

SOURce<cnum>:POWer:STARt <num>

(Read-Write) Sets the power sweep start power. Must also set SENS:SWE:TYPE POWer and SOURce:POWer:STOP <num>**.**

Parameters

<cnum> Any existing channel number. If unspecified, value is set to 1

<num> Start power.

Note: The range of settable power values depends on the PNA model and if source attenuators are installed. To determine the range of values, perform a query using MAX, then MIN, as an argument. (SOUR:POW:ATT:AUTO must be set to ON) Example: SOURce:POWer:STARt? MIN

Actual achievable leveled power depends on frequency.

Examples

```
SOUR:POW:STAR -15
source2:power:start -7
```

Query Syntax SOURce<cnum>:POWer:STARt?

Return Type Numeric

Overlapped? No

Default 0 dBm

SOURce<cnum>:POWer:STOP <num>

(Read-Write) Sets the power sweep stop power. Must also set: SENS:SWE:TYPE POWer and SOURce:POWer:START <num>**.**

Parameters

<cnum> Any existing channel number. If unspecified, value is set to 1

<num> Stop power.

Note: The range of settable power values depends on the PNA model and if source attenuators are installed. To determine the range of values, perform a query using MAX, then MIN, as an argument. (SOUR:POW:ATT:AUTO must be set to ON) Example: SOURce:POWer:STOP? MAX

Actual achievable leveled power depends on frequency.

Examples

```
SOUR:POW:STOP -15
source2:power:stop -7
```

Query Syntax SOURce<cnum>:POWer:STOP?

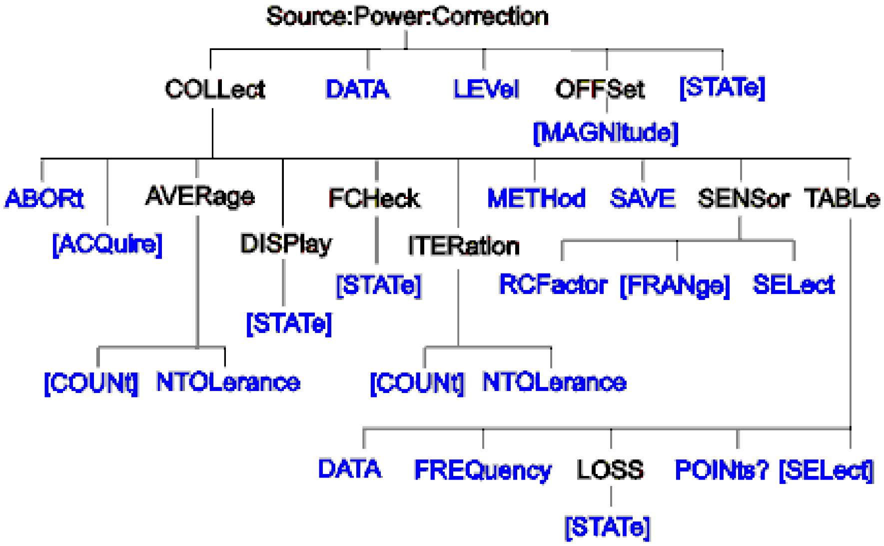
Return Type Numeric

Overlapped? No

Default 0 dBm

Source:Power:Correction Commands

Controls the source power correction features of the analyzer.



- Click on a blue keyword to view the command details.
- See a List of all commands in this block.
- See an example program using these commands.
- See a template for creating your own Power Meter Driver
- Learn about Source Cal

Note: The SOURce:POWer:CORRection:COLLect:ACQuire command, used to step the PNA and read a power meter, cannot be sent over the GPIB unless the power meter is connected to a different GPIB interface. See the alternative methods described in the command details.

SOURce<cnum>:POWer<port>:CORRection:COLLect:ABORt

SOURce<cnum>:POWer<port>:CORRection:COLLect:ABORt

(Write-only) Aborts a source power calibration sweep that is in progress.

Parameters

<cnum> Any existing channel number. If unspecified, value is set to 1

<port> Port number to correct for source power. If unspecified, value is set to 1.

Examples

```
SOUR:POW:CORR:COLL:ABOR
source1:power2:correction:collect:abort
```

Query Syntax Not Applicable

Overlapped? No

Default Not Applicable

SOURce<cnum>:POWer<port>:CORRection:COLLect[:ACQuire] <char>

(Write-only) Initiates a source power cal acquisition sweep using the power sensor attached to the specified channel (A or B) on the power meter.

Note: This command, used to step the PNA and read a power meter, cannot be sent over the GPIB unless the power meter is connected to a different GPIB interface. Use one of the following methods to perform this command or its equivalent:

- Connect the power meter to the PNA using a USB / GPIB interface (Agilent 82357A).
- SCPI programming of the PNA using a LAN Client interface (see example)
- Send SCPI commands through the COM interface using the SCPI String Parser object.
- Directly control the Power Meter and PNA to step frequency; then acquire and store the Power reading. (see example)

Parameters

<cnum> Any existing channel number. If unspecified, value is set to 1

<port> Port number to correct for source power. If unspecified, value is set to 1.

<char> Choose from:

ASENsor - Sensor on power meter channel A

BSENsor - Sensor on power meter channel B

Examples

```
SOUR:POW:CORR:COLL ASEN
source1:power2:correction:collect:acquire bsensor
```

Query Syntax Not Applicable

Overlapped? No

Default Not Applicable

SOURce<cnum>:POWer<port>:CORRection:COLLect:AVERage[:COUNt] <num>

(Read-Write) Specifies the maximum number of power readings that are taken at each stimulus point to allow for power meter settling. Each reading is averaged with the previous readings at that stimulus point. When this average meets the Average:NTOLerance value or this number of readings has been made, the average is returned as the valid reading.

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1
<port>	Port number to correct for source power. If unspecified, value is set to 1.
<num>	Maximum number of readings to make to allow for settling. Choose any number between 3 and 25.

Examples

```
SOUR:POW:CORR:COLL:AVER 2
source1:power2:correction:collect:average:count 3
```

Query Syntax SOURce:POWer:CORRection:COLLect:AVERage[:COUNt]?

Return Type Numeric

Overlapped? No

Default 3

SOURce<cnum>:POWer<port>:CORRection:COLLect:AVERage:NTOLerance <num>

(Read-Write) Each power reading is averaged with the previous readings at each stimulus point. When the average meets this nominal tolerance value or the max number of readings has been made, the average is returned as the valid reading.

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1
<port>	Port number to correct for source power. If unspecified, value is set to 1.
<num>	Power meter settling tolerance value in dB. Choose any number between 0 and 5.

Examples

```
SOUR:POW:CORR:COLL:AVER:NTOL .05
source1:power2:correction:collect:average:ntolerance .003
```

Query Syntax SOURce:POWer:CORRection:COLLect:AVERage:NTOLerance?

Return Type Numeric

Overlapped? No

Default .050 dBm

SOURce<cnum>:POWer:CORRection:COLLect:DISPlay[:STATe] <ON | OFF>

(Read-Write) Enables and disables the display of power readings on the PNA screen. Send this command BEFORE you begin a source power cal acquisition. After the source power cal data is acquired, this setting is reset to ON.

Parameters

<cnum> Any existing channel number. If unspecified, value is set to 1

<ON|OFF> **ON (1)** Source power calibration dialog box is shown on the PNA screen. Power meter readings are plotted against the Tolerance value as limit lines.

OFF (0) - Source power calibration dialog box is NOT shown on the PNA screen.

Examples

```
SOUR:POW:CORR:COLL:DISP ON
source1:power2:correction:collect:display:state off
```

Query Syntax SOURce:POWer:CORRection:COLLect:DISPlay[:STATe]?

Return Type Boolean (1 = ON, 0 = OFF)

Overlapped? No

Default ON (1)

SOURce<cnum>:POWer:CORRection:COLLect:FCHeck[:STATe] <ON | OFF>

(Read-Write) Enables and disables frequency checking of source power cal acquisition sweeps. ONLY use when you have more than one power sensor.

Parameters

<cnum> Any existing channel number. If unspecified, value is set to 1

<ON|OFF>	**ON (1)** turns source power cal frequency checking ON. A requested acquisition will only succeed for those frequency points which fall within a frequency range specified for the power sensor being used. An acquisition will pause in mid-sweep if the frequency is about to exceed the maximum frequency limit specified for that sensor. When the sweep is paused in this manner, a sensor connected to the other channel input of the power meter can be connected to the measurement port in place of the previous sensor, and used to complete the sweep. However, the maximum frequency specified for the second sensor would need to be sufficient for the sweep to complete. Frequency limits are specified using the SOUR:POW:CORR:COLL:SEN command. **OFF (0)** - turns source power cal frequency checking OFF. An acquisition will use just one power sensor for the entire sweep, regardless of frequency.
Examples	`SOUR:POW:CORR:COLL:FCH ON` `source1:power2:correction:collect:fcheck:state off`
Query Syntax	SOURce:POWer:CORRection:COLLect:FCHeck[:STATe]?
Return Type	Boolean (1 = ON, 0 = OFF)
Overlapped?	No
Default	OFF (0)

SOURce<cnum>:POWer<port>:CORRection:COLLect:ITERation[:COUNt] <num>

(Read-Write) Sets the maximum number of readings to take at each data point for iterating the source power. Power readings will continue to be made, and source power adjusted, until a reading is within the iteration tolerance value or this max number of readings has been met. The last value to be read is the valid reading for that data point.

Parameters	
<cnum>	Any existing channel number. If unspecified, value is set to 1
<port>	Port number to correct for source power. If unspecified, value is set to 1.
<num>	Maximum number of readings. Choose any number between 1 and 100.
Examples	`SOUR:POW:CORR:COLL:ITER 2` `source1:power2:correction:collect:iteration 3`
Query Syntax	SOURce:POWer:CORRection:COLLect:ITERation[:COUNt]?
Return Type	Numeric
Overlapped?	No
Default	1

SOURce<cnum>:POWer<port>:CORRection:COLLect:ITERation:NTOLerance <num>

(Read-Write) Sets the maximum desired deviation from the sum of the test port power and the offset value. Power readings will continue to be made, and source power adjusted, until a reading is within this tolerance value or the max number of readings has been met. The last value to be read is the valid reading for that data point.

Parameters	
<cnum>	Any existing channel number. If unspecified, value is set to 1
<port>	Port number to correct for source power. If unspecified, value is set to 1.
<num>	Tolerance value in dBm. Choose any number between 0 and 5
Examples	`SOUR:POW:CORR:COLL:ITER:NTOL .005` `source1:power2:correction:collect:iteration:ntolerance .1`
Query Syntax	SOURce:POWer:CORRection:COLLect:ITERation:NTOLerance?
Return Type	Numeric
Overlapped?	No
Default	.05

SOURce<cnum>:POWer<port>:CORRection:COLLect:METHod <char>

(Read-Write) Selects the calibration method to be used for the source power cal acquisition.

Parameters	
<cnum>	Any existing channel number. If unspecified, value is set to 1
<port>	Port number to correct for source power. If unspecified, value is set to 1.
<char>	Choose from: **NONE** - No Cal method **PMETer** - Power Meter is used for all readings. (same as "fast iteration" box not checked on dialog box) **PMReceiver** - Power meter for the first iteration; then use the reference receiver for remaining readings if necessary (same as "fast iteration" box checked on dialog box)
Examples	`SOUR:POW:CORR:COLL:METH PMET` `source1:power2:correction:collect:method pmreceiver`
Query Syntax	SOURce:POWer:CORRection:COLLect:METHod?
Return Type	Character
Overlapped?	No

Default NONE

SOURce<cnum>:POWer<port>:CORRection:COLLect:SAVE

(Write-only) Applies the array of correction values after a source power calibration sweep has completed. The source power correction will then be active on the specified source port for channel <cnum>. This command does NOT save the correction values. To save correction values, save an instrument / calibration state (*.cst file) after performing a source power cal.

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1
<port>	Port number to correct for source power. If unspecified, value is set to 1.

Examples

```
SOUR:POW:CORR:COLL:SAVE
source1:power2:correction:collect:save
```

Query Syntax Not Applicable

Overlapped? No

Default Not Applicable

SOURce<cnum>:POWer:CORRection:COLLect:<pmChan>SENsor[:FRANge] <num1>,<num2>

(Read-Write) Specifies the frequency range over which the power sensors connected to the specified channels (A and B) of the power meter can be used (minimum frequency, maximum frequency). If the power meter has only a single channel, that channel is considered channel A.

Parameters

<cnum> Any existing channel number. If unspecified, value is set to 1

<pmChan> Power Meter channel. Choose from:

A - Channel A

B - Channel B

<num1> Minimum frequency for the sensor. If a frequency unit is not specified, Hz is assumed. No limits are placed on this value.

<num2> Maximum frequency for the sensor. If a frequency unit is not specified, Hz is assumed. No limits are placed on this value.

Examples

```
SOUR:POW:CORR:COLL:ASEN 100E3, 3E9
source1:power2:correction:collect:bsensor:frange 10 MHz, 18 GHz
```

Query Syntax	SOURce:POWer:CORRection:COLLect:ASENsor[:FRANge]? SOURce:POWer:CORRection:COLLect:BSENsor[:FRANge]?
Return Type	Numeric
Overlapped?	No
Default	0,0

SOURce<cnum>:POWer:CORRection:COLLect:<pmChan>SENsor:RCFactor <num>

(Read-Write)) Specifies the reference cal factor for the power sensor connected to channel A or B of the power meter. If the power meter has only a single channel, that channel is considered channel A.

Note: If the sensor connected to the specified channel of the power meter contains cal factors in EPROM (such as the Agilent E-series power sensors), those will be the cal factors used during the calibration sweep. The reference cal factor value associated with this command, and any cal factors entered into the PNA for that sensor channel, will not be used.

Parameters	
<cnum>	Any existing channel number. If unspecified, value is set to 1
<pmChan>	Power Meter channel. Choose from: **A** - Channel A **B -** Channel B
<num>	Reference cal factor in percent. Choose any number between 1 and 150.
Examples	**SOUR:POW:CORR:COLL:ASEN:RCF 98.7** **source1:power2:correction:collect:bsensor:rcfactor 105**
Query Syntax	SOURce:POWer:CORRection:COLLect:ASENsor:RCFactor? SOURce:POWer:CORRection:COLLect:BSENsor:RCFactor?
Return Type	Numeric
Overlapped?	No
Default	100

SOURce<cnum>:POWer:CORRection:COLLect:<pmChan>SENsor:SELect

(Read-Write) Sets and returns the power sensor channel (A or B) to be used. This performs the same function as the **Use this sensor only** checkbox in the Power Sensor Settings dialog.

Note: This write portion of this command is only necessary when performing an SMC calibration.

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1
<pmChan>	Power Meter channel. Choose from: **A -** Channel A **B -** Channel B
Examples	**SOUR:POW:CORR:COLL:<pmChan>SEN:SEL 'Write** **source1:power2:correction:collect:bsensor:select? 1e9 'Read**
Query Syntax	SOURce:POWer:CORRection:COLLect:ASENsor:SELect? <Frequency> SOURce:POWer:CORRection:COLLect:BSENsor:SELect? <Frequency> Returns a boolean 1 or 0 (true or false) indicating whether the sensor is to be used at the specified frequency. If frequency checking is OFF, then the <Frequency> parameter is ignored. The query returns if the sensor is selected for ALL frequencies.
Return Type	Numeric
Overlapped?	No
Default	Not Applicable

SOURce<cnum>:POWer:CORRection:COLLect:TABLe:DATA <data>

(Read-Write) Read or write data into the selected table. If the selected table is a power sensor table, the data is interpreted as cal factors in units of percent. If the loss table is selected, the data is interpreted as loss in units of dB.

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1
<data>	Data to write into the selected table.
Examples	**SOURce:POWer:CORRection:COLLect:TABLe:DATA 0.12, 0.34, 0.56**
Query Syntax	SOURce<cnum>:POWer:CORRection:COLLect:TABLe:DATA? If the selected table is currently empty, no data is returned.
Return Type	Numeric - one number per table segment
Overlapped?	No

Default Not Applicable

SOURce<cnum>:POWer:CORRection:COLLect:TABLe:FREQuency <data>

(Read-Write) Read or write frequency values for the selected table (cal factor table for a power sensor, or the loss compensation table).

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1
<data>	Frequency data to write into the selected table.
Examples	`SOURce:POWer:CORRection:COLLect:TABLe:FREQuency 10E6, 1.5E9, 9E9`
Query Syntax	SOURce<cnum>:POWer:CORRection:COLLect:TABLe:FREQuency? If the selected table is currently empty, no data is returned.
Return Type	Numeric - one number per table segment
Overlapped?	No
Default	Not Applicable

SOURce<cnum>:POWer:CORRection:COLLect:TABLe:LOSS[:STATe] <ON | OFF>

(Read-Write) Indicates whether or not to adjust the power readings using the values in the loss table during a source power cal sweep.

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1
<ON|OFF>	**ON (or 1) -** turns use of the loss table ON. **OFF (or 0) -** turns use of the loss table OFF.
Examples	`SOUR:POW:CORR:COLL:TABL:LOSS ON` `source1:power2:correction:collect:table:loss:state off`
Query Syntax	SOURce:POWer:CORRection:COLLect:TABLe:LOSS[:STATe]?
Return Type	Boolean (1 = ON, 0 = OFF)
Overlapped?	No
Default	OFF (0)

SOURce<cnum>:POWer:CORRection:COLLect:TABLe:POINts?

(Read-only) Returns the number of segments that are currently in the selected table.

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1
Examples	**`SOUR:POW:CORR:COLL:TABL:POIN?`** **`source1:power2:correction:collect:table:points?`**
Return Type	Numeric
Overlapped?	No
Default	0

SOURce<cnum>:POWer:CORRection:COLLect:TABLe[:SELect] <char>

(Read-Write) Selects which table (cal factor table for a power sensor, or the loss compensation table) you want to write to or read from. Read or write using SOURce:POWer:CORRection:COLLect:TABLe:FREQuency and SOURce:POWer:CORRection:COLLect:TABLe:DATA

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1
<char>	Choose from: **NONE** - No table selected **ASENsor** - Cal Factor table for Power Sensor A **BSENsor** - Cal Factor table for Power Sensor B **LOSS** - Loss compensation table
Examples	**`SOUR:POW:CORR:COLL:TABL ASEN`** **`source1:power2:correction:collect:table:select bsensor`**
Query Syntax	SOURce:POWer:CORRection:COLLect:TABLe[:SELect]?
Return Type	Character
Overlapped?	No
Default	NONE

SOURce<cnum>:POWer<port>:CORRection:DATA <data>

(Read-Write) Writes and reads source power calibration data.

When querying source power calibration data, if no source power cal data exists for the specified channel and source port, no data is returned.

If a change in the instrument state causes interpolation and/or extrapolation of the source power cal, the correction data associated with this command correspond to the new instrument state (interpolated and/or extrapolated data).

If the channel is sweeping the source backwards, then the first data point is the highest frequency value; the last data point is the lowest. Use the SENS:X:VALues? command to return the X-axis values in the displayed order.

Parameters	
<cnum>	Any existing channel number. If unspecified, value is set to 1
<port>	Port number to correct for source power. If unspecified, value is set to 1.
<data>	Correction Data
Examples	**`SOURce1:POWer2:CORRection:DATA 0.12, -0.34, 0.56`**
Query Syntax	SOURce<cnum>:POWer<port>:CORRection:DATA?
Return Type	Depends on FORMat:DATA command
Overlapped?	No
Default	Not Applicable

SOURce<cnum>:POWer<port>:CORRection:LEVel <num>

(Read-Write) Specifies the power level that is expected at the desired reference plane (DUT input or output). This is not used for segment sweep with independent power levels or power sweeps.

Note: Although this command still works, it is recommended that you specify cal power by setting the test port power and offset value.

Parameters	
<cnum>	Any existing channel number. If unspecified, value is set to 1
<port>	Port number to correct for source power. If unspecified, value is set to 1.
<num>	Cal power level in dBm. Because this could potentially be at the output of a device-under-test, no limits are placed on this value here. It is realistically limited by the specifications of the device (power sensor) that will be used for measuring the power. The power delivered to the PNA receiver must never exceed PNA specifications for the receiver!
Examples	**`SOUR:POW:CORR:LEV 10`** **`source1:power2:correction:level 0 dbm`**
Query Syntax	SOURce:POWer:CORRection:LEVel?

Return Type Numeric

Overlapped? No

Default 0 dBm

SOURce<cnum>:POWer<port>:CORRection:OFFSet[:MAGNitude] <num>

(Read-Write) Sets or returns a power level offset from the PNA test port power. This can be a gain or loss value (in dB) to account for components you connect between the source and the reference plane of your measurement. For example, specify 10 dB to account for a 10 dB amplifier at the input of your DUT.

Cal power is the sum of the test port power setting and this offset value. Following the calibration, the PNA power readouts are adjusted to the cal power.

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1
<port>	Port number to correct for source power. If unspecified, value is set to 1.
<num>	Gain or loss value in dB. Choose a value between -200 and 200

Examples

```
SOUR:POW:CORR:OFFS 10
source1:power2:correction:offset:magnitude -3
```

Query Syntax SOURce:POWer:CORRection:OFFSet[:MAGNitude]?

Return Type Numeric

Overlapped? No

Default 0 dB

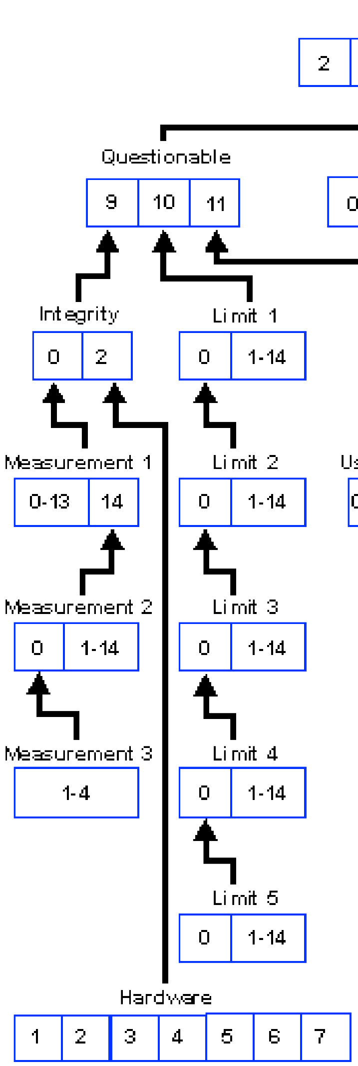
SOURce<cnum>:POWer<port>:CORRection[:STATe] <ON|OFF>

(Read-Write) Enables and disables source power correction for the specified port on the specified channel.

Parameters	
<cnum>	Any existing channel number. If unspecified, value is set to 1
<port>	Port number to correct for source power. If unspecified, value is set to 1.
<ON|OFF>	ON (or 1) turns source power correction ON. OFF (or 0) - turns source power correction OFF.
Examples	**SOUR:POW:CORR ON** **source1:power2:correction:state off**
Query Syntax	SOURce:POWer:CORRection[:STATe]?
Return Type	Boolean (1 = ON, 0 = OFF)
Overlapped?	No
Default	OFF (0)

Status Register Commands

The status registers enable you to query the state of selected events that occur in the analyzer.



te: This documentation requires familiarity with the "Standard Status Data Structure - Register Model" as defined in IEEE Std 488.2-1992.

- Click on a blue keyword to view the command details.
- See a List of all commands in this block.
- Learn about Status Registers

Note: Any bit not shown in the registers is not used but may be reserved for future use.

Status Byte Register

Summarizes the states of the other registers and monitors the analyzer's output queue. It also generates **service requests**. The Enable register is called the Service Request Enable Register.

Commands	**Description**
*CLS	Clears ALL "event" registers and the SCPI Error / Event queue. The corresponding ENABLE registers are unaffected.
*STB?	Reads the value of the analyzer's status byte. The byte remains after being read.
*SRE?	Reads the current state of the Service Request **Enable** Register.
*SRE <*num*>	Sets bits in the Service Request **Enable** register. The current setting of the SRE register is stored in non-volatile memory. Use *SRE 0 to clear the enable. <*num*> Combined value of the weights for bits to be set.

Bit	Weight	Description	Bit is set to 1 when the following conditions exist:
2	4	Error / Event queue Summary (EAV)	the Error / Event queue is not empty. To read the the error message, use SYST:ERR?
3	8	Questionable Register Summary	any enabled bit in the **questionable** event status register is set to 1
4	16	Message Available	the output queue is not empty
5	32	Standard Event Register Summary	any enabled bit in the **standard** event status register is set to 1
6	64	Request Service	any of the other bits in the status byte register is set to 1 (used to alert the controller of a service request within the analyzer). This bit cannot be disabled.
7	128	Operation Register Summary	any enabled bit in the standard **operation** event status register is set to 1

STATus:QUEStionable:<keyword>

Summarizes conditions that monitor the quality of measurement data.

<keyword>	Example
:CONDition?	`STAT:QUES:COND?`
:ENABle <bits>	`STAT:QUES:ENAB 1024`
[:EVENt]?	`STAT:QUES?`
:NTRansition <bits>	`STAT:QUES:NTR 1024`
:PTRansition <bits>	`STAT:QUES:PTR 0`

Bit	Weight	Description	Bit is set to 1 when the following conditions exist:
9	512	Integrity Reg summary	any enabled bit in the **Integrity** event register is set to 1
10	1024	Limit Registers summary	any enabled bit in the **Limit** event registers is set to 1
11	2048	Define Registers summary	any enabled bit in the **Define** event registers is set to 1

STATus:QUEStionable:INTegrity <keyword>

Summarizes conditions in the Measurement Integrity register.

<keyword>	Example
:CONDition?	`STAT:QUES:INT:COND?`
:ENABle <bits>	`STAT:QUES:INT:ENAB 1024`
[:EVENt]?	`STAT:QUES:INT?`
:NTRansition <bits>	`STAT:QUES:INT:NTR 1024`
:PTRansition <bits>	`STAT:QUES:INT:PTR 0`

Bit	Weight	Description	Bit is set to 1 when the following conditions exist:
0	1	Measurement Summary	any bit in the **Measurement Integrity** event register is set to 1
2	2	Hardware Summary	any bit in the **Hardware** event register is set to 1

STATus:QUEStionable:INTegrity:HARDware<keyword>

Monitors the status of hardware failures.

<keyword>	Example
:CONDition?	`STAT:QUES:INT:HARD:COND?`
:ENABle <bits>	`STAT:QUES:INT:HARD:ENAB 1024`
[:EVENt]?	`STAT:QUES:INT:HARD?`
:NTRansition <bits>	`STAT:QUES:INT:HARD:NTR 1024`
:PTRansition <bits>	`STAT:QUES:INT:HARD:PTR 0`

Bit	Weight	Description	Bit is set to 1 when the following conditions exist:
1	2	Phase Unlock	the source has lost phaselock, possibly caused by a reference channel open or a hardware failure.
2	4	Unleveled	the source power is unleveled. This could be caused by a source set for more power than it can deliver at the tuned frequency. Or it could be caused by a hardware failure.
3	8	Not used	N/A
4	16	EE Write Failed	an attempted write to the EEPROM has failed, possibly caused by a hardware failure.
5	32	Not used	N/A
6	64	Ramp Cal Failed	the analyzer was unable to calibrate the analog ramp generator due to a possible hardware failure.
7	128	Not used	N/A

STATus:QUEStionable:INTegrity:MEASurement<n> <keyword>

Monitors the lag between changing a channel settings and when the data is ready to query out.

When you change the channel state (start/stop freq, bandwidth, and so on), then the questionable bit for that channel gets set. This indicates that your desired channel state does not yet match the data you would get if querying a data trace. When the next complete sweep has been taken (without aborting in the middle), and the data trace matches the channel state that produced it, the bit is cleared for that channel.

<n>	Measurement register number. Choose from 1 to 3
<keyword>	**Example**
:CONDition?	`STAT:QUES:INT:MEAS1:COND?`
:ENABle <bits>	`STAT:QUES:INT:MEAS2:ENAB 1024`
[:EVENt]?	`STAT:QUES:INT:MEAS3?`
:NTRansition <bits>	`STAT:QUES:INT:MEAS2:NTR 1024`
:PTRansition <bits>	`STAT:QUES:INT:MEAS1:PTR 0`

		Measurement Register <n>			
Bit	**Weight**	**1**	**2**	**3**	**Bit is set to 1 when the following conditions exist:**
0	1	1	Summary from Meas Reg 3		a setting change on this channel has occurred and the data does not yet reflect that change.
1	2	2	15	29	a setting change on this channel has occurred and the data does not yet reflect that change.
2	4	3	16	30	a setting change on this channel has occurred and the data does not yet reflect that change.
3	8	4	17	31	a setting change on this channel has occurred and the data does not yet reflect that change.
4	16	5	18	32	a setting change on this channel has occurred and the data does not yet reflect that change.
5	32	6	19		a setting change on this channel has occurred and the data does not yet reflect that change.
6	64	7	20		a setting change on this channel has occurred and the data does not yet reflect that change.
7	128	8	21		a setting change on this channel has occurred and the data does not yet reflect that change.

8	256	9	22		a setting change on this channel has occurred and the data does not yet reflect that change.
9	512	10	23		a setting change on this channel has occurred and the data does not yet reflect that change.
10	1024	11	24		a setting change on this channel has occurred and the data does not yet reflect that change.
11	2048	12	25		a setting change on this channel has occurred and the data does not yet reflect that change.
12	4096	13	26		a setting change on this channel has occurred and the data does not yet reflect that change.
13	8192	14	27		a setting change on this channel has occurred and the data does not yet reflect that change.
14	16384	Summary from Meas Reg 2	28		a setting change on this channel has occurred and the data does not yet reflect that change.

STATus:QUEStionable:LIMit<n> <keyword>

Monitors and summarizes the status of limit line failures. When a trace fails, the representative bit is set to 1. Bit 0 is used to summarize failures in the registers that follow. For example, Limit 3 register, bit 0, summarizes the failures from registers 4 and 5.
All enable bits are set to 1 by default. To find the measurement number, use Calc:Par:Mnum

<n>	Limit register: Choose from 1 to 5.
<keyword>	**Example**
:CONDition?	`STAT:QUES:LIM4:COND?`
:ENABle <bits>	`STAT:QUES:LIM1:ENAB 1024`
[:EVENt]?	`STAT:QUES:LIM3?`
:NTRansition <bits>	`STAT:QUES:LIM2:NTR 1024`
:NTRansition?	`STAT:QUES:LIM1:NTR?`
:PTRansition <bits>	`STAT:QUES:LIM5:PTR 0`
:PTRansition?	`STAT:QUES:LIM1:PTR?`

		Limit Register <n>					
Bit	**Weight**	**1**	**2**	**3**	**4**	**5**	**Bit is set to 1 when the following conditions exist:**
0	1	2, 3, 4, 5	3, 4, 5	4, 5	5		Summary - Any point from these registers fails
		Trace Numbers					
1	2	1	15	29	43	57	any point on trace fails the limit test
2	4	2	16	30	44	58	any point on trace fails the limit test
3	8	3	17	31	45	59	any point on trace fails the limit test
4	16	4	18	32	46	60	any point on trace fails the limit test
5	32	5	19	33	47	61	any point on trace fails the limit test
6	64	6	20	34	48	62	any point on trace fails the limit test
7	128	7	21	35	49	63	any point on trace fails the limit test
8	256	8	22	36	50	64	any point on trace fails the limit test
9	512	9	23	37	51		any point on trace fails the limit test
10	1024	10	24	38	52		any point on trace fails the limit test
11	2048	11	25	39	53		any point on trace fails the limit test
12	4096	12	26	40	54		any point on trace fails the limit test
13	8192	13	27	41	55		any point on trace fails the limit test
14	16384	14	28	42	56		any point on trace fails the limit test

STATus:QUEStionable:DEFine<keyword>

Summarizes conditions in the Questionable:Define:User<1|2|3> event registers.

<keyword>	**Example**
:CONDition?	`STAT:QUES:DEF:COND?`
:ENABle <bits>	`STAT:QUES:DEF:ENAB 1024`
[:EVENt]?	`STAT:QUES:DEF?`
:NTRansition <bits>	`STAT:QUES:DEF:NTR 1024`
:PTRansition <bits>	`STAT:QUES:DEF:PTR 0`

Bit	Weight	Description	Bit is set to 1 when the following conditions exist:
1	2	USER1	any bit in the **USER1** event register is set to 1
2	4	USER2	any bit in the **USER2** event register is set to 1
3	8	USER3	any bit in the **USER3** event register is set to 1

STATus:QUEStionable:DEFine:USER<1|2|3><keyword>

Monitors conditions that you define and map in any of the three QUES:DEF:USER event registers.

<keyword>	**Example**
:ENABle <bits>	`STAT:QUES:DEF:USER1:ENABle 1024`
[:EVENt]?	`STAT:QUES:DEF:USER1?`
:MAP <bit>,<error>	`STAT:QUES:DEF:USER1:MAP 0,-113 'when error -113 occurs, bit 0 in USER1 will set to 1.`

Bit	Weight	Description	Bit is set to 1 when the following conditions exist:
0	1	for user	user defined
1	2	for user	user defined
2	4	for user	user defined
3	8	for user	user defined
4	16	for user	user defined
5	32	for user	user defined
6	64	for user	user defined
7	128	for user	user defined
8	256	for user	user defined
9	512	for user	user defined
10	1024	for user	user defined
11	2048	for user	user defined
12	4096	for user	user defined
13	8192	for user	user defined
14	16384	for user	user defined

Standard Event Status Register

Monitors "standard" events that occur in the analyzer. This register can only be cleared by:

- a Clear Command (*CLS).
- reading the Standard Enable Status Register (*ESE?).
- a power-on transition. The analyzer clears the register and then records any transitions that occur, including setting the Power On bit (7).

Commands	Description
*ESE?	Reads the settings of the standard event **ENABLE** register.
*ESE ***<bits>***	Sets bits in the standard event **ENABLE** register. The current setting is saved in non-volatile memory. ***<bits>*** The sum of weighted bits in the register. Use *ESE 0 to clear the enable register.
*ESR?	Reads and clears the **EVENT** settings in the Standard Event Status register.
*OPC	Sets bit 0 when the overlapped command is complete. (see Understanding Command Synchronization / OPC).
*OPC?	Operation complete query - read the Operation Complete bit (0).

Bit	Weight	Description	Bit is set to 1 when the following conditions exist:
0	1	Operation Complete	the two following events occur **in order:** 1. the *OPC command is sent to the analyzer 2. the analyzer completes all pending overlapped commands
1	NA	Request Control	Not Supported - the analyzer application is not configured to control GPIB operation
2	4	Query Error	a query error is detected indicating: - an attempt to read data from the output queue when no data was present **OR** - data in the output queue was lost, as in an overflow
4	16	Execution Error	an execution error is detected indicating: - a <PROGRAM DATA> element was outside the legal range or inconsistent with the operation of the analyzer **OR** **-** the analyzer could not execute a valid command due to some internal condition
5	32	Command Error	a command error is detected indicating that the analyzer received a command that: • did not follow proper syntax • was misspelled • was an optional command it does not implement

7	128	Power ON	Power to the analyzer has been turned OFF and then ON since the last time this register was read.

STATus:OPERation<keyword>

Summarizes conditions in the Averaging and Operation:Define:User<1|2|3> event registers.

<keyword>	**Example**
:CONDition?	`STAT:OPER:COND?`
:ENABle <bits>	`STAT:OPER:ENAB 1024`
[:EVENt]?	`STAT:OPER?`
:NTRansition <bits>	`STAT:OPER:NTR 1024`
:PTRansition <bits>	`STAT:OPER:PTR 0`

Bit	Weight	Description	Bit is set to 1 when the following conditions exist:
8	256	Averaging summary	either enabled bit in the **Averaging summary** event register is set to 1
9	512	User Defined summary	
10	1024	Device summary	either enabled bit in the **Device summary** event register is set to 1

STATus:OPERation:AVERaging<n> <keyword>

Monitors and summarizes the status of Averaging on traces 1 to 64. When averaging for a trace is complete, the representative bit is set to 1. Bit 0 is used to summarize the status in the registers that follow. For example, Average 3 register, bit 0, summarizes the status from registers 4 and 5. All enable bits are set to 1 by default. To find the measurement number, use Calc:Par:Mnum.

<n> Averaging Register. Choose from 1 to 5

<keyword>	**Example**
:CONDition?	`STAT:OPER:AVER1:COND?`
:ENABle <bits>	`STAT:OPER:AVER1:ENAB 1024`
[:EVENt]?	`STAT:OPER:AVER1?`
:NTRansition <bits>	`STAT:OPER:AVER1:NTR 1024`
:PTRansition <bits>	`STAT:OPER:AVER1:PTR 0`

		Averaging Register <n>					
Bit	**Weight**	**1**	**2**	**3**	**4**	**5**	**Bit is set to 1 when the following conditions exist:**
0	1	2, 3, 4, 5	3, 4, 5	4, 5	5		any enabled bit in these registers is set to 1(Summary Bit)
		Trace Numbers					
1	2	1	15	29	43	57	Averaging on this trace is complete
2	4	2	16	30	44	58	Averaging on this trace is complete
3	8	3	17	31	45	59	Averaging on this trace is complete
4	16	4	18	32	46	60	Averaging on this trace is complete
5	32	5	19	33	47	61	Averaging on this trace is complete
6	64	6	20	34	48	62	Averaging on this trace is complete
7	128	7	21	35	49	63	Averaging on this trace is complete
8	256	8	22	36	50	64	Averaging on this trace is complete
9	512	9	23	37	51		Averaging on this trace is complete
10	1024	10	24	38	52		Averaging on this trace is complete
11	2048	11	25	39	53		Averaging on this trace is complete
12	4096	12	26	40	54		Averaging on this trace is complete
13	8192	13	27	41	55		Averaging on this trace is complete
14	16384	14	28	42	56		Averaging on this trace is complete

STATus:OPERation:DEFine<keyword>

Summarizes conditions in the OPERation:Define:User<1|2|3> event registers.

<keyword>	**Example**
:CONDition?	`STAT:OPER:DEF:COND?`
:ENABle <bits>	`STAT:OPER:DEF:ENAB 12`
[:EVENt]?	`STAT:OPER:DEF?`
:NTRansition <bits>	`STAT:OPER:DEF:NTR 12`
:PTRansition <bits>	`STAT:OPER:DEF:PTR 0`

Bit	Weight	Description	Bit is set to 1 when the following conditions exist:
1	2	USER1	any bit in the **USER1** event register is set to 1
2	4	USER2	any bit in the **USER2** event register is set to 1
3	8	USER3	any bit in the **USER3** event register is set to 1

STATus:OPERation:DEFine:USER<1|2|3><keyword>

Monitors conditions that you define and map in any of the three OPER:DEF:USER event registers.

<keyword>	**Example**
:ENABle <bits>	`STAT:OPER:DEF:USER1:ENAB 1024`
[:EVENt]?	`STAT:OPER:DEF:USER1?`
:MAP <bit>,<error>	`STAT:OPER:DEF:USER1:MAP 0,-113 'when error -113 occurs, bit 0 in USER1 will set to 1.`

Bit	Weight	Description	Bit is set to 1 when the following conditions exist:
0	1	for user	user defined
1	2	for user	user defined
2	4	for user	user defined
3	8	for user	user defined
4	16	for user	user defined
5	32	for user	user defined
6	64	for user	user defined
7	128	for user	user defined
8	256	for user	user defined
9	512	for user	user defined
10	1024	for user	user defined
11	2048	for user	user defined
12	4096	for user	user defined
13	8192	for user	user defined
14	16384	for user	user defined

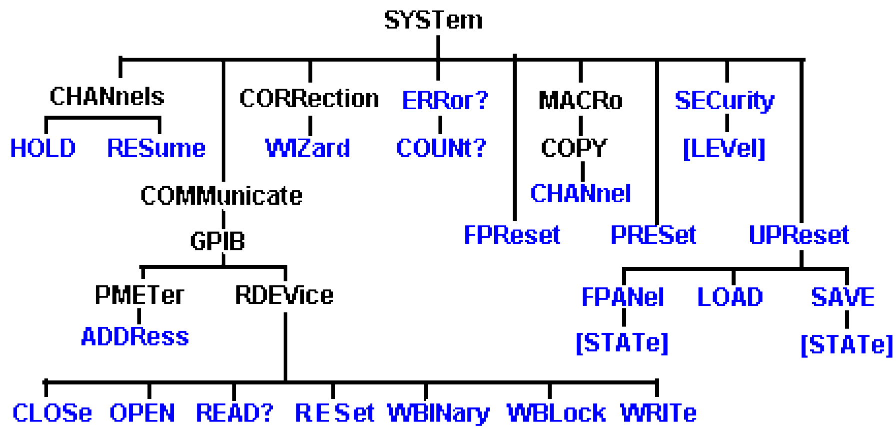
STATus:OPERation:DEVice<keyword>

Summarizes conditions in the OPERation:DEVice event registers.

<keyword>	Example
:CONDition?	`STAT:OPER:DEV:COND?`
:ENABle <bits>	`STAT:OPER:DEV:ENAB 16`
[:EVENt]?	`STAT:OPER:DEV?`
:NTRansition <bits>	`STAT:OPER:DEV:NTR 16`
:PTRansition <bits>	`STAT:OPER:DEV:PTR 0`

Bit	Weight	Description	Bit is set to 1 when the following conditions exist:
0	1	Unused	
1	2	Unused	
2	4	Unused	
3	8	Unused	
4	16	Sweep Completed	When sweep is complete
5	32	Unused	
6	64	Unused	
7	128	Unused	
8	256	Unused	
9	512	Unused	
10	1024	Unused	
11	2048	Unused	
12	4096	Unused	
13	8192	Unused	
14	16384	Unused	

System Commands



- Click on a blue keyword to view the command details.
- See a List of all commands in this block.
- Learn about Preset

SYSTem:CHANNels:HOLD

(Write-only) Places all channels in hold mode. To place a single channel in hold mode, use SENS:SWE:MODE.

Examples	**SYST:CHAN:HOLD**
Query Syntax	Not Applicable
Overlapped?	No
Default	Not Applicable

SYSTem:CHANNels:RESume

(Write-only) Resumes the trigger mode of all channels that was in effect before sending SYSTem:CHANNels:HOLD (must be sent before SYST:CHAN:RESume).

Examples	**SYST:CHAN:RES**
Query Syntax	Not Applicable
Overlapped?	No
Default	Not Applicable

SYSTem:COMMunicate:GPIB:PMETer:ADDRess <num>

(Read-Write) Specifies the GPIB address of the power meter to be used in a source power calibration. When performing a source power cal, the PNA will search VISA interfaces that are configured in the Agilent IO LIbraries on the PNA.

Parameters	
<num>	GPIB address of the power meter. Choose any integer between 0 and 30.
Examples	`SYST:COMM:GPIB:PMET:ADDR 13` `system:communicate:gpib:pmeter:address 14`
Query Syntax	SYSTem:COMMunicate:GPIB:PMETer:ADDRess?
Return Type	Numeric
Overlapped?	No
Default	13

SYSTem:COMMunicate:GPIB:RDEVice:CLOSe <ID>

(Write only) Closes the remote GPIB session. This command should be sent when ending every successful OPEN session.

Parameters	
<ID>	Session identification number that was returned with the OPEN? command.
Examples	See an example program
Query Syntax	Not Applicable
Overlapped?	No
Default	Not Applicable

SYSTem:COMMunicate:GPIB:RDEVice:OPEN <bus>, <addr>, <timeout>

(Read-Write) Initiates a GPIB pass-through session. First send this OPEN command, then send the OPEN query to read the session ID number. An existing GPIB pass-through session remains open after an instrument preset.

To learn more about GPIB pass-through capability, see the example program.

Parameters

<bus>	Bus ID number. You can find the USB-GPIB adapter bus number by looking at the dialog that appears when the USB-GPIB device is connected.
<addr>	GPIB Address of the device to be controlled
<timeout>	The amount of time (in milliseconds) to wait for a response from the remote device after sending a command. A "timeout" error is displayed after this time has passed without a response.
Examples	See an example program
Query Syntax	SYSTem:COMMunicate:GPIB:RDEVice:OPEN? Returns the session identification number that is used when communicating with this device.
Return Type	Numeric
Overlapped?	No
Default	Not Applicable

SYSTem:COMMunicate:GPIB:RDEVice:READ? <ID>

(Read-only) Returns data from the GPIB pass-through device.

Parameters

<ID>	Session identification number that was returned with the OPEN? command.
Examples	See an example program
Return Type	String
Overlapped?	No
Default	Not Applicable

SYSTem:COMMunicate:GPIB:RDEVice:RESet

(Write-only)   Performs the same function as SYST:COMM:GPIB:RDEV:CLOS except that ALL pass-through sessions are closed.

Examples `SYST:COMM:GPIB:RDEV:RES`

Query Syntax Not Applicable

Overlapped? No

Default Not Applicable

SYSTem:COMMunicate:GPIB:RDEVice:WBINary <ID>,<data>

(Write-only) Sends data to a GPIB pass-through device. This command requires a header that specifies the size of the data to be written. The header (described below) is not passed along to the device.

Use this command if too many embedded quotes prevent you from using SYST:COMM:GPIB:RDEV:WRIT.

Use SYST:COMM:GPIB:RDEV:OPEN to open the pass through session.

Parameters

<ID> Session identification number that was returned with the OPEN? command.

<data> Data to be sent to the GPIB pass-through device. Use the following syntax:

```
#<num digits><byte count><data bytes><NL><END>
```

<num_digits> specifies how many digits are contained in <byte_count>

<byte_count> specifies how many data bytes will follow in <data bytes>

Examples `SYSTem:COMMunicate:GPIB:RDEVice:WBINary 101,#17ABC+XYZ<nl><end>`

- always sent before data.

1 - specifies that the byte count is one digit (7).

7 - specifies the number of data bytes that will follow, not counting <NL><END>.

ABC+XYZ - Data block

<nl><end> - always sent at the end of block data.

Query Syntax Not Applicable

Overlapped? No

Default Not Applicable

SYSTem:COMMunicate:GPIB:RDEVice:WBLock <ID>,<data>

(Write-only) Same as SYSTem:COMM:GPIB:RDEV:WBIN (above) but the header **IS** passed along to the device. Use this command if too many embedded quotes prevent you from using SYST:COMM:GPIB:RDEV:WRIT.

Parameters	
<ID>	Session identification number that was returned with the OPEN? command.
<data>	Data to be sent to the GPIB pass-through device. See previous command.
Examples	See previous example.
Query Syntax	Not Applicable
Overlapped?	No
Default	Not Applicable

SYSTem:COMMunicate:GPIB:RDEVice:WRITe <ID>,<string>

(Write-only) Sends ASCII string data to the GPIB pass-through device.

Parameters	
<ID>	Session identification number that was returned with the OPEN? command.
<string>	Commands to be sent to the GPIB pass-through device.
Examples	See an example program
Query Syntax	Not Applicable
Overlapped?	No
Default	Not Applicable

SYSTem:CORRection:WIZard <char>

(Write-only) Launches either the Calibration Wizard or the Version 2 Calibration Kit File Manager dialog box.

Parameters

<char>	Choose from: MAIN - Launches the Calibration Wizard CKIT - Launches the Version 2 Calibration Kit File Manager dialog box. Both display on the PNA screen.
Examples	**SYST:CORR:WIZ MAIN** **system:correction:wizard ckit**
Query Syntax	Not Applicable?
Overlapped?	No
Default	Not Applicable

SYSTem:ERRor?

(Read-only) Returns the next error in the error queue. Each time the analyzer detects an error, it places a message in the error queue. When the `SYSTEM:ERROR?` query is sent, one message is moved from the error queue to the output queue so it can be read by the controller. Error messages are delivered to the output queue in the order they were received. The error queue is cleared when any of the following conditions occur:

- When the analyzer is switched ON.
- When the `*CLS` command is sent to the analyzer.
- When all of the errors are read.

If the error queue overflows, the last error is replaced with a `"Queue Overflow"` error. The oldest errors remain in the queue and the most recent error is discarded.

Examples	**SYST:ERR?** **system:error?**
Overlapped?	No
Default	Not applicable

SYSTem:ERRor:COUNt?

(Read-only) Returns the number of errors in the error queue. Use SYST:ERR? to read an error.

Examples	**SYST:ERR:COUN?** **system:error:count?**
Overlapped?	No
Default	Not applicable

SYSTem:FPReset

(Write-only) Performs a standard Preset, then deletes the default trace, measurement, and window. The PNA screen becomes blank.

Examples	`SYST:FPR` `system:fpreset`
Overlapped?	No
Default	Not applicable

SYSTem:MACRo:COPY:CHANnel<cnum>[:TO] <num>

(Write-only) Sets up channel <num> as a copy of channel <cnum>. Learn more about copy channels.

Parameters	
<cnum>	Any existing channel number. If unspecified, value is set to 1.
<num>	Number of the channel which is to become a copy of channel <cnum>.
Examples	`SYST:MACR:COPY:CHAN1 2` `system:macro:copy:channel2:to 3`
Query Syntax	Not Applicable
Overlapped?	No
Default	Not Applicable

SYSTem:PRESet

(Write-only) Deletes all traces, measurements, and windows. In addition, resets the analyzer to factory defined default settings and creates a S11 measurement named "CH1_S11_1". For a list of default settings, see Preset.

Regardless of the state of the User Preset Enable checkbox, the SYST:PRESet command will always preset the PNA to the factory preset settings, and SYST:UPReset will always perform a User Preset.

If the PNA display is disabled with DISP:ENAB OFF then SYST:PRES will NOT enable the display.

This command performs the same function as *RST.

Examples	`SYST:PRES` `system:preset`
Overlapped?	No
Default	Not applicable

SYSTem:SECurity[:LEVel] <char>

(Read-Write) Sets and returns the display of frequency information on the PNA screen and printouts. Learn more about security level.

Parameters	
<char>	Choose from: **None -** ALL frequency information is displayed **Low -** NO frequency information is displayed. Frequency information can be redisplayed using the Security Setting dialog box or this command.. **High -** NO frequency information is displayed. Frequency information can be redisplayed ONLY by performing a Preset or recalling an instrument state with None or Low security settings, or using this command.
Examples	`SYST:SEC LOW` `system:security:level high`
Query Syntax	SYSTem:SECurity[:LEVel]?
Return Type	Character
Overlapped?	No
Default	None

SYSTem:UPReset

(Write-only) Performs a User Preset. There must be an active User Preset state file (see Load and Save) or an error will be returned. Learn more about User Preset.

Regardless of the state of the User Preset Enable checkbox, the SYST:PRESet command will always preset the PNA to the factory preset settings, and SYST:UPReset will always perform a User Preset.

Examples	`SYST:UPReset` `system:upreset`
Query Syntax	Not Applicable
Overlapped?	No
Default	Not Applicable

SYSTem:UPReset:FPANel[:STATe] <bool>

(Read-Write)   'Checks' and 'clears' the enable box on the User Preset dialog box. This only affects subsequent Presets from the front panel user interface.

Regardless of the state of the User Preset Enable checkbox, the SYST:PRESet command will always preset the PNA to the factory preset settings, and SYST:UPReset will always perform a User Preset.

Parameters

<bool>	Front Panel User Preset State. Choose from: **0** – User Preset OFF **1** – User Preset ON
Examples	**`SYST:UPR:FPAN 1`** **`system:upreset:fpanel:state 0`**
Query Syntax	SYSTem:UPREset:FPANel[:STATe]?
Return Type	Boolean
Overlapped?	No
Default	0

SYSTem:UPReset:LOAD <file>

(Read-Write) Loads an existing instrument state file (.sta or .cst) to be used for User Preset. Subsequent execution of SYSTem:UPReset will cause the PNA to assume this instrument state.

Regardless of the state of the User Preset Enable checkbox, the SYST:PRESet command will always preset the PNA to the factory preset settings, and SYST:UPReset will always perform a User Preset.

Learn more about User Preset.

Parameters

<file>	String - Name of the file to be loaded. The default folder "C:\Program Files\Agilent\Network Analyzer\Documents" is used if unspecified. Change the default folder name using MMEMory:CDIRectory.
Examples	**`SYST:UPR:LOAD '1MHzto20GHzUserPreset.cst'`** **`system:upreset:load 'C:\Documents and Settings\Administrator\My Documents\NewUserPreset.cst'`**
Query Syntax	Not Applicable
Overlapped?	No
Default	Not Applicable

SYSTem:UPReset:SAVE[:STATe]

(Write-only) Saves the current instrument settings as UserPreset.sta. Subsequent execution of SYSTem:UPReset will cause the PNA to assume this instrument state.

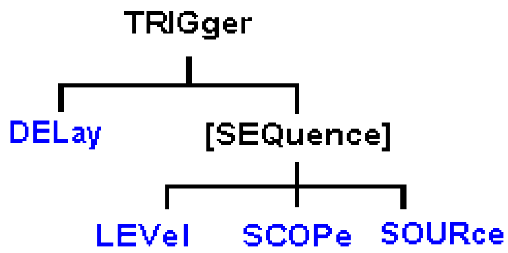
Regardless of the state of the User Preset Enable checkbox, the SYST:PRESet command will always preset the PNA to the factory preset settings, and SYST:UPReset will always perform a User Preset.

Learn more about User Preset.

Examples	`SYST:UPR:SAVE` `system:upreset:save:state`
Query Syntax	Not Applicable
Overlapped?	No
Default	Not Applicable

Trigger Commands

Starts or ends a measurement sequence.



- Click on a blue keyword to view the command details.
- See a List of all commands in this block.
- Learn about Triggering

TRIGger:DELay <num>

(Read-Write) Sets and reads the trigger delay for all channels. This delay is only applied while in External Trigger mode. After an external trigger is applied, the start of the sweep is held off for an amount of time equal to the delay setting plus any inherent latency.

To apply delay to a single channel, use SENS:SWE:TRIG:DELay

Parameters

<num>	Delay value in seconds. Choose from 0 to 1.
Examples	**TRIG:DEL .0003** **Sets the trigger delay to 300 microseconds. The sweep will not start until approximately 300 microseconds after an external trigger is applied.**
Query Syntax	TRIGger:DELay?
Return Type	Numeric
Overlapped?	No
Default	0

TRIGger[:SEQuence]:LEVel <char>

OBSOLETE COMMAND - Replaced with CONTrol:SIGNal.command.

(Read-Write) Triggers either on a **High or Low** level trigger signal. This setting only has an effect when TRIG:SOURce EXTernal is selected.

Parameters	
<char>	Choose from: **HIGH -** analyzer triggers on TTL **High** **LOW -** analyzer triggers on TTL **Low**
Examples	`TRIG:LEV HIGH` `trigger:sequence:level low`
Query Syntax	TRIGger[:SEQuence]:LEVel?
Return Type	Character
Overlapped?	No
Default	LOW

TRIGger[:SEQuence]:SCOPe <char>

(Read-Write) Specifies whether triggers are applied to all channels or the current channel.

Parameters	
<char>	Choose from: **ALL -** triggers all channels. Also sets SENS:SWEep:TRIG:POINt OFF on **ALL** channels. **CURRent -** trigger only one channel at a time. With each trigger signal, the channel is incremented to the next triggerable channel.
Examples	`TRIG:SCOP ALL` `trigger:sequence:scope current`
Query Syntax	TRIGger[:SEQuence]:SCOPe?
Return Type	Character
Overlapped?	No
Default	ALL

TRIGger[:SEQuence]:SOURce <char>

(Read-Write) Sets the source of the sweep trigger signal. This command is a super-set of INITiate:CONTinuous, which can NOT set the source to External. To configure external triggering, see CONTrol:SIGNal.

Parameters	
<char>	Choose from: **EXTernal -** external (rear panel) source. **IMMediate** - internal source sends continuous trigger signals **MANual -** sends one trigger signal when manually triggered from the front panel or INIT:IMM is sent.
Examples	`TRIG:SOUR EXT` `trigger:sequence:source immediate`
Query Syntax	TRIGger[:SEQuence]:SOURce?
Return Type	Character
Overlapped?	No
Default	IMMediate

Catalog Measurements using SCPI

This Visual Basic Program does the following:

- Catalogs the currently defined measurements, windows, and traces
- Selects a measurement for further definition
- Adds a Title to the window

To run this program, you need:

- An established GPIB interface connection

See Other SCPI Example Programs

```
Dim Meas as String
Dim Win as String
Dim Trace as String

'Read the current measurements in Channel 1
GPIB.Write "CALCulate1:PARameter:CATalog?"
Meas = GPIB.Read
MsgBox ("Ch1 Measurments: " & Meas)

'Read the current windows
GPIB.Write "DISPlay:CATalog?"
Win = GPIB.Read
MsgBox ("Windows: " & Win)

'Read current traces in window 1
GPIB.Write "DISPlay:WINDow1:CATalog?"
Trace = GPIB.Read
MsgBox ("Traces in Window1: " & Win)
```

Create a Measurement using SCPI

This VBScript program creates a new S21 measurement and displays it on the PNA screen.

The SCPI commands in this example are sent over a COM interface using the SCPIStringParser object. You do NOT need a GPIB connection to run this example.

This VBScript (*.vbs) program can be run as a macro in the PNA. To do this, copy the following code into a text editor file such as Notepad and save it on the PNA hard drive as NewMeas.vbs. Learn how to setup and run the macro.

See Other SCPI Example Programs

```
Dim app
Dim scpi
' Create / Get the PNA application.
Set app = CreateObject("AgilentPNA835x.Application")
Set scpi = app.ScpiStringParser

' A comment
'Preset the analyzer
scpi.Execute ("SYST:FPReset")
' Create and turn on window 1
scpi.Execute ("DISPlay:WINDow1:STATE ON")
'Define a measurement name, parameter
scpi.Execute ("CALCulate:PARameter:DEFine 'MyMeas',S21")
'Associate ("FEED") the measurement name ('MyMeas') to WINDow (1), and give the new
TRACe a number (1).
scpi.Execute ("DISPlay:WINDow1:TRACe1:FEED 'MyMeas'")
```

Setup Sweep Parameters using SCPI

This Visual Basic program sets up sweep parameters on the Channel 1 measurement.
To run this program, you need:

- An established GPIB interface connection

See Other SCPI Example Programs

```
GPIB.Write "SYSTem:PRESet"
'Select the measurement
GPIB.Write "CALCulate:PARameter:SELect 'CH1_S11_1'"
'Set sweep type to linear
GPIB.Write "SENSe1:SWEep:TYPE LIN"

'Set IF Bandwidth to 700 Hz
GPIB.Write "SENSe1:BANDwidth 700"

'Set Center and Span Freq's to 4 GHz
GPIB.Write "SENSe1:FREQuency:CENTer 4ghz"
GPIB.Write "SENSe1:FREQuency:SPAN 4ghz"

'Set number of points to 801
GPIB.Write "SENSe1:SWEep:POINts 801"

'Set sweep generation mode to Analog
GPIB.Write "SENSe1:SWEep:GENeration ANAL"

'Set sweep time to Automatic
GPIB.Write "SENSe1:SWEep:TIME:AUTO ON"

'Query the sweep time
GPIB.Write "SENSe1:SWEep:TIME?
SweepTime = GPIB.Read
```

Setup the Display using SCPI

This Visual Basic program:

- Sets data formatting
- Turns ON the Trace, Title, and Frequency Annotation
- Autoscales the Trace
- Queries Per Division, Reference Level, and Reference Position
- Turn ON and set averaging
- Turn ON and set smoothing

To run this program, you need:

- An established GPIB interface connection

See Other SCPI Example Programs

```
GPIB.Write "SYSTem:PRESet"

'Select the measurement
GPIB.Write "CALCulate:PARameter:SELect 'CH1_S11_1'"

'Set the Data Format to Log Mag
GPIB.Write ":CALCulate1:FORMat MLOG"

'Turn ON the Trace, Title, and Frequency Annotation
GPIB.Write "Display:WINDow1:TRACe1:STATe ON"
GPIB.Write "DISPlay:WINDow1:TITLe:STATe ON"
GPIB.Write "DISPlay:ANNotation:FREQuency ON"

'Autoscale the Trace
GPIB.Write "Display:WINDow1:TRACe1:Y:Scale:AUTO"

'Query back the Per Division, Reference Level, and Reference Position
GPIB.Write "DISPlay:WINDow1:TRACe1:Y:SCALe:PDIVision?"
Pdiv = GPIB.Read
GPIB.Write "DISPlay:WINDow1:TRACe1:Y:SCALe:RLEVel?"
Rlev = GPIB.Read
GPIB.Write "DISPlay:WINDow1:TRACe1:Y:SCALe:RPOSition?"
Ppos = GPIB.Read

'Turn ON, and average five sweeps
GPIB.Write "SENSe1:AVERage:STATe ON"
GPIB.Write "SENSe1:AVERage:Count 5"

'Turn ON, and set 20% smoothing aperture
GPIB.Write "CALCulate1:SMOothing:STATe ON"
GPIB.Write "CALCulate1:SMOothing:APERture 20"
```

See Other SCPI Example Programs

Channels, Windows, and Measurements using SCPI

SOURce and most SENSe commands act on the **channel** that is specified in the command. Channel 1 is default if not specified. There can be up to four channels present (numbers 1 to 4) in the analyzer at the same time.

Most **DISPlay** commands act on the **window and trace** specified in the command. Window1 and Trace1 are default if not specified. There can be up to four windows present (numbers 1 to 4) and up to four traces present in each window (numbers 1 to 4) at the same time.

CALCulate commands act on the **selected measurement** in the specified channel. Select the measurement for each channel using CALCulate<channel number>:PARameter:SELect <meas name>. You can select one measurement in each channel. Therefore, you can have up to four measurements selected at the same time.

The following Visual Basic program does the following:

- Presets the analyzer
- Create 2 windows
- Create 2 Measurements
- Feed the measurements to windows / traces
- Change frequency ranges for channels
- Select both measurements
- Turn marker 1 ON for each measurement

To run this program, you need:

- An established GPIB interface connection

```
GPIB.Write "SYSTem:PREset"

'Create Measurements
GPIB.Write "CALCulate1:PARameter:DEFine 'Meas1',S11"
GPIB.Write "CALCulate2:PARameter:DEFine 'Meas2',S21"

' Turn on windows - creates if new
GPIB.Write "DISPlay:WINDow1:STATE ON"
GPIB.Write "DISPlay:WINDow2:STATE ON"

'Associate ("FEED") the measurement name('Meas1') to WINDow(1), and give the new
TRACe a number(1).
GPIB.Write "DISPlay:WINDow1:TRACe1:FEED 'Meas1'"
GPIB.Write "DISPlay:WINDow2:TRACe2:FEED 'Meas2'"

'Change each channel's frequency range
GPIB.Write "SENSe1:FREQuency:SPAN 1e9"
GPIB.Write "SENSe2:FREQuency:SPAN 2e9"

'Select both measurements
GPIB.Write "CALCulate1:PARameter:SELect 'Meas1'"
GPIB.Write "CALCulate2:PARameter:SELect 'Meas2'"

'Turn marker 1 ON for each measurement
GPIB.Write "CALCulate1:MARKer:STATe ON"
```

```
GPIB.Write "CALCulate2:MARKer:STATE ON"
```

Perform a Guided Calibration using SCPI

This VBScript program performs a Guided Calibration using ECal **or** Mechanical standards. This example includes optional ECal orientation features.

The SCPI commands in this example are sent over a COM interface using the SCPIStringParser object. You do NOT need a GPIB connection to run this example.

This VBScript (*.vbs) program can be run as a macro in the PNA. To do this, copy the following code into a text editor file such as Notepad and save it on the PNA hard drive as Guided.vbs. Learn how to setup and run the macro.

```
' Performing a Guided 2-port cal (Ports 1 and 2)
TwoPortGuidedCal
Sub TwoPortGuidedCal
Dim app
Dim scpi
Dim connList
Dim selectedConn
Dim kitList
Dim selectedKit
Dim message

' Create / Get the PNA application.
Set app = CreateObject("AgilentPNA835x.Application")
Set scpi = app.ScpiStringParser

' Query the list of connectors that the PNA system recognizes
connList = scpi.Execute("sens:corr:coll:guid:conn:cat?")

' Format the list with linefeed characters in place of the commas
connList = FormatList(connList)

message = "Enter your DUT connector for Port 1.  Choose from this list:"
message = message & Chr(10) & Chr(10) & connList

' Select the connector for Port 1
selectedConn = InputBox(message)
If selectedConn = "" Then Exit Sub
scpi.Execute "sens:corr:coll:guid:conn:port1 '" & selectedConn & "'"

message = "Enter your DUT connector for Port 2.  Again, choose from this list:"
message = message & Chr(10) & Chr(10) & connList

' Select the connector for Port 2
selectedConn = InputBox(message)
If selectedConn = "" Then Exit Sub
scpi.Execute "sens:corr:coll:guid:conn:port2 '" & selectedConn & "'"

' Note: If your PNA has more than 2 ports, then uncomment
' one or both of these next two lines.
'scpi.Execute "sens:corr:coll:guid:conn:port3 ""Not used"" "
'scpi.Execute "sens:corr:coll:guid:conn:port4 ""Not used"" "

' Query the list of acceptable cal kits and
' ECal module characterizations for Port 1.
kitList = scpi.Execute("sens:corr:coll:guid:ckit:port1:cat?")
```

```
' Format the list with linefeed
' characters in place of the commas
kitList = FormatList(kitList)

message = "Enter your cal kit or ECal module characterization for Port 1.  "
message = message & "Choose from this list:"
message = message & Chr(10) & Chr(10) & kitList

' Select the Cal Kit or ECal module
' characterization to use for Port 1.
selectedKit = InputBox(message)
If selectedKit = "" Then Exit Sub
scpi.Execute "sens:corr:coll:guid:ckit:port1 '" & selectedKit & "'"

' Query the list of acceptable cal kits
' and ECal module characterizations for Port 2.
kitList = scpi.Execute("sens:corr:coll:guid:ckit:port2:cat?")
' Format the list with linefeed characters in place of the commas
kitList = FormatList(kitList)

message = "Enter your cal kit or ECal module characterization for Port 2.  "
message = message & "Choose from this list:"
message = message & Chr(10) & Chr(10) & kitList

' Select the Cal Kit or ECal module
' characterization to use for Port 2.
selectedKit = InputBox(message)
If selectedKit = "" Then Exit Sub
scpi.Execute "sens:corr:coll:guid:ckit:port2 '" & selectedKit & "'"

' This next block of commented code
' shows optional functions when using ECal.
' Send these "sens:corr:pref" commands prior to the
' "sens:corr:coll:guid:init" command.

' Read ECAL information from ECal module #1 on the USB bus
' about the Agilent factory characterization data

'module1Info = scpi.Execute("sens:corr:coll:ckit:inf? ECAL1,CHAR0")
'MsgBox "Description of ECal Module #1:" & Chr(10) & Chr(10) & module1Info

' The following command enables auto orientation of
' the ECal module (The PNA senses which port of the
' module is connected to which port of the PNA).

'scpi.Execute "sens:corr:pref:ecal:ori ON"

' However, if you are measuring at very low power levels where
' the PNA may fail to sense the module's orientation, then turn auto
' orientation OFF and specify how the module is connected.
' "A1,B2" indicates Port A of the module is connected
' to PNA Port 1 and Port B is connected to PNA Port 2).
'scpi.Execute "sens:corr:pref:ecal:ori OFF"
'scpi.Execute "sens:corr:pref:ecal:pmap ECAL1,'A1,B2'"
' End of optional ECal setup
```

```
' Select the thru method of "Default".  This instructs the PNA to
' determine which thru standard measurement technique to use
' based upon the selected connectors and
' calibration kit(s) and the PNA model number.
scpi.Execute "sens:corr:coll:guid:meth default"

' Initiate the calibration and query the number of steps
scpi.Execute "sens:corr:coll:guid:init"
numSteps = scpi.Execute("sens:corr:coll:guid:steps?")
MsgBox "Number of steps is " + CStr(numSteps)

' Measure the standards
For i = 1 To numSteps
step = "Step " + CStr(i) + " of " + CStr(numSteps)
strPrompt = scpi.Execute("sens:corr:coll:guid:desc? " + CStr(i))
MsgBox strPrompt, vbOKOnly, step
scpi.Execute "sens:corr:coll:guid:acq STAN" + CStr(i)
Next

' Conclude the calibration
scpi.Execute "sens:corr:coll:guid:save"
MsgBox "Cal is done!"

End Sub

Function FormatList(list)
Dim tokens
' Strip the leading and trailing quotation
' marks from the list string
list = Mid(list, 2, Len(list) - 3)
' Tokenize the comma-delimited list string
' into an array of the individual substrings
tokens = Split(list, ",")
' Rebuild the list string, placing linefeed
' characters where the commas were,
' using Trim to remove leading and trailing spaces.
list = ""
For i = 0 To UBound(tokens)
  tokens(i) = Trim(tokens(i))
  list = list & tokens(i) & Chr(9)
  If i < UBound(tokens) Then
    i = i + 1
    tokens(i) = Trim(tokens(i))
    list = list & tokens(i) & Chr(10)
  End If
Next
FormatList = list
End Function
```

Perform a Guided 1-Port Cal on Port 2

This VBScript program does the following:

1. Clear measurements from the PNA
2. Create a new S22 measurement
3. Set an instrument state
4. Select the connector types
5. Select a cal kit
6. Initiate a Guided calibration
7. Display a prompt to connect each standard
8. Save the calibration to a newly created cal set

Note: This example illustrates an important step when calibrating a reflection measurement in the reverse direction. You MUST create a reverse (S22) measurement and have it be the active (selected) measurement on the channel that is being calibrated. This is not necessary for any calibrating any other measurement parameter.

The SCPI commands in this example are sent over a COM interface using the SCPIStringParser object. You do NOT need a GPIB connection to run this example.

This VBScript (*.vbs) program can be run as a macro in the PNA. To do this, copy the following code into a text editor file such as Notepad and save it on the PNA hard drive as Guided.vbs. Learn how to setup and run the macro.

```
Dim App
Set App = CreateObject("AgilentPNA835x.Application")
App.Preset

Dim step
Dim Parser
Dim prompt
Dim txtDat
Dim Chan

Rem Clear old measurements
App.Reset

Rem Create a new Measurement
Set Parser = App.SCPIStringParser
Parser.Parse "DISPlay:WINDow1:STATE ON"
Parser.Parse "CALCulate:PARameter:DEFine 'MyMeas',S22"
Parser.Parse "DISPlay:WINDow1:TRACe1:FEED 'MyMeas'"

Rem Initialize state
Set Chan = App.ActiveChannel
Chan.StartFrequency = 200e6
Chan.StopFrequency = 1.5e9
Chan.IFBandwidth = 1000
step = 3
```

```
Rem Begin a guided calibration
Parser.Parse "SENS:CORR:COLL:GUID:CONN:PORT1 'Not used'"
Parser.Parse "SENS:CORR:COLL:GUID:CONN:PORT2 'Type N (50) male'"
Parser.Parse "SENS:CORR:COLL:GUID:CKIT:PORT1 ''"
Parser.Parse "SENS:CORR:COLL:GUID:CKIT:PORT2 '85054D'"
Parser.Parse "SENS:CORR:COLL:GUID:INIT"

Rem Query the number of steps
txtDat = Parser.Parse("SENS:CORR:COLL:GUID:STEP?")

Rem Display the number of steps
MsgBox("Number of steps is " + txtDat)

Rem Set the loop counter limit
step = txtDat

Rem  Measure the standards
For i = 1 To step
If i= 1 Then
prompt = Parser.Parse("sens:corr:coll:guid:desc? 1")
MsgBox(prompt)
Parser.Parse ("sens:corr:coll:guid:acq STAN1")
ElseIf i = 2 then
prompt = Parser.Parse("sens:corr:coll:guid:desc? 2")
MsgBox(prompt)
Parser.Parse ("sens:corr:coll:guid:acq STAN2")
ElseIf i = 3 then
prompt = Parser.Parse("sens:corr:coll:guid:desc? 3")
MsgBox(prompt)
Parser.Parse ("sens:corr:coll:guid:acq STAN3")
End If
Next

Rem All standards have been measured.  Save the result
Parser.Parse "SENS:CORR:COLL:GUID:SAVE"
MsgBox("The calibration has been completed")
```

Perform an Unguided ECal

This VBScript program performs an Unguided Full 2-Port ECal.

The SCPI commands in this example are sent over a COM interface using the SCPIStringParser object. You do NOT need a GPIB connection to run this example.

This VBScript (*.vbs) program can be run as a macro in the PNA. To do this, copy the following code into a text editor file such as Notepad and save it on the PNA hard drive as Unguided.vbs. Learn how to setup and run the macro.

```
Set pna = CreateObject("AgilentPNA835x.Application")
Set scpi = pna.ScpiStringParser
' Preset the analyzer
scpi.Execute "SYSTem:PRESet"

' Start frequency of 10 MHz
scpi.Execute "SENSe:FREQuency:STARt 10E6"

' Stop frequency of 9 GHz
scpi.Execute "SENSe:FREQuency:STOP 9E9"

' Select the preset S11 measurement
scpi.Execute "CALCulate:PARameter:SELect 'CH1_S11_1'"
' Read the information about the Agilent factory
' characterization data of ECal module #1 on the USB bus
module1Info = scpi.Execute("SENSe:CORRection:COLLect:CKIT:INFormation? ECAL1,CHAR0")

' Prompt for the ECal module
MsgBox "Description of ECal Module #1:" & Chr(10) & Chr(10) & module1Info & _Chr(10)
& Chr(10) & "Make port connections to the ECal module, then press enter"
' ECal full 1 port and 2 port
' Choose a Calibration Type (comment out one of these)
scpi.Execute "SENSe:CORRection:COLLect:METHod refl3"
scpi.Execute "SENSe:CORRection:COLLect:METHod SPARSOLT"
' Specify to have the PNA automatically determine which port of the
' ECal module is connected to which port of the PNA.
scpi.Execute "SENSe:CORRection:PREFerence:ECAL:ORIentation ON"
' Alternatively, if you are measuring at very low power levels where
' the PNA fails to sense the module's orientation, you may need to turn
' off the auto orientation and specify how the module is connected (as in
' these next two commented lines of code -- "A1,B2" would indicate Port A
' of the module is connected to Port 1 and Port B is connected to Port 2).
'scpi.Execute "SENSe:CORRection:PREFerence:ECAL:ORIentation OFF"
'scpi.Execute "SENSe:CORRection:PREFerence:ECAL:PMAP ECAL1,'A1,B2'"
' Acquire and store the calibration terms.  *OPC? causes a "+1" to be
' returned when finished.  CHAR0 indicates to use the Agilent factory
' characterized data within the ECal module (as opposed to a user characterization).
x = scpi.Execute("SENSe:CORRection:COLLect:ACQuire ECAL1,CHAR0;*OPC?")
MsgBox "Done with calibration."
```

Perform an Unguided 2-Port Mechanical Cal

This VBScript program performs an Unguided, Full 2-Port, calibration using ONE set of mechanical calibration standards.

The SCPI commands in this example are sent over a COM interface using the SCPIStringParser object. You do NOT need a GPIB connection to run this example.

This VBScript (*.vbs) program can be run as a macro in the PNA. To do this, copy the following code into a text editor file such as Notepad and save it on the PNA hard drive as Unguided.vbs. Learn how to setup and run the macro.

```
Set App = CreateObject("AgilentPNA835x.Application")
Set Scpi = App.SCPIStringParser

'Initialize state
Scpi.Execute ("SYSTem:PRESet")

'Select the Preset measurement
Scpi.Execute ("CALCulate:PARameter:SELect 'CH1_S11_1'")

'Set the calibration method
Scpi.Execute ("SENSe:CORRection:COLLect:METHod SPARSOLT")

'Select a cal kit
Scpi.Execute ("SENSe:CORRection:COLLect:CKIT:SELect 1")

'Set acquisition to FORWARD
Scpi.Execute ("SENSe:CORRection:SFORward ON")

'Measure the standards in forward direction
MsgBox "Connect OPEN to Port 1; then press OK"
Scpi.Execute ("SENSe:CORRection:COLLect:ACQuire stan1")

MsgBox "Connect SHORT to Port 1; then press OK"
Scpi.Execute ("SENSe:CORRection:COLLect:ACQuire stan2")

MsgBox "Connect LOAD to Port 1; then press OK"
Scpi.Execute ("SENSe:CORRection:COLLect:ACQuire stan3")

'Set acquisition to REVERSE
Scpi.Execute ("SENSe:CORRection:SFORward OFF")

'Measure the standards in reverse direction
MsgBox "Connect OPEN to Port 2; then press OK"
Scpi.Execute ("SENSe:CORRection:COLLect:ACQuire stan1")

MsgBox "Connect SHORT to Port 2; then press OK"
Scpi.Execute ("SENSe:CORRection:COLLect:ACQuire stan2")

MsgBox "Connect LOAD to Port 2; then press OK"
Scpi.Execute ("SENSe:CORRection:COLLect:ACQuire stan3")

'Measure the thru standard
MsgBox "Connect THRU between Ports 1 and 2; then press OK"
Scpi.Execute ("SENSe:CORRection:COLLect:ACQuire stan4")
```

```
'All standards have been measured.  Save the result
Scpi.Execute ("SENS:CORR:COLL:SAVE")
MsgBox "The calibration has been completed"
```

Perform an Unguided 1-Port Cal on Port 2

This VBScript program does the following:

1. Clear measurements from the PNA
2. Create a new S22 measurement
3. Set an instrument state
4. Select a cal kit
5. Initiate an Unguided calibration
6. Display a prompt to connect each standard
7. Save the calibration to a newly created cal set

Note: This example illustrates an important step when calibrating a reflection measurement in the reverse direction. You MUST create a reverse (S22) measurement and have it be the active (selected) measurement on the channel that is being calibrated. This is not necessary for any calibrating any other measurement parameter.

The SCPI commands in this example are sent over a COM interface using the SCPIStringParser object. You do NOT need a GPIB connection to run this example.

This VBScript (*.vbs) program can be run as a macro in the PNA. To do this, copy the following code into a text editor file such as Notepad and save it on the PNA hard drive as Unguided.vbs. Learn how to setup and run the macro.

```
Dim App
Set App = CreateObject("AgilentPNA835x.Application")
App.Preset

Dim Parser
Dim Chan

Rem Clear old measurements
App.Reset

Rem Create a new Measurement
Set Parser = App.SCPIStringParser
Parser.Parse "DISPlay:WINDow1:STATE ON"
Parser.Parse "CALCulate:PARameter:DEFine 'MyMeas',S22"
Parser.Parse "DISPlay:WINDow1:TRACe1:FEED 'MyMeas'"

Rem Initialize state
Set Chan = App.ActiveChannel
Chan.StartFrequency = 200e6
Chan.StopFrequency = 1.5e9
Chan.IFBandwidth = 1000

Rem Begin an unguided calibration
Rem Set the calibration method
Parser.Parse "SENSe:CORRection:COLLect:METHod REFL3"

Rem Turn off continuous sweep
Parser.Parse "INITiate:CONTinuous OFF"
```

```
Rem Select a cal kit
Parser.Parse "SENSe:CORRection:COLLect:CKIT:SELect 1"

Rem  Measure the standards
MsgBox("Connect OPEN to port 2.  Then press OK")
Parser.Parse ("sens:corr:coll:acq STAN1")

MsgBox("Connect SHORT to port 2.  Then press OK")
Parser.Parse ("sens:corr:coll:acq STAN2")

MsgBox("Connect LOAD to port 2.  Then press OK")
Parser.Parse ("sens:corr:coll:acq STAN3")

Rem All standards have been measured.  Save the result
Parser.Parse "SENS:CORR:COLL:SAVE"

Rem Turn ON continuous sweep
Parser.Parse "INITiate:CONTinuous ON"
MsgBox("The calibration has been completed")
```

Perform a Source Power Cal using SCPI

Programming the PNA using COM or using SICL/VISA over LAN (as in this example) leaves the PNA free to control GPIB devices as needed. This Visual Basic program demonstrates:

- Performing a source power calibration of Port 2 for Channel 1.
- Reading the calibration data.

Learn more about Power Calibrations.

See an example that Uploads a Source Power Cal.

Other SCPI Example Programs

To run this program, you need:

- One of the following power meters connected to the PNA through GPIB: E4416A, E4417A, E4418A/B, E4419A/B, 437B, 438A, EPM-441A, EPM-442A

Note: If your power meter is other than these, you can create your own Power Meter Driver using our template.

- Your PC and PNA both connected to a LAN (for communicating with each other).
- The SICL and VISA components of Agilent's I/O Libraries software installed on your PC (both are included when you install the software, unless you already have another vendor's VISA installed. Then specify Full SICL and VISA installation to overwrite the other vendor's VISA.
- The module visa32.bas added to your VB project.
- A form with two buttons: cmdRun and cmdQuit.
- A VISA interface configured on your remote PC to control the PNA. This could be GPIB interface or a VISA LAN Client.

Note: The SOURce:POWer:CORRection:COLLect:ACQuire command, used to step the PNA and read a power meter, cannot be sent over the GPIB unless the power meter is connected to a different GPIB interface. See the alternative methods described in the command details.

```
'Session to VISA Default Resource Manager
Private defRM As Long
'Session to PNA
Private viPNA As Long
'VISA function status return code
Private status As Long

Private Sub Form_Load()
defRM = 0
End Sub

Private Sub cmdRun_Click()
```

```
' String to receive data from the PNA.
' Dimensioned large enough to receive scalar comma-delimited values
' for 21 frequency points (20 ASCII characters per point)
Dim strReply As String * 420

Dim strStimulus, strCalValue
Dim strResult As String

' Open the VISA default resource manager
status = viOpenDefaultRM(defRM)
If (status < VI_SUCCESS) Then HandleVISAError

' ' Open a session (viPNA) to the PNA at "address 16" on the VISA
' interface configured as "GPIB1" on this PC.  This could be a
' VISA LAN Client pointing to the SICL LAN Server on the PNA, or
' an actual GPIB interface on this PC connected to the PNA GPIB
' (in which case the power meter would need to be connected to a
' different GPIB interface on the PNA, such as the Agilent 82357A
' USB-to-GPIB).
status = viOpen(defRM, "GPIB0::16::INSTR", 0, 0, viPNA)
If (status < VI_SUCCESS) Then HandleVISAError

' Set the number of sweep points to 21 on Channel 1.
status = myGPIBWrite(viPNA, "SENS1:SWE:POIN 21")
If (status < VI_SUCCESS) Then HandleVISAError

' Specify the GPIB address of the power meter
' that will be used in performing the calibration.
status = myGPIBWrite(viPNA, "SYST:COMM:GPIB:PMET:ADDR 13")
If (status < VI_SUCCESS) Then HandleVISAError

' Turn use of the loss table OFF (this assumes there is
' virtually no loss in the RF path to the power sensor
' due to a splitter, coupler or adapter).
status = myGPIBWrite(viPNA, "SOUR:POW:CORR:COLL:TABL:LOSS OFF")
If (status < VI_SUCCESS) Then HandleVISAError

' Turn frequency checking OFF (so one power sensor is used for the entire cal
' acquisition sweep regardless of frequency span).
status = myGPIBWrite(viPNA, "SOUR:POW:CORR:COLL:FCH OFF")
If (status < VI_SUCCESS) Then HandleVISAError

' Specify a nominal power accuracy tolerance (NTOLerance) in dB for the calibration,
' and the maximum number (COUNt) of iterations to adjust power at each point,
' attempting to achieve within tolerance of the desired power.  If at any stimulus
' point the power fails to reach within the set tolerance of the desired power
' after the maximum number of iterations, the power at that point will be set to the
' value determined by the last iteration (the Source Power Cal dialog box will
' indicate the FAIL, but we can still apply the cal if desired when it's complete).
' Each iteration is based upon a SETTLED power reading (see comments preceding the
' next two commands below).
status = myGPIBWrite(viPNA, "SOUR1:POW2:CORR:COLL:ITER:NTOL 0.1")
If (status < VI_SUCCESS) Then HandleVISAError
status = myGPIBWrite(viPNA, "SOUR1:POW2:CORR:COLL:ITER:COUN 3")
If (status < VI_SUCCESS) Then HandleVISAError

' The worst-case window of power uncertainty (for a calibration which meets
```

```
' tolerance) is the sum of the iteration tolerance and the power meter settling
' tolerance (which is described below).

' At each stimulus point, the PNA takes power meter readings and determine when
' they have settled by comparing the magnitude difference between consecutive
' readings versus a nominal dB tolerance limit (NTOLerance) on that magnitude
' difference.  When consecutive readings are within tolerance of each other, or
' if they are not within tolerance but we've taken a maximum number of readings
' (COUNt), the PNA does a weighted average of the readings taken at that stimulus
' point and that is considered our settled power reading.
status = myGPIBWrite(viPNA, "SOUR1:POW2:CORR:COLL:AVER:NTOL 0.1")
If (status < VI_SUCCESS) Then HandleVISAError
status = myGPIBWrite(viPNA, "SOUR1:POW2:CORR:COLL:AVER:COUN 5")
If (status < VI_SUCCESS) Then HandleVISAError

' Specify if the cal power level is offset (positive value for a gain, negative
' value for a loss) from the PNA port power setting on the channel when no source
' power cal is active.  This is to account for components between the PNA test
' port and cal reference plane.  In this example, we will calibrate at the PNA
' test port, so there is no offset (it is zero).

status = myGPIBWrite(viPNA, "SOUR1:POW2:CORR:OFFS 0 DB")
If (status < VI_SUCCESS) Then HandleVISAError

' Specify the method (type of devices) that will be used to perform the cal.
' Choose either power meter (PMETer), or power meter and receiver (PMReceiver).
' PMReceiver uses the power meter for the first iteration of each point and
' the PNA's reference receiver for subsequent iterations, so is much faster
' than using power meter only.  But the power meter accounts for compression
' when calibrating at the output of an active device, whereas the reference
' receiver cannot unless it is coupled to the cal reference plane (on a PNA
' which allows direct access to the receivers).
status = myGPIBWrite(viPNA, "SOUR1:POW2:CORR:COLL:METH PMET")
'status = myGPIBWrite(viPNA, "SOUR1:POW2:CORR:COLL:METH PMR")
If (status < VI_SUCCESS) Then HandleVISAError

' Show the source power cal dialog during the source power cal acquisition.
' (this is the default, so this command is only necessary if this setting
' may have been changed beforehand, perhaps by another program).
status = myGPIBWrite(viPNA, "SOUR1:POW2:CORR:COLL:DISP ON")
If (status < VI_SUCCESS) Then HandleVISAError

' Perform the source power cal acquisition sweep using the sensor attached to
' Channel A of the power meter (asking for an OPC reply when it’s done).  This
' assumes that the power sensor is already connected to Port 2 of the PNA.
' We'll put up an hourglass cursor while waiting for the acquire to complete.
Screen.MousePointer = vbHourglass
status = myGPIBWrite(viPNA, "SOUR1:POW2:CORR:COLL:ACQ ASEN;*OPC?")
If (status < VI_SUCCESS) Then HandleVISAError

' In the process of beginning a source power cal acquisition, the PNA searches
' for the power meter on VISA interfaces configured in the Agilent I/O Libraries
' on the PNA.  One of those interfaces is the SICL/VISA LAN server, so if this
' program is using that interface, we need to ensure our program is not pending
' an operation on that interface when the PNA wants to search it.  So this
' Wait subroutine suspends execution of our program (for 6000 milliseconds =
' 6 seconds), giving the PNA time to search that interface and discover that the
```

```
' power meter is not there (the 6 seconds is just to be safe, the search actually
' takes only a few seconds).
' Note: If instead of using the VISA LAN server interface, you are having this
' program communicate with the PNA via it's GPIB interface (which requires the
' power meter be connected to a different GPIB interface on the PNA, such as the
' Agilent 82357A USB-to-GPIB), then this Wait is not needed.
Wait 6000

' The PNA sends an OPC reply ("+1") when the cal acquisition is complete, so
' our Read operation will wait on the reply until it is received.  We need to
' set the VISA timeout value long enough to give our Read sufficient time to
' complete before a timeout error occurs.  For this example, let's try setting
' the limit to 60000 milliseconds (60 seconds).
status = viSetAttribute(viPNA, VI_ATTR_TMO_VALUE, 60000)
If (status < VI_SUCCESS) Then HandleVISAError
status = myGPIBRead(viPNA, strReply)
If (status < VI_SUCCESS) Then HandleVISAError

' Change mouse cursor from hourglass back to normal
Screen.MousePointer = vbDefault

' Conclude the calibration.  This applies the cal data to PNA channel memory,
' and turns the correction ON for Port 2 on Channel 1,
' but does NOT save the calibration.
status = myGPIBWrite(viPNA, "SOUR1:POW2:CORR:COLL:SAVE")
If (status < VI_SUCCESS) Then HandleVISAError

' At this point, if you choose to save the instrument state as a ".CST" file,
' the calibration will be saved with the instrument state in that file.

' Prepare for doing data transfer in ASCII format.
status = myGPIBWrite(viPNA, "FORM:DATA ASCII")
If (status < VI_SUCCESS) Then HandleVISAError

' Read the stimulus values from Channel 1.
status = myGPIBWrite(viPNA, "SENS1:X?")
If (status < VI_SUCCESS) Then HandleVISAError
status = myGPIBRead(viPNA, strReply)
If (status < VI_SUCCESS) Then HandleVISAError

' Tokenize the reply string into an array containing the values
strStimulus = Split(strReply, ",")

' Read the source power correction data.
status = myGPIBWrite(viPNA, "SOUR1:POW2:CORR:DATA?")
If (status < VI_SUCCESS) Then HandleVISAError
status = myGPIBRead(viPNA, strReply)
If (status < VI_SUCCESS) Then HandleVISAError

' Tokenize the reply string into an array containing the values
strCalValue = Split(strReply, ",")

' Print the data using a message box (here, Chr returns the ASCII characters
' for Tab (9) and Linefeed (10)).
strResult = "Stimulus" & Chr(9) & Chr(9) & "Cal Value" & Chr(10)
For i = 0 To UBound(strStimulus)
strResult = strResult & Val(strStimulus(i)) & Chr(9) & Val(strCalValue(i)) & Chr(10)
```

```
Next
MsgBox strResult
End Sub

Private Function myGPIBWrite(ByVal viHandle As Long, ByVal strOut As String) As Long

' The "+ Chr$(10)" appends an ASCII linefeed character to the
' output, for terminating the write transaction.
myGPIBWrite = viVPrintf(viHandle, strOut + Chr$(10), 0)
End Function

Private Function myGPIBRead(ByVal viHandle As Long, strIn As String) As Long
myGPIBRead = viVScanf(viHandle, "%t", strIn)
End Function

Sub HandleVISAError()
Dim strVisaErr As String * 200
Call viStatusDesc(defRM, status, strVisaErr)
MsgBox "*** Error : " + strVisaErr, vbExclamation

' Close the resource manager session (which also closes
' the session to the PNA).

If defRM <> 0 Then Call viClose(defRM)
End
End Sub


Public Sub Wait(ByVal mS_delay As Long)

Dim t0 As Single
t0 = Timer
Do While Timer - t0 < mS_delay / 1000
Dim dummy As Integer
dummy = DoEvents() ' if we cross midnight, back up one day
If Timer < t0 Then t0 = t0 - 86400
Loop
End Sub
```

Uploading a Source Power Cal using SCPI

Programming the PNA using COM or using SICL/VISA over LAN (as in this example) leaves the PNA free to control GPIB devices as needed. This Visual Basic program demonstrates:

- Uploading a source power calibration of Port 2 for Channel 1.
- Reading the calibration data.

Learn more about Power Calibrations

Other SCPI Example Programs

To run this program, you need:

- Your PC and PNA both connected to a LAN (if using VISA LAN server / client).
- The SICL and VISA components of Agilent's I/O Libraries software installed on your PC (both are included when you install the software, unless you already have another vendor's VISA installed. Then specify Full SICL and VISA installation to overwrite the other vendor's VISA.
- The module visa32.bas added to your VB project.
- A form with two buttons: cmdRun and cmdQuit.
- A VISA interface configured on your remote PC to control the PNA. This could be GPIB interface or a VISA LAN Client.

```
'Session to VISA Default Resource Manager
Private defRM As Long
'Session to PNA
Private viPNA As Long
'VISA function status return code
Private status As Long
Private Sub Form_Load()
defRM = 0
End Sub
Private Sub cmdRun_Click()

' String to receive data from the PNA.
' Dimensioned large enough to receive scalar comma-delimited values
' for 21 frequency points (20 ASCII characters per point)
Dim strReply As String * 420
Dim strPower As String, strCalPower As String
Dim strStimulus, strCalValue
Dim strResult As String

' Open the VISA default resource manager
status = viOpenDefaultRM(defRM)
If (status < VI_SUCCESS) Then HandleVISAError

' Open a session (viPNA) to the PNA at "address 16" on the VISA
' interface configured as "GPIB0" on this PC.
```

```
status = viOpen(defRM, "GPIB0::16::INSTR", 0, 0, viPNA)
If (status < VI_SUCCESS) Then HandleVISAError

' Set the number of sweep points to 2 on Channel 1.
status = myGPIBWrite(viPNA, "SENS1:SWE:POIN 2")
If (status < VI_SUCCESS) Then HandleVISAError

' Ensure there's currently no source power cal on for this channel and port.
status = myGPIBWrite(viPNA, "SOUR1:POW2:CORR OFF")
If (status < VI_SUCCESS) Then HandleVISAError

' Specify if the cal power level is offset (positive value for a gain, negative
' value for a loss) from the PNA port power setting on the channel when no source
' power cal is active.  This is to account for components between the PNA test
' port and cal reference plane.  In this example, let's set up our calibration
' at the output of an amplifier with 15 dB gain.
status = myGPIBWrite(viPNA, "SOUR1:POW2:CORR:OFFS 15 DB")
If (status < VI_SUCCESS) Then HandleVISAError

' Prepare for doing data transfer in ASCII format.
status = myGPIBWrite(viPNA, "FORM:DATA ASCII")
If (status < VI_SUCCESS) Then HandleVISAError

' Send our source power correction data to the PNA.  For purpose of simplicity
' in this example, we'll set up for no correction (0) at our start stimulus and
' 0.5 dB at our stop stimulus (recall that our sweep currently has just 2 points).
status = myGPIBWrite(viPNA, "SOUR1:POW2:CORR:DATA 0,0.5")
If (status < VI_SUCCESS) Then HandleVISAError

' Set the number of sweep points to 21 on Channel 1.
status = myGPIBWrite(viPNA, "SENS1:SWE:POIN 21")
If (status < VI_SUCCESS) Then HandleVISAError

' Read the fixed power level for this port on Channel 1.
status = myGPIBWrite(viPNA, "SOUR1:POW2:LEV?")
If (status < VI_SUCCESS) Then HandleVISAError
status = myGPIBRead(viPNA, strReply)
If (status < VI_SUCCESS) Then HandleVISAError
strPower = strReply

' Turn the source power cal on.
status = myGPIBWrite(viPNA, "SOUR1:POW2:CORR ON")
If (status < VI_SUCCESS) Then HandleVISAError

' Again read the fixed power level for this port on Channel 1
' (with our calibration turned on, this should now include the 15 dB offset
' we indicated our power amplifier provides).
status = myGPIBWrite(viPNA, "SOUR1:POW2:LEV?")
If (status < VI_SUCCESS) Then HandleVISAError
status = myGPIBRead(viPNA, strReply)
If (status < VI_SUCCESS) Then HandleVISAError
strCalPower = strReply

' Read the stimulus values from Channel 1.
status = myGPIBWrite(viPNA, "SENS1:X?")
If (status < VI_SUCCESS) Then HandleVISAError
status = myGPIBRead(viPNA, strReply)
```

```
If (status < VI_SUCCESS) Then HandleVISAError

' Tokenize the reply string into an array containing the values
strStimulus = Split(strReply, ",")

' Read back the source power correction data, now interpolated for 21 points
status = myGPIBWrite(viPNA, "SOUR1:POW2:CORR:DATA?")
If (status < VI_SUCCESS) Then HandleVISAError
status = myGPIBRead(viPNA, strReply)
If (status < VI_SUCCESS) Then HandleVISAError

' Tokenize the reply string into an array containing the values
strCalValue = Split(strReply, ",")

' Print the data using a message box (here, Chr returns the ASCII characters
' for Tab (9) and Linefeed (10)).
strResult = "PNA port power = " & Val(strPower) & Chr(10)
strResult = strResult & "Power at reference plane = " & Val(strCalPower) & Chr(10)
 Chr(10)
strResult = strResult & "Stimulus" & Chr(9) & Chr(9) & "Cal Value" & Chr(10)
For i = 0 To UBound(strStimulus)
 strResult = strResult & Val(strStimulus(i)) & Chr(9) & Val(strCalValue(i)) &
Chr(10)
Next
MsgBox strResult
End Sub
Private Sub cmdQuit_Click()

' Close the resource manager session (which also closes
' the session to the PNA).
If defRM <> 0 Then Call viClose(defRM)

' End the program
End
End Sub
Private Function myGPIBWrite(ByVal viHandle As Long, ByVal strOut As String) As Long

' The "+ Chr$(10)" appends an ASCII linefeed character to the
' output, for terminating the write transaction.
myGPIBWrite = viVPrintf(viHandle, strOut + Chr$(10), 0)
End Function
Private Function myGPIBRead(ByVal viHandle As Long, strIn As String) As Long
myGPIBRead = viVScanf(viHandle, "%t", strIn)

' Remove trailing linefeed character
If Right(strIn, 1) = Chr(10) Then strIn = Left(strIn, Len(strIn) - 1)
End Function
Sub HandleVISAError()
Dim strVisaErr As String * 200
Call viStatusDesc(defRM, status, strVisaErr)
MsgBox "*** Error : " + strVisaErr, vbExclamation

' Close the resource manager session (which also closes
' the session to the PNA).
If defRM <> 0 Then Call viClose(defRM)
End
End Sub
```

Perform a Sliding Load Calibration using GPIB

This Visual Basic program does a **only** the sliding load portion of a Calibration.
To run this program, you need:

- An established GPIB interface connection
- A measurement and calibration routine to call this sub-program
- STAN3 set up as a sliding load standard

See Other SCPI Example Programs

```
Sub slide()
'Measure the sliding load for at least 5 and no more than 7 slides
'Note that "SLSET" and "SLDONE" must be executed before the actual acquisition of a
slide
MsgBox "Connect Sliding Load; set to Position 1; then press OK"
GPIB.Write "SENS:CORR:COLL SLSET"
GPIB.Write "SENS:CORR:COLL STAN3;"

MsgBox "Set Sliding Load to position 2; then press OK"
GPIB.Write "SENS:CORR:COLL SLSET"
GPIB.Write "SENS:CORR:COLL STAN3;"

MsgBox "Set Sliding Load to position 3; then press OK"
GPIB.Write "SENS:CORR:COLL SLDONE"
GPIB.Write "SENS:CORR:COLL STAN3;"
End Sub
```

ECALConfidence Check using SCPI

This Visual Basic program performs a complete ECAL confidence check.

To run this program, you need:

- An established GPIB interface connection
- Agilent's VISA or National Instrument's VISA installed on your PC
- The module visa32.bas added to your VB project.
- A form with two buttons: cmdRun and cmdQuit
- A calibrated S11 1-port or N-port measurement active on Channel 1
- Window 1 is visible

See Other SCPI Example Programs

```
'Session to VISA Default Resource Manager
Private defRM As Long
'Session to PNA
Private viPNA As Long
'VISA function status return code
Private status As Long

Private Sub Form_Load()
   defRM = 0
End Sub

Private Sub cmdRun_Click()
'String to receive data from the PNA
Dim strReply As String * 200

' Open the VISA default resource manager
status = viOpenDefaultRM(defRM)
If (status < VI_SUCCESS) Then HandleVISAError

' Open a VISA session (viPNA) to the PNA at GPIB address 16.
status = viOpen(defRM, "GPIB0::16::INSTR", 0, 0, viPNA)
If (status < VI_SUCCESS) Then HandleVISAError

' Need to set the VISA timeout value to give all our GPIB Reads
' sufficient time to complete before a timeout error occurs.
' For this example, let's try setting the limit to
' 10000 milliseconds (10 seconds).
status = viSetAttribute(viPNA, VI_ATTR_TMO_VALUE, 10000)
If (status < VI_SUCCESS) Then HandleVISAError

' Get the catalog of all the measurements currently on Channel 1.
status = myGPIBWrite(viPNA, "CALC1:PAR:CAT?")
If (status < VI_SUCCESS) Then HandleVISAError
status = myGPIBRead(viPNA, strReply)
If (status < VI_SUCCESS) Then HandleVISAError

' If an S11 measurement named "MY_S11" doesn't already exist,
' then create it.
If InStr(strReply, "MY_S11") = 0 Then
   status = myGPIBWrite(viPNA, "CALC1:PAR:DEF MY_S11,S11")
   If (status < VI_SUCCESS) Then HandleVISAError
End If
```

```
strReply = ""

' Get the catalog of all the trace numbers currently active
' in Window 1.
status = myGPIBWrite(viPNA, "DISP:WIND1:CAT?")
If (status < VI_SUCCESS) Then HandleVISAError
status = myGPIBRead(viPNA, strReply)
If (status < VI_SUCCESS) Then HandleVISAError

' If a trace number 4 already exists in Window 1, then this
' will remove it.
If InStr(strReply, "4") > 0 Then
   status = myGPIBWrite(viPNA, "DISP:WIND1:TRAC4:DEL")
   If (status < VI_SUCCESS) Then HandleVISAError
End If

' Set trace number 4 to MY_S11.
status = myGPIBWrite(viPNA, "DISP:WIND1:TRAC4:FEED MY_S11")
If (status < VI_SUCCESS) Then HandleVISAError

' Set up trace view so we are viewing only the data trace.
status = myGPIBWrite(viPNA, "DISP:WIND1:TRAC4 ON")
If (status < VI_SUCCESS) Then HandleVISAError
status = myGPIBWrite(viPNA, "DISP:WIND1:TRAC4:MEM OFF")
If (status < VI_SUCCESS) Then HandleVISAError

' Select MY_S11 as the measurement to be used for the
' Confidence Check.
status = myGPIBWrite(viPNA, "SENS1:CORR:CCH:PAR MY_S11")
If (status < VI_SUCCESS) Then HandleVISAError

' Acquire the S11 confidence check data from ECal Module A
' into the memory buffer (asking for an OPC reply when it's done).
status = myGPIBWrite(viPNA, "SENS1:CORR:CCH:ACQ ECAL1;*OPC?")
If (status < VI_SUCCESS) Then HandleVISAError

' The PNA sends an OPC reply ("+1") when the confidence data
' acquisition into memory is complete, so this Read is waiting on
' the reply until it is received.
status = myGPIBRead(viPNA, strReply)
If (status < VI_SUCCESS) Then HandleVISAError

' Turn on trace math so the trace shows data divided by memory.
' You can be confident the S11 calibration is reasonably good if
' the displayed trace varies no more than a few tenths of a dB
' from 0 dB across the entire span.
status = myGPIBWrite(viPNA, "CALC1:MATH:FUNC DIV")
If (status < VI_SUCCESS) Then HandleVISAError
End Sub

Private Sub cmdQuit_Click()
' Turn off trace math
status = myGPIBWrite(viPNA, "CALC1:MATH:FUNC NORM")
If (status < VI_SUCCESS) Then HandleVISAError

' Conclude the confidence check to set the ECal module
' back to it's idle state.
```

```
status = myGPIBWrite(viPNA, "SENS1:CORR:CCH:DONE")
If (status < VI_SUCCESS) Then HandleVISAError

' Close the resource manager session (which also closes
' the session to the PNA).
If defRM <> 0 Then Call viClose(defRM)

' End the program
End
End Sub

Private Function myGPIBWrite(ByVal viHandle As Long, ByVal strOut As String) As Long
' The "+ Chr$(10)" appends an ASCII linefeed character to the output, for
' terminating the write transaction.
myGPIBWrite = viVPrintf(viHandle, strOut + Chr$(10), 0)
End Function

Private Function myGPIBRead(ByVal viHandle As Long, strIn As String) As Long
myGPIBRead = viVScanf(viHandle, "%t", strIn)
End Function

Sub HandleVISAError()
Dim strVisaErr As String * 200
Call viStatusDesc(defRM, status, strVisaErr)
MsgBox "*** Error : " + strVisaErr, vbExclamation
End
End Sub
```

Create and Calibrate a VMC Measurement

This VB Script example creates and calibrates a Vector mixer measurement. To run this example **without modification** you need the following:

- A Mixer setup file saved on the PNA: C:\Program Files\Agilent\Network Analyzer\Documents\Mixer\MyMixer.mxr.
- If the mixer file uses an external LO source, it must also be attached and configured.
- An ECAL module that covers the frequency range of the measurement.

The SCPI commands in this example are sent over a COM interface using the SCPIStringParser object. You do NOT need a GPIB connection to run this example. However, some modification may be necessary to make the program run on a traditional GPIB Interface.

This VBScript (*.vbs) program can be run as a macro in the PNA. To do this, copy the following code into a text editor file such as Notepad and save it on the PNA hard drive as VMC.vbs. Learn how to setup and run the macro.

```
Dim app
Dim scpi
' Create / Get the PNA application.
Set app = CreateObject("AgilentPNA835x.Application")
Set scpi = app.ScpiStringParser
'---Create a Vector Mixer Measurement
'First, delete all measurements on the channel
scpi.Execute "CALC:PAR:DEL:ALL"
'Create a forward scalar mixer measurement and configure
'it in channel 1.
'The first parameter is a unique identifying string
'(specified by the user) to allow subsequent
'commands to be directed at this specific measurement.
SCPI.Execute "CALC:CUST:DEF ""My VC21"", ""FCA.VMCMeas"""
'Setup the new measurement as the 2nd trace in the active window
SCPI.Execute "DISP:WIND:TRAC2:FEED ""My VC21"""
'Make the new trace the active measurement
SCPI.Execute "CALC:PAR:SEL ""My VC21"""
'The parameters of the mixer measurement can now be configured.
'This can be done by either using the SENS:MIX commands
'for each of the parameters or by loading a mixer setup file.
'This example loads a mixer setup file. The path name
'for the mixer file may be loaded from other mapped drives.
SCPI.Execute "SENS:MIXer:Load ""C:\Program Files\Agilent\Network
Analyzer\Documents\Mixer\MyMixer.mxr"""

'--------------Perform A Vector Mixer Calibration-------------
'Initialize an VMC guided calibration for session number 6
SCPI.Execute "SENS:CORR:COLL:SESS6:INIT ""VMC"""
'This sets the VMC operation to full system cal as opposed to
'performing a mixer characterization only.
SCPI.Execute "SENS:CORR:COLL:SESS6:VMC:OPER ""CAL"""
'This example uses ECal for the 2-port cal portion of the procedure
'To use a mechanical kit you will have to use the following command:
'SCPI.Execute  "SENS:CORR:COLL:SESS6:SMC:TWOPort:OPTion""MECH"""
SCPI.Execute "SENS:CORR:COLL:SESS6:VMC:TWOPort:OPTion ""ECAL"""
'If you select the mechanical method then you also have to
```

```
'specify the connector types and the cal kits for each of the ports.
'The comments below show an example of how that is done:
'SCPI.Execute "SENS:CORR:COLL:SESS6:CONN:PORT1:SEL ""APC 3.5 male"""
'SCPI.Execute "SENS:CORR:COLL:SESS6:CONN:PORT2:SEL ""APC 3.5 female"""
'SCPI.Execute "SENS:CORR:COLL:SESS6:CKIT:PORT1:SEL ""85052D"""
'SCPI.Execute "SENS:CORR:COLL:SESS6:CKIT:PORT2:SEL ""85052D"""
'Choose the between ECal or Mechanical calibration for the
'Mixer Characterization portion of the VMC cal.
SCPI.Execute "SENS:CORR:COLL:SESS6:VMC:MIX:CHAR:CAL:OPT ""ECAL"""
'This command sets the port mapping for the ECal to be used
'during the Mixer Characterization portion of the VMC cal.
'It is a required command if in the previous command the option
'was set to 'ECAL'.  The only valid port maps are either 'A1'
'or 'B1'.
SCPI.Execute "SENS:CORR:COLL:SESS6:VMC:MIX:ECAL:PORT 1, ""A1"""
'Specify the ECal module and the ECal characterization for the
'two port calibration portion of this session.  FCA calibrations
'currently only support ECal module number 1.  In this example
'the factory characterization is used by specifying 0 for the
'characterization number.
SCPI.Execute "SENS:CORR:COLL:SESS6:VMC:TWOP:ECAL:CHAR 1,0"
'Specify the ECal module and the ECal characterization for the
'Mixer Characterization portion of this session.  FCA calibrations
'currently only support ECal module number 1.  In this example
'the factory characterization is used by specifying 0 for the
'characterization number.
SCPI.Execute "SENS:CORR:COLL:SESS6:VMC:MIX:ECAL:CHAR 1,0"
'Specify the thru measurement method.  This applies to both ECal
'and mechanical calibrations.  For ECal 'DEFAULT' will use the ECal
'thru.  Other choices may be used depending on the genders and types
'of the connectors on the test interface.
SCPI.Execute "SENS:CORR:COLL:SESS6:VMC:TWOP:METH ""DEFAULT"""
'Omit the isolation part of the 2-port cal
SCPI.Execute "SENS:CORR:COLL:SESS6:VMC:TWOP:OMIT 1"
'Turn on auto orientation for the ECal
SCPI.Execute "SENS:CORR:COLL:SESS6:VMC:TWOP:ECAL:ORI:STATE 1"

'Tell the wizard to generate and report the number of steps in this
'cal session
Dim steps
Dim desc
'Determine the number of steps required to complete the calibration.
'First send the write command, then the query.
SCPI.Execute "SENS:CORR:COLL:SESS6:STEP"
steps = SCPI.Execute ("SENS:CORR:COLL:SESS6:STEP?")
For i = 1 To steps
  'Display the prompt for each step
  desc = SCPI.Execute ("SENS:CORR:COLL:SESS6:DESC? " & CStr(i))
 MsgBox (desc)
 'Perform the measurement for each step
  SCPI.Execute "SENS:CORR:COLL:SESS6:ACQ " & CStr(i)
Next
Dim calset
'Finish the cal and save the calset
calset = SCPI.Execute ("SENS:CORR:COLL:SESS6:SAVE?")
Msgbox ("VMC Cal Complete!")
```

Create and Cal an SMC Measurement

This Visual Basic example creates and calibrates a scalar mixer measurement.

To run this example **without modification** you need the following:

- A Mixer setup file saved on the PNA: C:\Program Files\Agilent\Network Analyzer\Documents\Mixer\MyMixer.mxr.
- If the mixer file uses an external LO source, it must also be attached and configured.
- An ECAL module that covers the frequency range of the measurement.
- A power meter must be attached to the PNA. If this example is run in the PNA, the power meter does not need to be attached using a GPIB/USB interface.

The SCPI commands in this example are sent over a COM interface using the SCPIStringParser object. You do NOT need a GPIB connection to run this example. However, some modification is necessary to make the program run on a traditional GPIB Interface. For example, during the power meter portion of this calibration, scpi.Execute will not process a command until the power meter routine has completed. Traditional GPIB would require a serial polling technique to ensure the routine has completed before proceeding.

This VBScript (*.vbs) program can be run as a macro in the PNA. To do this, copy the following code into a text editor file such as Notepad and save it on the PNA hard drive as SMC.vbs. Learn how to setup and run the macro.

```
Dim app
Dim scpi
' Create / Get the PNA application.
Set app = CreateObject("AgilentPNA835x.Application")
Set scpi = app.ScpiStringParser
'---Create a Scalar Mixer Forward Measurement
'First, delete all measurements on the channel
scpi.Execute "CALC:PAR:DEL:ALL"
'Create a forward scalar mixer measurement and configure it in
'channel 1.  The first parameter is a unique
'identifying string (specified by the user) to allow subsequent
'commands to be directed at this specific measurement.
scpi.Execute "CALC:CUST:DEF ""My SC21"", ""SMC_Forward.SMC_ForwardMeas"""
'For reverse measurements, use the following:
'scpi.Execute "CALC:CUST:DEF ""My SC12"", ""SMC_Reverse.SMC_ReverseMeas"""
'Setup the new measurement as the 2nd trace in the active window
scpi.Execute "DISP:WIND:TRAC2:FEED ""My SC21"""
'Make the new trace the active measurement
scpi.Execute "CALC:PAR:SEL ""My SC21"""
'The parameters of the mixer measurement can now be configured.
'This can be done by either using the individual SENS:MIX commands
'for each of the parameters or by loading a mixer setup file.  This
'example loads a mixer setup file. The path name
'for the mixer file may be loaded from other mapped drives
scpi.Execute "SENS:MIXer:Load ""C:\Program Files\Agilent\Network
Analyzer\Documents\Mixer\MyMixer.mxr"""
'--------------Perform A Scalar Mixer Calibration---------------------
'Initialize an SMC guided calibration for session number 6
scpi.Execute "SENS:CORR:COLL:SESS6:INIT ""SMC"""
'Select to use an ECal for the 2-port cal portion of the procedure
'To use a mechanical kit you will have to use the following command:
'scpi.Execute "SENS:CORR:COLL:SESS6:SMC:TWOPort:OPTion ""MECH"""
```

```
scpi.Execute "SENS:CORR:COLL:SESS6:SMC:TWOPort:OPTion ""ECAL"""
'If you select the mechanical method then you also have to
'specify the connector types and the cal kits for each of the ports.
'The comments below show an example of how that is done:
'scpi.Execute "SENS:CORR:COLL:SESS6:CONN:PORT1:SEL ""APC 3.5 male"""
'scpi.Execute "SENS:CORR:COLL:SESS6:CONN:PORT2:SEL ""APC 3.5 female"""
'scpi.Execute "SENS:CORR:COLL:SESS6:CKIT:PORT1:SEL ""85052D"""
'scpi.Execute "SENS:CORR:COLL:SESS6:CKIT:PORT2:SEL ""85052D"""
'Specify the ECal module and the ECal characterization for this
'session.  FCA calibrations currently only support ECal module
'number 1.  In this example the factory characterization is used
'by specifying 0 for the characterization number.
scpi.Execute "SENS:CORR:COLL:SESS6:SMC:ECAL:CHAR 1,0"
'Specify the thru measurement method.  This applies to both ECal
'and mechanical calibrations.  For ECal 'DEFAULT' will use the ECal
'thru.  Other choices may be used depending on the genders and types
'of the connectors on the test interface.
scpi.Execute "SENS:CORR:COLL:SESS6:SMC:TWOP:METH ""DEFAULT"""
'Omit the isolation part of the 2-port cal
scpi.Execute "SENS:CORR:COLL:SESS6:SMC:TWOP:OMIT 1"
'Turn on auto orientation for the ECal
scpi.Execute "SENS:CORR:COLL:SESS6:SMC:TWOP:ECAL:ORI:STATE 1"
'Set the Source port for the power cal to 1.  If the
'measurement is an SMC Forward the valid port settings are '1' and
''BOTH' and if the measurement is an SMC Reverse, the valid port settings
'are '2' and 'BOTH'.
scpi.Execute "SENS:CORR:COLL:SESS6:SMC:PWRC:SRCP ""1"""
'Tell the wizard to generate and report the number of steps in this
'cal session
Dim steps
Dim desc
'Determine the number of steps required to complete the calibration.
'First send the write command, then the query.
scpi.Execute "SENS:CORR:COLL:SESS6:STEP"
steps = scpi.Execute ("SENS:CORR:COLL:SESS6:STEP?")
For i = 1 To steps
'Display the prompt for each step
desc = scpi.Execute ("SENS:CORR:COLL:SESS6:DESC? " & CStr(i))
MsgBox (desc)
'Perform the measurement for each step
scpi.Execute "SENS:CORR:COLL:SESS6:ACQ " & CStr(i)
Next
Dim calset
'Finish the cal and save the calset
calset = scpi.Execute ("SENS:CORR:COLL:SESS6:SAVE?")
Msgbox ("SMC Cal Complete!")
```

Create an SMC Fixed Output Measurement

This VB Script example creates a calibrated SMC fixed output measurement using a controlled LO. Then a single sweep is taken and data is retrieved. You can run a VBScript (*.vbs) program from the PNA using Macros.

The SCPI commands in this example are sent over a COM interface using the SCPIStringParser object.

Fixed output measurements are only supported on SMC (not VMC) measurements. Fixed output measurements require that an external LO source be sweeping and synchronized with the PNA source. FCA can perform this synchronization using the external source configuration settings.

The fastest, and recommended, method of controlling the LO source is Hardware List (BNC) triggering mode. However, in this mode, FCA channels will not respond to manual triggers. Therefore, the example uses the following mechanism to trigger a sweep:

```
Write "SENS:SWE:MODE HOLD"    'place channel 1 in HOLD mode
Write "INIT:CONT ON"          'place PNA in internal trigger mode
Write "SENS:SWE:GRO:COUNT 1"  'specify that the group count is 1 sweep
Write "SENS:SWE:MODE GROUPS"  'execute group count (1 sweep)
Write "*OPC?"       'wait until the sweep is complete
Read
```

Both VMC and SMC measurements require that mixer parameters be setup before making the measurement. The mixer parameters are not applied until calling SENS:MIX:CALC

You can also setup the mixer parameters using SENS:MIX:LOAD. With this interface, the specified mixer file is loaded and immediately applied.

To run this program, copy the following code into a text editor and save it as a *.vbs file.

```
option explicit

' Setup infrastructure to use the SCPI over COM
dim app
set app = createobject("Agilentpna835x.application")
dim p
set p = app.scpistringparser
dim returnStr

sub Write (command)
if len(returnStr) <> 0 then
err.Raise 55,"Write","Query Unterminated"
end if
returnStr = p.parse(command)
end sub

sub WriteIgnoreError(command)
returnStr = p.Execute(command)
p.Parse("SYST:ERR?") ' clear error queue
end sub

function Read
if len(returnStr) = 0 then
err.Raise 55,"Read","Bad read"
end if
Read = returnStr
returnStr = ""
end function

Write "SYST:PRES"
```

```
' When programming in remote mode, hold mode is recommended
Write "SENS:SWE:MODE HOLD"

' Select calibration. Use execute so that errors get ignored
WriteIgnoreError "SENS:CORR:CSET:GUID '{C9080B34-EF1F-4ED5-B07D-250B45962B99}'"

' Delete the standard measurement
Write "CALC:PAR:DEL:ALL"

' Create an SC21 measurement
Write "CALC:CUST:DEF 'MySMC', 'SMC_Forward.SMC_ForwardMeas'"
Write "DISP:WIND:TRACE:FEED 'MySMC'"
Write "CALC:PAR:SEL 'MySMC'"
Write "CALC:CUST:MOD 'SC21'"

' Set number of points to 11
Write "SENS:SWE:POIN 11 "

' Setup the mixer parameters for a swept LO, fixed output measurement
Write "SENS:MIX:INP:FREQ:START 200e6"
Write "SENS:MIX:INP:FREQ:STOP 700e6"
Write "SENS:MIX:LO:FREQ:MODE Swept"
Write "SENS:MIX:OUTPUT:FREQ:FIX 3.4e9"
Write "SENS:MIX:OUTP:FREQ:SID HIGH"
Write "SENS:MIX:INP:POW -17"
Write "SENS:MIX:LO:POW 10"
' Specify the LO name, for controlled LO.
' This name is setup in the External Source Config Dialog
Write "SENS:MIX:LO:NAME '8360'"
' The CALC method calculates the LO frequency from the other parameters,
' It also applies the mixer parameters to the channel.
Write "SENS:MIX:CALC LO_1"

' Create an S11 in the same channel
Write "CALC:CUST:DEF 'MyS11', 'SMC_Forward.SMC_ForwardMeas'"
Write "DISP:WIND:TRACE2:FEED 'MyS11'"
Write "CALC:PAR:SEL 'MyS11'"
Write "CALC:CUST:MOD 'S11'"

' Create an IPwr in the same channel
Write "CALC:CUST:DEF 'MyIPwr', 'SMC_Forward.SMC_ForwardMeas'"
Write "DISP:WIND:TRACE3:FEED 'MyIPwr'"
Write "CALC:PAR:SEL 'MyIPwr'"
Write "CALC:CUST:MOD 'IPwr'"

' Create an OPwr in the same channel
Write "CALC:CUST:DEF 'MyOPwr', 'SMC_Forward.SMC_ForwardMeas'"
Write "CALC:PAR:SEL 'MyOPwr'"
Write "DISP:WIND:TRACE4:FEED 'MyOPwr'"
Write "CALC:CUST:MOD 'OPwr'"

' Perform a single sweep, synchronously. When *OPC returns, the sweep is done
Write "SENS:SWE:GRO:COUN 1"
Write "SENS:SWE:MODE GROUPS"
Write "*OPC?"
Read
```

```
' Retrieve the SC21 data
Write "CALC:PAR:SEL 'MySMC'"
Write "CALC:DATA? SDATA"
dim data
data = Read()
wscript.echo(data)

'Retrieve the S11 data
Write "CALC:PAR:SEL 'MyS11'"
Write "CALC:DATA? SDATA"
data = Read()
wscript.echo(data)
```

Getting and Putting Data using SCPI

This Visual Basic Program does the following:

- Reads data from the analyzer
- Puts the data back into memory
- To see the data on the analyzer after running the program, from the front panel click: **Trace - Math/Memory - Memory Trace**

To run this program, you need:

- An established GPIB interface connection

See Other SCPI Example Programs

Note: To change the read and write location of data, removing the comment from the beginning of ONE of the lines, and replace the comment in the beginning of the SDATA and SMEM lines.

```
Private Sub ReadWrite_Click()
Dim i As Integer
Dim t As Integer
Dim q As Integer
Dim dat As String
Dim cmd As String
Dim datum() As Double

GPIB.Configure
GPIB.Write "SYSTem:PRESet;*wai"

'Select the measurement
GPIB.Write "CALCulate:PARameter:SELect 'CH1_S11_1'"

'Read the number of data points
GPIB.Write "SENSe1:SWEep:POIN?"
numpts = GPIB.Read

'Turn continuous sweep off
GPIB.Write "INITiate:CONTinuous OFF"

'Take a sweep
GPIB.Write "INITiate:IMMediate;*wai"

'Ask for the Data

'PICK ONE OF THESE LOCATIONS TO READ
'GPIB.Write "CALCulate:DATA? FDATA" 'Formatted Meas
'GPIB.Write "CALCulate:DATA? FMEM" 'Formatted Memory
GPIB.Write "CALCulate:DATA? SDATA" 'Corrected, Complex Meas
'GPIB.Write "CALCulate:DATA? SMEM" 'Corrected, Complex Memory
'GPIB.Write "CALCulate:DATA? SCORR1" 'Error-Term Directivity

'Number of values returned per data point
'q = 1 ' Pick this if reading FDATA or FMEM
q = 2 ' Otherwise pick this

'Parse the data
ReDim datum(q, numpts)
```

```
For i = 0 To numpts - 1
  For t = 0 To q - 1
  'Read the Data
  dat = GPIB.Read(20)
  'Parse it into an array
  datum(t, i) = Val(dat)
  Next t
Next i

'PUT THE DATA BACK IN
GPIB.Write "format ascii"

'PICK ONE OF THESE LOCATIONS TO PUT THE DATA
'cmd = "CALCulate:DATA FDATA," 'Formatted Meas
'cmd = "CALCulate:DATA FMEM," 'Formatted Memory
'cmd = "CALCulate:DATA SDATA," 'Corrected, Complex Meas
cmd = "CALCulate:DATA SMEM," 'Corrected, Complex Memory
'cmd = "CALCulate:DATA SCORR1," 'Error-Term Directivity

For i = 0 To numpts - 1
  For t = 0 To q - 1
  If i = numpts - 1 And t = q - 1 Then
  cmd = cmd & Format(datum(t, i))
  Else
  cmd = cmd & Format(datum(t, i)) & ","
  End If
  Next t
Next i

GPIB.Write cmd
End Sub
```

Establish a VISA Session

This Visual Basic program demonstrates how to send a SCPI command using VISA and the Agilent IO libraries. To run this program, you need:

- Your PC and PNA both connected to a LAN (for communicating with each other).
- The SICL and VISA components of Agilent's I/O Libraries software installed on your PC. Both are included when you install the software, unless you already have another vendor's VISA installed. Then specify Full SICL and VISA installation to overwrite the other vendor's VISA.
- The module visa32.bas added to your VB project. After you install VISA, the module will be located at C:\VXIPNP\WINNT (or equivalent)\INCLUDE\Visa32.bas
- A form with two buttons: cmdRun and cmdQuit.
- Your PC configured to be a VISA LAN Client, and the SICL Server capability enabled on the PNA. See Configure for VISA and SICL

See Other SCPI Example Programs

Note: This example is a piece of a larger VISA program that performs a source power calibration.

```
'Session to VISA Default Resource Manager
Private defRM As Long
'Session to PNA
Private viPNA As Long
'VISA function status return code
Private status As Long

Private Sub Form_Load()
defRM = 0
End Sub

Private Sub cmdRun_Click()
' String to receive data from the PNA.
' Dimensioned large enough to receive scalar comma-delimited values
' for 21 frequency points (20 ASCII characters per point)
Dim strReply As String * 420

' Open the VISA default resource manager
status = viOpenDefaultRM(defRM)
If (status < VI_SUCCESS) Then HandleVISAError

' Open a VISA session (viPNA) to the SICL LAN server
' at "address 16" on the PNA pointed to by the "GPIB0"
' VISA LAN Client on this PC.
' CHANGE GPIB0 TO WHATEVER YOU PNA IS SET TO
status = viOpen(defRM, "GPIB0::16::INSTR", 0, 0, viPNA)
If (status < VI_SUCCESS) Then HandleVISAError

' Need to set the VISA timeout value to give all our calls to
' myGPIBRead sufficient time to complete before a timeout
' error occurs.
' For this example, let's try setting the limit to
' 30000 milliseconds (30 seconds).
status = viSetAttribute(viPNA, VI_ATTR_TMO_VALUE, 30000)
```

```
If (status < VI_SUCCESS) Then HandleVISAError


' Preset the PNA
status = myGPIBWrite(viPNA, "SYST:PRES")
If (status < VI_SUCCESS) Then HandleVISAError


' Print the data using a message box
MsgBox strReply
End Sub

Private Sub cmdQuit_Click()
' Close the resource manager session (which also closes
' the session to the PNA).
If defRM <> 0 Then Call viClose(defRM)

' End the program
End
End Sub

Private Function myGPIBWrite(ByVal viHandle As Long, ByVal strOut As String) As Long
' The "+ Chr$(10)" appends an ASCII linefeed character to the
' output, for terminating the write transaction.
myGPIBWrite = viVPrintf(viHandle, strOut + Chr$(10), 0)
End Function

Private Function myGPIBRead(ByVal viHandle As Long, strIn As String) As Long
myGPIBRead = viVScanf(viHandle, "%t", strIn)
End Function

Sub HandleVISAError()
Dim strVisaErr As String * 200
Call viStatusDesc(defRM, status, strVisaErr)
MsgBox "*** Error : " + strVisaErr, vbExclamation
End
End Sub
```

See Other SCPI Example Programs

Status Reporting using SCPI

This Visual Basic program demonstrates two methods of reading the analyzer's status registers:

- Polled Bit Method - reads the Limit1 register continuously.
- SRQ Method - enables an interrupt of the program when bit 6 of the status byte is set to 1. The program then queries registers to determine if the limit line failed.

To run this program, you need:

- An established GPIB interface connection
- A form with two buttons: Poll and SRQ Method
- A means of causing the limit line to fail, assuming it passes initially.

```
Private Sub Poll_Click()
' POLL THE BIT METHOD
' Clear status registers
GPIB.Write "*CLS"

'Loop FOREVER
Do
  DoEvents
  GPIB.Write "STATus:QUEStionable:LIMit1:EVENt?"
  onn = GPIB.Read
Loop Until onn = 2

MsgBox "Limit 1 Failed "
End Sub


Private Sub SRQMethod_Click()
'SRQ METHOD
GPIB.Write "SYSTem:PRESet"
GPIB.Write "CALCulate:PARameter:SELect 'CH1_S11_1'"
 low down the trace
GPIB.Write "SENS:BWID 150"

'Setup limit line
GPIB.Write "CALC:LIM:DATA 2,3e9,6e9,-2,-2"
GPIB.Write "CALC:LIMit:DISP ON"
GPIB.Write "CALC:LIMit:STATe ON"

' Clear status registers.
GPIB.Write "*CLS;*wai"
' Clear the Service Request Enable register.
GPIB.Write "*SRE 0"
' Clear the Standard Event Status Enable register.
GPIB.Write "*ESE 0"

' Enable questionable register, bit(10) to report to the status byte.
GPIB.Write "STATus:QUEStionable:ENABle 1024"
```

```
' Enable the status byte register bit3 (weight 8) to notify controller
GPIB.Write "*SRE 8"

' Enable the onGPIBNotify event
GPIB.NotifyMask = cwGPIBRQS
GPIB.Notify
End Sub

----------------------------------------------------
Private Sub GPIB_OnGPIBNotify(ByVal mask As Integer)
' check to see what failed
' was it the analyzer?
GPIB.Write "*STB?"
onn = GPIB.Read
If onn <> 0 Then
' If yes, then was it the questionable register?
  GPIB.Write "STATus:QUEStionable:EVENt?"
  onn = GPIB.Read
  ' Determine if the limit1 register, bit 8 is set.
  If onn = 1024 Then
  'if yes, then was it trace 1?
  GPIB.Write "STAT:QUES:LIMIT1:EVEN?"
  onn = GPIB.Read
  If onn = 2 Then MsgBox ("Limit Line1 Failed")
  End If
End If
End Sub
```

Create New Cal Kit using SCPI

When creating new cal kits programmatically, the order in which cal kit commands are sent can be important. For example to create a kit with opens, shorts, loads, and thrus. Be sure to use the following sequence for each newly defined standard.

1. Programmatically select the standard number
2. Programmatically select the standard type.
3. Program the cal standard's values.
4. Repeat steps 1, 2, 3 for additional new standards being defined.

```
10   !
20   !
30   ! This example program demonstrates how to create
40   ! new PNA calibration kits.
50   !
60   ! 1) Select a kit not previously defined
70   ! 2) Define open, short, load, and thru cal standards
80   !    Note:  Each of the newly defined standards is assigned
90   !    a default connector name.  These default connector names
100  !    will be replaced in subsequent steps.
110  ! 3) Use the delete connector command to remove default
120  !    connector names.
130  ! 4) Add connectors.  Specify:
140  !      Start and Stop Freq
150  !      Z - Impedance
160  !      sex - MALE, FEMALE, NONE
170  !      media - COAX, WAVE
180  !      cutoff - Frequency for waveguide
190  ! 5) Assign the appropriate connector to each standard
200  ! 6) Modify the class assignments for the standards defined
210  ! 7) Verify the kit values
220  !
230  ! Additional Note:  After setting each new cal kit value, it is
240  ! recommended that the program periodically perform queries to
250  ! verify the new values.
260  !
270  ! This will prevent program synchronization issues that can affect
280  ! final values stored within new cal kits.
290  !
300  !-----------------------------------------------------------
310  !
320  ! Set up I/O path
330   ASSIGN @Na TO 716
340   DIM Calkname$[80],Conn$[80]
350   INTEGER Calkitnum
360  !
370   CLEAR SCREEN
380  !
390  !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
400  ! Designate the kit selection to be used for performing cal's
410   OUTPUT @Na;":sens:corr:ckit:count?"
420   ENTER @Na;Calkitnum
430   Calkitnum=Calkitnum+1
440   OUTPUT @Na;":sens:corr:coll:ckit "&VAL$(Calkitnum)
450  !
460  ! Name this kit with your own name
470   OUTPUT @Na;":sens:corr:coll:ckit:name ""Special 2.4 mm Model 85056"""
480  !
490  !
500   DISP "Defining kit std 1..."
510  ! Now set up standard #1
520   OUTPUT @Na;":sens:corr:coll:ckit:stan 1"
530   OUTPUT @Na;":sens:corr:coll:ckit:stan:type SHORT"
540   Get_std
550   OUTPUT @Na;":sens:corr:coll:ckit:stan:char coax"
560   OUTPUT @Na;":sens:corr:coll:ckit:stan:label ""My Short"""
570   Get_label
580  !
590   DISP "Defining kit std 2..."
600  ! Now set up standard #2
610   OUTPUT @Na;":sens:corr:coll:ckit:stan 2"
620   OUTPUT @Na;":sens:corr:coll:ckit:stan:type OPEN"
630   Get_std
640   OUTPUT @Na;":sens:corr:coll:ckit:stan:char coax"
650   OUTPUT @Na;":sens:corr:coll:ckit:stan:label ""My Open"""
660   Get_label
670  !
680   DISP "Defining kit std 3..."
690  ! Now set up standard #3
700   OUTPUT @Na;":sens:corr:coll:ckit:stan 3"
710   OUTPUT @Na;":sens:corr:coll:ckit:stan:type LOAD"
720   Get_std
730   OUTPUT @Na;":sens:corr:coll:ckit:stan:char coax"
740   OUTPUT @Na;":sens:corr:coll:ckit:stan:label ""My Fixed Load"""
750   Get_label
760  !
770   DISP "Defining kit std 4..."
780  ! Now set up standard #4
790   OUTPUT @Na;":sens:corr:coll:ckit:stan 4"
800   OUTPUT @Na;":sens:corr:coll:ckit:stan:type THRU"
810   Get_std
820   OUTPUT @Na;":sens:corr:coll:ckit:stan:char coax"
830   OUTPUT @Na;":sens:corr:coll:ckit:stan:label ""My Thru"""
840   Get_label
850  !
860   DISP "Defining kit std 5..."
870  ! Now set up standard #5
880   OUTPUT @Na;":sens:corr:coll:ckit:stan 5"
890   OUTPUT @Na;":sens:corr:coll:ckit:stan:type SLOAD"
900   Get_std
910   OUTPUT @Na;":sens:corr:coll:ckit:stan:char coax"
920   OUTPUT @Na;":sens:corr:coll:ckit:stan:label ""Sliding Load"""
```

```
930   Get_label
940  !
950   DISP "Defining kit std 6..."
960  ! Now set up standard #6
970  !
980   OUTPUT @Na;":sens:corr:coll:ckit:stan 6"
990   OUTPUT @Na;":sens:corr:coll:ckit:stan:type SHORT"
1000  Get_std
1010  OUTPUT @Na;":sens:corr:coll:ckit:stan:char coax"
1020  OUTPUT @Na;":sens:corr:coll:ckit:stan:label ""Short"""
1030  Get_label
1040 !
1050  DISP "Defining kit std 7..."
1060 ! Now set up standard #7
1070  OUTPUT @Na;":sens:corr:coll:ckit:stan 7"
1080  OUTPUT @Na;":sens:corr:coll:ckit:stan:type SHORT"
1090  Get_std
1100  OUTPUT @Na;":sens:corr:coll:ckit:stan:char coax"
1110  OUTPUT @Na;":sens:corr:coll:ckit:stan:label ""Short"""
1120  Get_label
1130 !
1140  DISP "Defining kit std 8..."
1150 ! Now set up standard #8
1160 !
1170  OUTPUT @Na;":sens:corr:coll:ckit:stan 8"
1190  OUTPUT @Na;":sens:corr:coll:ckit:stan:type ARBI"
1200  Get_std
1210  OUTPUT @Na;":sens:corr:coll:ckit:stan:char coax"
1220  OUTPUT @Na;":sens:corr:coll:ckit:stan:TZR 15;"
1230  OUTPUT @Na;":sens:corr:coll:ckit:stan:TZI -9;"
1240  OUTPUT @Na;":sens:corr:coll:ckit:stan:label ""Z Load"""
1250  Get_label
1260 !
1270 !
1280 !
1290 ! First remove any old connector names
1300  OUTPUT @Na;":sens:corr:coll:ckit:conn:del"
1310 ! Verify that no connectors are currently installed
1320  OUTPUT @Na;":sens:corr:coll:ckit:conn:cat?"
1330  ENTER @Na;Conn$
1340  PRINT "Verify empty list: ";Conn$
1350 !
1360 ! Define your new connectors
1370  OUTPUT @Na;":sens:corr:coll:ckit:conn:add ""PSC
2.4"",0HZ,999GHZ,50.0,MALE,COAX,0.0"
1380  OUTPUT @Na;":sens:corr:coll:ckit:conn:add ""PSC
2.4"",0HZ,999GHZ,50.0,FEMALE,COAX,0.0"
1390 !
1400 ! Verify that the new connectors are installed
1410  OUTPUT @Na;":sens:corr:coll:ckit:conn:cat?"
1420  ENTER @Na;Conn$
1430  PRINT "Verify new connectors: ";Conn$
1440  DISP ""
```

```
1450 !
1460  DISP "Defining conn std 1..."
1470 ! Now set up standard #1
1480  OUTPUT @Na;":sens:corr:coll:ckit:stan 1"
1490  Verify_std
1500  OUTPUT @Na;":sens:corr:coll:ckit:conn:snam ""PSC 2.4"",FEMALE,1"
1510  Print_connector
1520 !
1530  DISP "Defining conn std 2..."
1540 ! Now set up standard #2
1550  OUTPUT @Na;":sens:corr:coll:ckit:stan 2"
1560  Verify_std
1570  OUTPUT @Na;":sens:corr:coll:ckit:conn:snam ""PSC 2.4"",FEMALE,1"
1580  Print_connector
1590 !
1600  DISP "Defining conn std 3..."
1610 ! Now set up standard #3
1620  OUTPUT @Na;":sens:corr:coll:ckit:stan 3"
1630  Verify_std
1640  OUTPUT @Na;":sens:corr:coll:ckit:conn:snam ""PSC 2.4"",FEMALE,1"
1650  Print_connector
1660 !
1670  DISP "Defining conn std 4..."
1680 ! Now set up standard #4
1690  OUTPUT @Na;":sens:corr:coll:ckit:stan 4"
1700  Verify_std
1710  OUTPUT @Na;":sens:corr:coll:ckit:conn:snam ""PSC 2.4"",FEMALE,1"
1720  OUTPUT @Na;":sens:corr:coll:ckit:conn:snam ""PSC 2.4"",MALE,2"
1730  Print_connector
1740 !
1750  DISP "Defining conn std 5..."
1760 ! Now set up standard #5
1770  OUTPUT @Na;":sens:corr:coll:ckit:stan 5"
1780  OUTPUT @Na;":sens:corr:coll:ckit:stan:label ""Sliding Load"""
1790  Verify_std
1800  OUTPUT @Na;":sens:corr:coll:ckit:conn:snam ""PSC 2.4"",MALE,1"
1810  Print_connector
1820 !
1830  DISP "Defining conn std 6..."
1840 ! Now set up standard #6
1850 !
1860  OUTPUT @Na;":sens:corr:coll:ckit:stan 6"
1870  Verify_std
1880  OUTPUT @Na;":sens:corr:coll:ckit:conn:snam ""PSC 2.4"",MALE,1"
1890  Print_connector
1900 !
1910  DISP "Defining conn std 7..."
1920 ! Now set up standard #7
1930  OUTPUT @Na;":sens:corr:coll:ckit:stan 7"
1940  Verify_std
1950  OUTPUT @Na;":sens:corr:coll:ckit:conn:snam ""PSC 2.4"",MALE,1"
1960  Print_connector
1970 !
```

```
1980  DISP "Defining conn std 8..."
1990 ! Now set up standard #8
2000  OUTPUT @Na;":sens:corr:coll:ckit:stan 8"
2010  Verify_std
2020  OUTPUT @Na;":sens:corr:coll:ckit:conn:snam ""PSC 2.4"",MALE,1"
2030  Print_connector
2040 !
2050  DISP "Class assignments..."
2060 !
2070 ! Designate the "order" associated with measuring the standards
2080 !
2090 !    Set Port 1, 1st standard measured to be standard #2
2100  OUTPUT @Na;":sens:corr:coll:ckit:order1 2"
2110 !    Set Port 1, 2nd standard measured to be standard #1
2120  OUTPUT @Na;":sens:corr:coll:ckit:order2 1,6,7"
2130 !    Set Port 1, 3nd standard measured to be standard #3 and #5
2140  OUTPUT @Na;":sens:corr:coll:ckit:order3 3,5"
2150 !    Set Port 1, 4th standard measured to be standard #4
2160  OUTPUT @Na;":sens:corr:coll:ckit:order4 4"
2170 !
2180 !    Set Port 2, 1st standard measured to be standard #2
2190  OUTPUT @Na;":sens:corr:coll:ckit:order5 2"
2200 !    Set Port 2, 2nd standard measured to be standard #1
2210  OUTPUT @Na;":sens:corr:coll:ckit:order6 1,6,7"
2220 !    Set Port 2, 3nd standard measured to be standard #3 and #6
2230  OUTPUT @Na;":sens:corr:coll:ckit:order7 3,5"
2240 !    Set Port 2, 4th standard measured to be standard #4
2250  OUTPUT @Na;":sens:corr:coll:ckit:order8 4"
2260 !
2270 !    Set Port 1, 1st standard
2280  OUTPUT @Na;":sens:corr:coll:ckit:olabel1 ""MyOpen1"""
2290 !    Set Port 1, 2nd standard
2300  OUTPUT @Na;":sens:corr:coll:ckit:olabel2 ""MyShorts1"""
2310 !    Set Port 1, 3nd standard
2320  OUTPUT @Na;":sens:corr:coll:ckit:olabel3 ""MyLoads1"""
2330 !    Set Port 1, 4th standard measured to be standard #4
2340  OUTPUT @Na;":sens:corr:coll:ckit:olabel4 ""MyThru1"""
2350 !
2360 !    Set Port 2, 1st standard
2370  OUTPUT @Na;":sens:corr:coll:ckit:olabel5 ""MyOpen2"""
2380 !    Set Port 2, 2nd standard
2390  OUTPUT @Na;":sens:corr:coll:ckit:olabel6 ""MyShorts2"""
2400 !    Set Port 2, 3nd standard
2410  OUTPUT @Na;":sens:corr:coll:ckit:olabel7 ""MyLoads2"""
2420 !    Set Port 2, 4th standard
2430  OUTPUT @Na;":sens:corr:coll:ckit:olabel8 ""MyThrus2"""
2440 !
2450  BEEP
2460  DISP "Done!"
2470  END
2480  SUB Get_label
2490    OUTPUT 716;":sens:corr:coll:ckit:stan:label?"
2500    ENTER 716;Label$
```

```
2510    PRINT Label$
2520  SUBEND
2530 !
2540  SUB Get_std
2550    OUTPUT 716;":sens:corr:coll:ckit:stan:type?"
2560    ENTER 716;Type$
2570    PRINT Type$
2580  SUBEND
2590 !
2600  SUB Print_connector
2610    DIM Nam$[40]
2620    OUTPUT 716;":sens:corr:coll:ckit:conn:sname?"
2630    ENTER 716;Nam$
2640    PRINT Nam$
2650  SUBEND
2660 !
2670  SUB Verify_std
2680    OUTPUT 716;":sens:corr:coll:ckit:stan:label?"
2690    ENTER 716;Label$
2700  SUBEND
2710 !
```

Modify a Calibration Kit using SCPI

This Visual Basic program:

- Modifies Calibration kit number 3
- Completely defines standard #4 (thru)

To run this program, you need:

- An established GPIB interface connection

See Other SCPI Example Programs

```
'Modifying cal kit number 3
Calkitnum = 3

'Designate the kit selection to be used for performing cal's
GPIB.Write "SENSe:CORRection:COLLect:CKIT:SELect " & Val(Calkitnum)

'Reset to factory default values.
GPIB.Write "SENSe:CORRection:COLLect:CKIT:RESet " & Val(Calkitnum)

'Name this kit with your own name
GPIB.Write "SENSe:CORRection:COLLect:CKIT:NAME 'My Cal Kit'"

'Assign standard numbers to calibration classes
'Set Port 1, class 1 (S11A) to be standard #8
GPIB.Write "SENSe:CORRection:COLLect:CKIT:ORDer1 8"
'Set Port 1, class 2 (S11B) to be standard #7
GPIB.Write "SENSe:CORRection:COLLect:CKIT:ORDer2 7"
'Set Port 1, class 3 (S11C) to be standard #3
GPIB.Write "SENSe:CORRection:COLLect:CKIT:ORDer3 3"
'Set Port 1, class 4 (S21T) to be standard #4
GPIB.Write "SENSe:CORRection:COLLect:CKIT:ORDer4 4"
'Set Port 2, class 1 (S22A) to be standard #8
GPIB.Write "SENSe:CORRection:COLLect:CKIT:ORDer5 8"
'Set Port 2, class 2 (S22B) to be standard #7
GPIB.Write "SENSe:CORRection:COLLect:CKIT:ORDer6 7"
'Set Port 2, class 3 (S22C) to be standard #3
GPIB.Write "SENSe:CORRection:COLLect:CKIT:ORDer7 3"
'Set Port 2, class 4 (S12T) to be standard #4
GPIB.Write "SENSe:CORRection:COLLect:CKIT:ORDer8 4"

'Set up Standard #4 completely
'Select Standard #4; the rest of the commands act on it
GPIB.Write "SENSe:CORRection:COLLect:CKIT:STANdard 4"
GPIB.Write "SENSe:CORRection:COLLect:CKIT:STANdard:FMIN 300KHz"
GPIB.Write "SENSe:CORRection:COLLect:CKIT:STANdard:FMAX 9GHz"
GPIB.Write "SENSe:CORRection:COLLect:CKIT:STANdard:IMPedance 50"
GPIB.Write "SENSe:CORRection:COLLect:CKIT:STANdard:DELay 1.234 ns"
GPIB.Write "SENSe:CORRection:COLLect:CKIT:STANdard:LOSS 23e6"
GPIB.Write "SENSe:CORRection:COLLect:CKIT:STANdard:C0 0"
GPIB.Write "SENSe:CORRection:COLLect:CKIT:STANdard:C1 1"
GPIB.Write "SENSe:CORRection:COLLect:CKIT:STANdard:C2 2"
GPIB.Write "SENSe:CORRection:COLLect:CKIT:STANdard:C3 3"
```

```
GPIB.Write "SENSe:CORRection:COLLect:CKIT:STANdard:L0 10"
GPIB.Write "SENSe:CORRection:COLLect:CKIT:STANdard:L1 11"
GPIB.Write "SENSe:CORRection:COLLect:CKIT:STANdard:L2 12"
GPIB.Write "SENSe:CORRection:COLLect:CKIT:STANdard:L3 13"
GPIB.Write "SENSe:CORRection:COLLect:CKIT:STANdard:LABel 'My Special Thru'"
GPIB.Write "SENSe:CORRection:COLLect:CKIT:STANdard:TYPE THRU"
GPIB.Write "SENSe:CORRection:COLLect:CKIT:STANdard:CHARacteristic Coax"
```

GPIB using Visual C++

See Other SCPI Example Programs

```
/*
 * This example assumes the user's PC has a National Instruments GPIB board. The
example is comprised of three basic parts:
 *
 * 1. Initialization
 * 2. Main Body
 * 3. Cleanup
 *
 * The Initialization portion consists of getting a handle to the PNA and then doing
a GPIB clear of the PNA.
 *
 * The Main Body consists of the PNA SCPI example.
 *
 * The last step, Cleanup, releases the PNA for front panel control.
 */

#include <stdio.h>
#include <stdlib.h>

/*
 * Include the WINDOWS.H and DECL-32.H files. The standard Windows
 * header file, WINDOWS.H, contains definitions used by DECL-32.H and
 * DECL-32.H contains prototypes for the NI GPIB routines and constants.
 */
#include <windows.h>
#include "decl-32.h"

#define ERRMSGSIZE 1024 // Maximum size of SCPI command string
#define ARRAYSIZE 1024 // Size of read buffer

#define BDINDEX 0 // Board Index of GPIB board
#define PRIMARY_ADDR_OF_PNA 16 // GPIB address of PNA
#define NO_SECONDARY_ADDR 0 // PNA has no Secondary address
#define TIMEOUT T10s // Timeout value = 10 seconds
#define EOTMODE 1 // Enable the END message
#define EOSMODE 0 // Disable the EOS mode

int pna;
char ValueStr[ARRAYSIZE + 1];
char ErrorMnemonic[21][5] = {"EDVR", "ECIC", "ENOL", "EADR", "EARG",
  "ESAC", "EABO", "ENEB", "EDMA", "",
  "EOIP", "ECAP", "EFSO", "", "EBUS",
  "ESTB", "ESRQ", "", "", "", "ETAB"};


void GPIBWrite(char* SCPIcmd);
char *GPIBRead(void);
void GPIBCleanup(int Dev, char* ErrorMsg);


int main()
{
```

```
char *opc;
char *result;
char *value;

/*
 * ==========================================
* INITIALIZATION SECTION
 * ==========================================
 */

  /*
  * The application brings the PNA online using ibdev. A device handle,pna, is
returned and is used in all subsequent calls to the PNA.
  */
  pna = ibdev(BDINDEX, PRIMARY_ADDR_OF_PNA, NO_SECONDARY_ADDR,
  TIMEOUT, EOTMODE, EOSMODE);
  if (ibsta & ERR)
  {
  printf("Unable to open handle to PNA\nibsta = 0x%x iberr = %d\n",
  ibsta, iberr);
  return 1;
  }

  /*
  * Do a GPIB Clear of the PNA. If the error bit ERR is set in ibsta, call
GPIBCleanup with an error message.
  */
  ibclr (pna);
  if (ibsta & ERR)
  {
  GPIBCleanup(pna, "Unable to perform GPIB clear of the PNA");
  return 1;
  }

/*
 * ==========================================
 * MAIN BODY SECTION
 * ==========================================
 */

  // Reset the analyzer to instrument preset
  GPIBWrite("SYSTem:FPRESET");

  // Create S11 measurement
  GPIBWrite("CALCulate1:PARameter:DEFine 'My_S11',S11");

  // Turn on Window #1
  GPIBWrite("DISPlay:WINDow1:STATe ON");

  // Put a trace (Trace #1) into Window #1 and 'feed' it from the measurement
  GPIBWrite("DISPlay:WINDow1:TRACe1:FEED 'My_S11'");

  // Setup the channel for single sweep trigger
  GPIBWrite("INITiate1:CONTinuous OFF;*OPC?");
  opc = GPIBRead();
  GPIBWrite("SENSe1:SWEep:TRIGger:POINt OFF");
```

```
  // Set channel parameters
  GPIBWrite("SENSe1:SWEep:POINts 11");
  GPIBWrite("SENSe1:FREQuency:STARt 1000000000");
  GPIBWrite("SENSe1:FREQuency:STOP 2000000000");

  // Send a trigger to initiate a single sweep
  GPIBWrite("INITiate1;*OPC?");
  opc = GPIBRead();

  // Must select the measurement before we can read the data
  GPIBWrite("CALCulate1:PARameter:SELect 'My_S11'");

  // Read the measurement data into the "result" string variable
  GPIBWrite("FORMat ASCII");
  GPIBWrite("CALCulate1:DATA? FDATA");
  result = GPIBRead();

  // Print the data to the display console window
printf("S11(dB) - Visual C++ SCPI Example for PNA\n\n");
  value = strtok(result, ",");
while (value != NULL)
  {
  printf("%s\n", value);
  value = strtok(NULL, ",");
  }

/*
 * ========================================
 * CLEANUP SECTION
 * ========================================
 */

  /* The PNA is returned to front panel control. */
  ibonl(pna, 0);

  return 0;
}

/*
 * Write to the PNA
 */
void GPIBWrite(char* SCPIcmd)
{
int length;
char ErrorMsg[ERRMSGSIZE + 1];
length = strlen(SCPIcmd) ;

  ibwrt (pna, SCPIcmd, length);
  if (ibsta & ERR)
  {
  strcpy(ErrorMsg, "Unable to write this command to PNA:\n");
  strcat(ErrorMsg, SCPIcmd);

  GPIBCleanup(pna, ErrorMsg);
  exit(1);
  }
```

```
}

/*
 * Read from the PNA
 */
char* GPIBRead(void)
{
  ibrd (pna, ValueStr, ARRAYSIZE);
  if (ibsta & ERR)
   {
  GPIBCleanup(pna, "Unable to read from the PNA");
  exit(1);
   }
else
  return ValueStr;
}

/*
 * After each GPIB call, the application checks whether the call succeeded. If an
NI-488.2 call fails, the GPIB driver sets the corresponding bit in the global status
variable. If the call failed, this procedure prints an error message, takes the PNA
offline and exits.
 */
void GPIBCleanup(int Dev, char* ErrorMsg)
{
  printf("Error : %s\nibsta = 0x%x iberr = %d (%s)\n",
  ErrorMsg, ibsta, iberr, ErrorMnemonic[iberr]);
  if (Dev != -1)
   {
  printf("Cleanup: Returning PNA to front panel control\n");
  ibonl (Dev, 0);
   }
}
```

Create a Custom Power Meter Driver

Note: This topic requires that you have a working knowledge of Visual Basic.

This topic will help you create your own power meter driver for use with Source Power Calibration on the PNA. If you are using one of the following Power Meters to perform a Source Power Calibration, you do NOT need to create your own driver:

E4416A, E4417A, E4418A/B, E4419A/B, 437B, 438A, EPM-441A, EPM-442A

Your Power Meter driver will be created from a template written in Visual Basic using VISA over the GPIB bus.

Note: This procedure applies to Visual Basic 6.0. Applicability to Visual Basic .NET has not yet been investigated.

- **Prepare Template Files**
- **Modify Template Files**
- **Compile, Copy, and Register, Your New Driver**
- **Test Your new Driver**

Other SCPI Example Programs

Prepare Template Files

1. Copy all the files from the PNA hard drive C:\Program Files\Agilent\Network Analyzer\Automation\Power Meter Driver Template folder, to a folder on your development PC.
2. In Visual Basic click **File**, then **Open Project…**, find **MyPowerMeter.vbp** (a file you copied from the PNA). Click **Open**. This is a VB ActiveX EXE template, which you will fill in to become your driver.
3. Click **Project**, then **MyPowerMeter Properties**. Click the **General** tab.
4. Overwrite the Project Name with a name of your own choosing. This will be the name of your driver's type library (also the default name of your exe).

Note If the name of your exe does not match the VB Project Name with which it was compiled, registration of the exe on the PNA will not succeed.

5. Set the Project Description. After building your driver if you wish to test it using VB, this is the string that will show up in the VB References list of your test project, and also in the lower pane of the VB Object Browser.
6. Set the Thread Pool size to 1 thread.
7. Click **OK** to close the project properties dialog.
8. From the VB **Project** menu, click **References…** Ensure that **Agilent PNA Power Meter 1.0 Type Library** and **VISA Library** are checked. Click **OK.**

Note: Agilent's implementation of VISA is installed as part of the Agilent I/O Libraries on the PNA. For help on VISA, go to the Windows Start button on your PNA, select Programs, Agilent IO Libraries, VISA Help.

Modify Template Files

From Visual Basic **View** menu click **Project Explorer**. Expand the **Modules** and **Class Modules** folders. Ensure there is one module (WinAPI) and one class module (PowerMeter).

Let's look at the WinAPI module first.

1. In the **Project Explorer** window, click **WinAPI**.
2. From the **View** menu click **Code**.

There is only one line of code you should need to modify in this module: the value of the string constant named sIDSEARCH. The comments preceding the declaration of that string describe how to change it. The rest of this module contains functions which will use the Microsoft Windows API to insure proper registration of your driver on the PNA. If you know of other Windows API functions you feel might be helpful to call from within your PowerMeter class module (to help in formatting data, for example), this module would be the place to declare them.

Now let's look at the class module.

1. In the Project Explorer window, click **PowerMeter**.
2. From the **View** menu click **Properties Window.** The **Instancing** property must be set to MultiUse. This allows other applications to create objects from this class, such that one instance of your driver EXE can supply more than one such object at a time.
3. From the **View** menu click **Code.**

Do NOT modify the Interfaces to IPowerMeter subroutines and functions. PNA source power cal expects to find these interfaces as they are currently defined.

The only members that you need to supply code to are those containing **"Your code here"** comments.

In addition, comments have been provided at the beginning of each member to describe the information that member needs to be read from or written to the power meter.

To get an idea of how communicate with the power meter using the VISA functions **viWrite** and **viRead**, examine the code which has been implemented for you in IPowerMeter_Connect, IPowerMeter_QueryMeter, and IPowerMeter_WriteMeter.

Compile, Copy, and Register Your New Driver

When your driver is ready to run, you will first need to compile it into an EXE.

From the File menu select **Make exe.**

After compiling, the following will instruct VB to use the same ID (GUID) every time you re-compile your project.

1. From the **Project** menu, click **PowerMeter Properties**.
2. On the **Component** tab, select **Binary Compatibility** and click **...**
3. Browse to and select your project EXE. Click **Open.**
4. Click **OK** to close **Project Properties.**
5. Save your project.
6. Copy your driver EXE file to a folder on your PNA (do NOT use C:\Program Files\Agilent\Network Analyzer\Automation\Power Meter Driver Template folder).

7. Run the EXE file. A message box will pop up reporting whether or not registration was successful. If not successful, it will make a suggestion on what to fix.

When your driver is properly registered, PNA Source Power Cal should be able to associate it with the ID string of your power meter.

Test Your Power Meter Driver

We have also provided a Visual Basic project to test your new Power Meter driver. This project individually calls every IPowerMeter method and property in your driver to verify that it performs correctly. Before running the test your PC and PNA must be configured to communicate using DCOM.

1. Connect your PC and the PNA to LAN.
2. Add your PC logon to the PNA. Both logons and password must match to communicate using DCOM. See Additional PNA users.
3. Configure your driver using DCOM Config on the PNA. This will give you permission to launch and access the driver. See Configure for COM-DCOM Programming.

Modify the Test Project

1. In Visual Basic click **File**, then **Open Project…**, find **MyPowerMeterTest.vbp** (a file you copied from the PNA). Click **Open**.
2. From the **Project** menu, click **References…** From the list, find and check your new Power Meter Driver. (It should have been registered on your PC when you successfully made your driver EXE.) Click **OK.**
3. From the **View** menu click **Code.**
4. Modify the **CreateObject** line as follows:
   Replace **MyPowerMeter** with the Project Name that you chose for your driver
   Replace **MyPNA** with the Computer Name of your PNA.
   For example:

```
Set PowerMeterObj = CreateObject("AcmeBrand.PowerMeter", "AGILENT-PNA123")
```

(This assumes that you kept **PowerMeter** as class module name in your driver.)

Run the Test Project

Ensure your power meter is connected to the PNA with a GPIB cable.

Put the PNA in system controller mode:

1. From the PNA **System** menu point to **Configure** then click **SICL/GPIB**.
2. In the GPIB box click **System Controller**.

Run the test project. If there are no errors, the driver is created successfully. If there are errors, try to figure out what went wrong and fix it. Then re-compile, re-copy the .exe to the PNA, and re-run the test. You should not need to re-register the driver or re-modify the test program.

GPIB Pass-Through Example

The SCPI SYSTem commands used in this example allow you to send GPIB commands to another GPIB device through the PNA. The other GPIB device cannot be connected to the GPIB bus through the PNA rear panel if the PNA is being controlled by a remote PC using that connector. The other device would typically be connected to the PNA using a USB/GPIB interface.

This VB Script example uses the COM SCPIStringParser object. However, this is not critical to the use of these commands; they can be sent using the normal syntax of your programming environment. Using the SCPIStringParser over LAN allows you to communicate with GPIB devices without requiring your remote PC to have a GPIB interface card installed.

Although this method of pass-through works for most applications, there are a couple of limitations:

- All data is transferred using ASCII format. Therefore, transferring large blocks of data is very slow.
- Only read and write functions are possible. Service Interrupts are not supported.

See Other SCPI Example Programs

```
option explicit
dim app
set app = CreateObject("AgilentPNA835x.Application")

dim p
set p = app.ScpiStringParser

' Open a new GPIB session on Bus:2 Device:14 Timeout: 100ms
p.Parse "SYST:COMM:GPIB:RDEV:OPEN 2,14,100"
dim handleAsStr

' Retrieve the handle (ID number)
handleAsStr = p.Parse ("SYST:COMM:GPIB:RDEV:OPEN?")

' Convert the handle to an integer
dim handleAsInt
handleAsInt = CInt(handleAsStr)

' Send the "*IDN?" query
p.Parse "SYST:COMM:GPIB:RDEV:WRITE " & handleAsInt & ",'*IDN?'"

' Read its results
dim idn
idn = p.Parse("SYST:COMM:GPIB:RDEV:READ? " & handleAsInt)
msgbox idn

' Close the GPIB session
p.Parse "SYST:COMM:GPIB:RDEV:CLOSE " & handleAsInt
```

PNA as Controller and Talker / Listener

This Visual Basic Program uses VISA to do the following:

- This Visual Basic Program uses VISA to do the following:
- Control the PNA using a VISA LAN Client interface on the PNA.
- Control another instrument using the PNA as GPIB controller.
- Queries both the analyzer and other instrument to identify themselves with *IDN?

Note: This program can be modified to work from a remote PC to control both instruments. In that case, set up the PNA to be a talker/listener.

To run this program, you need to do the following:

- Add module **visa32.bas** to the VB project. It is located on the analyzer at C:\Program Files\HP\VXIPNP\WINNT\Include\VISA32.bas
- Configure the PNA for VISA / SICL
- Set up the PNA to be GPIB system controller.
  1. On the **System** menu, point to **Configure**. Click **SICL / GPIB**
  2. Click **System Controller**
- Connect another instrument to the analyzer through a GPIB cable with Primary address of 13 on GPIB0 interface

See Other SCPI Example Programs

```
Sub main()

'This application run from onboard the PNA
'can control both the PNA and another GPIB instrument.
'
'To run this program the module visa32.bas must be added
'to the project.

'VISA function status return code
Dim status As Long
'Session to Default Resource Manager
Dim defRM As Long
'Session to instrument
Dim viPNA As Long
'Session to other GPIB instrument
Dim viInstrument As Long
'String to hold results
Dim strRes As String * 200
On Error GoTo ErrorHandler

status = viOpenDefaultRM(defRM)
If (status < VI_SUCCESS) Then GoTo VisaErrorHandler


'Open the session to the PNA
status = viOpen(defRM, "GPIB1::16::INSTR", 0, 0, viPNA)
```

```
If (status < VI_SUCCESS) Then GoTo VisaErrorHandler

'Ask for the PNA's ID.
status = viVPrintf(viPNA, "*IDN?" + Chr$(10), 0)
If (status < VI_SUCCESS) Then GoTo VisaErrorHandler

'Read the ID as a string.
status = viVScanf(viPNA, "%t", strRes)
If (status < VI_SUCCESS) Then GoTo VisaErrorHandler
'Display the results
MsgBox "PNA is: " + strRes


'Open the session to the other instrument
status = viOpen(defRM, "GPIB0::13::INSTR", 0, 0, viInstrument)
If (status < VI_SUCCESS) Then GoTo VisaErrorHandler

'Ask for the instrument's ID.
status = viVPrintf(viInstrument, "*IDN?" + Chr$(10), 0)
If (status < VI_SUCCESS) Then GoTo VisaErrorHandler

'Read the ID as a string.
status = viVScanf(viPNA, "%t", strRes)
If (status < VI_SUCCESS) Then GoTo VisaErrorHandler

'Display the results
MsgBox "Other instrument is: " + strRes
' Close the resource manager session (which closes everything)
Call viClose(defRM)
End

ErrorHandler:
'Display the error message
MsgBox "*** Error : " + Error$, MB_ICONEXCLAMATION
End

VisaErrorHandler:
Dim strVisaErr As String * 200
Call viStatusDesc(defRM, status, strVisaErr)
MsgBox "*** Error : " + strVisaErr

End
End Sub
```

GPIB Fundamentals

The General Purpose Interface Bus (GPIB) is a system of hardware and software that allows you to control test equipment to make measurements quickly and accurately. This topic contains the following information:

- **The GPIB Hardware Components**
- **The GPIB / SCPI Programming Elements**
- **How to Configure for GPIB / SICL**
- **LCL- RMT Operation Label**
- **Specifications**
- **GPIB Interface Capability Codes**

Note: All of the topics related to programming assume that you already know how to program, preferably using a language that can control instruments.

Other Topics about GPIB Concepts

The GPIB Hardware Components

The system bus and its associated interface operations are defined by the IEEE 488 standard. The following sections list and describe the main pieces of hardware in a GPIB system:

Instruments

The analyzer is configured as a Talker / Listener by default.

- **Talkers** are instruments that can be addressed to send data to the controller.
- **Listeners** are instruments that can be addressed to receive a command, and then respond to the command. All devices on the bus are required to listen.

GPIB Addresses

Every GPIB instrument must have its own unique address on the bus. The analyzer address (716) consists of two parts:

1. **The Interface select code** (typically 7) indicates which GPIB port in the system controller is used to communicate with the device.
2. **The primary address** (16) is set at the factory. You can change the primary address of any device on the bus to any number between 0 and 30. To change the analyzer address click **System \ Configure \ SICL-GPIB**

The secondary address is sometimes used to allow access to individual modules in a modular instrument system, such as a VXI mainframe. The analyzer does not have secondary addresses.

Controllers

Controllers specify the instruments that will be the talker and listener in a data exchange. The controller of the bus must have a GPIB interface card to communicate on the GPIB.

- The **Active Controller** is the computer or instrument that is currently controlling data exchanges.

- The **System Controller** is the only computer or instrument that can take control and give up control of the GPIB to another computer or instrument, which is then called the active controller.

The PNA can NOT be passed control of the GPIB. However, you can communicate with other GPIB devices through the PNA using one of, or a combination of, the following methods:

- Use the SCPI SYST:COMM:GPIB:RDEV: commands.
- Use VISA or SICL over LAN to accomplish this. See an example.
- Use USB / GPIB Interface

Cables

GPIB Cables are the physical link connecting all of the devices on the bus. There are eight data lines in a GPIB cable that send data from one device to another. There are also eight control lines that manage traffic on the data lines and control other interface operations.

You can connect instruments to the controller in any arrangement with the following limitations:

- Do not connect more than 15 devices on any GPIB system. This number can be extended with the use of a bus extension.
- Do not exceed a total of 20 meters of total cable length or 2 meters per device, whichever is less.
- Avoid stacking more than three connectors on the back panel of an instrument. This can cause unnecessary strain on the rear-panel connector.

The GPIB / SCPI Programming Elements

The following software programming elements combine to become a GPIB program:

- **GPIB / SCPI Commands**
- **Programming Statements**
- **Instrument Drivers**

GPIB Commands

The GPIB command is the basic unit of communication in a GPIB system. The analyzer responds to three types of GPIB commands:

1. IEEE 488.1 Bus-management Commands

These commands are used primarily to tell some or all of the devices on the bus to perform certain interface operations.

All of the functions that can be accomplished with these commands can also be done with IEEE 488.2 or SCPI commands. Therefore, these commands are not documented in this Help system. For a complete list of IEEE 488.1 commands refer to the IEEE 488 standard. **Examples** of IEEE 488.1 Commands

- **`CLEAR` - Clears the bus of any pending operations**
- **`LOCAL` - Returns instruments to local operation**

2. IEEE 488.2 Common Commands

These commands are sent to instruments to perform interface operations. An IEEE 488.2 common command consists of a single mnemonic and is preceded by an asterisk (*). Some of the commands have a query form which adds a "?" after the command. These commands ask the instrument for the current setting. See a complete list of the Common Commands that are recognized by the analyzer. **Examples** of IEEE 488.2 Common Commands

- `*OPC  - Operation Complete`
- `*RST  - Reset`
- `*OPT? - Queries the option configuration`

3. SCPI Commands

The Standard Commands for Programmable Instruments (SCPI) is a set of commands developed in 1990. The standardization provided in SCPI commands helps ensure that programs written for a particular SCPI instrument are easily adapted to work with a similar SCPI instrument. SCPI commands tell instruments to do device specific functions. For example, SCPI commands could tell an instrument to make a measurement and output data to a controller. **Examples** of SCPI Commands:

```
CALCULATE:AVERAGE:STATE ON
SENSE:FREQUENCY:START?
```

For more information on SCPI:

- The Rules and Syntax of SCPI Commands provides more detail of the SCPI command structure.
- SCPI Command Tree is a complete list of the SCPI commands for the analyzer

Programming Statements

SCPI commands are included with the language specific I/O statements to form program statements. The programming language determines the syntax of the programming statements. SCPI programs can be written in a variety of programming languages such as VEE, HP BASIC, or C++. **Example** of a Visual Basic statement:

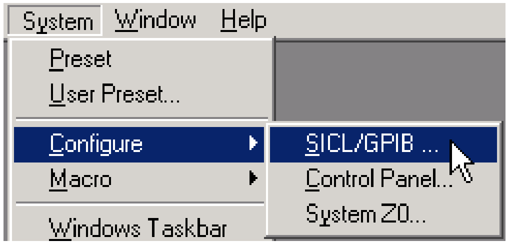
- `GPIB.Write "SOURCE:FREQUENCY:FIXED 1000 MHz"`

Note about examples

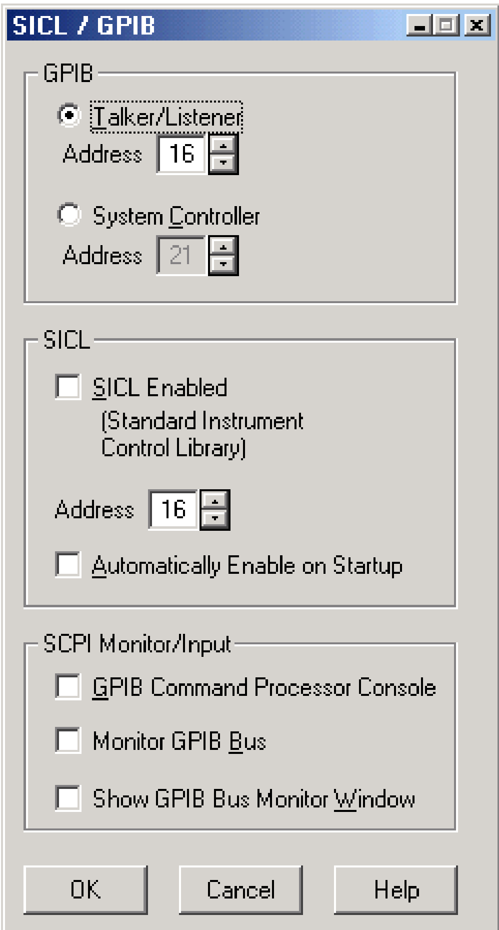
Instrument Drivers

Instrument drivers are subroutines that provide routine functionality and can be reused from program to program. GPIB industry leaders have written standards for use by programmers who develop drivers. When programmers write drivers that comply with the standards, the drivers can be used with predictable results. To comply with the standard, each instrument driver must include documentation describing its functionality and how it should be implemented.

How to configure for SICL / GPIB Operation



Learn more about using the front panel interface



SICL / GPIB dialog box help

GPIB

Talker/Listener Sets the PNA to receive and send GPIB/SCPI messages to the system controller (external computer).

Talker/Listener Address Sets the PNA talker/listener GPIB address.

System Controller Sets the PNA as the system controller, controlling GPIB communications of external devices. Learn about the PNA as controller.

System Controller Address Sets the PNA system controller GPIB address.

SICL

SICL Enabled When checked, the analyzer is capable of running GPIB programs on its computer to control analyzer functions. The programs must be run from a GPIB-capable programming environment (VEE, Visual Basic). This mode does not allow control of external GPIB instruments. To uncheck this box, exit the PNA application - (Click File, then Exit). The PNA restarts with the SICL enabled box unchecked unless **Automatically Enable on Startup** is checked.

Learn more about Configuring for VISA and SICL.

Address Sets the PNA address.

Automatically Enable on Startup When checked, SICL Enabled is automatically selected when starting the PNA application.

SCPI Monitor / Input

GPIB Command Processor Console Launches a window that is used to send single SCPI/GPIB commands.

Note: Press **Control**+**Z** , then enter, to close the console window.

- Type a valid command, with appropriate arguments and press enter.
- Use the arrow keys to recall previous commands.
- The console window may launch behind the PNA application. Press **Control**+**Tab** to bring the console window to the top.

Monitor GPIB Bus Enables monitoring activity on the GPIB.

Show GPIB Bus Monitor Window Shows and hides the window monitoring GPIB activity.

LCL and RMT Operation

The analyzer **LCL** and **RMT** (Local and Remote) operation labels appear in the lower right corner of the status bar.

Note: The status bar is NOT visible when the analyzer is preset. To make the bar visible, click **View** then **Status Bar**

- **LCL** appears when not under SCPI control
- **RMT** appears when under SCPI control. The RMT label does NOT appear when under COM control

  RMT disables the front panel keys except for the **Macro/Local** key.

Pressing the Macro / Local key returns the analyzer to Local (front panel) operation.

The IEEE488.1 "GTL" (go to local) command also returns the analyzer to Local (front panel) operation.

The IEEE488.1 "LLO" (local lockout) command disables the front panel Local button.

GPIB Specifications

Interconnected devices - Up to 15 devices (maximum) on one contiguous bus.

Interconnection path - Star or linear (or mixed) bus network, up to 20 meters total transmission path length or 2 meters per device, whichever is less.

Message transfer scheme - Byte-serial, bit-parallel, asynchronous data transfer using an interlocking 3-wire handshake.

Maximum data rate - 1 megabyte per second over limited distances, 250 to 500 kilobytes per second typical maximum over a full transmission path. The devices on the bus determine the actual data rate.

Address capability - Primary addresses, 31 Talk and 31 Listen; secondary addresses, 961 Talk and 961 Listen. There can be a maximum of 1 Talker and up to 14 Listeners at a time on a single bus. See also previous section on GPIB addresses.

GPIB Interface Capability Codes

The IEEE 488.1 standard requires that all GPIB compatible instruments display their interface capabilities on the rear panel using codes. The codes on the analyzer, and their related descriptions, are listed below:

SH1 full source handshake capability

AH1 full acceptor handshake capability

T6 basic talker, serial poll, no talk only, unaddress if MLA (My Listen Address)

TEO no extended talker capability

L4 basic listener, no listen only, unaddress if MTA (My Talk Address)

LEO no extended listener capability

SR1 full service request capability

RL1 full remote / local capability

PPO **no parallel poll capability**

DC1 full device clear capability

DT1 full device trigger capability

C1 system controller capability

C2 send IFC (Interface Clear) and take charge controller capability

C3 send REN (Remote Enable) controller capability

C4 respond to SRQ (Service Request)

The Rules and Syntax of SCPI

Most of the commands used for controlling instruments on the GPIB are SCPI commands. The following sections will help you learn to use SCPI commands in your programs.

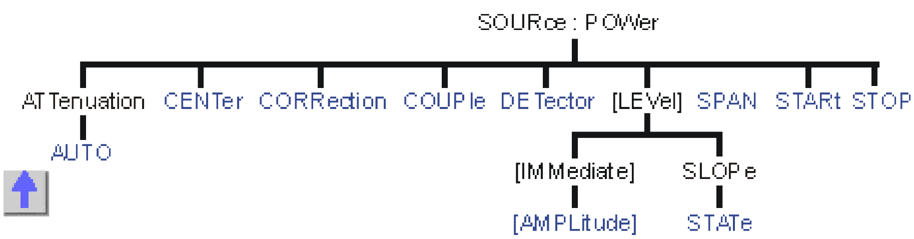
- Branches on the Command Tree
- Command and Query
- Multiple Commands
- Command Abbreviation
- Bracketed (Optional) Keywords
- Vertical Bars (Pipes)
- MIN and MAX Parameters

Other Topics about GPIB Concepts

Branches on the Command Tree

All major functions on the analyzer are assigned keywords which are called ROOT commands. (See GPIB Command Finder for a list of SCPI root commands). Under these root commands are branches that contain one or more keywords. The branching continues until each analyzer function is assigned to a branch. A root command and the branches below it is sometimes known as a subsystem.

For example, the following graphic shows the SOURce subsystem. Under the `SOURce` and `POWer` keywords are several branch commands.



Sometimes the same keyword, such as `STATE`, is used in several branches of the command tree. To keep track of the current branch, the analyzer's command parser uses the following rules:

- **Power On and Reset** - After power is cycled or after `*RST`, the current path is set to the root level commands.
- **Message Terminators** - A message terminator, such as a <NL> character, sets the current path to the root command level. Many programming language output statements send message terminators automatically. Message terminators are described in Sending Messages to the Analyzer.
- **Colon (:)** - When a colon is between two command keywords, it moves the current path down one level in the command tree. For example, the colon in `:SOURCE:POWER` specifies that `POWER` is one level below `SOURCE`. When the colon is the first character of a command, it specifies that the following keyword is a root level command. For example, the colon in `:SOURCE` specifies that source is a root level command.

Note: You can omit the leading colon if the command is the first of a new program line. For example, the following two commands are equivalent:
`SOUR:POW:ATT:AUTO`
`:SOUR:POW:ATT:AUTO`

- **<WSP>** - Whitespace characters, such as <tab> and <space>, are generally ignored. There are two important exceptions:
  - Whitespace inside a keyword, such as `:CALC ULATE`, is not allowed.
  - Most commands end with a parameter. You must use whitespace to separate these ending parameters from commands. **Always refer to the command documentation**. In the following example, there is whitespace between `STATE` and `ON`.

```
CALCULATE1:SMOOTHING:STATE ON
```

- **Comma (,)** - If a command requires more than one parameter, you must separate adjacent parameters using a comma. For example, the `SYSTEM:TIME` command requires three values to set the analyzer clock: one for hours, one for minutes, and one for seconds. A message to set the clock to 8:45 AM would be `SYSTEM:TIME 8,45,0.` Commas do not affect the current path.
- **Semicolon(;)** - A semicolon separates two commands in the same message without changing the current path. See Multiple Commands later in this topic.
- **IEEE 488.2 Common Commands** - Common commands, such as `*RST`, are not part of any subsystem. An instrument interprets them in the same way, regardless of the current path setting.

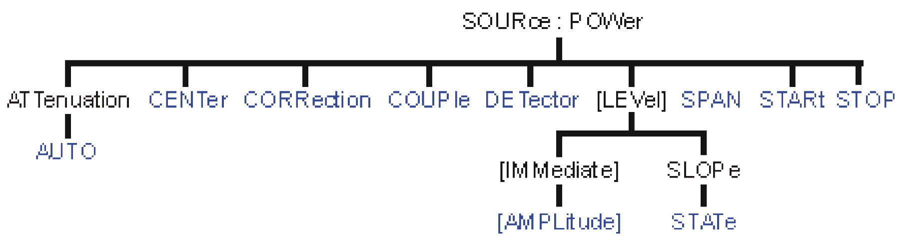
Command and Query

A SCPI command can be an Event command, Query command (a command that asks the analyzer for information), or both. The following are descriptions and examples of each form of command. GPIB Command Finder lists every SCPI command that is recognized by the analyzer, and its form.

Form	Examples
Event commands - cause an action to occur inside the analyzer.	`:INITIATE:IMMEDIATE`
Query commands - query only; there is no associated analyzer state to set.	`:SYSTem:ERRor?`
Command and query - set or query an analyzer setting. The query form appends a question mark (?) to the set form	`:FORMat:DATA ! Command` `:FORMat:DATA? ! Query`

Multiple Commands

You can send multiple commands within a single program message. By separating the commands with semicolons the current path does not change. The following examples show three methods to send two commands:



1. **Two program messages:**

```
SOURCE:POWER:START 0DBM
SOURCE:POWER:STOP 10DBM
```

2. **One long message**. A colon follows the semicolon that separates the two commands causing the command parser to reset to the root of the command tree. As a result, the next command is only valid if it includes the entire keyword path from the root of the tree:

```
SOURCE:POWER:START 0DBM;:SOURCE:POWER:STOP 10DBM
```

3. **One short message**. The command parser keeps track of the position in the command tree. Therefore, you can simplify your program messages by including only the keyword at the same level in the command tree.

```
SOURCE:POWER:START 0DBM;STOP 10DBM
```

Common Commands and SCPI Commands

You can send Common commands and SCPI commands together in the same message. (For more information on these types of commands see GP-IB Fundamentals.) As in sending multiple SCPI commands, you must separate them with a semicolon.

Example of Common command and SCPI commands together

```
*RST;SENSE:FREQUENCY:CENTER 5MHZ;SPAN 100KHZ
```

Command Abbreviation

Each command has a long form and an abbreviated short form. The syntax used in this Help system use uppercase characters to identify the short form of a particular keyword. The remainder of the keyword is lower case to complete the long form.

```
SOUR - Short form
SOURce - Long form
```

Either the complete short form or complete long form must be used for each keyword. However, the keywords used to make a complete SCPI command can be a combination of short form and long form.

The following is **unacceptable** - The first three keywords use neither short or long form.

```
SOURc:Powe:Atten:Auto on
```

The following is **acceptable** - All keywords are either short form or long form.

```
SOUR:POWer:ATT:AUTO on
```

In addition, the analyzer accepts lowercase and uppercase characters as equivalent as shown in the following equivalent commands:

```
source:POW:att:auto ON
Source:Pow:Att:Auto on
```

Optional [Bracketed] Keywords

You can omit some keywords without changing the effect of the command. These optional, or default, keywords are used in many subsystems and are identified by brackets in syntax diagrams.

Example of Optional Keywords

The `HCOPy` subsystem contains the optional keyword `IMMediate` at its first branching point. Both of the following commands are equivalent:

```
"HCOPY:IMMEDIATE"
"HCOPY"
The syntax in this Help system looks like this:
HCOPy[:IMMediate]
```

Vertical Bars | Pipes

Vertical bars, or "pipes", can be read as **"or"**. They are used in syntax diagrams to separate alternative parameter options.

Example of Vertical Bars:

```
SOURce:POWer:ATTenuation:AUTO <on|off>
```

Either `ON` or `OFF` is a valid parameter option.

MIN and MAX Parameters

The special form parameters "MINimum" and "MAXimum" can be used with **some** commands in the analyzer, as noted in the command documentation. The short form (min) and long form (minimum) of these two keywords are equivalent.

- **MAX**imum refers to the largest value that the function can currently be set to
- **MIN**imum refers to the smallest value that the function can currently be set to.

For example, the following command sets the start frequency to the smallest value that is currently possible:

```
SENS:FREQ:START MIN
```

In addition, the max and min values can also be queried for these commands.

For example, the following command returns the smallest value that Start Frequency can currently be set to:

```
SENS:FREQ:START? MIN
```

An error will be returned if a numeric parameter is sent that exceeds the MAX and MIN values.

For example, the following command will return an "Out of range" error message.

```
SENS:FREQ:START 1khz
```

Calibrating the PNA Using SCPI

There are several ways to calibrate the PNA using SCPI depending on your measurement needs. As from the Cal Wizard, you can perform a Guided Cal, Unguided Cal, or ECal. This topic explains the differences in these calibration choices when using SCPI commands.

Guided Calibrations

- Use Sens:Corr:Coll:Guided commands.
- Full 1,2,3,4-port SOLT and TRL calibrations - No response cals.
- All of the advanced calibration features (Thru method, specify DUT connectors and Cal kits for each port, port pairings).

Unguided Calibrations

- Sens:Correction commands.
- 1-port, 2-port, Response.
- Can select 2 sets of standards.
- TRL is NOT recommended.

ECal

From the Cal Wizard or from a SCPI program, ECal is fast, accurate, and very repeatable. Unlike from the Cal Wizard, you can use SCPI to perform ECal using either the Guided or Unguided commands. The Unguided commands are easiest to use. However, the following situations require that you use the Guided commands.

- To maximize accuracy, all ECal calibrations on the PNA perform an Unknown Thru measurement of the ECal module Thru state **IF** the PNA model being used has 1 reference receiver per port. If your PNA does NOT have 1 reference receiver per port, use Guided ECal commands and specify a Thru method.
- If your ECal module connectors do NOT match the DUT connectors, and you choose not to perform a User Characterization, use Guided ECal commands and specify the Thru method.

ECAL Notes:

- When using either Guided or Unguided ECal commands under low power situations, use the Orientation settings. The Guided example shows the use of these commands. When using Unguided, they must appear before the Acquire command.
- The frequency range of the measurement must be within the range of the ECal module. Otherwise, the calibration will fail.
- Although we recently provided the command, you do NOT have to send the ECal module state command. The ECal algorithm switches ECal states automatically.
- All of these ECal choices are listed in the Programming Command Search function in this Help file.

See Using ECal to learn about all of the ECal features.

See SCPI Calibration Examples

Getting Data from the Analyzer

Data is sent from the analyzer in response to program queries. Data can be short response messages, such as analyzer settings, or large blocks of measurement data. This topic discusses how to read query responses and measurement data from the analyzer in the most efficient manner.

- **Response Message Syntax**
- **Clearing the Output Queue**
- **Response Data Types**
- **Transferring Measurement Data**

Note: Some PCs use a modification of the IEEE floating point formats with the byte order reversed. To reverse the byte order for data transfer into a PC, use the FORMat:BORDer command.

Other Topics about GPIB Concepts

Response Message Syntax

Responses sent from the analyzer contain data, appropriate punctuation, and message terminators.

<NL><^END> is always sent as a response message terminator. Most programming languages handle these terminators transparent to the programmer.

Response messages use commas and semicolons as separators in the following situations:

- a comma separates response data items when a single query command returns multiple values

```
FORM:DATA? 'Query
ASC, +0 'Analyzer Response
```

- a semicolon separates response data when multiple queries are sent within the same messages

```
SENS:FREQ:STAR?;STOP? --Example Query
+1.23000000E+008; +7.89000000E+008<NL><^END> 'Analyzer Response
```

Clearing the Output Queue

After receiving a query, the analyzer places the response message in it's output queue. Your program should read the response immediately after the query is sent. This ensures that the response is not cleared before it is read. The response is cleared when one of the following conditions occur:

- When the query is not properly terminated with an ASCII carriage return character or the GPIB <^END> message.
- When a second program query is sent.
- When a program message is sent that exceeds the length of the input queue
- When a response message generates more response data than fits in the output queue.
- When the analyzer is switched ON.

Response Data Types

The analyzer sends different response data types depending on the parameter being queried. You need to know the type of data that will be returned so that you can declare the appropriate type of variable to accept the data. For more information on declaring variables see your programming language manual. The GPIB Command Finder lists every GPIB command and the return format of data in response to a query. The analyzer returns the following types of data:

- **Numeric Data**
- **Character Data**
- **String Data**
- **Block Data**

Numeric Data

All numeric data sent over the GPIB is ASCII character data. Your programming environment may convert the character data to numeric data for you. Boolean data (1 | 0) is a type of numeric data.

Character Data

Character data consists of ASCII characters grouped together in mnemonics that represent specific analyzer settings. The analyzer always returns the short form of the mnemonic in upper-case alpha characters. Character data looks like string data. Therefore, refer to the GPIB Command Finder to determine the return format for every command that can be queried.

Example of Character Data

```
MLOG
```

String Data

String data consists of ASCII characters. String parameters can contain virtually any set of ASCII characters. When sending string data to the analyzer, the string **must** begin with a single quote (') or a double quote (") and end with the same character (called the delimiter).

Note: The analyzer responds best to all special characters if the string is enclosed in single quotes. If quotes are not used, the analyzer will convert the text to uppercase. The analyzer may not respond as you expect.

The analyzer always encloses data in double quotes when it returns string data.

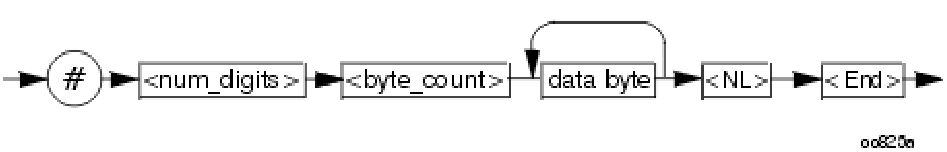
Example of String Data

GPIB.Write "DISP:WINDow:TITLe:DATA?"

```
“This is string response data.”
```

Block Data

Block data is used to transfer measurement data. Although the analyzer will accept either definite length blocks or indefinite length blocks, it always returns definite length block data in response to queries unless the specified format is ASCII. The following graphic shows the syntax for definite block data:



<num_digits> specifies how many digits are contained in <byte_count>
<byte_count> specifies how many data bytes will follow in <data bytes>

Example of Definite Block Data

#17ABC+XYZ<nl><end>

- always sent before definite block data

1 - specifies that the byte count is one digit (7)

7 - specifies the number of data bytes that will follow, not counting <NL><END>

<NL><END> - always sent at the end of block data

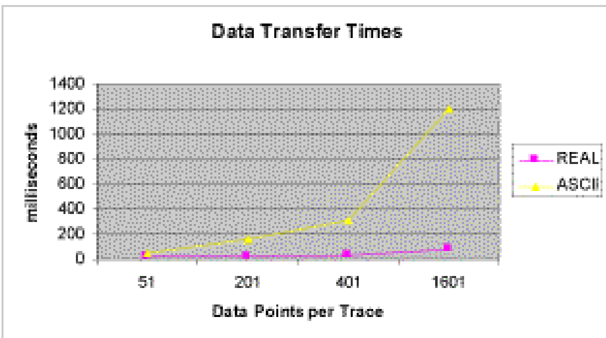
Transferring Measurement Data

Measurement data is blocks of numbers that result from an analyzer measurement. Measurement data is available from various processing arrays within the analyzer. For more information on the analyzer's data processing flow, see Accessing Data Map. Regardless of which measurement array is read, transferring measurement data is done the same.

When transferring measurement data, the FORMat:DATA command allows you to choose from the following two data types:

- REAL
- ASCII

The following graphic shows the differences in transfer times between the two:



REAL Data

REAL data (also called floating-point data) types transfer faster. This is because REAL data is binary and takes about half the space of ASCII data. The disadvantage of using REAL data is that it requires a header that must be read. See definite length block data. The binary floating-point formats are defined in the IEEE 754-1985 standard. The following choices are available in REAL format:

- **REAL,32** - IEEE 32-bit format - single precision (not supported by HP BASIC)
- **REAL,64** - IEEE 64-bit format - double precision

ASCII Data

The easiest and slowest way to transfer measurement data is to use ASCII data. ASCII data is sent if the data contains both numbers and characters (the setting of FORMat:DATA is ignored). ASCII data is separated by commas.

Understanding Command Synchronization

The analyzer takes more time to process some commands than others:

- **Sequential** commands are processed quickly and in the order in which they are received.
- **Overlapped** commands take longer to process. Therefore, they allow the program to do other tasks while waiting. However, the programmer may want to prevent the analyzer from processing new commands until the overlapped command has completed. This is called "synchronizing" the analyzer and controller.

Note: The analyzer has two overlapped commands:
INITitate:IMMediate

SENSe:SWEep:MODE GROUPS (when INIT:CONT is ON)

The analyzer's queues store commands and responses waiting to be processed. Using the analyzer's queues and controlling the processing sequence of overlapped commands is called synchronizing the analyzer and the controller. This topic discusses how and when synchronizing should be performed.

- **Analyzer Queues**
- **Synchronizing Overlapped Commands**

Other Topics about GPIB Concepts

Analyzer Queues

Queues are memory buffers that store messages until they can be processed. The analyzer has the following queues:

- **Input Queue**
- **Output Queue**
- **Error Queue**

Input Queue

The controller sends statements to the analyzer without regard to the amount of time required to execute the statements. The input queue is very large (31k bytes). It temporarily stores commands and queries from the controller until they are read by the analyzer's command parser. The input queue is cleared when the analyzer is switched ON.

Output Queue

When the analyzer parses a query, the response is placed in the output queue until the controller reads it. Your program should immediately read the response or it may be cleared from the output queue. The following conditions will clear a query response:

- When a second query is sent before reading the response to the first. This does not apply when multiple queries are sent in the same statement.

- When a program statement is sent that exceeds the length of the input queue.
- When a response statement generates more data than fits in the output queue.
- When the analyzer is switched ON.

Error Queue

Each time the analyzer detects an error, it places a message in the error queue. When the `SYSTEM:ERROR?` query is sent, one message is moved from the error queue to the output queue so it can be read by the controller. Error messages are delivered to the output queue in the order they were received. The error queue is cleared when any of the following conditions occur:

- When the analyzer is switched ON.
- When the `*CLS` command is sent to the analyzer.
- When all of the errors are read.

If the error queue overflows, the last error is replaced with a `"Queue Overflow" error.` The oldest errors remain in the queue and the most recent error is discarded.

Synchronizing Overlapped Commands

GPIB commands are executed and processed by the analyzer in the order they are received. Commands can be divided into two broad classes:

- **Overlapped commands** generally take extended time to process by the analyzer. Examples of functions that have overlapped commands are printing and making measurements. Because they take longer to process, they allow the execution of subsequent commands while the overlapped command is still in progress. However, the programmer may want to prevent the analyzer from processing new commands until the overlapped command has completed. This is called "synchronizing" the analyzer and controller.

- **Sequential commands** are generally processed quickly by the analyzer. Therefore, they prevent the processing of subsequent commands until the sequential command has been completely processed. **These commands do NOT require synchronization.**

Synchronization Methods

When To Synchronize

Synchronization Methods

The following common commands are used to synchronize the analyzer and controller. Examples are included that illustrate the use of each command in a program. See the SCPI command details to determine if a command is an overlapped command.

***WAI**

***OPC?**

***OPC**

***WAI**

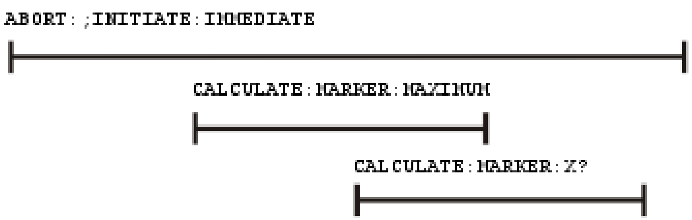
The *WAI command:

- **Stops the analyzer** from processing subsequent device commands until all overlapped commands are completed.
- **It does NOT stop the controller** from sending commands to this and other devices on the bus. This is the easiest method of synchronization.

Example of the *WAI command

```
GPIB.Write "ABORT;:INITIATE:IMMEDIATE" 'Restart the measurement.
GPIB.Write "CALCULATE:MARKER:SEARCH:MAXIMUM" 'Search for max amplitude.
GPIB.Write "CALCULATE:MARKER:X?" 'Which frequency?
```

The following timeline shows how the processing times of the three commands relate to each other:

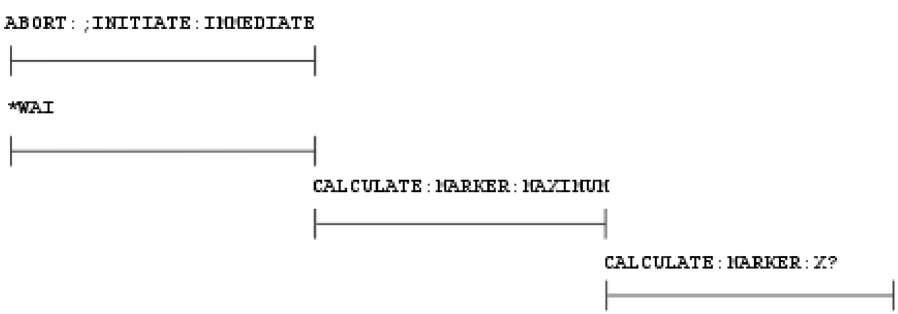


`INITIATE:IMMEDIATE` is an overlapped command; it allows the immediate processing of the sequential command, `CALCULATE:MARKER:SEARCH:MAXIMUM`. However, the `INITIATE:IMMEDIATE` is not considered complete until the measurement is complete. Therefore, the marker searches for maximum amplitude before the measurement completes. **The `CALCULATE:MARKER:X?` query could return an inaccurate value.**

To solve the problem, insert a `*WAI` command.

```
GPIB.Write "ABORT;:INITIATE:IMMEDIATE" 'Restart the measurement.
GPIB.Write "*WAI" 'Wait until complete.
GPIB.Write "CALCULATE:MARKER:MAXIMUM" 'Search for max amplitude.
GPIB.Write "CALCULATE:MARKER:X?" 'Which frequency
```

The timeline now looks like this:



The `*WAI` command keeps the `MARKER:SEARCH:MAXIMUM` from taking place until the measurement is completed. The `CALCULATE:MARKER:X?` query returns the correct value.

Note: Although *WAI stops the analyzer from processing subsequent commands, it does not stop the controller. The controller could send commands to other devices on the bus.

***OPC?**

The *OPC? query **stops the controller until all pending overlapped commands are completed.**

In the following example, the **Read** statement following the *OPC? query will not complete until the analyzer responds, which will not happen until all pending overlapped commands have finished. Therefore, the analyzer and other devices receive no subsequent commands. A "**1**" is placed in the analyzer output queue when the analyzer completes processing an overlapped command. The "**1**" in the output queue satisfies the **Read** command and the program continues.

Example of the *OPC? query **Click**

***OPC**

The *OPC command **allows the analyzer and the controller** to process commands while processing the overlapped command.

When the analyzer completes processing an overlapped command, the *OPC command sets bit 0 of the standard event register to 1 . This requires polling of status bytes or use of the service request (SRQ) capabilities of your controller. See Reading the Analyzer's Status Registers for more information about the standard event status register, generating SRQs, and handling interrupts.

Note: Be careful when sending commands to the analyzer between the time you send *OPC and the time you receive the interrupt. Some commands could jeopardize the integrity of your measurement. It also could affect how the instrument responds to the previously sent *OPC.

Example of polled bit and SRQ processes.

When To Synchronize the Analyzer and Controller

Although a command may be defined as an overlapped command, synchronization may not be required. The need to synchronize depends upon the situation in which the overlapped command is executed. The following section describes situations when synchronization is required to ensure a successful operation.

- **Completion of a Measurement**
- **Measurements with External Trigger**
- **Averaged Measurements**

Completion of a Measurement

To synchronize the analyzer and controller to the completion of a measurement, use the `ABORT;INITIATE:IMMEDIATE` command sequence to initiate the measurement.

This command sequence forces data collection to start (or restart) under the current measurement configuration. A restart sequence, such as `ABORT;INITIATE:IMMEDIATE` is an overlapped command. It is complete when all operations initiated by that restart command sequence, including the measurement, are finished. The `*WAI`, `*OPC?` and `*OPC` commands allow you to determine when a measurement is complete. This ensures that valid measurement data is available for further processing.

Measurements with External Trigger

To use an external trigger, synchronize the analyzer and controller before the trigger is supplied to the measurement. Setup the analyzer to receive a trigger from an external source (wired to the EXTERNAL TRIGGER connector on the rear panel. The trigger system is armed by GPIB with INITIATE:IMMEDIATE. (Because the source of the trigger has been specified as external, this command "readies" the analyzer for a trigger but it does not actually generate the trigger.).

Averaged Measurements

Averaged measurements are complete when the average count is reached. The average count is reached when the specified number of individual measurements is combined into one averaged measurement result. Use synchronization to determine when the average count has been reached.

If the analyzer continues to measure and average the results after the average count is reached, use synchronization to determine when each subsequent measurement is complete.

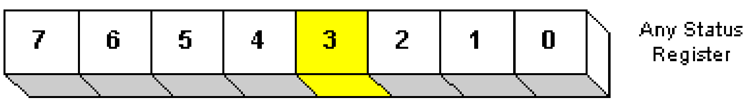
Reading the Analyzer's Status Register

The analyzer has several status registers that your program can read to know when specific events occur. There are two methods of reading the status registers in the analyzer: the Polled Bit method and the Service Request method.

- **Polled Bit Method**
- **Service Request Method**
- **Setting and Reading Bits in Status Registers**
- **Positive and Negative Transitions**
- **Status Commands**

Other Topics about GPIB Concepts

Most of the status registers in the analyzer are sixteen bits. For simplicity, this topic will illustrate their use with 8-bit registers. Bits in registers represent the status of a different conditions inside of the analyzer. In the following graphic, a register is represented by a row of boxes; each box represents a bit. Bit 3 is ON.



The Polled Bit Method

With the Polled Bit Method, your program **continually** monitors a bit in the status register that represents the condition of interest to you. When the analyzer sets the bit to 1, your program immediately sees it and responds accordingly.

Advantage: This method requires very little programming.

Disadvantage: This method renders your program unavailable to do anything other than poll the bit of interest until the condition occurs.

Procedure:

1. Decide which condition to monitor. The Status Commands topic lists all of the possible conditions that can be monitored in the analyzer.

2. Determine the command and the bit that will monitor the command.

3. Construct a loop to poll that bit until it is set.

4. Construct the routine to respond when the bit is set.

The Service Request (SRQ) Method

Your program enables the bits in the status registers representing the condition of interest. When the condition occurs, the analyzer actively interupts your program from whatever it is doing, and an event handler in your program responds accordingly. Do this method if you have several conditions you want to monitor or the conditions are such that it is not practical to wait for the condition to occur.

Advantage: This method frees your program to do other things until the condition occurs. The program is interupted to respond to the condition.

Disadvantage: This method can require extensive programming depending on the number and type of conditions that you want to monitor.

Procedure:

1. Decide which conditions to monitor. The Status Commands topic lists all of the possible analyzer conditions that can be monitored.

2. Set the **enable** bits in the **summary** registers and the **status byte** register.

> **Enabling** is like making power available to a light **-** without power available, the switch can be activated, but the light won't turn ON. In the analyzer, without enabling, the condition may occur, but the controller won't see it unless it is enabled.
>
> The condition, and the bit in the **summary** registers in the reporting path, must be enabled. Summary This is like streams (conditions) flowing into rivers (summary registers), and rivers flowing into the ocean (controller). See the diagram of status registers in Status Commands.
>
> Bit 6 of the **status byte** register is the only bit that can interrupt the controller. When **any** representative bit in the status byte register goes ON, bit 6 is automatically switched ON.

4. Enable your program to interupt the controller, This is done several ways depending on the programming language and GPIB interface card you use. An example program is provided showing how this is done with in Visual Basic with a National Instruments GPIB card.

5. Construct a subroutine to handle the interrupt event. If you are monitoring more than one condition in your system, your event handler must determine which condition caused the interupt. Use the *SPE command to determine the instrument that caused the interupt and then poll the summary registers, and then condition registers to determine the cause of the interupt.

Setting and Reading Bits in Status Registers

Both methods for reading status registers requires that you read bits out of the status registers. Most of the analyzers status registers contain 16 bits, numbered 0 to 15. Each bit has a weighted value. The following example shows how to set the bits in a 8-bit status register.

8-bit register

Bit	0	1	2	3	**4**	**5**	6	7
Weight	1	2	4	8	**16**	**32**	64	128

We want to set bits 4 and 5 in the Standard Event Status Enable register.

Step	**Example**
	16 + 32 = 48
command. (see Status Commands)	

Positive and Negative Transitions

Transition registers control what type of in a condition register will set the corresponding bit in the event register.

- **Positive** transitions (**0 to 1**) are only reported to the event register if the corresponding positive transition bit is set to 1.
- **Negative** transitions (**1 to 0**) are only reported to the event register if the corresponding negative transition bit is set to 1.
- Setting **both** transition bits to 1 causes both **positive and negative** transitions to be reported.

Transition registers are read-write and are unaffected by *CLS (clear status) or queries. They are reset to their default settings at power-up and after *RST and SYSTem:PRESet commands. The **following are the default settings** for the transition registers:

- All Positive Transition registers = 1
- All Negative Transition registers = 0

This means that by default, the analyzer will latch all event registers on the negative to positive transition (0 to 1).

The following is an example of why you would set transitions registers:

A critical measurement requires that you average 10 measurements and then restart averaging. You decide to poll the averaging bit. When averaging is complete, the bit makes a positive transition. After restart, you poll the bit to ensure that it is set back from 1 to 0, a negative transition. You set the negative transition bit for the averaging register.

Configure for SCPI LAN using SICL / VISA

- PNA Supported Interfaces
- Agilent I/O Libraries
- SICL / VISA Programs Running on the PNA
- Configure the PNA for SICL / VISA
- Configure the External Controller

Other Topics about GPIB Concepts

PNA Supported Interfaces

The PNA supports the following interfaces for SICL / VISA communication:

- **LAN** - as a remote GPIB interface. The PNA LAN is presented as a virtual GPIB interface. It does NOT support simple TCPIP-based control. Therefore, when configuring the Agilent IO libraries on your PC, add a **REMOTE GPIB** interface, which uses the LAN client interface.
- **GPIB** - requires that your external controller have a GPIB card.

Note: For optimum LAN interface performance, use COM to control the PNA. SCPI commands can be sent to the PNA using the COM SCPIStringParser object.

The following interfaces are NOT supported:

- **USB -** Although the PNA uses USB to control peripherals, it cannot be controlled over USB.
- **Serial**

Important Note:

To enable VISA or SICL communication over LAN, you must do the following:

1. On the PNA, click **System,** point to **Configure,** then click **SICL/GPIB**.
2. Check **SICL Enabled.** To automatically enable SICL when the PNA is booted, check **Automatically enable on Startup**.
3. Click **OK.**

The PNA is now ready to be controlled over LAN.

Learn more about this dialog box.

Agilent I/O Libraries

The Agilent I/O libraries includes the drivers to allow you to communicate with Agilent test instruments. Every PNA is shipped with the 'M" version of the Agilent I/O libraries installed. We recommend you do NOT upgrade the Agilent I/O libraries on the PNA as unexpected result may occur. If you choose to upgrade the Agilent I/O libraries on the PNA, do NOT change the default folder path in the InstallShield Wizard.

To communicate with the PNA, the Agilent I/O libraries must also be installed on your external controller. To purchase the Agilent I/O libraries, or download a free upgrade, go to www.agilent.com and search for IO Libraries. Scroll to find Software, Firmware & Drivers.

SICL / VISA Programs Running on the PNA

You can run your SICL / VISA program on the PNA to control the PNA. Although the Agilent I/O libraries are already installed on the PNA, it is configured as the **Host**. You must also configure a SICL or VISA LAN **Client** interface on the PNA, specifying the LAN hostname of that same PNA.

If your program uses the COM interface to VISA, and is compiled on a PC with the Agilent IO Libraries Suite (version 14 or later), and the resulting executable is copied and run on the PNA, it will produce a “type mismatch error”. This is because the PNA has the ‘M’ version of Agilent I/O libraries. The following Visual Basic code is an example of how to avoid this error when communicating with the PNA from within the PNA:

```
Dim rm As IResourceManager
Dim fmio As IFormattedIO488
Set rm = CreateObject("AgilentRM.SRMCls")
Set fmio = CreateObject("VISA.BasicFormattedIO")
Set fmio.IO = rm.Open("GPIB0::22")
fmio.WriteString "*IDN?" & Chr(10)
MsgBox fmio.ReadString()
```

Controlling the PNA over LAN while controlling other instruments over GPIB

The PNA can NOT be both a controller and talker/listener on the same GPIB bus. Using SICL / VISA, you can use LAN to control the PNA, leaving the PNA free to use the rear-panel GPIB interface to control other GPIB devices.

Configure the PNA for SICL / VISA

1. On the PNA, click **System** then check **Windows Taskbar**
2. Click **Start** then point to **Program Files, Agilent IO Libraries**, then click **IO Config**
3. Select each GPIB Interface and click **Edit** to verify (or make) the default settings in the following table. These settings are REQUIRED when using a 82357A USB / GPIB Interface with the PNA.
4. When complete, click **OK** to close the edit dialog.
5. Click **OK** to close the IO Config dialog.

VISA Interface Name	SICL Interface Name	Dialog box title	Description
GPIB0	gpib0	GPIB Using NI-488.2	PNA Rear-panel GPIB connector. This GPIB interface can be used to control the PNA **OR** for the PNA to control external equipment. IT CAN NOT DO BOTH IN THE SAME PROGRAM. Learn more about pass-through options.
GPIB1	hpib7	Internal Instrument Configuration	Internal interface for programs running on the PNA to control itself.

Configure the External Controller

Please refer to the Agilent I/O libraries documentation to learn how to configure your controller to communicate with the PNA. These links can show you how to find the following PNA information:

- PNA full computer name
- GPIB Address
- IP Address

This example program can help test your VISA configuration.

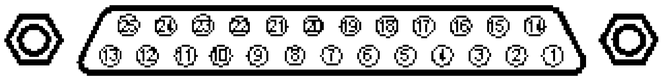
Auxiliary I/O Connector

General Description

This DB-25 male connector provides a variety of analog I/O, digital I/O, timing I/O, and supply lines. You can change the settings on the Auxiliary IO connector through SCPI and COM programming commands. The settings are NOT accessible through the front-panel keys or display menu.

Note: AUX IO settings remain Instrument Preset will clear the

Note: The AUX IO configuration settings REMAIN after an Instrument Preset and Hibernation. However, Preset will clear the DAC values. The settings will revert to their default settings ONLY after the PNA is restarted, or until they are changed by you. AUX IO settings are saved and recalled with Instrument State.



Pin	Name	Description
1	ACOM	Ground reference for analog signals
2	Analog Out 2	-10 to +10Vdc output, 10mA max
3	Analog Out 1	-10 to +10Vdc output, 10mA max
4	no connect	for future enhancements
5	DCOM	Ground reference for digital signals
6	reserved	for future enhancements
7	reserved	for future enhancements
8	reserved	for future enhancements
9	+5V	+5Vdc output, 100mA max.
10	Pass/Fail Write Strobe	Indicates pass/fail line is valid (active low)
11	Sweep End	Indicates sweep is done (programmable modes)
12	Pass/Fail	Indicates pass/fail (programmable logic, modes and scope)
13	Output Port Write Strobe	Writes I/O port data (active low)
14	Analog In	-10 to +10VDC analog input
15	ACOM	Ground reference for analog signals
16	Power Button In	Grounding replicates front panel power button press
17	DCOM	Ground reference for digital signals
18	Ready for Trigger	Indicates ready for external trigger (active low)
19	External Trigger In	Measurement trigger input (programmable to be active high or low)

20	Footswitch In	Active low input latches a user-readable status bit.
21	+22V	+22Vdc output, 100mA max.
22	In/Out port C0	General purpose input / output
23	In/Out port C1	General purpose input / output
24	In/Out port C2	General purpose input / output
25	In/Out port C3	General purpose input / output

ACOM (pins 1, 15)

Description

Analog common (ground) - To be used with the Analog Out and Analog In lines.

ACOM and DCOM are connected to system ground at a star ground point inside the analyzer.

Analog Out 1, 2 (pins 2, 3)

Description

Two analog outputs programmable to +/-10V; I_{out}<10mA; R_{out}=100 ohms

12-bit DACs with voltage resolution of approximately 5mV/count.

The DACs are set to constant values using SCPI or COM, and can be read using SCPI or COM commands.

Preset state for both pins is 0 volts.

HW Details

Looking into this output pin is a 100-ohm series resistor followed by two diodes tied to +/-15V for static protection, then the output or an op-amp.

The voltage output is provided by a 12-bit DAC with an op amp buffer.

Specifics:

- Maximum output current = 10mA
- Settling time = 3us

Timing

The DACs are set after the last data point is measured, during retrace. If the analyzer is in single sweep mode, the DACs are set as part of the presweep process, before the sweep is triggered.

DCOM (pins 5, 17)

Description

Digital common (ground).

Used with the digital input and output lines.

ACOM and DCOM are connected to system ground at a star ground point inside the analyzer.

Pins 6, 7, 8

Description

Reserved

+5V (pin 9)

Description

+5V nominal output (100mA max).

Protected by self-healing fuse:

Pass/Fail Write Strobe (pin 10)

Description

See Handler IO connector.

Sweep End (pin 11)

Description

See Handler IO connector.

Pass/Fail (pin 12)

Description

See Handler IO connector.

Output Port Write Strobe (pin 13)

Description

See Handler IO connector.

Analog In (pin 14)

Description

Analog input, +/-10V range, Rin=100k ohm

Bandwidth = 40kHz (2-pole lowpass filter).

This analog input may be read using the SCPI or COM commands.

HW Details

Looking into this pin there is 1k-ohm series resistor followed by 100k-ohm resistor to ground, static protection diodes after the 1k resistor limit the signal to +/-15V, then a high impedance buffer and active filter limiting the bandwidth to 40kHz with a lowpass filter.

Power Button In (pin 16)

Description

Short this pin to ground to replicate a front panel power button key press.

HW Details

Looking into the pin there is a 215-ohm series resistor followed by a 10k pull-up to the 3V standby supply, static protection diodes to the 0V/5V and then connects to the front panel power key circuit.

CAUTION: Because this line is internally pulled up to 3V, it should not be driven by a TTL driver.

Timing

Grounding this line for 1us to 2 seconds will simulate pressing the front panel power button.

Grounding this line for >4 seconds will perform a hard reset (similar to a personal computer) and is not recommended.

Ready for Trigger (pin 18)

Description

TTL output.

Active Low signal indicates that system is ready for an external trigger.

Remains High if system is not in External Trigger mode.

Goes High after an External Trigger is acknowledged.

Goes Low after the system has finished with its measurements, the source has been set up, and the next data point is ready to be measured.

HW Details

Looking into this pin there is a 215-ohm series resistor followed by a 10k pullup, diodes to 0V/5V for static protection, then the output of an "ABT" TTL buffer.

This line is enabled only when the analyzer is in External Trigger mode.

Refer to External Trigger In (following pin) for more information.

Timing

Refer to External Trigger In (following pin)

External Trigger In (pin 19)

Description

This input accepts level trigger signals (High / Low) on all PNA models, or edge trigger signals on some PNA models.

The external trigger configuration is set from the front panel, SCPI or COM.

For more information, see External triggering.

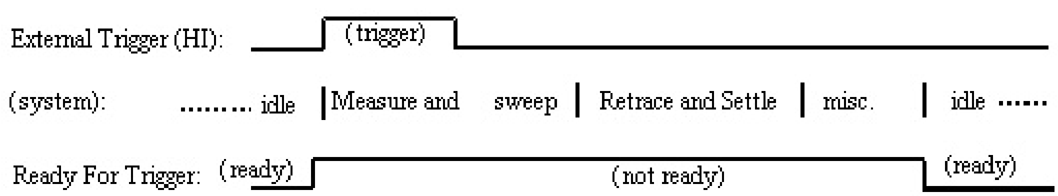
A single trigger is achieved by asserting the external trigger for a period from 1us to 50us. Continuous triggering is achieved by holding the external trigger in the "asserted" mode (either Low or High).

HW Details

Looking into this pin is a 215-ohm series resistor followed by a 4.64k pullup, 1000pF to ground and then a "FAST" TTL buffer input.

Timing

A level trigger width should be between 1us and 50us.



Footswitch In (pin 20)

Description

TTL input.

A Low level input such as shorting this line to ground using a footswitch (where the input stays low for >1us) will be latched.

The latched status may be read using the SCPI or COM commands.

Only one footswitch press can be latched (remembered) by the system.

Reading the latch status will reset it if Footswitch In has returned to a high level.

HW Details

Looking into this pin is a 215-ohm series resistor followed by a 4.64k pullup to 5V and 1000pF to ground. This line is an input to a "FAST" TTL buffer.

Timing

Footswitch In must be Low for at least 1us.

+22V (pin 21)

Description

+22V nominal output (100mA max).
Protected by self-healing fuse.

In/Out Port C0-C3 (pins 22-25)

Description

See Handler IO connector

External Test Set I/O Connector

General Description

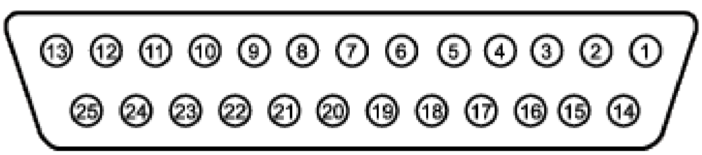
This DB-25 female connector is used to control external test sets. The external test set bus consists of 13 multiplexed address and data lines, three control lines, and an open-collector interrupt line. The Test Set IO is not compatible with the 8753 test sets.

You can change the settings on the External Test Set IO connector through SCPI and COM programming commands. The settings are NOT accessible through the front-panel keys or display menu.

Notes:

- The External Test Set pin settings are NOT affected by Instrument State Save/Recall or Instrument Preset.
- At PNA Power Up and return from Hibernation, the External Test Set bus data lines, address lines, and control lines are set HIGH, and no strobe lines are pulsed.

Caution: Do not mistake this connector with a Parallel Printer port. A printer may be damaged if connected to this port.



Pin	Name	Description
1	SEL0	Test set select bit 0; tied to GND
2	Sweep Holdoff In	TTL input - state may be read with SCPI or COM command
3	AD12	Address and latched data
4	AD10	Address and latched data
5	AD9	Address and latched data
6	AD8	Address and latched data
7	GND	0V
8	LAS	TTL output Low = Address Strobe
9	AD4	Address and latched data
10	AD3	Address and latched data
11	AD2	Address and latched data
12	GND	0V

13	Interrupt In	TTL input - state may be read with a SCPI or COM command
14	No connect	**CAUTION**: Older PNAs have +22v on this line; this will damage a printer.
15	SEL1	Test set select bit 1; tied to GND
16	SEL2	Test set select bit 2; tied to GND
17	AD11	Address and latched data
18	SEL3	Test set select bit 3; tied to GND
19	AD7	Address and latched data
20	AD6	Address and latched data
21	AD5	Address and latched data
22	AD0	Address and latched data
23	AD1	Address and latched data
24	LDS	TTL output - active low data strobe
25	RLW	TTL output - high-read, low write

SEL0-SEL3 (pins 1,15,16,18)

Description

Selects addresses of test sets that are "daisy chained" to this port. The select code is set to zero at the PNA connector and is incremented by one as it goes through each successive external test set. Therefore, the first test set in the chain has address zero and so on, for up to 16 test sets.

HW Details

Connected to ground inside the PNA.

Timing

None

Sweep Holdoff In (pin 2)

Description

Input line used by the test set for holding off a sweep. Holding off a sweep is one way of introducing a delay that allows an external device to settle before the PNA starts taking data. You must write a program that will query the line and perform the delay. The program needs to query the line and keep PNA from sweeping while the line remains low. When a subsequent query detects that the line went high the program would then trigger the PNA to start the sweep.

Use either Single or External trigger mode to control the PNA sweep.

HW Details

This pin has a series 215-ohms resistor followed by 4.7k-ohm pull-up and then an "ABT" TTL buffered register.

Timing

This input is not latched by the PNA hardware. Therefore the input level must be held at the desired state by the test set until it's read by your program.

AD0-AD12 (pins 3-6, 9-11, 17, 19-23)

Description

Thirteen lines are used to output data addresses or input / output data. Several SCPI and COM commands are available for reading and writing to these lines. You can choose to use commands where the PNA provides the appropriate timing signals needed for strobing the addresses and data. Or you can choose to control the timing signal directly. The timing signals are RLW, LAS and LDS. If you decide to do direct control refer to the corresponding SCPI and COM command details. Close attention to detail is needed to insure the desired results.

After a write command, lines AD0-AD12 are left in the state they were programmed. Default setting for Mode is Read / Input).

After a read command, lines AD0-AD12 are left in input mode. While in this mode an external test set attached to the IO is free to set the level on each line.

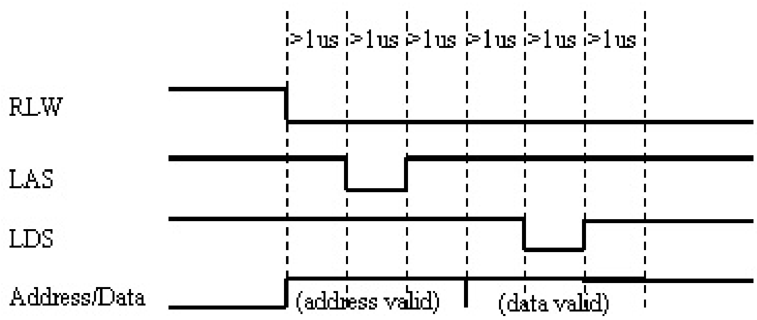
HW Details

Each of these I/O pins has a series 215-ohm resistor followed by 4.7k-ohm pull-up resistor.

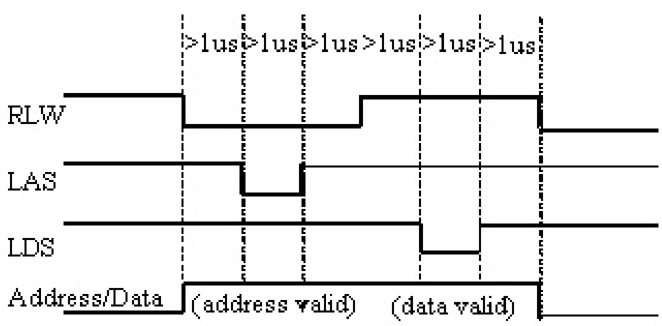
Write/Read is implemented by an output tri-state TTL buffer / latch for latching and enabling write data in parallel with a TTL input buffer for reading.

Timing

Output Address and data setup and hold times are 1us minimum.



Address & Data I/O Write



Address & Data I/O Read - Data must be valid for 1us before and after strobe

---

GND (pins 7, 12)

Description

Two ground pins used as ground references by the test set.

HW Details

Connected to digital ground.

Timing

None.

LAS (Low Address Strobe) (pin 8)

Description

This line has two behaviors that are command dependent. Refer to the SCPI and COM commands for further details.

In one behavior LAS is one of the lines used by the PNA to provide appropriate timing for writing Address and Data to the Test Set. In this case LAS is controlled automatically by the PNA and is intended to be used as the strobe for the Address. When LAS is low, lines AD0 - AD12 represent the Address. LAS will return to its normally high state when the transaction is finished.

In the second behavior the PNA will NOT provide appropriate timing. In this case LAS is controlled directly by the user through a SCPI or COM command. When the transaction is finished LAS is left set to the state it was programmed to until another command changes it. (Default for LAS is TTL High).

HW Details

This output pin is driven by a TTL latched buffer with a series 215-ohm resistor followed by 2.15k-ohm pull-up.

Timing

Strobe length, setup and hold times are all 1us minimum.

See the description for AD0-AD12 for more timing information.

Interrupt In (pin 13)

Description

Query this line with a SCPI or COM command.

HW Details

This line is a non-latched TTL input, has series 215-ohms followed by 4.64k-ohm pullup.

Timing

The Test Set must maintain at the desired TTL level until its read.

(pin 14) No Connect (previously +22V)

WARNING: Early versions of the PNA had +22v on this pin. **Connecting a printer to this port will usually damage the printer.**

Description

+22V, 100mA max. The 25-pin D connector is the same as a computer parallel printer port connector. Pin (14) corresponds to a printer's "autofeed" line. **Connecting a printer to this port will damage the printer if +22v is present** since printers requires less than 5V on all control lines.

HW Details

No connect

Timing

None

LDS (Low Data Strobe) (pin 24)

Description

This line has two behaviors that are command dependent. Refer to the External Test Set IO SCPI and COM commands for further details. (Default setting for LDS is TTL High)

In one behavior LDS is one of lines used by the PNA to provide appropriate timing for writing Address and Data to the Test Set. In this case LDS is controlled automatically by the PNA and is intended to be used as the strobe for the Data. When LDS is low, lines AD0 - AD12 represents Data. LDS will return to its normally high state when the transaction is finished.

In the second behavior the PNA will NOT provide appropriate timing. In this case LDS is controlled directly by the user through a SCPI or COM command. When the transaction is finished the LDS is left set to the state it was programmed to.

HW Details

This output pin is driven by a TTL latched buffer with a series 215-ohm resistor followed by 2.15k-ohm pull-up.

Timing

Strobe length, setup and hold times are all 1us minimum.

See the description for AD0-AD12 for more timing information.

RLW (pin 25)

Description

This line is the output for the Read Write signal. It has two behaviors that are command dependent. Refer to the External Test Set IO SCPI and COM commands for further details. (Default setting for RLW is TTL High)

In one behavior RWL is controlled automatically by the PNA during a Read Write operation. When RLW is low, lines AD0 - AD12 represent output Data. When RLW is high, the lines represent input Data.

In the second behavior the PNA does NOT provide the timing. The user must control it directly through the SCPI or COM command. In this case the line is left set to the state it was programmed to.

HW Details

This pin is a TTL latched output with a series 215-ohm resistor followed by 2.15k-ohm pull-up resistor.

Timing

Strobe length, setup and hold times are all 1us minimum.

See the description for AD0-AD12 for more timing information.

Material Handler I/O Connector

This rectangular 36-pin female connector provides communication signals between the PNA and a material parts handler. You can change the settings on the Material Handler IO connector using SCPI and COM commands. The settings are NOT accessible through the front-panel keys or display menu.

- **Overview - Controlling a Material Handler**
- **Pin Assignments**
- **Pin Descriptions**
- **Timing Diagrams**
- **Input Output Electrical Characteristics**

Note: On early PNAs this connector is labeled "GPIO". It is covered to indicate that the connector is not functional.

Overview - Controlling a Material Handler

The PNA is capable of interacting with an external material handler or part handler. This allows the PNA to be used in an automated test environment, where devices to be tested are inserted into a test fixture by a part handler, and sorted into pass/fail bins by the handler after testing is complete. By connecting the part handler to the PNAs Auxiliary or Material Handler I/O ports, the PNA and part handler can synchronize their activities in a way that makes automated testing possible.

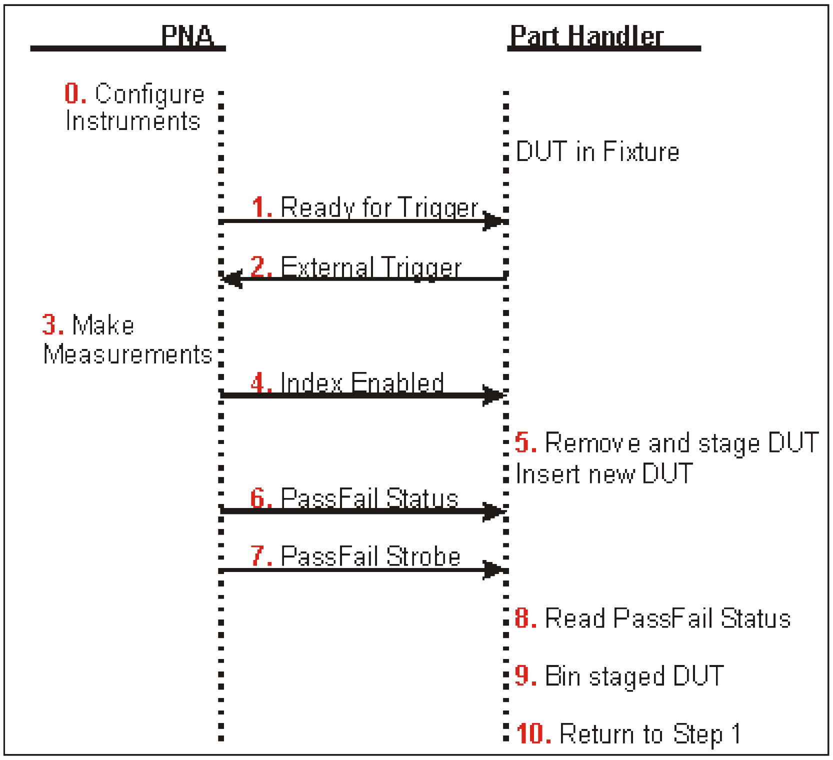
PNA and Part Handler Preparation

1. Define the measurements you want to make.
2. Define limits for each of the measurements.
3. Configure the PNAs Material Handler port so that it is compatible with your part handler. This usually involves setting the handler logic, pass/fail logic, pass/fail scope, and pass/fail mode. These settings are made remotely using SCPI or COM commands.
4. Use a cable to connect the PNA to your part handler.
5. Put the PNA in External Trigger mode.
6. Load parts in handler per manufacturer instructions.

Note: The Material Handler configuration settings REMAIN after an Instrument Preset and Hibernation. The settings will revert to their default settings ONLY after the PNA is restarted, or until they are changed by you. Material Handler settings are saved and recalled with Instrument State.

Flow Diagram

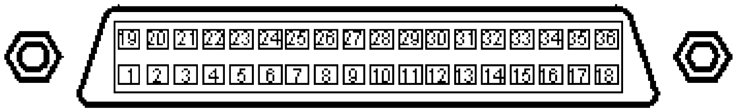
The following diagram and descriptions summarizes the events that occur during automated testing. 'DUT' refers to Device Under Test.



Text Descriptions

0. (Optional). The PNA sends values out the Material Handler and/or Auxiliary I/O connectors to configure external instruments. The A,B,C, and D ports of the Material Handler can be used to control devices used in testing, such as step attenuators, part handlers, or even the DUT itself. Also, the DAC1 and DAC2 lines on the Aux I/O connector can be used to provide bias voltages for devices and instruments. If you wish to use the Material Handler or Aux connectors for testing, you will need to write a program to send values out the various lines and ports, as there is no activity on these lines by default.
1. The part handler receives a Ready for Trigger signal from the PNA. This indicates that the PNA is properly configured and ready to take a measurement.
2. The part handler sends an External Trigger signal to the PNA. This signals that the part handler has settled, and allows the PNA to begin taking measurements.
3. The PNA takes measurements on all triggerable channels.
4. The Index line on the material handler goes to a Low state, which means that all required data has been collected by the PNA.
5. The part handler removes the DUT from the test fixture, and inserts a new DUT into the fixture. This operation is often referred to as part handler indexing. The device just tested is staged (removed from the fixture and prepared for binning), and the next part to be tested is put into the fixture. The removed DUT cannot be assigned to a Pass/Fail bin yet, as the Pass/Fail status is not available.
6. The PNA sends the Pass/Fail Status.
7. The PNA sends the Pass/Fail Strobe meaning that the Pass/Fail status has been determined.
8. The part handler reads the Pass/Fail Status line.
9. The part handler bins the staged part based on the Pass/Fail Status.
10. The test process repeats at step 1, waiting for Ready for Trigger from the PNA.

Material Handler IO Pin Assignments



There are three different Handler IO pin assignment configurations depending on the PNA model:

- **Type 1** - All 3 GHz, 6 GHz, and 9 GHz PNA models. You can change the pinout configuration to Type 2 on these models. This requires opening the instrument and changing a connector internally. Refer to the procedure in the Service Guide, Chapter 7. You can download a copy of the Service Guide from our Web site at http://www.agilent.com/find/pna. **Caution:** Changing this connection should be done by qualified service personnel.

- **Type 2** - All PNA models EXCEPT 3 GHz, 6 GHz, and 9 GHz and N5230 Opt 240, 245.

- **Type 3** - N5230 Options: 020,025,120,125, 240, 245

See PNA models and options.

Shaded/bold indicates changes from Type 1

Note: A slash (/) preceding the signal names indicates that the signal uses negative (active low) logic. A low pulse is a logical 1.

Pin	Type 1	Type 2	Type 3
1	Ground	Ground	Ground
2	/ INPUT1	/ INPUT1	/ INPUT1
3	/ OUTPUT1	/ OUTPUT1	/ OUTPUT1
4	/ OUTPUT2	/ OUTPUT2	/ OUTPUT2
5	/ Output port A0	/ Output port A0	/ Output port A0
6	/ Output port A1	/ Output port A1	/ Output port A1
7	/ Output port A2	/ Output port A2	/ Output port A2
8	/ Output port A3	/ Output port A3	/ Output port A3
9	/ Output port A4	/ Output port A4	/ Output port A4
10	/ Output port A5	/ Output port A5	/ Output port A5
11	/ Output port A6	/ Output port A6	/ Output port A6
12	/ Output port A7	/ Output port A7	/ Output port A7
13	/ Output port B0	/ Output port B0	/ Output port B0
14	/ Output port B1	/ Output port B1	/ Output port B1
15	/ Output port B2	/ Output port B2	/ Output port B2

16	/ Output port B3	/ Output port B3	/ Output port B3
17	/ Output port B4	/ Output port B4	/ Output port B4
18	no connect	**/ Output port B5**	**/ Ext. Trigger**
19	/ Output port B5	**/ Output port B6**	/ Output port B5
20	/ Output port B6	**/ Output port B7**	**/ Output port B6 - / Index Signal**
21	/ Output port B7	**/ In/Out port C0**	**/ Output port B7 / Ready for Trigger**
22	/ In/Out port C0	**/ In/Out port C1**	/ In/Out port C0
23	/ In/Out port C1	**/ In/Out port C2**	/ In/Out port C1
24	/ In/Out port C2	**/ In/Out port C3**	/ In/Out port C2
25	/ In/Out port C3	**/ In/Out port D0**	/ In/Out port C3
26	/ In/Out port D0	**/ In/Out port D1**	/ In/Out port D0
27	/ In/Out port D1	**/ In/Out port D2**	/ In/Out port D1
28	/ In/Out port D2	**/ In/Out port D3**	/ In/Out port D2
29	/ In/Out port D3	**Port C Status**	/ In/Out port D3
30	Port C Status	**Port D Status**	Port C Status
31	Port D Status	**/ Output Port Write Strobe**	Port D Status
32	/ Output Port Write Strobe	**no connect**	/ Output Port Write Strobe
33	/ Pass/Fail	/ Pass/Fail	/ Pass/Fail
34	/ Sweep End	**+5V**	/ Sweep End
35	+5V	**/ Sweep End**	+5V
36	/ Pass/Fail Write Strobe	/ Pass/Fail Write Strobe	/ Pass/Fail Write Strobe

Pin Descriptions

Input1

When this Input line receives a Low pulse from the material handler, data is latched on the OUTPUT1 and OUTPUT2 lines. See OUTPUT1|2 Data Output Write Timing

Note: Type 1 and Type 2 Behavior: The Input line responds to a High (rising edge) pulse.

The Input Line activity can be read:

SCPI	COM
CONTrol:HANDler:INPut?	get_Input1 Method

Output1, Output2

See OUTPUT1|2 Data Output Write Timing

The **current** state of these latched TTL outputs may be set High or Low (Default setting) using the (non-user) SCPI put_Output (COM) commands.

The **next** state (following a negative edge on the INPUT1 line) may be pre-loaded to High or Low (Default setting) using the user commands.

For example, on the next negative pulse on the INPUT1 line, you want the OUTPUT1 line to go from 0 to 1. To do this:

```
CONT:HAND:OUTP1:DATA 0   'Force the OUTPUT1 line to 0
CONT:HAND:OUTP1:USER 1   'Set the OUTPUT1:USER buffer to 1, indicating the next state
```

	SCPI	COM
Write User Data	CONT:HAND:OUTP<pin>:USER	**put** Output Method
Read last value written	CONT:HAND:OUTP<pin>:USER	get_Output Method.
Write non-user data	CONT:HAND:OUTP<pin>:DATA	**put** Output Method
Read last value written	CONT:HAND:OUTP<pin>:DATA	get_Output Method

Output Ports A and B

These two general purpose, 8-bit output ports are used to write data to the material handler. When any line changes state, all output lines are latched to the I/O connector as the Output Write Strobe goes Low.

The default state for data is Low.

See Data Output Write Timing Diagram

Set Port Logic:

The logic for the data lines can be set to either: Positive (1 = High) or Negative (1 = Low). This setting affects all data ports. They cannot be set independently.

SCPI	COM
CONTrol:HANDler:LOGic	PortLogic Property

Combine to read or write data to Port F:

Ports A and B can be virtually combined to write data to one 16-bit I/O port **F.**

SCPI	COM
CONTrol:HANDler:F <num>	put Port (F)

Input/Output Ports C and D

These two general purpose 4-bit Input/Output ports are used to write data (Output) or read data (Input). These lines could be used to write to an external device such as a step attenuator.

When any line changes state, all output lines are latched to the I/O connector as the Output Write Strobe goes Low. See Data Output Write Timing

The four lines of Port C are connected internally to the Auxiliary IO connector.

Set Input | Output Mode:

Each port may be independently defined as Output or Input.

SCPI	COM
CONTrol:HANDler:C:MODE	PortMode Property
CONTrol:HANDler:D:MODE	

Set Port Logic:

The logic for the data lines can be set to either: Positive (1 = High) or Negative (1 = Low). This setting affects all data ports. They cannot be set independently.

SCPI	COM

CONTrol:HANDler:LOGic	PortLogic Property

Read or write data:

Ports C and D can be virtually combined to read or write data to one 8-bit I/O port **E.** When combined, **both** C and D ports must be set to either INPUT or OUTPUT mode.

SCPI	COM
CONTrol:HANDler:<port>[:DATa]>	get_Port(x) put Port (x)

Port C Status, Port D Status

These two output lines indicate the Read / Write mode of the C and D ports.

- A Low level indicates that the associated port is in **INPUT** mode (read only).
- A High level indicates that the associated port is in **OUTPUT** mode (write only).

These logic of these status outputs cannot be changed.

See Input/Output Ports C and D to learn how to set I/O Mode

See Data Output Write Timing

Output Port Write Strobe

This Output line goes Low to write data from Ports A and B and Ports C and D when a change is detected on any of the data lines.

These logic of this strobe output cannot be changed.

This line is shared with Auxiliary IO connector.

See Data Output Write Timing

External Trigger

When trigger source is set to external, this Input line accepts a trigger signal from the material handler. This usually means that a part is in place and ready to be tested.

See Trigger Timing Diagram

Index

A Low signal on this Output line indicates to the material handler that the measurement is complete. This usually means that the handler can connect the next device. However, measurement data is not available until data is calculated. See Trigger Timing Diagram.

Set Function:

This line also serves as a data line. Set the function using the following commands:

SCPI	COM
CONTrol:HANDler:INDex:STATe	IndexState

Ready for Trigger

When this output line goes low, it indicates to the material handler that the PNA is ready for a trigger signal.

See Trigger Timing Diagram

See Pass/Fail Timing Diagram

Set Function:

This line also serves as a data line. Set the function using the following commands:

SCPI	COM
CONTrol:HANDler:RTRigger:STATe	ReadyForTriggerState

Pass/Fail State

This Output line indicates to the handler whether the limit test has passed or failed.

Pass/Fail state is valid only when the limit test function is ON and while Pass/Fail strobe line is Low. See Pass/Fail Timing Diagram

This line is shared with the Auxiliary IO connector.

Set Pass / Fail Logic:

- Positive Logic: High=Pass, Low=Fail. (Default setting)
- Negative Logic: High=Fail, Low=Pass.

SCPI	COM
CONTrol:HANDler:PASSfail:LOGic	PassFailLogic Property

Set Default Conditions:

- **PASS**- the line stays in PASS state. When a device fails, then the line goes to fail after the Sweep End line is asserted.
- **FAIL**- the line stays in FAIL state. When a device passes, then the line goes to PASS state after the Sweep End line is asserted.
- **No Wait-** the line stays in PASS state. When a device fails, then the line goes to fail IMMEDIATELY. (Default setting)

SCPI	COM
CONTrol:HANDler:PASSfail:MODE	PassFailMode Property

Set Pass / Fail Scope:

- **Channel scope:** The line resets to the default state after the measurements on a channel have completed.
- **Global scope:** The line resets to the default state after the measurements on all triggerable channels have completed. (Default setting)

SCPI	COM
CONTrol:HANDler:PASSfail:SCOPe	PassFailScope Property

Pass/Fail Write Strobe

A Low pulse indicates that Pass/Fail line is valid and the Pass / Fail State is output to the material handler.

This line is shared with the Auxiliary IO connector.

The Pass/Fail Strobe is fixed in duration and timing. However, when the strobe occurs depends on the Pass/Fail Mode and Pass/Fail Scope (Channel or Global) settings. See Pass/Fail State

See Pass/Fail Timing Diagram

+5V

+5V nominal output (100mA max).

Protected by self-healing fuse.

Sweep End

This output line indicates the status of the PNA sweep. The sweep includes sweeping the source and taking data.

- **Low** (falling edge) indicates that the specified sweep event has finished. This does NOT indicate that all calculations have finished.
- **High** indicates that the specified sweep event is active.

See Trigger Timing Diagram

This line is shared with the Auxiliary IO connector.

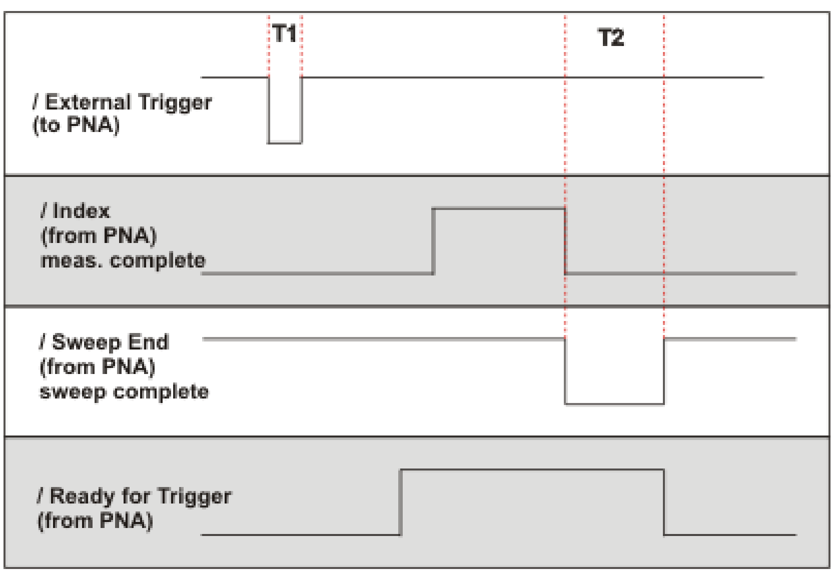
Set Sweep Event Mode:

- **Sweep**: indicates that a single source sweep has finished. (Default setting)
- **Channel**: indicates that a single channel has finished.
- **Global**: indicates that all enabled channels have finished.

SCPI	COM
CONTrol:HANDler:SWEepend	SweepEndMode Property

Timing Diagrams

Trigger Timing

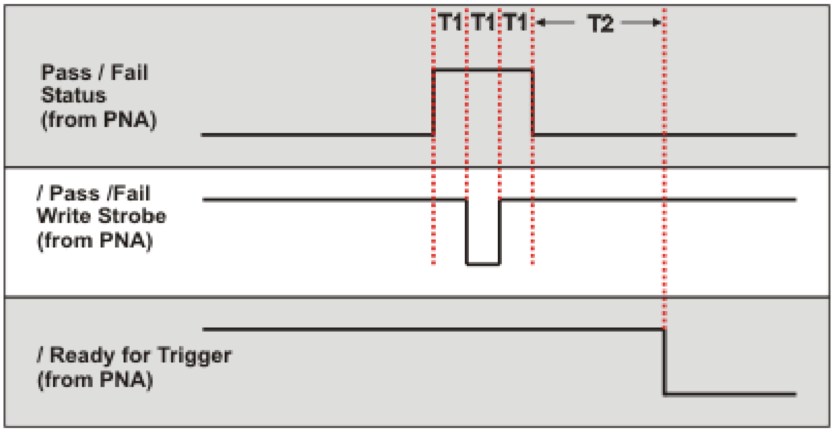


All signals are active low.

T1 = 1 ms External Trigger pulse width

T2 > 10ms Sweep End pulse width (both High and Low)

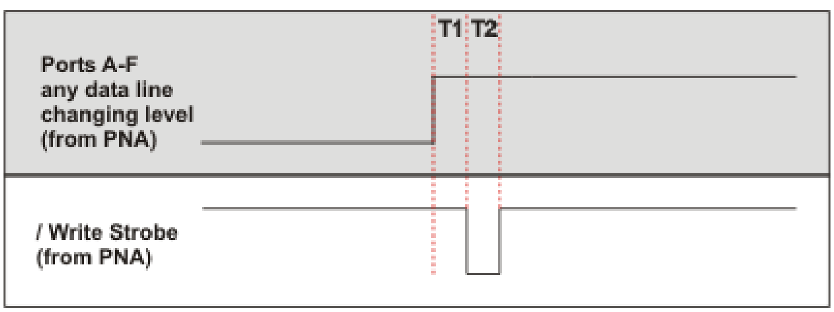
Pass / Fail Timing



T1 = 1 ms Pulse width and response time of Pass / Fail Strobe

T2 > 10 ms Ready for Trigger lag

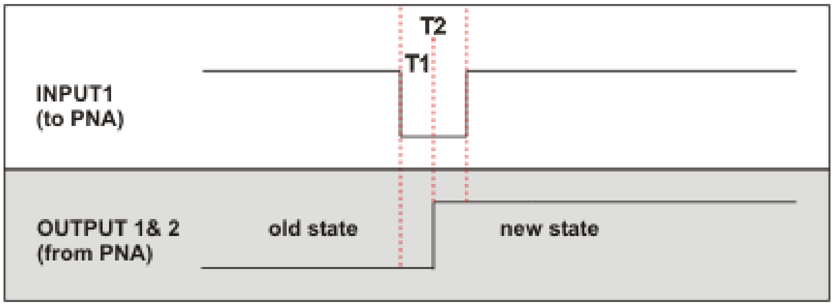
Ports A-F Data Output Write Timing



T1 = 1 ms Write Strobe response time

T2 = 1 ms Write Strobe pulse width

OUTPUT1|2 Data Output Write Timing



The old state to new state transition can be either low to high (as shown) or high to low.

T1 = .6 ms Output1|2 response time

T2 = 1 ms Input1 Strobe pulse width

Input / Output Electrical Characteristics

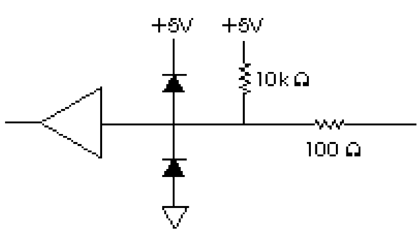
All Material Handler I/O Input and Output lines are TTL compatible.

Input and Input/Output lines

Lines carrying information IN (or bidirectional) to the PNA from the material handler.

Maximum Input Voltages:	-0.5 V to 5.5 V
TTL High level:	2.0 V to 5.0 V
TTL Low level:	0 V to 0.5 V

PNA **Input** and **Input/Output** Circuit Diagram



To Handler I/O Port

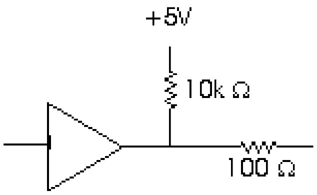
Note: The INPUT1 line does NOT have the 10K pullup resistor.

Output Lines

Lines carrying information OUT of the PNA to the material handler.

Maximum Output Current:		-10 mA to 10 mA
Output Current	TTL High level:	-5 mA
	TTL Low level:	3 mA
Output Voltage	TTL High level:	2.0 V to 3.3 V
	TTL Low level:	0 V to 0.8 V

PNA **Output** Circuit Diagram



To Handler I/O Port

COM versus SCPI

There are two methods you can use to remotely control the PNA: COM and SCPI. The following topics can help you choose the method that best meets your needs:

- **Software Connection**
- **Physical Connection**
- **Selecting a Method**
- **Programming Languages**

Other Topics about COM Concepts

Software Connection

COM uses a binary protocol, allowing you to directly invoke a PNA feature. This is more efficient than SCPI. For example, the following statement calls directly into the PNA, executing the routine GetIDString.

```
PNA.GetIDString()
```

SCPI is a text based instrument language. To retrieve the ID string, you would send the following text string to the PNA:

```
IbWrite( "*IDN?")
```

The PNA SCPI parser would first decode this text string to determine that the user has asked for the PNA to identify itself. Then the parser would call GetIDString().

The Physical Connection

Internal Control

With either COM or SCPI, the best throughput is attained by using the PNA's internal PC to execute your test code. However, if your test code uses too much system resources (CPU cycles and/or memory), this will slow the PNA's performance.

Using the SICL I/O Libraries, you can also connect to the PNA from a program running on the PNA.

External Control

You can control the PNA from a remote PC using either COM or SCPI.

COM - (Component Object Model) can be used to access any program like the PNA (835x.exe) or library (.dll) that exposes its features using a COM compliant object model. These programs or libraries are called "servers". Programs (like your remote program on your PC) that connect to and use the features of these servers are called "clients."

With COM, the server and the client do not need to reside on the same machine. DCOM, or distributed COM, is easy to configure and makes the location of the server transparent to the client. When you access the PNA from a remote computer, you are using DCOM. In this case, the mechanical transport is a LAN (local area network).

SCPI - Using a GPIB interface card in a remote computer, you can connect to the instrument using a GPIB cable. There are some constraints on the length of this cable and the number of instruments that can be daisy-chained together.

Using the Agilent SICL I/O libraries, you can connect to the instrument over a LAN connection.

(LAN or INTERNAL) You can send SCPI commands using COM with the ScpiStringParser object.

Selecting a Method

You should almost always choose COM for the following reasons:

- COM executes faster most of the time.
- COM is generally easier to use. The latest development tools embrace COM and know how to make your life easier with integrated development environments that show automation syntax as you type.
- As time goes on, more emphasis will be put on COM as the preferred programming paradigm.

But choosing a connection method depends on your situation. Here are some additional things to consider:

1. If you want to use the PNA to control other GPIB instruments, you may want to use COM as the means of talking to the PNA. In GPIB, the PNA can not be configured as both **System Controller** and **talker/listener**. Because the PNA does not support pass control mode, only one mode can be used at a time.
2. If you have legacy code written in SCPI for another network analyzer, you may be able to leverage that code to control the PNA. However, the PNA uses a different platform than previous Agilent Network Analyzers. Therefore, not all commands have a direct replacement. See the PNA Code Translator Application.

Programming Languages

You can program the PNA with either COM or SCPI using several languages. The most common include:

Agilent VEE - With this language you can send text based SCPI commands and also use automation. VEE 6.0 or later is recommended.

Visual Basic - This language has great support for automation objects and can be used to drive SCPI commands. The use of VISA drivers for your GPIB hardware interface will make the task of sending SCPI commands easier.

C++ - This language can do it all. It is not as easy to use as the above two, but more flexible.

Code Translator Application

The Code Translator (CXL) Application allows you to control the PNA using your 8753, 872x, or 8510 (Legacy) GPIB test programs. The CXL Application runs on the PNA and processes each legacy command, real time, by first recognizing it as a valid legacy commands, and then executing the equivalent PNA command if one exists.

The CXL Application will also translate your Legacy programs into PNA SCPI and COM commands.

The CXL Application works with HP-VEE, Visual Basic, and RMB programs. The CXL Application must be used with PNA revision firmware A.04.00 or later.

Installing the CXL Application

Download the CXL application from http://na.tm.agilent.com/pna/programming/. To install the CXL application, double click on the InstallShield package icon.

The CXL application has its own help file that includes a command cross-reference for 8510/8753/872x models to PNA commands.

Network Analyzer Basics

This multimedia presentation is intended to run on the PNA.

Click to proceed with Network Analyzer Basics.

To view this presentation on an external PC, copy the all of the files from the PNA C:\WINNT\Help\NABasics to the same location on your PC's hard drive. The total file size is approximately 100MB.

Connector Care

Proper connector care is critical for accurate and repeatable measurements. The following information will help you preserve the precision and extend the life of your connectors - saving both time and money.

- Connector Care Quick Reference Guide
- Connector Cleaning Supplies
- Safety Reminders
- About Connectors
- Gaging Fundamentals
- Connector Care Procedures

Connector Care Quick Reference Guide

Handling and Storing Connectors	
Do	**Do Not**
Keep connectors clean Protect connectors with plastic end caps Keep connector temperature same as analyzer	Touch mating-plane surfaces Set connectors contact-end down Store connectors loose in box or drawer
Visual Inspection	
Do	**Do Not**
Inspect connectors with magnifying glass. Look for metal debris, deep scratches or dents	Use a connector with a bent or broken center conductor Use a connector with deformed threads
Cleaning Connectors	
Do	**Do Not**
Clean surfaces first with clean, dry compressed air Use lint-free swab or brush Use minimum amount of alcohol Clean outer conductor mating surface and threads	Use high pressure air (>60 psi) Use any abrasives Allow alcohol into connector support beads Apply lateral force to center conductor
Gaging Connectors	
Do	**Do Not**

Inspect and clean gage, gage master and device tested Use correct torque wrench zero gage before use Use multiple measurements and keep record of readings	Use an out of specification connector Hold connector gage by the dial
Making Connections	
Do	**Do Not**
Align connectors first Rotate only the connector nut Use correct torque wrench	Cross thread the connection Twist connector body to make connection Mate different connector types

Connector Care and Cleaning Supplies

Listed below are products commonly used for connector cleaning. To order these and other connector care products, see Analyzer Accessories .

Description	Agilent P/N
Lint-Free Swabs, small 100 ct.	9301-1243
IPA 99.5% alcohol, 30 ml bottle	8500-5344
Compressed Air, 235 ml can	8500-6659

Safety Reminders

When cleaning connectors:

- Always use protective eyewear when using compressed air or nitrogen.
- Keep isopropyl alcohol away from heat, sparks and flame. Use with adequate ventilation. Avoid contact with eyes, skin and clothing.
- Avoid electrostatic discharge (ESD). Wear a grounded wrist strap (having a 1 MΩ series resistor) when cleaning device, cable or test port connectors.

About Connectors

- **Connector Service Life**
- **Connector Grades and Performance**
- **Adapters as Connector Savers**
- **Connector Mating Plane Surfaces**

Connector Service Life

Even though calibration standards, cables, and test set connectors are designed and manufactured to the highest standards, all connectors have a limited service life. This means that connectors can become defective due to wear during normal use. For best results, all connectors should be inspected and maintained to maximize their service life.

Visual Inspection should be performed each time a connection is made. Metal particles from connector threads often find their way onto the mating surface when a connection is made or disconnected. See Inspection procedure.

Cleaning the dirt and contamination from the connector mating plane surfaces and threads can extend the service life of the connector and improve the quality of your calibration and measurements. See Cleaning procedure.

Gaging connectors not only provides assurance of proper mechanical tolerances, and thus connector performance, but also indicate situations where the potential for damage to another connector may exist. See Gaging procedure.

Proper connector care and connection techniques yield:

- Longer Service Life
- Higher Performance
- Better Repeatability

Connector Grades and Performance

The three connector grades (levels of quality) for the popular connector families are listed below. Some specialized types may not have all three grades.

- **Production** grade connectors are the lowest grade and the least expensive. It is the connector grade most commonly used on the typical device under test (DUT). It has the lowest performance of all connectors due to its loose tolerances. This means that production grade connectors should always be carefully inspected before making a connection to the analyzer. Some production grade connectors are not intended to mate with metrology grade connectors.

- **Instrument** grade is the middle grade of connectors. It is mainly used in and with test instruments, most cables and adapters, and some calibration standards. It provides long life with good performance and tighter tolerances. It may have a dielectric supported interface and therefore may not exhibit the excellent match of a metrology grade connector.

- **Metrology** grade connectors have the highest performance and the highest cost of all connector grades. This grade is used on calibration standards, verification standards, and precision adapters. Because it is a high precision connector, it can withstand many connections and disconnections and, thus, has the longest life of all connector grades. This connector grade has the closest material and geometric specifications. Pin diameter and pin depth are very closely specified. Metrology grade uses an air dielectric interface and a slotless female contact which provide the highest performance and traceability.

Note: In general, Metrology grade connectors should not be mated with Production grade connectors.

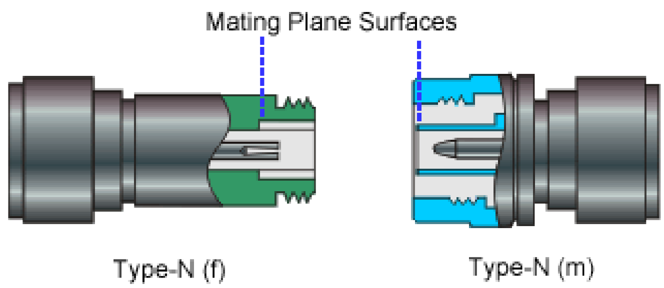
Adapters as Connector Savers

Make sure to use a high quality (Instrument grade or better) adapter when adapting a different connector type to the analyzer test ports. It is a good idea to use an adapter even when the device under test is the same connector type as the analyzer test ports. In both cases, it will help extend service life, and protect the test ports from damage and costly repair.

The adapter must be fully inspected before connecting it to the analyzer test port and inspected and cleaned frequently thereafter. Because calibration standards are connected to the adapter, the adapter should be the highest quality to provide acceptable RF performance and minimize the effects of mismatch.

Connector Mating Plane Surfaces

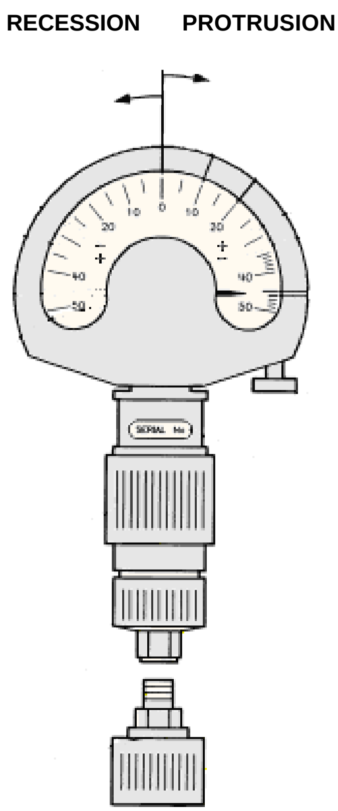
An important concept in RF and microwave measurements is the reference plane. For a network analyzer, this is the surface that all measurements are referenced to. At calibration, the reference plane is defined as the plane where the mating plane surfaces of the measurement port and the calibration standards meet. Good connections (and calibrations) depend on perfectly flat contact between connectors at all points on the mating plane surfaces (as shown in the following graphic).



Gaging Fundamentals

Connector gages are important tools used to measure center conductor pin depth in connectors. Connector pin depth, measured in terms of recession or protrusion, is generally the distance between the mating plane and the end of the center conductor, or the shoulder of the center conductor for a stepped male pin.

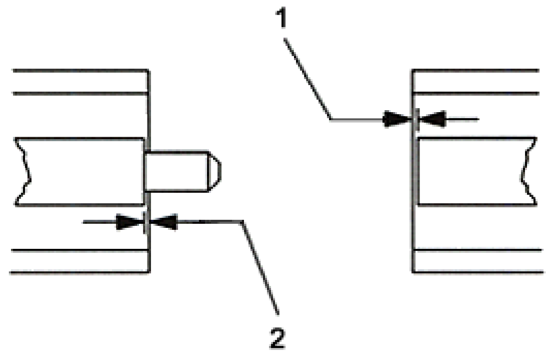
Typical Connector Gage



Recession and Protrusion

Pin depth is negative (recession) if the center conductor is recessed below the outer conductor mating plane, usually referred to as the "reference plane". Pin depth is positive (protrusion) if the center conductor projects forward from the connector reference plane.

Pin Depth



1. Recession of female contact
2. Recession of male pin shoulder

Difference with Type-N Connectors

Type-N connectors have the mating plane of the center conductors offset from the connector reference plane. In this case the zero setting "gage masters" generally offset the nominal distance between the center conductor mating plane and the connector reference plane.

When to Gage Connectors

- Before using a connector or adapter the first time.
- When visual inspection or electrical performance suggests the connector interface may be out of range.
- After every 100 connections, depending on use.

Connector Gage Accuracy

Connector gages (those included with calibration and verification kits), are capable of performing coarse measurements only. This is due to the repeatability uncertainties associated with the measurement. It is important to recognize that test port connectors and calibration standards have mechanical specifications that are extremely precise. Only special gaging processes and electrical testing (performed in a calibration lab) can accurately verify the mechanical characteristics of these devices. The pin depth specifications in the Agilent calibration kit manuals provide a compromise between the pin depth accuracy required, and the accuracy of the gages. The gages shipped with calibration and verification kits allow you to measure connector pin depth and avoid damage from out-of-specification connectors.

Note: Before gaging any connector, the mechanical specifications provided with that connector or device should be checked.

To Gage Connectors

1. Wear a grounded wrist strap (having a 1 MΩ series resistor).
2. Select proper gage for device under test (DUT).
3. Inspect and clean gage, gage master, and DUT.
4. Zero the connector gage.
   a. While holding gage by the barrel, carefully connect gage master to gage. Finger-tighten connector nut only.
   b. Use proper torque wrench to make final connection. If needed, use additional wrench to prevent gage

master (body) from turning. Gently tap the barrel to settle the gage.

c. The gage pointer should line up exactly with the zero mark on gage. If not, adjust "zero set" knob until gage pointer reads zero. On gages having a dial lock screw and a movable dial, loosen the dial lock screw and move the dial until the gage pointer reads zero. Gages should be zeroed before each set of measurements to make sure zero setting has not changed.

d. Remove gage master.

5. Gage the device under test.

   a. While holding gage by the barrel, carefully connect DUT to gage. Finger-tighten connector nut only.

   b. Use proper torque wrench to make final connection and, if needed, use additional wrench to prevent DUT (body) from turning. Gently tap the barrel to settle the gage.

   c. Read gage indicator dial for recession or protrusion and compare reading with device specifications.

Caution: If the gage indicates excessive protrusion or recession, the connector should be marked for disposal or sent out for repair.

---

6. For maximum accuracy, measure the device a minimum of three times and take an average of the readings. After each measurement, rotate the gage a quarter-turn to reduce measurement variations.

7. If there is doubt about measurement accuracy, be sure the temperatures of the parts have stabilized. Then perform the cleaning, zeroing, and measuring procedure again.

Connector Care Procedures

Inspecting Connectors

Cleaning Connectors

Making Connections

Using a Torque Wrench

Handling and Storing Connectors

To Inspect Connectors

Wear a grounded wrist strap (having a 1 MΩ series resistor).

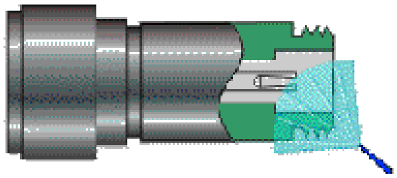
Use a magnifying glass (≥10X) and inspect connector for the following:

- Badly worn plating or deep scratches
- Deformed threads
- Metal particles on threads and mating plane surfaces
- Bent, broken, or mis-aligned center conductors
- Poor connector nut rotation

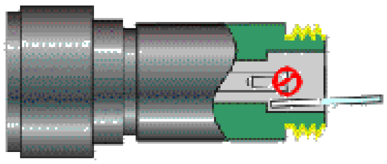
Caution: A damaged or out-of-specification device can destroy a good connector attached to it even on the first connection. Any connector with an obvious defect should be marked for disposal or sent out for repair.

To Clean Connectors

1. Wear a grounded wrist strap (having a 1 MΩ series resistor).
2. Use clean, low-pressure air to remove loose particles from mating plane surfaces and threads. Inspect connector thoroughly. If additional cleaning is required, continue with the following steps.



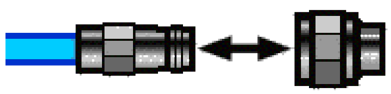
3. Moisten–do not saturate–a lint-free swab with isopropyl alcohol. See Cleaning Supplies for recommended type.
4. Clean contamination and debris from mating plane surfaces and threads. When cleaning interior surfaces, avoid exerting pressure on center conductor and keep swab fibers from getting trapped in the female center conductor.



5. Let alcohol evaporate–then use compressed air to blow surfaces clean.
6. Inspect connector. Make sure no particles or residue remains.
7. If defects are still visible after cleaning, the connector itself may be damaged and should not be used. Determine the cause of damage before making further connections.

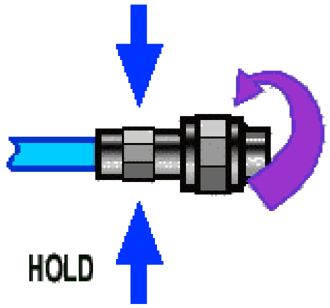
To Make Connections

1. Wear a grounded wrist strap (having a 1 MΩ series resistor).
2. Inspect, clean, and gage connectors. All connectors must be undamaged, clean, and within mechanical specification.
3. Carefully align center axis of both devices. The center conductor pin–from the male connector–must slip concentrically into the contact finger of the female connector.

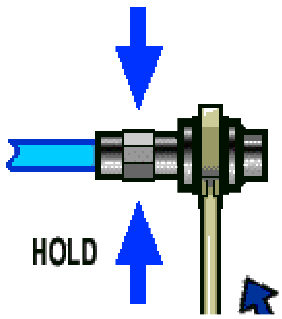


4. Carefully push the connectors straight together so they can engage smoothly. Rotate the connector nut (not

the device itself) until finger-tight, being careful not to cross the threads.



5. Use a torque wrench to make final connection. Tighten until the "break" point of the torque wrench is reached. Do **not** push beyond initial break point. Use additional wrench, if needed, to prevent device body from turning.

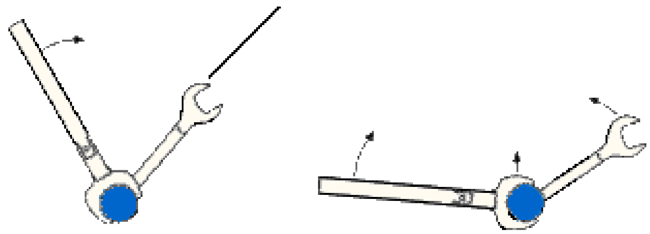


To Separate a Connection

1. Support the devices to avoid any twisting, rocking or bending force on either connector.
2. Use an open-end wrench to prevent the device body from turning.
3. Use another open-end wrench to loosen the connector nut.
4. Complete the disconnection by hand, turning only the connector nut.
5. Pull the connectors straight apart.

To Use a Torque Wrench

1. Make sure torque wrench is set to the correct torque setting.
2. Position torque wrench and a second wrench (to hold device or cable) within 90° of each other before applying force. Make sure to support the devices to avoid putting stress on the connectors.



3. Hold torque wrench lightly at the end of handle–then apply force perpendicular to the torque wrench handle. Tighten until the "break" point of the torque wrench is reached. Do **not** push beyond initial break point.

To Handle and Store Connectors

- Install protective end caps when connectors are not in use.
- Never store connectors, airlines, or calibration standards loose in a box. This is a common cause of connector damage.
- Keep connector temperature the same as analyzer. Holding the connector in your hand or cleaning connector with compressed air can significantly change the temperature. Wait for connector temperature to stabilize before using in calibration or measurements.
- Do not touch mating plane surfaces. Natural skin oils and microscopic particles of dirt are difficult to remove from these surfaces.
- Do not set connectors contact-end down on a hard surface. The plating and mating plane surfaces can be damaged if the interface comes in contact with any hard surface.
- Wear a grounded wrist strap and work on a grounded, conductive table mat. This helps protect the analyzer and devices from electrostatic discharge (ESD).

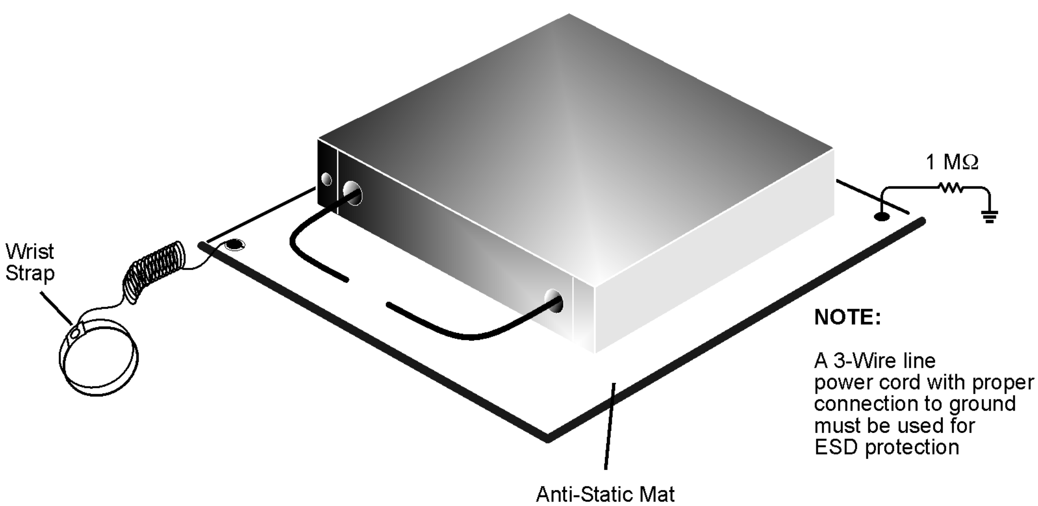
Electrostatic Discharge (ESD) Protection

Protection against electrostatic discharge (ESD) is essential while removing or connecting cables to the network analyzer. Static electricity can build up on your body and can easily damage sensitive internal circuit elements when discharged. Static discharges too small to be felt can cause permanent damage. To prevent damage to the instrument:

- *Always* have a grounded, conductive table mat in front of your test equipment.
- *Aways* wear a grounded wrist strap, connected to a grounded conductive table mat, having a 1 MO resistor in series with it, when making test setup connections.
- *Aways* wear a heel strap when working in an area with a conductive floor. If you are uncertain about the conductivity of your floor, wear a heel strap.
- *Aways* ground yourself before you clean, inspect, or make a connection to a static-sensitive device or test port. You can, for example, grasp the grounded outer shell of the test port or cable connector briefly.
- *Aways* ground the center conductor of a test cable before making a connection to the analyzer test port or other static-sensitive device. This can be done as follows:

    1. Connect a short (from your calibration kit) to one end of the cable to short the center conductor to the outer conductor.
    2. While wearing a grounded wrist strap, grasp the outer shell of the cable connector.
    3. Connect the other end of the cable to the test port and remove the short from the cable.

The following graphic shows a typical ESD protection setup using a grounded mat and wrist strap. Refer to Analyzer Accessories for part numbers.



esd_setup

Impedance Matching Model

Impedance matching is a procedure used in circuit design to match unequal source and load impedances, thereby optimizing the power delivered to the load from the source. Impedance matching is accomplished by inserting matching networks into a circuit between the source and the load.

- **Introduction to the Model**
- **Impedance Matching Model**
- **Description of Exercises**
- **Smith Chart Circuit Elements Paths**
- **Forbidden Regions of the Smith Chart**

Other Tutorials topics

Introduction

In this model, Smith Charts are used to visualize the interactive process of impedance matching to optimize transmitted power in simple circuits. Simple series/shunt, inductance/capacitance matching networks are used, and you can interactively adjust the values of corresponding L and C components. Adjusting the matching network components changes the reflectance of the overall circuit. The reflectance of each part of the circuit is indicated on the Smith Chart as a red or blue ball.

As you adjust the sliders and modify the component values, the model calculates new values for the circuit reflectance and moves the red and blue balls on the Smith Chart. The goal of each exercise is to move the reflectance point from the center of the Smith Chart, which represents either the load or source, into the appropriate red and blue rings which represent the desired matching condition. You can select three different impedance matching problems of increasing difficulty by clicking on one of the three labeled tabs.

Impedance Matching Model

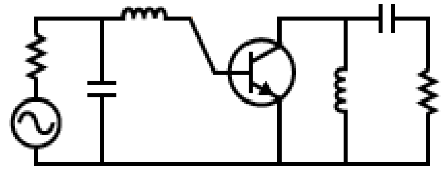
ximize this window for optimum viewing. **Click** if the Impedance model is not visible.

Description of Exercises

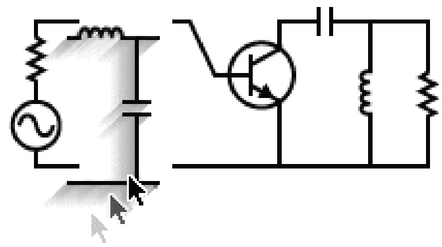
L–C Matching Network

The first exercise lets you use the Smith Chart to perform basic impedance matching between a resistive source and a resistive load. A simple series-inductance shunt-capacitance network is used to match the 50 ohm source to the 300 ohm load. The source reflectance of the circuit looking from the load toward the source is represented by the red ball, while the 300 ohm load is indicated by the stationary red ring.

The objective of the exercise is to interactively match these two impedances by adjusting the L and C sliders. The model will provide graphical feedback by moving the red ball indicating circuit reflectance on the Smith Chart. Adjust the series L and shunt C sliders to move the reflectance point from the center of the Smith Chart to the matching impedance position inside the red ring. You can study the Smith Chart Circuit Element Paths below for hints on how different circuit elements change circuit reflectance on the Smith Chart.



The second exercise provides the impedance matching experience of optimizing the transducer power gain of a transistor amplifier. Matching the 50 ohm source to the input reflectance of the transistor, s^*_{11}, and matching the 50 ohm load to the output reflectance of the transistor, s^*_{22}, optimizes the power delivered from the source, through the transistor, to the load. You are required to match both the input red ball and output blue ball of the transistor separately. Adjust the component values to move both reflectance points to their proper positions within the red and blue rings.

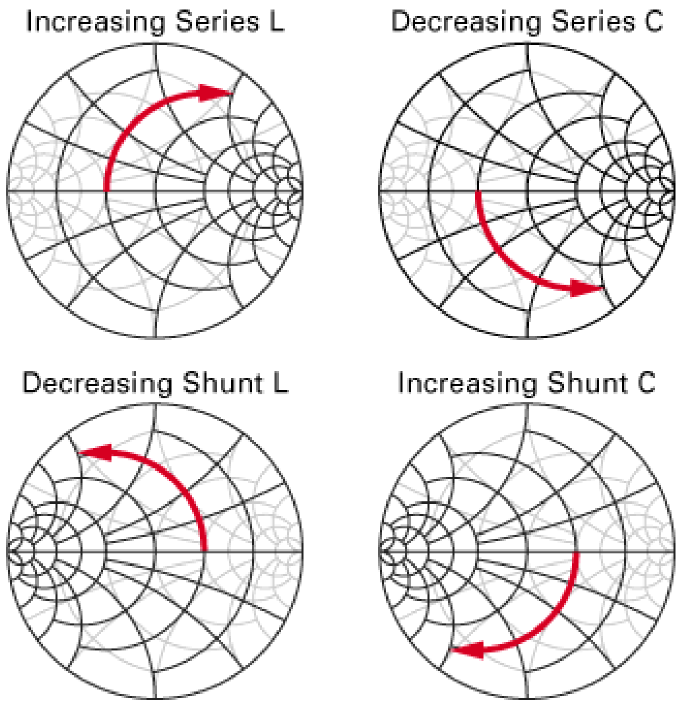


The third part of the interactive impedance matching model is a collection of exercises involving a modular circuit. You begin by constructing a circuit with either one or two modular drag-and-drop matching network components. Once the matching networks have been added to the circuit, the sliders will become active and allow you to adjust the component values. Then you will engage in impedance matching for the circuit you have just created! There are 8 different circuits you can construct and there are 5 different value pairs for s^*_{11} and s^*_{22}on the Smith Chart, altogether 40 impedance matching exercises. **You will find that not all matching networks will work!** For some of the circuits you will be able to construct, you will not be able to position the red ball within the red ring or the blue ball within the blue ring. To determine in advance which matching networks will work, take a close look at the Forbidden Regions of the Smith Chart below. There are 5 different location pairs for s^*_{11} and s^*_{22} corresponding to different frequencies that can be matched. Use the frequency indicator to select an operating frequency, and then drag-and-drop appropriate matching networks into the circuit and adjust the component values to move both reflectance points to their proper positions within the red and blue rings.

Smith Chart Circuit Elements Paths

The graphs below demonstrate how the various shunt and series L and C components change the circuit reflectance on the Smith Chart. Assuming the given component is the last component in the matching network, the circuit reflectance will move as indicated along constant resistance or constant conductance circles.

You can think of impedance matching using the Smith Chart as driving a car to a specific destination in Smith Town - a city were none of the streets are straight! By adjusting circuit components in appropriate order, we can constrain the circuit reflectance to paths along constant resistance or constant conductance circles. Just like road signs can direct a car along the circular streets of Smith Town, so can we reach the matching impedance condition in a straightforward and deterministic way.

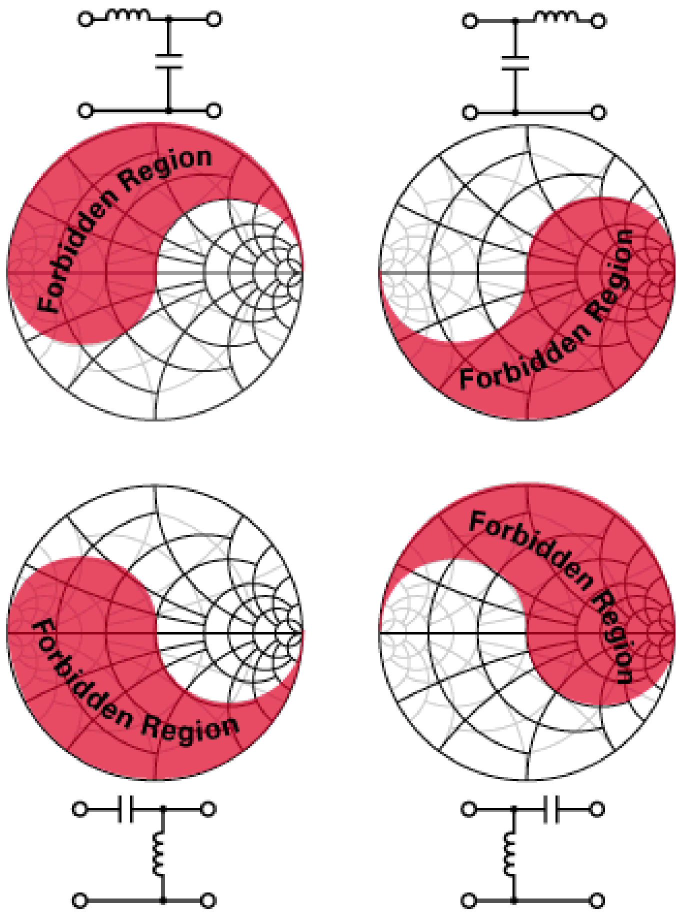


Forbidden Regions of the Smith Chart

For a given load reflectance, only certain L-C matching networks will be capable of transforming the source impedance to the load impedance. In fact, for any load reflectance, exactly two of the four possible L-C matching networks in the Transistor Amplifier-II model above will be able to do the matching job. But which two?

The charts below can be used to determine which matching networks will work in a given load situation. If the load reflectance lies within the forbidden region of the Smith Chart for the indicated matching network, then that network cannot perform the required matching operation. **You cannot drive your car into the forbidden neighborhoods of Smith Town! They are unpaved!**

Use these charts to determine which matching network should be used. First, visually locate the position of the load reflectance from the Transistor Amplifier-II model above on each of the four color Smith Charts below. Then, eliminate the two networks whose forbidden regions overlap the reflectance point, and use one of the remaining two networks to perform the impedance match.


Forbidden Region
Forbidden Region
Forbidden Region
Forbidden Region

Reflected Waves Along a Transmission line

When a sine wave from an RF signal generator is placed on a transmission line, the signal propagates toward the load. This signal, shown here in yellow, appears as a set of rotating vectors, one at each point on the transmission line.

Maximize this window for optimum viewing. **Click** if the applet is not visible.

In our example, the transmission line has a characteristic impedance of 50 ohms. If we choose a load of 50 ohms, then the amplitude of the signal will not vary with position along the line. Only the phase will vary along the line, as shown by the rotating vectors in yellow.

If the load impedance does not perfectly match the characteristic impedance of the line, there will be a reflected signal that propagates toward the source. At any point along the transmission line, that signal also appears to be a constant voltage whose phase is dependent upon physical position along the line.

The voltage seen at one particular point on the line will be the vector sum of the transmitted and reflected sinusoids. We can demonstrate this by looking at two examples.

Example 1: Perfect Match: 50 Ohms

Set the terminating resistor to 50 ohms by using the "down arrow" dialog box. Notice there is no reflection. We have a perfect match. Each rotating vector has a normalized amplitude of 1. If we were to observe the waveform at any point with a perfect measuring instrument, we would see equal sine wave amplitudes anywhere along the transmission line. The signal amplitudes are indicated by the green line.

Example 2: Mismatched Load: 200 Ohms

Now let's intentionally create a mismatched load. Set the terminating resistor to 200 ohms by using the down arrow. Hit the PLAY button and notice the change in the reflected waveform. If it were possible to measure just the reflected wave, we would see that its amplitude does not vary with position along the line. The only difference between the reflected (blue) signal, say at point z6 and point z4, is the phase.

But the amplitude of the resultant waveform, indicated by the standing wave (green), is not constant along the entire line because the transmitted and reflected signals (yellow and blue) combine. Since the phase between the transmitted and reflected signals varies with position along the line, the vector sums will be different, creating what's called a "standing wave".

With the load impedance at 200 ohms, a measuring device placed at point z6 would show a sine wave of constant amplitude. The sine wave at point z4 would also be of constant amplitude, but its amplitude would differ from that of the signal at point z6. And the two would be out of phase with each other. Again, the difference is shown by the green line, which indicates the amplitude at that point on the transmission line.

The impedance along the line also changes, as shown by the points labeled z1 through z7.

Reflection Measurements

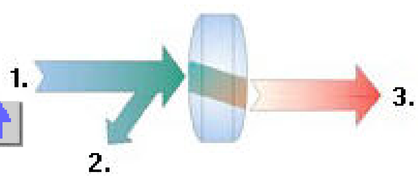
Reflection measurements are an important part of network analysis.

- What are Reflection Measurements?
- Why Make Reflection Measurements?
- Expressing Reflected Waves
  - Return Loss
  - VSWR
  - Reflection Coefficient
  - Impedance
  - Summary of Expressions

See other Tutorials

What are Reflection Measurements?

To understand reflection measurements, it is helpful to think of traveling waves along a transmission line in terms of a lightwave analogy. We can imagine incident light striking some optical component like a clear lens. Some of the light is reflected off the surface of the lens, but most of the light continues on through the lens. If the lens had mirrored surfaces, then most of the light would be reflected and little or none would be transmitted.



1. Incident **2.** Reflected **3.** Transmitted

With RF energy, reflections occur when the impedance of two mated devices are not the same. A reflection measurement is the ratio of the reflected signal to the incident signal. Network analyzers measure the incident wave with the R (for reference) channel and the reflected wave with the A channel. Therefore, reflection is often shown as the ratio of A over R (A/R). We can completely quantify the reflection characteristics of our device under test (DUT) with the amplitude and phase information available at both the A and R channel. In S-parameter terminology, S11 is a reflection measurement of port1 of the device (the input port); S22 is a reflection measurement of the port 2 (the output port)

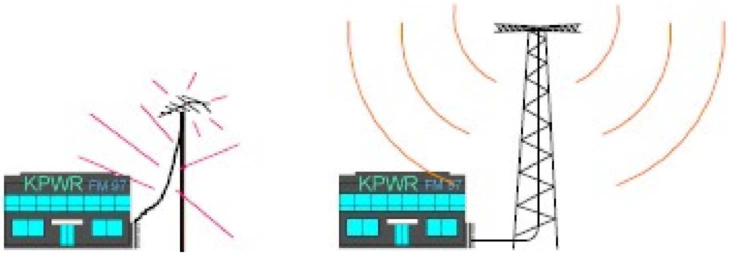
Why Make Reflection Measurements?

One reason we make reflection measurements to assure efficient transfer of RF power. We do this because:

1. RF energy is not cheap. When energy is reflected, that means less energy is transmitted to where it is

intended to go.
2. If the reflected energy is large, it can damage components, like amplifiers.

For example, in the following graphic, the radio station on the left is not operating at peak efficiency. The amplifier impedance is not the same as the transmission line, and the transmission line impedance is not the same as the antenna. Both of these conditions cause high reflected power. This condition results in less transmitted power, and the high reflected power could damage the amplifier.



The radio station on the right installed properly "matched" transmission line and antenna. Very little of the transmitted signal is reflected, resulting in increased broadcast power, more listeners, more advertising revenue, and more profit. The amplifier, transmission, and antenna all need to be measured to ensure that reflected power is minimized.

Expressing Reflected Waves

After making a reflection measurement, the reflection data can be expressed in a number of ways, depending on what you are trying to learn. The various expressions are all calculated by the analyzer from the same reflection measurement data. Each method of expressing reflection data can be graphically displayed in one or more formats. For more information, see display formats.

Return Loss

The easiest way to convey reflection data is return loss. Return loss is expressed in dB, and is a scalar (amplitude only) quantity. Return loss can be thought of as the absolute value or dB that the reflected signal is below the incident signal. Return loss varies between infinity for a perfect impedance match and 0 dB for an open or short circuit, or a lossless reactance. For example, using the log magnitude format on the analyzer, the measured reflection value on the screen may be -18dB. The minus sign is ignored when expressing return loss, so the component is said to have 18dB of return loss.

VSWR

Two waves traveling in opposite directions on the same transmission line cause a "standing wave". This condition can be measured in terms of the voltage standing wave ratio (VSWR or SWR for short). VSWR is defined as the maximum reflected voltage over the minimum reflected voltage at a given frequency. VSWR is a scalar (amplitude only) quantity. VSWR varies between one for a perfect match, and infinity for an open or short circuit or lossless reactance.

Reflection Coefficient

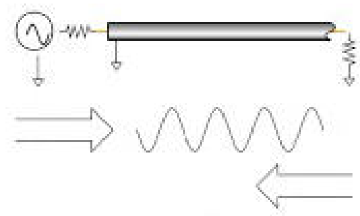
Another way of expressing reflection measurements is reflection coefficient gamma (Γ). Gamma includes both magnitude and phase.

The magnitude portion of gamma is called rho (ρ). Reflection coefficient is the ratio of the reflected signal voltage to the incident signal voltage. The range of possible values for ρ is between zero and one. A transmission line terminated in its characteristic impedance will have all energy transferred to the load; zero energy will be reflected and $\rho = 0$. When a transmission line terminated in a short or open circuit, all energy is reflected and $\rho = 1$. The value of rho is unitless.

Now for the phase information. At high frequencies, where the wavelength of the signal is smaller than the length of conductors, reflections are best thought of as waves moving in the opposite direction of the incident waves. The incident and reflected waves combine to produce a single "standing" wave with voltage that varies with position along the transmission line.

When a transmission line is terminated in its characteristic impedance (Zo) there is no reflected signal. All of the incident signal is transferred to the load, as shown in the following graphic. There is energy flowing in one direction along the transmission line.

Incident Voltage

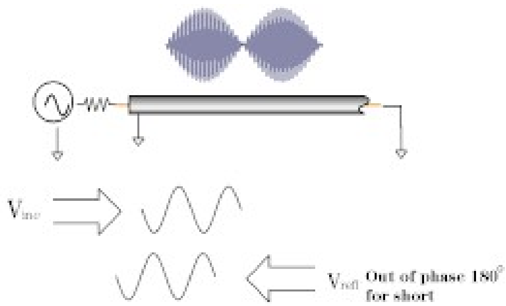


Zo

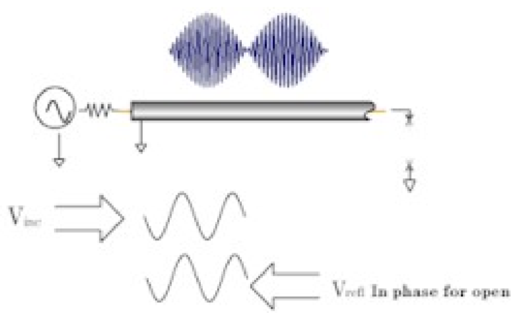
Reflected Voltage = 0

(All the incident power is absorbed in the load)

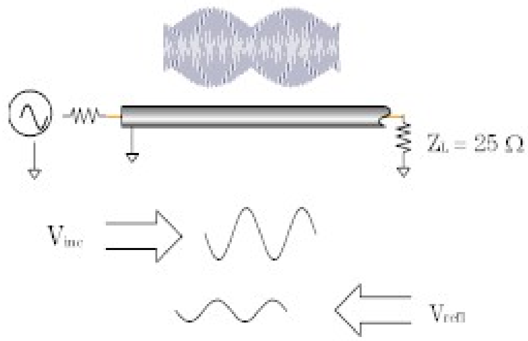
When a transmission line is terminated in a short circuit termination, all of the energy is reflected back to the source. The reflected wave is equal in magnitude to the incident wave (ρ = 1). The voltage across any short circuit is zero volts. Therefore, the voltage of the reflected wave will be 180 degrees out of phase with the incident wave, canceling the voltage at the load.



When a transmission line is terminated in an open circuit termination, all of the energy is reflected back to the source. The reflected wave is equal in magnitude to the incident wave (ρ = 1). However, no current can flow in an open circuit. Therefore, the voltage of the reflected wave will be in phase with the voltage of the incident wave.



When a transmission line is terminated in a 25 ohm resistor, some but not all of the incident energy will be absorbed, and some will be reflected back towards the source. The reflected wave will have an amplitude 1/3 that of the incident wave and the voltage of the two waves will be out of phase by 180 degrees at the load. The phase relationship will change as a function of distance along the transmission line from the load. The valleys of the standing wave pattern will no longer go to zero, and the peaks will be less than that of the open / short circuit.

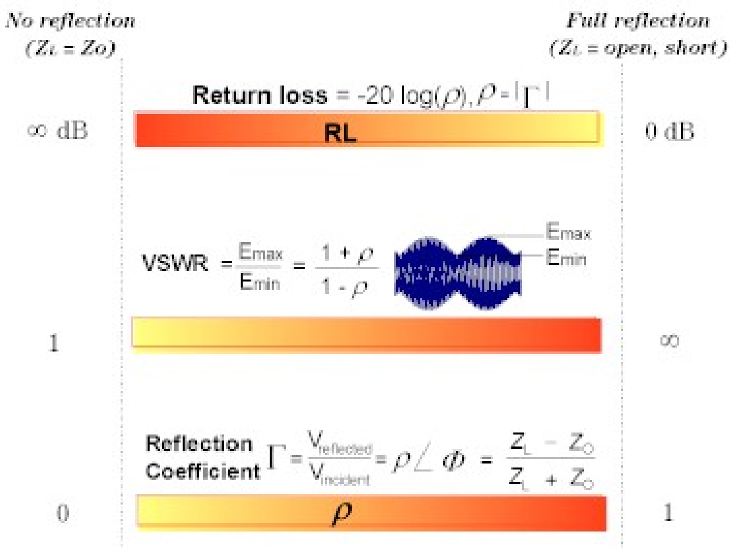


For more information, see Phase Measurements.

Impedance

Impedance is another way of expressing reflection data. For more information on Impedance, see Smith Charts.

Summary of the Expressions of Reflection Measurements:



Phase Measurements

Knowledge of both magnitude and phase characteristics is needed for successful higher-level component integration.

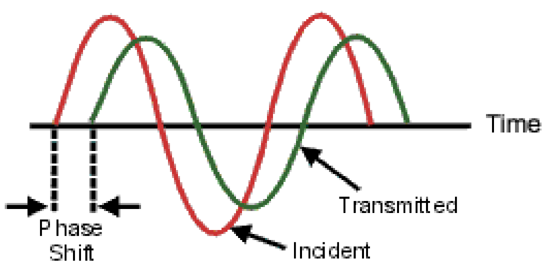
- **What are Phase Measurements?**
- **Why Measure Phase?**
- **Using the Analyzer's Phase Format**
- **Types of Phase Measurements**

See other Tutorials

What are Phase Measurements?

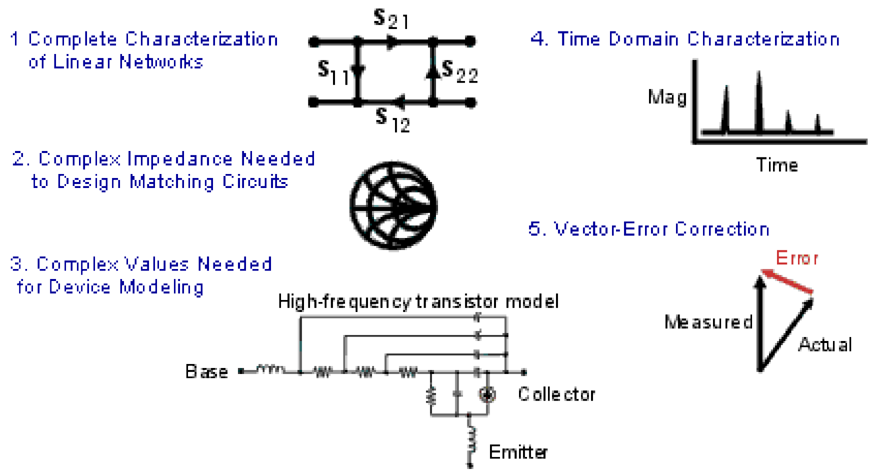
Phase measurements are made using S-parameters, just like amplitude measurements. A phase measurement is a relative (ratio) measurement and not an absolute measurement. Phase measurements compare the phase of the signal going into a device (the incident signal) to the phase of the device's response signal. The response signal can be either reflected or transmitted. Assuming an accurate calibration has been performed, the difference in phase between the two signals (known as phase shift) is a result of the electrical characteristics of the device under test.

The following graphic shows the phase shift (in time or degrees) between an incident signal and a transmitted signal (as might be seen on an oscilloscope display).



Why Measure Phase?

Measuring phase is a critical element of network analysis. The following graphic lists five reasons for measuring both magnitude and phase.



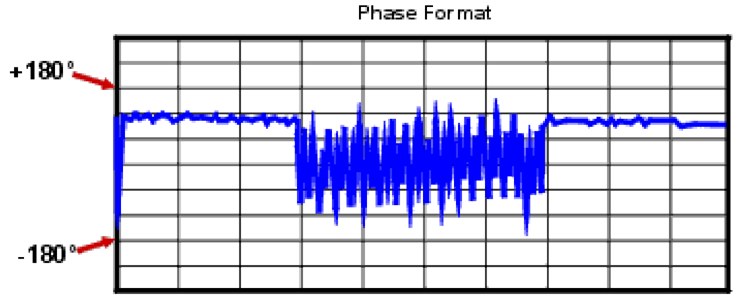
When used in communications systems to pass signals, components or circuits must not cause excessive signal distortion. This distortion can be:

- Linear, where flat magnitude and linear phase shift versus frequency is not maintained over the bandwidth of interest.
- Nonlinear, such as AM-to-PM conversion.

It is important to measure how reflective a component or circuit is, to ensure that it transmits or absorbs energy efficiently. Measuring the complex impedance of an antenna is a good example.

Using the Analyzer's Phase Format

The analyzer's phase format displays a phase-versus-frequency or phase-versus-power measurement. The analyzer does not display more than ±180 degrees phase difference between the reference and test signals. As the phase value varies between +180 degrees and -180 degrees, the analyzer display creates the sawtooth pattern as shown in the following graphic.



The sawtooth pattern does not always reach +180 degrees and -180 degrees. This is because the measurement is made at discrete frequencies, and the data point at +180 degrees and -180 degrees may not be measured for the selected sweep.

Types of Phase Measurements

Complex impedance data is information such as resistance, reactance, phase, and magnitude that can be determined from an S11 or S22 measurement. Complex impedance data can be viewed using either the Smith Chart format or the Polar format.

AM-to-PM conversion is a measure of the amount of undesired phase deviation (PM) that is caused by amplitude variations (AM) of the system. AM-to-PM conversion is usually defined as the change in output phase for a 1-dB increment in the input power to an amplifier (i.e. at the 1 dB gain compression point). This is expressed in degrees-per-dB (°/dB).

Deviation from linear phase is a measure of phase distortion caused by a device. Ideally, the phase shift through a device is a linear function of frequency. The amount of variation from this theoretical phase shift is known as its deviation from linear phase (also called phase linearity).

Group delay is another way to look at phase distortion caused by a device. Group delay is a measure of transit time through a device at a particular frequency. The analyzer computes group delay from the derivative of the measured phase response.

Deviation from Linear Phase Versus Group Delay

Although deviation from linear phase and group delay are similar measurements, they each have their purpose.

The following are the advantages of deviation from linear phase measurements:

- Less noisy than group delay.
- Able to characterize devices that pass phase modulated signals, and show units of phase rather than units of seconds.

The following are the advantages of group delay measurements:

- More easily interpreted indication of phase distortion than deviation from linear phase.
- Able to most accurately characterize a device under test. This is because in determining group delay, the analyzer calculates the slope of the phase ripple, which is dependent on the number of ripples which occur per unit of frequency. Comparing two phase responses with equal peak-to-peak phase ripple, the response with the larger phase slope results in:
    - More group delay variation.
    - More signal distortion.

Amplifier Parameters Reference

- Gain
- Gain Flatness
- Reverse Isolation
- Gain Drift Versus Time
- Deviation from Linear Phase
- Group Delay
- Return Loss (SWR, ρ)
- Complex Impedance
- Gain Compression
- AM-to-PM Conversion

Gain

$$\tau = \frac{V_{trans}}{V_{inc}}$$

$$\text{Gain (dB)} = -20\log_{10}|\tau|$$

$$\text{Gain (dB)} = P_{out}\text{ (dBm)} - P_{in}\text{ (dBm)}$$

The ratio of the amplifier's output power (delivered to a Z_0 load) to the input power (delivered from a Z_0 source). Z_0 is the characteristic impedance, in this case, 50Ω.

For small signal levels, the output power of the amplifier is proportional to the input power. Small signal gain is the gain in this linear region.

As the input power level increases and the amplifier approaches saturation, the output power reaches a limit and the gain drops. Large signal gain is the gain in this nonlinear region. See Gain Compression.

Gain Flatness

The variation of the gain over the frequency range of the amplifier. See Small Signal Gain and Flatness.

Reverse Isolation

The measure of transmission from output to input. Similar to the gain measurement except the signal stimulus is applied to the output of the amplifier. See Reverse Isolation.

Gain Drift versus Time (temperature, bias)

The maximum variation of gain as a function of time, with all other parameters held constant. Gain drift is also observed with respect to other parameter changes such as temperature, humidity or bias voltage.

Deviation from Linear Phase

The amount of variation from a linear phase shift. Ideally, the phase shift through an amplifier is a linear function of frequency. See Deviation from Linear Phase.

Group Delay

$$\tau_g\,(\text{sec}) = -\frac{\Delta\theta}{\Delta\omega}$$

$$= -\frac{1}{360} * \frac{\Delta\theta}{\Delta f}$$

The measure of the transit time through the amplifier as a function of frequency. A perfectly linear phase shift would have a constant rate of change with respect to frequency, yielding a constant group delay. See Group Delay.

Return Loss (SWR, Γ)

$$\Gamma = \frac{V_{refl}}{V_{inc}} = \rho\angle\theta$$

Reflection coefficient = ρ

Return loss (dB) = $-20\log_{10}\rho$

$$SWR = \frac{1+\rho}{1-\rho}$$

The measure of the reflection mismatch at the input or output of the amplifier relative to the system Z_0 characteristic impedance.

Complex Impedance

$$Z = \frac{1+\Gamma}{1-\Gamma} * Z_0$$

$$= -R + jX$$

Complex impedance (1+G). The amount of reflected energy from an amplifier is directly related to its impedance. Complex impedance consists of both a resistive and a reactive component. It is derived from the characteristic impedance of the system and the reflection coefficient. See Complex Impedance.

Gain Compression

An amplifier has a region of linear gain where the gain is independent of input power level (small signal gain). As the power is increased to a level that causes the amplifier to saturate, the gain decreases.

Gain compression is determined by measuring the amplifier’s 1 dB gain compression point (P_{1dB}) which is the output power at which the gain drops 1 dB relative to the small signal gain. This is a common measure of an amplifier’s power output capability. See Gain Compression.

AM-to-PM Conversion Coefficient

$$AM/PM = \frac{\Delta\theta}{\Delta P}$$

The amount of phase change generated in the output signal of an amplifier as a result of an amplitude change of the input signal.

The AM-to-PM conversion coefficient is expressed in units of degrees/dB at a given power level (usually P_{1dB}, which is the 1 dB gain compression point). See AM-PM Conversion.

AM-PM Conversion

The AM-PM conversion of an amplifier is a measure of the amount of undesired phase deviation (PM) that is caused by amplitude variations (AM) inherent in the system.

- **What Is AM-PM Conversion?**
- **Why Measure AM-PM Conversion**
- **Accuracy Considerations**
- **How to Measure AM-PM Conversion**

Other Tutorials topics

What Is AM-PM Conversion?

AM-to-PM conversion measures the amount of undesired phase deviation (PM) that is caused by amplitude variations (AM) of the system. For example, unwanted phase deviation (PM) in a communications system can be caused by:

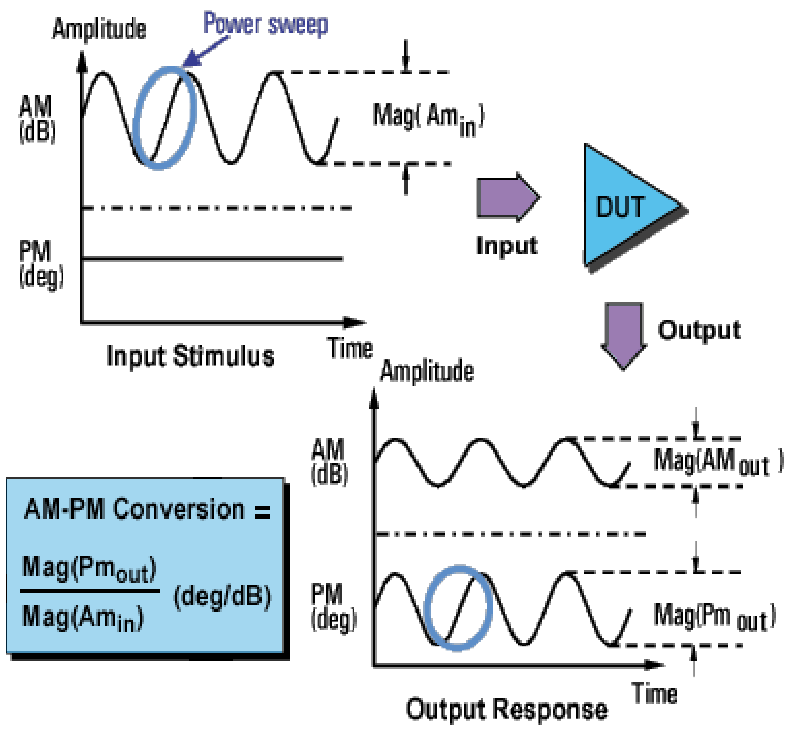
Unintentional amplitude variations (AM)

- Power supply ripple
- Thermal drift
- Multipath fading

Intentional modulation of signal amplitude

- QAM
- Burst modulation

AM-to-PM conversion is usually defined as the change in output phase for a 1-dB increment in the power-sweep applied to the amplifier's input (i.e. at the 1 dB gain compression point). It is expressed in degrees-per-dB (°/dB). An ideal amplifier would have no interaction between its phase response and the power level of the input signal.



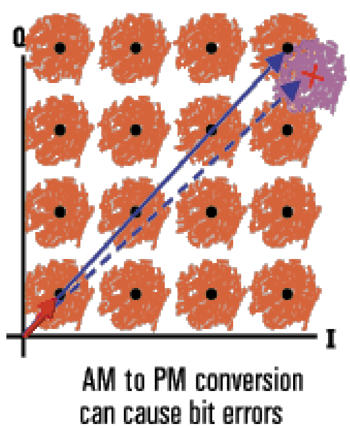
Why Measure AM-PM Conversion

AM-to-PM conversion is a critical parameter in systems where phase (angular) modulation is used, such as:

- FM
- QPSK
- 16QAM

It is a critical parameter because undesired phase deviation (PM) causes analog signal degradation, or increased bit-error rates (BER) in digital communication systems. While it is easy to measure the BER of a digital communication system, this measurement alone does not help you understand the underlying causes of bit errors. AM-to-PM conversion is one of the fundamental contributors to BER, and therefore it is important to quantify this parameter in communication systems.

Refer to the I/Q diagram below for the following discussion on how AM-to-PM conversion can cause bit errors.



- The desirable state change is from the small solid vector to the large solid vector.

- With AM-to-PM conversion, the large vector may actually end up as shown with the dotted line. This is due to phase shift that results from a change in the input power level.
- For a 64QAM signal as shown (only one quadrant is drawn), we see that the noise circles that surround each state would actually overlap, which means that statistically, some bit errors would occur.

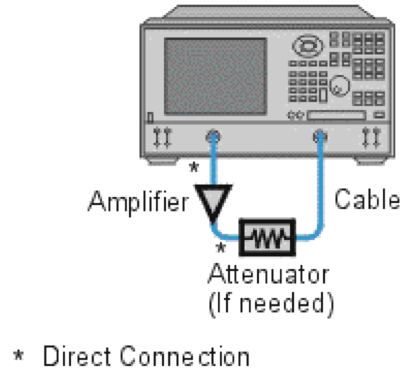
Accuracy Considerations

With this method of measuring AM-to-PM conversion, the modulation frequency is approximately the inverse of the sweep time. Even with the fastest power sweep available on most network analyzers, the modulation frequency ends up being fairly low (typically less than 10 Hz). This could cause a slight temperature change as the sweep progresses, especially if the amplifier has low thermal mass, typical of an unpackaged device. Results using this method could differ slightly if the nonlinear behavior of an amplifier is extremely sensitive to thermal changes. (The PNA series analyzers can make power sweeps <1 ms.)

- The amplifier may respond very differently at various temperatures. The tests should be done when the amplifier is at the desired operating temperature.
- The output power of the amplifier should be sufficiently attenuated if necessary. Too much output power could:
  - damage the analyzer receiver
  - exceed the input compression level of the analyzer receiver, resulting in inaccurate measurements
- Attenuation of the amplifier's output power can be accomplished using:
  - Attenuators
  - Couplers
- The frequency-response effects of the attenuators and couplers must be accounted for during calibration since they are part of the test system. Proper error-correction techniques can reduce these effects.
- The frequency response is the dominant error in an AM-to-PM conversion measurement setup. Performing a thru-response measurement calibration significantly reduces this error. For greater accuracy, perform a 2-port measurement calibration.

How to Measure AM-PM Conversion

1. Preset the analyzer.
2. Select an S21 measurement in the power-sweep mode.
3. Enter the start and stop power levels for the analyzer's power sweep. The start power level should be in the linear region of the amplifier's response (typically 10-dB below the 1-dB compression point). The stop power should be in the compression region of the amplifier's response.
4. Select an external attenuator (if needed) so the amplifier's output power will be sufficiently attenuated to avoid causing receiver compression or damage to the analyzer's port 2.
5. Connect the amplifier as shown in the following graphic, and provide the dc bias.



6. Select the analyzer settings for your amplifier under test in order to perform a swept-power gain compression measurement at a chosen frequency. See Gain Compression.

7. Remove the amplifier and perform a measurement calibration. Be sure to include the attenuator and cables in the calibration setup if they will be used when measuring the amplifier.

8. Save the instrument state to memory.

9. Reconnect the amplifier.

10. Use a reference marker to target the amplifier's input power at the 1-dB gain compression point. Select a second marker and adjust its stimulus value until its response is 1-dB below the reference marker.

11. Change the S_{21} measurement from a log magnitude format to a phase format (no new calibration is required).

12. Find the phase change between the markers. The value is the AM-to-PM conversion coefficient at the 1-dB gain compression point.

13. Print the data or save it to a disk.

Antenna Measurements

This topic describes how to setup the PNA to make S21 measurements on an array of antennas. Measurements can be made on up to 100 antenna arrays (Ports) and up to 15 discrete frequencies

Measurement Sequence

1. The PNA is set to a start frequency.
2. As the antenna moves, the PNA responds to each external trigger signal by measuring an antenna port.
3. When all ports are measured, the PNA increments to the next frequency
4. Again the PNA measures all ports, and so forth until all ports are measured at all frequencies in the forward direction.
5. As the antenna begins moving in the opposite direction, the same sequence occurs, except the PNA decrements in frequency until all ports are measured at all frequencies and the PNA is set back to the original start frequency.

Once setup, only external trigger signals are sent to the PNA. After each trigger, measurement data is stored in internal PNA memory.

How to set up the PNA

See the Antenna Macro to learn how to do this automatically.

1. On the **System** menu click **Preset**
2. On the **Sweep** menu point to **Trigger** then click **Trigger**
3. In Trigger Source click **External**
4. In Trigger Scope click **Channel**
5. Click **OK**

Forward Sweep

1. On the **Trace** menu click **New Trace**
2. Click **S21** then Channel Number **1**
3. On the **Sweep** menu point to **Trigger** then click **Trigger**
4. In Channel Trigger State check **Point Sweep**
5. Click **OK**
6. On the Sweep menu click **Sweep Type**:then **Segment Sweep**
7. Click **OK**

8. On the **View** menu point to **Tables** then click **Segment Table**
9. Do this 15 times **-** Sweep menu point to **Segment Table** then **Insert Segment**
10. For each Segment in the Segment table:
    1. Click **State**:and select **ON**
    2. Double click both **START** and **STOP** Frequency: (each new segment ascends in frequency)
    3. Double click **Points**: type Number of Ports (elements)

Reverse sweep

Repeat the following steps for each frequency: (up to 15)

- Increment the channel number (**X**) Starting with Channel 2
- Decrement the frequency (**F**)

1. On the **Trace** menu click **New Trace...**
2. Click **S21** then Channel Number ***X***
3. When a window contains four traces, check **Create in New Window**.
4. Click **OK**
5. On the **Sweep** menu point to **Trigger** then click **Trigger**
6. In Channel Trigger State check **Point Sweep**
7. Click **OK**
8. On the Sweep menu click **Sweep Type**:then **Segment Sweep**
9. Click **OK**
10. On the **View** menu point to **Tables** then click **Segment Table**
11. In the Segment table
    1. Click **State**:and select **ON**
    2. Double click both **START** and **STOP** Frequency ***F***
    3. Double click **Points**: type Number of Ports (elements)

Antenna Test Macro

This VB Script macro automates the setup of a PNA as documented in Antenna Measurements. This topic will show you how to copy this macro into a PNA and then run the macro.

For more information, see Using Macros.

Note: Print these instructions if viewing in the PNA. This topic will be covered by the Macro Setup dialog box.

1. Copy the following code into a Notepad file.

```
'Start copying here
Set rfna = CreateObject("AgilentPNA835x.Application")
Set chans = rfna.Channels
numfreqs = 15
portz = 3
rfna.Preset
rfna.Reset
rfna.TriggerSignal=3 ' Manual trigger
rfna.TriggerType=1   ' ChannelTrigger
'setup forward sweep
win = 1
trce = 1
rfna.CreateMeasurement 1, "S21", 2, win
Set chan = rfna.ActiveChannel
chan.TriggerMode = 0 'TriggerModePoint
chan.SweepType = 4   'SegmentSweep

For i = 1 To 15
chan.Segments.Add (i)
chan.Segments(i).State = True
chan.Segments(i).NumberOfPoints = portz
Next

chan.Segments(1).StartFrequency = 30000
chan.Segments(1).StopFrequency = 30000
chan.Segments(2).StartFrequency = 1000000
chan.Segments(2).StopFrequency = 1000000
chan.Segments(3).StartFrequency = 500000
chan.Segments(3).StopFrequency = 500000
chan.Segments(4).StartFrequency = 10000000
chan.Segments(4).StopFrequency = 10000000
chan.Segments(5).StartFrequency = 100000000
chan.Segments(5).StopFrequency = 100000000
chan.Segments(6).StartFrequency = 500000000
chan.Segments(6).StopFrequency = 500000000
chan.Segments(7).StartFrequency = 1000000000
chan.Segments(7).StopFrequency = 1000000000
chan.Segments(8).StartFrequency = 2000000000
chan.Segments(8).StopFrequency = 2000000000
chan.Segments(9).StartFrequency = 3000000000
chan.Segments(9).StopFrequency = 3000000000
chan.Segments(10).StartFrequency = 4000000000
chan.Segments(10).StopFrequency = 4000000000
chan.Segments(11).StartFrequency = 5000000000
chan.Segments(11).StopFrequency = 5000000000
```

```
chan.Segments(12).StartFrequency = 6000000000
chan.Segments(12).StopFrequency = 6000000000
chan.Segments(13).StartFrequency = 7000000000
chan.Segments(13).StopFrequency = 7000000000
chan.Segments(14).StartFrequency = 8000000000
chan.Segments(14).StopFrequency = 8000000000
chan.Segments(15).StartFrequency = 9000000000
chan.Segments(15).StopFrequency = 9000000000

'setup reverse sweep
'setup the channel
'start with ch 2
For ch = 2 To numfreqs + 1

'put four traces per window
trce = trce + 1
If trce = 5 Then
win = win + 1
trce = 1
End If

rfna.CreateMeasurement ch, "S21", 2, win

Set chan = rfna.ActiveChannel
chan.TriggerMode = 0 'TriggerModePoint
chan.SweepType = 4   'SegmentSweep
Set segs = chan.Segments

Select Case ch
Case 2
segs(1).StartFrequency = 9000000000
segs(1).StopFrequency = 9000000000
Case 3
segs(1).StartFrequency = 8000000000
segs(1).StopFrequency = 8000000000
Case 4
segs(1).StartFrequency = 7000000000
segs(1).StopFrequency = 7000000000
Case 5
segs(1).StartFrequency = 6000000000
segs(1).StopFrequency = 6000000000
Case 6
segs(1).StartFrequency = 5000000000
segs(1).StopFrequency = 5000000000
Case 7
segs(1).StartFrequency = 4000000000
segs(1).StopFrequency = 4000000000
Case 8
segs(1).StartFrequency = 3000000000
segs(1).StopFrequency = 3000000000
Case 9
segs(1).StartFrequency = 2000000000
segs(1).StopFrequency = 2000000000
Case 10
segs(1).StartFrequency = 1000000000
segs(1).StopFrequency = 1000000000
Case 11
```

```
segs(1).StartFrequency = 500000000
segs(1).StopFrequency = 500000000
Case 12
segs(1).StartFrequency = 100000000
segs(1).StopFrequency = 100000000
Case 13
segs(1).StartFrequency = 10000000
segs(1).StopFrequency = 10000000
Case 14
segs(1).StartFrequency = 500000
segs(1).StopFrequency = 500000
Case 15
segs(1).StartFrequency = 100000
segs(1).StopFrequency = 100000
Case 16
segs(1).StartFrequency = 30000
segs(1).StopFrequency = 30000
End Select

segs(1).NumberOfPoints = portz
Next

'End copying here
```

2. Save the file on the analyzer hard drive in the **C:\Documents** folder. Name the file **AntennaTest.vbs** . If viewing this on an external PC, you can transfer the file to the PNA with a floppy disk or by mapping a drive.
3. Close Notepad

Load the macro into the analyzer

1. In the analyzer application, click **System,** point to **Macro**, then click **Macro Setup**
2. Click on a blank line below the last entry. (There may be NO entry.)
3. Click **Edit**
4. In the **Macro Title** box**,** type **Antenna Test**
5. Click **Browse.**
6. Change **Files of Type** to **VBScript** Go to the **C:\Documents** folder.
7. Click **AntennaTest.vbs**
8. Click **OK**

9. Click **OK** on the Macro Setup dialog box.

To run the macro:

1. Press 

on the Front Panel **UTILITY** keys.
2. Click the Active Entry key labeled **Antenna Test**

Complex Impedance

When making an S_{11} or S_{22} measurement of your device under test, you can view complex-impedance data such as series resistance and reactance as well as phase and magnitude information. Complex impedance data can be viewed using either the Smith Chart format or the Polar format.

- What Is Complex Impedance?
- Accuracy Considerations
- How to Measure Complex Impedance

What Is Complex Impedance?

Complex-impedance data is information that can be determined from an S_{11} or S_{22} measurement of your device under test, such as:

- Resistance
- Reactance
- Phase
- Magnitude

The amount of power reflected from a device is directly related to the impedances of both the device and the measuring system. For example, the value of the complex reflection coefficient (Γ) is equal to 0 only when the device impedance and the system impedance are exactly the same (i.e. maximum power is transferred from the source to the load). Every value for Γ corresponds uniquely to a complex device impedance (as a function of frequency), according to the equation:

$$Z_L = [(1 + \Gamma) / (1 - \Gamma)] \times Z_0$$

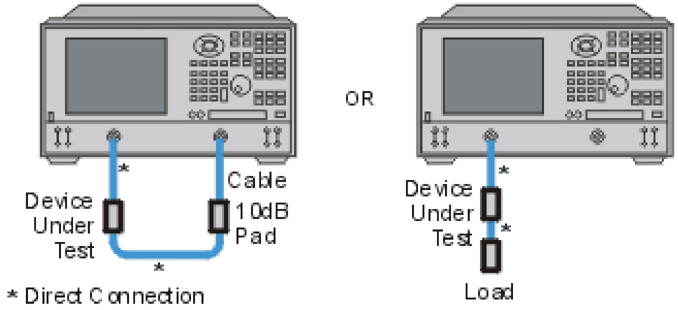
where Z_L is your test device impedance and Z_0 is the measuring system's characteristic impedance.

Complex Impedance is best viewed using either Polar or Smith Chart format.

Accuracy Considerations

- The Smith chart is most easily understood when used with a full scale value of 1.0.
- For greater accuracy when using markers in the Smith chart or polar formats, activate the discrete marker mode.
- The uncertainty of reflection measurements is affected by:
  - Directivity
  - Reflection tracking
  - Source match
  - Load match (with 2-port devices)

With a 2-port calibration, the effects of these factors are reduced. A 1-port calibration provides the same accuracy if the output of the device is well terminated. Refer to the graphic below for the following discussion.



- If you connect the device between both analyzer ports, it is recommended that you use a 10 dB pad on the output of the device to improve measurement accuracy. This is not necessary if you use a 2-port calibration since it corrects for load match.
- If you connect a two-port device to only one analyzer port, it is recommended that you use a high-quality load (such as a calibration standard) on the output of the device.

How to Measure Complex Impedance

1. Connect the device as shown in the previous graphic.
2. Preset the analyzer.
3. Set up, calibrate, and perform an S11 or S22 measurement.
4. View impedance data:
    a. Select the Smith Chart format.
    b. Scale the displayed measurement for optimum viewing.
    c. Position the marker to read the resistive and reactive components of the complex impedance at any point along the trace.
    d. Print the data or save it to a disk.
5. View the magnitude and phase of the reflection coefficient:
    a. Select the Smith chart format or the Polar format.
    b. Select either Lin Marker or Log Marker formats.
    c. Scale the displayed measurement for optimum viewing.
    d. Position the marker to read the frequency, magnitude, and phase of the reflection coefficient (Γ) at any point along the trace.
    e. Print the data or save it to a disk.

Deviation from Linear Phase

Deviation from linear phase is a measure of phase distortion. The electrical delay feature of the analyzer is used to remove the linear portion of the phase shift from the measurement. This results in a high-resolution display of the non-linear portion of the phase shift (deviation from linear phase).

What Is Linear Phase Shift?

What Is Deviation from Linear Phase?

Why Measure Deviation from Linear Phase?

Using Electrical Delay

Accuracy Considerations

See other Tutorials

What Is Linear Phase Shift?

Phase shift occurs because the wavelengths that occupy the electrical length of the device get shorter as the frequency of the incident signal increases. *Linear* phase-shift occurs when the phase response of a device is linearly proportional to frequency. Displayed on the analyzer, the phase-versus-frequency measurement trace of this ideal linear phase shift is a straight line. The slope is proportional to the electrical length of the device. Linear phase shift is necessary (along with a flat magnitude response) for distortionless transmission of signals.

What Is Deviation from Linear Phase?

In actual practice, many electrical or electronic devices will delay some frequencies more than others, creating non-linear phase-shift (distortion in signals consisting of multiple-frequency components). Measuring deviation from linear phase is a way to quantify this non-linear phase shift.

Since it is only the deviation from linear phase which causes phase distortion, it is desirable to remove the linear portion of the phase response from the measurement. This can be accomplished by using the electrical delay feature of the analyzer to mathematically cancel the electrical length of the device under test. What remains is the deviation from linear phase, or phase distortion.

Why Measure Deviation from Linear Phase?

The deviation from linear phase measurement accomplishes the following:

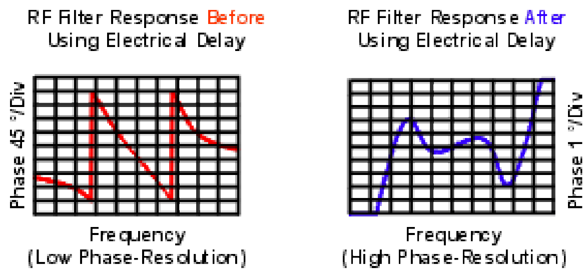
- Presents data in units of phase rather than units of seconds (group delay). For devices that pass modulated signals, units of phase may be most practical.
- Provides a less noisy measurement than a group delay measurement.

Using Electrical Delay

The electrical delay feature is the electronic version of the mechanical "line stretcher" of earlier analyzers. This feature does the following:

- Simulates a variable-length lossless transmission line, which is effectively added to or removed from the reference signal path.

- Compensates for the electrical length of the device under test.
- Flattens the measurement trace on the analyzer's display. This allows the trace to be viewed at high resolution in order to see the details of the phase nonlinearity.
- Provides a convenient method to view the deviation from linear phase of the device under test. See the following graphic.

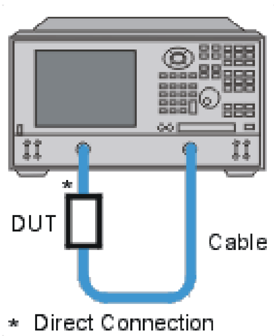


Accuracy Considerations

The frequency response of the test setup is the dominant error in a deviation from linear phase measurement. Performing a thru-response measurement calibration significantly reduces this error. For greater accuracy, perform a 2-port measurement calibration.

How to Measure Deviation from Linear Phase:

1. Preset the analyzer.
2. If your device under test is an amplifier, it may be necessary to adjust the analyzer's source power:
    - Set the analyzer's source power to be in the linear region of the amplifier's output response (typically 10-dB below the 1-dB compression point).
    - Select an external attenuator (if needed) so the amplifier's output power will be sufficiently attenuated to avoid causing receiver compression or damage to the analyzer's port 2.
3. Connect the device under test as shown in the following graphic.



3. Select an S21 measurement.
4. Select the settings for your device under test, including the following:

- Format: phase
- Scale: autoscale

5. Remove the device and perform a calibration.
6. Reconnect the device.
7. Scale the displayed measurement for optimum viewing.
8. Activate the electrical delay function to automatically flatten the phase trace. If desired, manually fine-tune electrical delay to adjust the flatness of the phase trace.
9. Use the markers to measure the maximum peak-to-peak deviation from linear phase.
10. Print the data or save it to a disk.

Group Delay

Group delay is a measure of phase distortion. Group delay is the actual transit time of a signal through a device under test as a function of frequency. When specifying group delay, it is important to specify the aperture used for the measurement.

- What is Group Delay?
- Group Delay versus Deviation from Linear Phase
- What Is Aperture?
- Accuracy Considerations
- How to Measure Group Delay

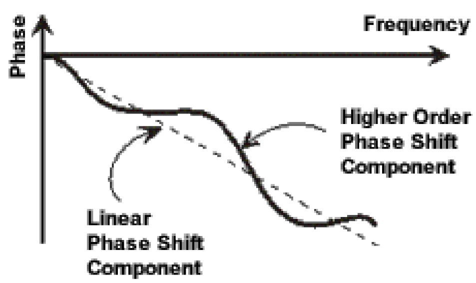
See other Amplifier Parameter topics

What Is Group Delay?

Group delay is:

- A measure of device phase distortion.
- The transit time of a signal through a device, versus frequency.
- The derivative of the device's phase characteristic with respect to frequency.

Refer to the graphic below for the following discussion:



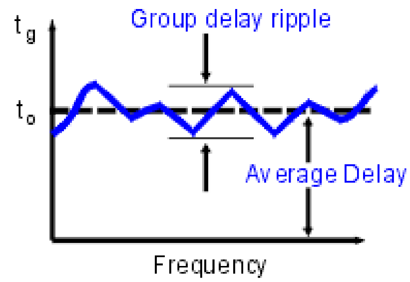
$$\text{Group Delay} = t_g = \frac{-d\varphi}{d\omega} \qquad \varphi \text{ in Radians}, \; \omega \text{ in Radians/Sec}$$

$$= \frac{-1}{360^\circ} \cdot \frac{d\Theta}{df} \qquad \Theta \text{ in Degrees}, \; f \text{ in Hz } (\omega = 2\pi f)$$

The phase characteristic of a device typically consists of both linear and higher order (deviations from linear) phase-shift components.

Linear phase-shift component:	**Higher-order phase-shift component:**
Represents average signal transit time.	Represents variations in transit time for different frequencies.
Attributed to electrical length of test device.	Source of signal distortion.

Refer to the graphic below for the following discussion:



In a group delay measurement:

- The linear phase shift component is converted to a constant value (representing the average delay).
- The higher order phase shift component is transformed into deviations from constant group delay (or group delay ripple).
- The deviations in group delay cause signal distortion, just as deviations from linear phase cause distortion.
- The measurement trace depicts the amount of time it takes for each frequency to travel through the device under test.

Refer to the following equation for this discussion on how the analyzer computes group delay:

$$\text{Group Delay} = tg = \frac{-d\varphi}{d\omega} = \frac{-1}{360^\circ} \cdot \frac{d\Theta}{df}$$

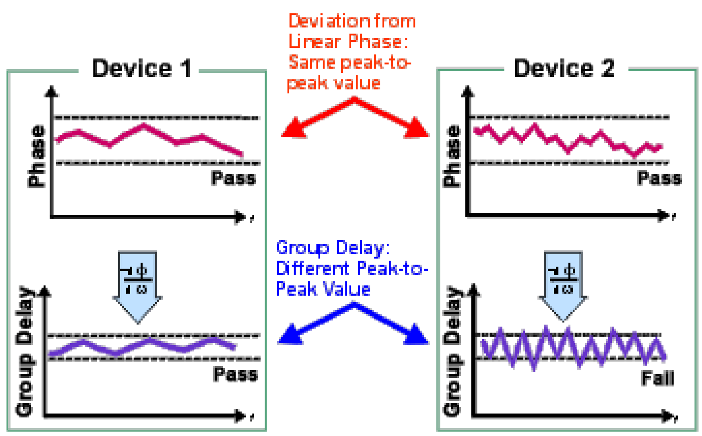
φ in Radians
ω in Radians/Sec
Θ in Degrees
f in Hz ($\omega = 2\pi f$)

- Phase data is used to find the phase change (-dϕ).
- A specified frequency aperture is used to find the frequency change (dω).
- Using the two values above, an approximation is calculated for the rate of change of phase with frequency.
- This approximation represents group delay in seconds (assuming linear phase change over the specified frequency aperture).

Group Delay versus Deviation from Linear Phase

Group delay is often a more accurate indication of phase distortion than Deviation from Linear Phase.

Deviation from linear phase results are shown in the upper region of the following graphic: Device 1 and device 2 have same value, despite different appearances.



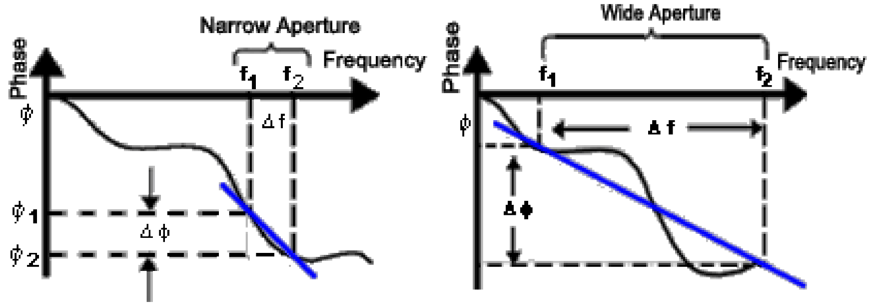
Group Delay results are shown in the lower region:

Device 1 and device 2 have different values of group delay. This is because in determining group delay, the analyzer calculates slope of phase ripple, which is dependent on number of ripples which occur per unit of frequency.

What Is Aperture?

During a group delay measurement, the analyzer measures the phase at two closely spaced frequencies and then computes the phase slope. The frequency interval (frequency delta) between the two phase measurement points is called the aperture. Changing the aperture can result in different values of group delay. The computed slope (delta phase) varies as the aperture is increased. This is why when you are comparing group delay data, you must know the aperture that was used to make the measurements.

Refer to the graphic below for the following discussion:



Narrow aperture:	**Wide aperture:**
Provides more fine detail in phase linearity.	Provides less fine detail in phase linearity because some phase response averaged-out or not measured.
Makes measurement susceptible to noise (smaller signal-to-noise ratio) and analyzer phase detector resolution.	Makes measurement less susceptible to noise (larger signal-to-noise ratio).

The analyzer's default setting for group delay aperture is the frequency span divided by the number of points across the display. There are two ways to set the aperture to a different value.

1. Adjust the number of measurement points or the frequency span.

- Increasing the number of points or reducing the frequency span narrows the aperture.
- Decreasing the number of points and/or increasing the frequency span widens the aperture.

Note: if the aperture is too wide (more than 180° of phase shift between adjacent frequency points), errors in group delay data will occur.

2. Use the analyzer's smoothing function.

- Performs a single-sweep, moving average of adjacent data-points over a specified percentage of the frequency span.
- Results in an action similar to changing the frequency interval between points.
- Allows a wider aperture because greater than 180º of phase shift can occur over the smoothing aperture.

Group delay measurements can be made on the following sweep types:

- Linear frequency

- List frequency sweep segment

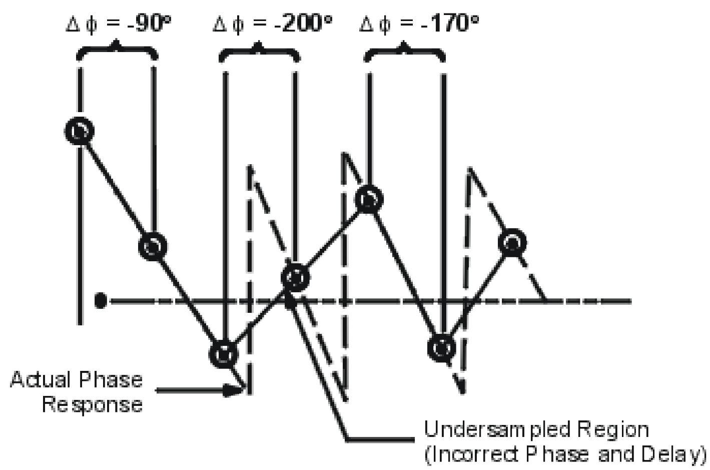
The group delay aperture varies depending on the frequency spacing and point density, therefore the aperture is not constant in segment sweep. In segment sweep, extra frequency points can be defined to ensure the desired aperture.

Accuracy Considerations

It is important to keep the phase difference between two adjacent measurement points less than 180° (see the following graphic). Otherwise, incorrect phase and delay information may result. Undersampling may occur when measuring devices with long electrical length. You can verify that the phase difference measured between two adjacent points is less than 180° by adjusting the following settings until the measurement trace no longer changes:

- Increase the number of points
- Narrow the frequency span

Electrical delay may also be used to compensate for this effect.

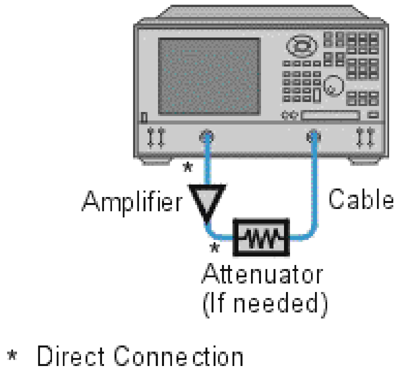


The frequency response is the dominant error in a group delay test setup. Performing a thru-response measurement calibration significantly reduces this error. For greater accuracy, perform a 2-port measurement calibration.

Particularly for an amplifier, the response may vary differently at various temperatures. The tests should be done when the amplifier is at the desired operating temperature.

How to Measure Group Delay

1. Preset the analyzer.
2. If your device under test is an amplifier, it may be necessary to adjust the analyzer's source power:
    - Set the analyzer's source power to be in the linear region of the amplifier's output response (typically 10-dB below the 1-dB compression point).
    - Select an external attenuator (if needed) so the amplifier's output power will be sufficiently attenuated to avoid causing receiver compression or damage to the analyzer's port 2.
3. Connect the device under test as shown in the following graphic.



4. Select an S_{21} measurement.
5. Select the settings for your device under test, including the following:
   - number of measurement points: maximum
   - format: delay
   - scale: autoscale
6. Remove the device under test and perform a measurement calibration.
7. Reconnect the device under test.
8. Scale the displayed measurement for optimum viewing.
9. Use the analyzer's smoothing feature to increase the aperture, reducing noise on the trace while maintaining meaningful detail. To increase the aperture:
   - Switch on the analyzer's smoothing feature.
   - Vary the smoothing aperture (up to 25% of the span swept).
10. Use the markers to measure group delay (expressed in seconds) at a particular frequency of interest.
11. Print the data or save it to a disk.

Small Signal Gain and Flatness

Small signal gain is the gain in the amplifier's linear region of operation. This is typically measured at a constant input power over a swept frequency. Gain flatness is the measure of the variation of gain over a specified frequency range.

What Is Gain?

What Is Flatness?

Why Measure Gain and Flatness?

Accuracy Considerations

How to Measure Gain and Flatness

See other Amplifier Parameter topics

What Is Gain?

RF amplifier gain is defined as the difference in power between the amplifier output signal and the input signal. It is assumed that both input and output impedances of the amplifier are the same as the characteristic impedance of the system.

- Gain is called S_{21} using S-parameter terminology
- Gain is expressed in dB-a logarithmic ratio of the output power relative to the input power.
- Gain can be calculated by subtracting the input from the output levels when both are expressed in dBm, which is power relative to 1 milliwatt.
- Amplifier gain is most commonly specified as a minimum value over a specified frequency range. Some amplifiers specify both minimum and maximum gain, to ensure that subsequent stages in a system are not under or over driven.

What Is Flatness?

Flatness specifies how much the amplifier's gain can vary over the specified frequency range. Variations in the flatness of the amplifier's gain can cause distortion of signals passing through the amplifier.

Why Measure Small-Signal Gain and Flatness?

Deviations in gain over the bandwidth of interest will induce distortion in the transmitted signal because frequency components are not amplified equally. Small-signal gain allows you to quantify the amplifier's gain at a particular frequency in a 50-ohm system. Flatness allows you to view the deviations in the amplifier's gain over a specified frequency range in a 50-ohm system.

Accuracy Considerations

- The amplifier may respond very differently at various temperatures. The tests should be done when the amplifier is at the desired operating temperature.

- The output power of the amplifier should be sufficiently attenuated if necessary. Too much output power could:
    - damage the analyzer receiver
    - exceed the input compression level of the analyzer receiver, resulting in inaccurate measurements.

Attenuation of the amplifier's output power can be accomplished using:

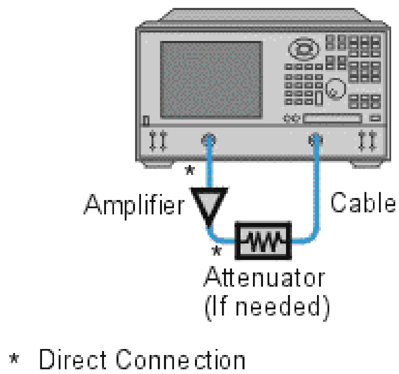
- attenuators
- couplers

The frequency-response effects and mismatches of the attenuators and couplers must be accounted for during calibration since they are part of the test system. Proper error-correction techniques can reduce these effects.

- The frequency response is the dominant error in a small-signal gain and flatness measurement setup. Performing a thru-response measurement calibration significantly reduces this error. For greater accuracy, perform a 2-port measurement calibration.
- Reducing IF bandwidth or using averaging improves measurement dynamic range and accuracy, at the expense of measurement speed.

How to Measure Gain and Flatness

1. Preset the analyzer.
2. Select an S21 measurement parameter.
3. Set the analyzer's source power to be in the linear region of the amplifier's output response (typically 10-dB below the 1-dB compression point).
4. Select an external attenuator (if needed) so the amplifier's output power will be sufficiently attenuated to avoid causing receiver compression or damage to the analyzer's port-2.



5. Connect the amplifier as shown in the following graphic, and provide the dc bias.
6. Select the analyzer settings for your amplifier under test.
7. Remove the amplifier and perform a measurement calibration. Be sure to include the attenuator and cables in the calibration setup if they will be used when measuring the amplifier.

8. Save the instrument-state to memory.
9. Reconnect the amplifier.
10. Scale the displayed measurement for optimum viewing and use a marker to measure the small signal gain at a desired frequency.
11. Measure the gain flatness over a frequency range by using markers to view the peak-to-peak ripple.
12. Print or save the data to a disk.
13. This type of measurement can be automated.

Gain Compression

Gain compression measures the level of input power applied to an amplifier that will cause a distorted output.

What Is Gain Compression?

Why Measure Gain Compression?

Accuracy Considerations

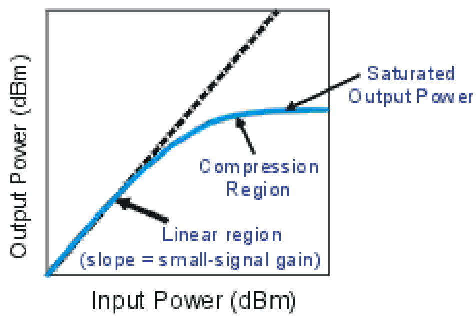
How to Measure Gain Compression

See other Amplifier Parameter topics

What Is Gain Compression?

Gain compression occurs when the input power of an amplifier is increased to a level that reduces the gain of the amplifier and causes a nonlinear increase in output power.

The analyzer has the ability to do power sweeps as well as frequency sweeps. Power sweeps help characterize the nonlinear performance of an amplifier. Refer to the graphic below (a plot of an amplifier's output power versus input power at a single frequency) for the following discussion.



- The amplifier has a linear region of operation where gain is constant and independent of power level. The gain in this region is commonly referred to as "small-signal gain."
- As the input power increases, the amplifier gain appears to decrease, and the amplifier goes into compression.
- The most common measurement of amplifier compression is the 1-dB compression point. This is defined as the input power (or sometimes the output power) which results in a 1-dB decrease in amplifier gain (relative to the amplifier's small-signal gain).

Why Measure Gain Compression?

When driven with a sinusoid, the output of an amplifier is no longer sinusoidal in the compression region. Some of the amplifier output appears in harmonics, rather than occurring only at the fundamental frequency of the input signal.

As input power is increased even more, the amplifier becomes saturated, and output power remains constant. At this point, further increases in amplifier input power result in no change in output power.

In some cases (such as with TWT amplifiers), output power actually decreases with further increases in input power after saturation, which means the amplifier has negative gain.

Since gain is desired in amplifier operation, it is important to know the limit of input signal that will result in gain compression.

Accuracy Considerations

The network analyzer must provide sufficient power to drive the amplifier into saturation. If you need a higher input-power level than the source of the analyzer can provide, use a preamplifier to boost the power level prior to the amplifier under test. (See High-Power Component Measurements.) If using a preamplifier, you can increase measurement accuracy in the following ways:

- Use a coupler on the output of the preamplifier so that a portion of the boosted input signal can be used for the analyzer's reference channel. This configuration removes the preamplifier's frequency response and drift errors from the measurement (by ratioing).
- Perform a thru-response calibration including the preamplifier, couplers, and attenuators in the test setup.

The output power of the amplifier should be sufficiently attenuated if necessary. Too much output power could:

- Damage the analyzer receiver
- Exceed the input compression level of the analyzer receiver

Attenuation of the amplifier's output power can be accomplished using:

- Attenuators
- Couplers

The frequency-response effects of the attenuators and couplers must be considered during calibration since they are part of the test system. Proper error-correction techniques can reduce these effects.

- The frequency response is the dominant error in a gain compression measurement setup. Performing a thru-response measurement calibration significantly reduces this error.
- The amplifier may respond very differently at various temperatures. The tests should be done when the amplifier is at the desired operating temperature.
- Reducing IF bandwidth or using measurement averages improves accuracy, at the expense of measurement speed.

How to Measure Gain Compression

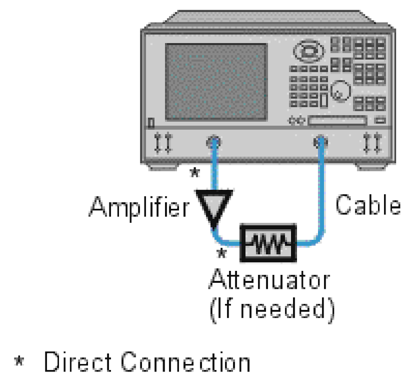
This procedure shows you how to make the following three measurements used to determine amplifier gain compression:

1. A Swept-Frequency Gain Compression measurement locates the lowest frequency at which the 1-dB gain compression first occurs.
2. A Swept-Power Gain Compression measurement shows the input power at which a in a 1-dB drop in gain occurs as a power ramp is applied to the amplifier at a particular frequency point (found in measurement 1).
3. An Absolute Power measurement shows the absolute power out (in dBm) at compression.

Swept-Frequency Gain Compression Measurement

A measurement of swept frequency gain compression locates the frequency point where 1-dB compression first occurs.

1. Preset the analyzer.
2. Select an S_{21} measurement parameter.
3. Set the analyzer's source power to be in the linear region of the amplifier's output response (typically 10-dB below the 1-dB compression point).
4. Select an external attenuator (if needed) so the amplifier's output power will be sufficiently attenuated to avoid causing receiver compression or damage to the analyzer's port-2.
5. Connect the amplifier as shown in the following graphic, and provide the dc bias.
6. Select the analyzer settings for your amplifier under test. To reduce the effects of noise, you may want to specify a narrower IF bandwidth.



7. Remove the amplifier and perform a thru-response calibration. Be sure to include the attenuator and cables in the calibration setup if they will be used when measuring the amplifier.
8. Save the instrument-state to memory.
9. Reconnect the amplifier.
10. Position a marker at approximately mid-span.
11. Adjust the analyzer's scale to 1 dB per division.
12. Store the trace in memory and display Data/Mem.
13. Gradually increase the source power until a 1-dB decrease in gain is observed at the first frequency over some portion of the trace.
14. Use markers to locate the frequency where the 1-dB decrease in gain first occurs. Note this frequency for use in the following measurement.
15. Print the data or save it to a disk.

Swept-Power Gain Compression Measurement

A swept-power gain compression measurement shows the input power resulting in a 1-dB drop in gain as a power ramp at a particular frequency (found in step 13 of the previous measurement) is applied to the amplifier.

1. If not already done, perform the previous measurement of swept-frequency gain compression.
2. Setup an S_{21} measurement in the power-sweep mode. Include the following settings:
   - Set the CW frequency to the frequency noted in step 14 of the previous measurement of swept-frequency gain compression.
   - Enter the start and stop power levels for the sweep. The start power should be in the linear region of the amplifier's response (typically 10 dB below the 1-dB compression point). The stop power should be in the compression region of the amplifier's response.
3. Adjust the scale to 1-dB per division.
4. Use markers (including reference marker) to find the input power where the 1-dB decrease in gain occurs.
5. Print the data or save it to a disk.

Absolute Output Power Measurement

An absolute-power measurement shows the absolute power-out (in dBm) of the amplifier at compression.

1. Select an unratioed (absolute) power measurement. Choose the B input if using the test setup in the previous graphic.
2. Retain the CW frequency used in the previous measurement of swept-power gain compression.
3. Set a marker to the input power level where the 1-dB decrease in gain occurs (found in step 4 of the previous measurement).
4. Scale the displayed measurement for optimum viewing.
5. Read the marker value to find the absolute output power of the amplifier (in dBm) where the 1-dB decrease in gain occurs.
6. Print the data or save it to a disk.

Note: The measurement calibration does not apply to absolute power. Therefore, if there is any attenuation external to the analyzer, you will have to correct for it manually.

Reverse Isolation

Reverse isolation is a measure of amplifier reverse transmission response- from output to input.

What is Reverse Isolation

Why Measure Reverse Isolation?

Accuracy Considerations

How to Measure Reverse Isolation

See other Tutorials

What is Reverse Isolation?

Reverse isolation is a measure of how well a signal applied to the device output is "isolated" from its input.

The measurement of reverse isolation is similar to that of forward gain, except:

- The stimulus signal is applied to the amplifier's output port.
- The response is measured at the amplifier's input port.

The equivalent S-parameter is S12.

Why Measure Reverse Isolation?

An ideal amplifier would have infinite reverse isolation-no signal would be transmitted from the output back to the input. However, reflected signals can pass through the amplifier in the reverse direction. This unwanted reverse transmission can cause the reflected signals to interfere with the desired fundamental signal flowing in the forward direction. Therefore, reverse isolation is important to quantify.

Accuracy Considerations

Since amplifiers often exhibit high loss in the reverse direction, generally there is no need for any attenuation that may have been used to protect the port 2 receiver during forward transmission measurements. Removing the attenuation will:

- Increase the dynamic range, resulting in improved measurement accuracy.
- Require a new calibration for maximum accuracy.

The RF source power can be increased to provide more dynamic range and accuracy.

Note: With the attenuation removed and the RF source power increased, a forward sweep could damage the analyzer's port 2 receiver. Do not perform a forward sweep or use 2-port calibration unless the forward power is set low enough to avoid causing port 2 receiver compression or damage.

If the isolation of the amplifier under test is very large, the transmitted signal level may be near the noise floor or crosstalk level of the receiver. To lower the noise floor:

- Use or increase measurement averages.

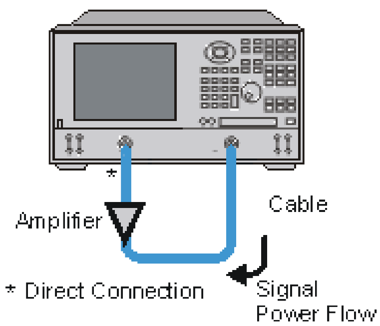
- Reduce the IF bandwidth of the analyzer.

Note: Reducing IF bandwidth or using averaging improves measurement dynamic range and accuracy, at the expense of reduced measurement speed.

- When crosstalk levels affect the measurement accuracy, reduce the crosstalk error term by performing a response and isolation calibration. When performing the isolation part of the calibration it is important to use the same average factor and IF bandwidth during the calibration and measurement.
- The frequency response of the test setup is the dominant error in a reverse isolation measurement. Performing a thru-response measurement calibration significantly reduces this error. This calibration can be done as part of the response and isolation calibration.
- The amplifier may respond very differently at various temperatures. The tests should be done when the amplifier is at the desired operating temperature.

How to Measure Reverse Isolation

1. Connect the amplifier as shown in the following graphic.



2. Preset the analyzer.
3. Select an S12 measurement.
4. Select the settings for your amplifier under test.
5. Remove the amplifier and perform a thru-response calibration or a response and isolation calibration.
6. Scale the displayed measurement for optimum viewing and use a marker to measure the reverse isolation at a desired frequency.
7. Print or save the data to a disk.

Absolute Output Power

An absolute output-power measurement displays absolute power versus frequency.

What is Absolute Output Power?

Why Measure Absolute Output Power?

Accuracy Considerations

How to Measure Absolute Output Power

See other Amplifier Parameters topics

What is Absolute Output Power?

An absolute-output power measurement displays the power present at the analyzer's input port. This power is absolute-it is not referenced (ratioed) to the incident or source power. In the log mag format, values associated with the grid's vertical axis are in units of dBm, which is the power measured in reference to 1 mW.

- 0 dBm = 1 mW
- -10 dBm = 100 µW
- +10 dBm = 10 mW

In the linear mag format, values associated with the grid's vertical axis are in units of watts (W).

Why Measure Absolute Output Power?

Absolute output power is measured when the amplifier's output must be quantified as absolute power rather than a ratioed relative power measurement. For example, during a gain compression measurement, it is typical to also measure absolute output power. This shows the absolute power out of the amplifier where 1-dB compression occurs.

Accuracy Considerations

The output power of the amplifier should be sufficiently attenuated if necessary. Too much output power could:

  - Damage the analyzer receiver
  - Exceed the input compression level of the analyzer receiver, resulting in inaccurate measurements.

Attenuation of the amplifier's output power can be accomplished using either attenuators or couplers

The amplifier may respond very differently at various temperatures. The tests should be done when the amplifier is at the desired operating temperature.

How to Measure Absolute Power

Do the following to measure absolute output power:

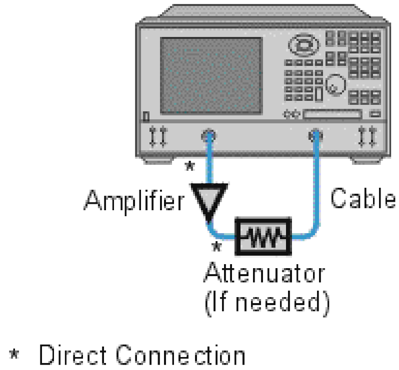
1. Preset the analyzer.

2. Select an unratioed power measurement (receiver B).

3. Set the analyzer's source power to 0 dBm.

4. Select an external attenuator (if needed) so the amplifier's output power will be sufficiently attenuated to avoid causing receiver compression or damage to the analyzer's port-2.

5. Connect the amplifier as shown in the following graphic, and provide the dc bias.



6. Select the analyzer settings for your amplifier under test.

7. Remove the amplifier and connect the measurement ports together. Store the data to memory. Be sure to include the attenuator and cables in the test setup if they will be used when measuring the amplifier.

8. Save the instrument state to memory.

9. Reconnect the amplifier.

10. Select the data math function Data/Memory.

11. Scale the displayed measurement for optimum viewing and use a marker to measure the absolute output-power at a desired frequency.

12. Print or save the data to a disk.

Time Domain

Time Domain allows you to view a device response as a function of time. The following are discussed in this topic:

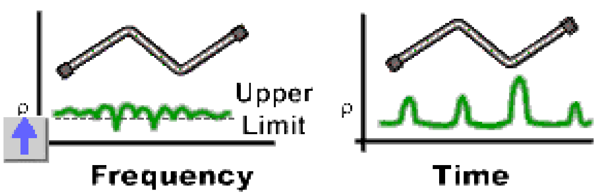
- Overview
- How the PNA Measures in the Time Domain
- Calibration for Time Domain
- Transmission Measurements
- Measurement Response Resolution
- Measurement Range and Alias Responses
- How to make Time Domain Settings
- Gating
- Window Settings

Note: Time Domain measurements are only available on PNAs with Option 010. See PNA Options

Overview

In normal operation, the PNA measures the characteristics of a test device as a function of frequency. With Time Domain (opt 010), the frequency information is used to calculate the inverse Fourier transform and display measurements with time as the horizontal display axis. The response values appear separated in time, allowing a different perspective of the test device's performance and limitations.

The graphic below compares the same cable reflection measurement data in both the frequency and time domain. The cable has two bends. Each bend creates a mismatch or change in the line impedance.



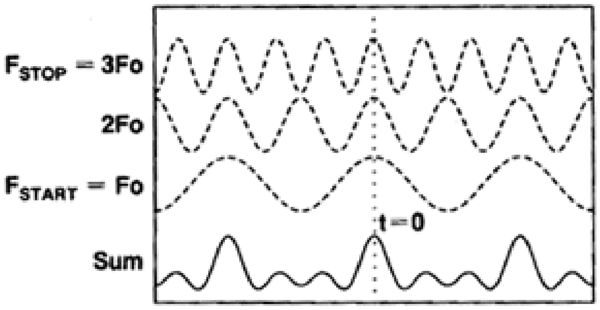
- The frequency domain S11 measurement shows reflections caused by mismatches in the cable. It is impossible to determine where the mismatches physically occur in the cable.
- The time domain response shows both the location and the magnitude of each mismatch. The responses indicate that the second cable bend is the location of a significant mismatch. This mismatch can be gated out, allowing you to view the frequency domain response as if the mismatch were not present. Distance Markers can be used to pinpoint the distance of the mismatch from the reference plane.

How the PNA Measures in the Time Domain

Time domain transform mode simulates traditional Time-Domain Reflectometry (TDR), which launches an impulse or step signal into the test device and displays the reflected energy on the TDR screen. By analyzing the magnitude, duration, and shape of the reflected waveform, you can determine the nature of the impedance variation in the test device.

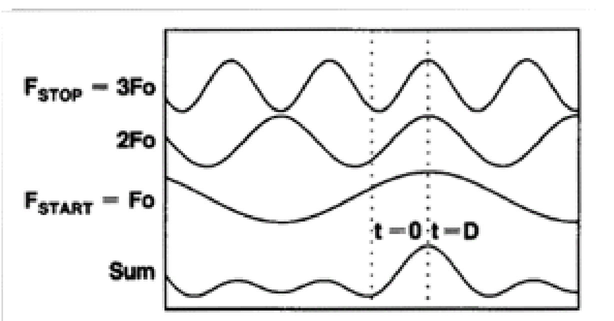
The PNA does not launch an actual incident impulse or step. Instead, a Fourier Transform algorithm is used to calculate time information from the frequency measurements. The following shows how this occurs.

A single frequency in the time domain appears as a sine wave. In the following graphic, as we add the fundamental frequency (F0), the first harmonic (2F0), and then the second harmonic (3F0), we can see a pulse taking shape in the Sum waveform. If we were to add more frequency components, the pulse would become sharper and narrower. When the PNA sends discrete frequencies to the test device, it is in effect, sending individual spectral pieces of a pulse separately to stimulate the test device.

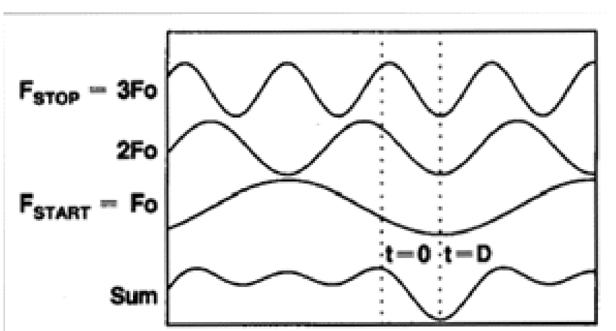


During an S11 reflection measurement, these incident signals reflect from the test device and are measured at the A receiver. This is when the time domain transform calculations are used to add the separate spectral pieces together.

For example, consider a short length of cable terminated with an open. All of the power in the incident signal is reflected, and the reflections are 'in-phase' with the incident signal. Each frequency component is added together, and we see the same pattern as the simulated incident would have looked (above). The magnitude of the reflection is related to the impedance mismatch and the delay is proportional to the distance to the mismatch. The x-axis (time) scale is changed from the above graphic to better show the delay.



Alternately, the same cable terminated with a short also reflects all of the incident power, but with a phase shift of 180 degrees. As the frequency components from the reflection are added together, the sum appears as a negative impulse delayed in time.



Calibration for Time Domain

For simplicity, we have discussed incident signals reflecting off discontinuities in the test device. By far the most common network analyzer measurement to transform to time domain is a ratioed S11 measurement. An S11 reflection measurement does not simply display the reflections measured at the A receiver - it displays the ratio (or difference) of the A receiver to the Reference receiver. In addition, the S11 measurement can also be calibrated to remove systematic errors from the ratioed measurement. This is critical in the time domain as the measurement plane, the point of calibration, becomes zero on the X-axis time scale. All time and distance data is presented in reference to this point. As a result, both magnitude and time data are calibrated and very accurate.

The following shows where the time domain transform occurs in the PNA data flow: (see graphic)

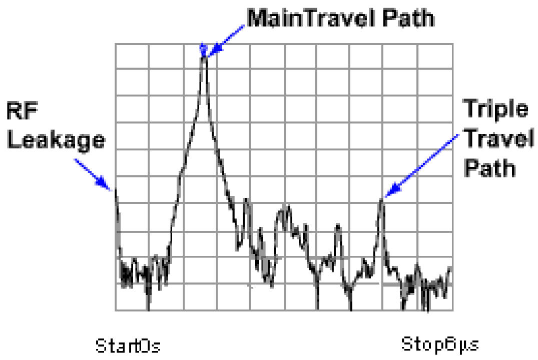
1. Acquire raw receiver (A and R1) data
2. Perform ratio (A/R1)
3. Apply calibration
4. Transform data to time domain
5. Display results

Therefore, although a time domain trace may be displayed, a calibration is always performed and applied to the frequency domain measurement which is not displayed.

Transmission Measurements

The most common type of measurement to transform is an S11 reflection measurement. However, useful information can be gained about a test device from a transformed S21 transmission measurement. The frequency components pass through the test device and are measured at the B receiver. If there is more than one path through the device, they would appear as various pulses separated in time.

For example, the following transmission measurement shows multiple paths of travel within a Surface Acoustic Wave (SAW) filter. The largest pulse (close to zero time) represents the propagation time of the shortest path through the device. It may not be the largest pulse or represent the desired path. Each subsequent pulse represents another possible path from input to output.



Triple travel is a term used to describe the reflected signal off the output, reflected again off the input, then finally reappearing at the output. This is best seen in a time domain S21 measurement.

Measurement Response Resolution

In the previous paragraphs, we have seen that using more frequency components causes the assembled waveform to show more detail. This is known as measurement response resolution, which is defined as the ability to distinguish between two closely spaced responses. The following settings also improve measurement response resolution.

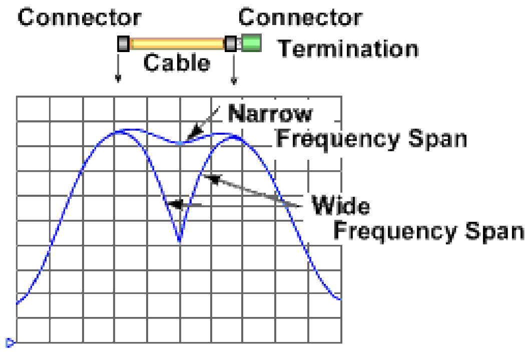
Note: Adjusting the transform time settings improves **display** resolution, but not measurement resolution.

Window Settings

Time Domain Mode

Frequency Span

The following graphic shows the effect of both a narrow and wide frequency span on the response resolution. The wider frequency span enables the analyzer to resolve the two connectors into separate, distinct responses.



Although a wider frequency span causes better measurement resolution, the measurement range becomes limited.

Increasing the frequency range can cause a measurement calibration to become invalid. Be sure to adjust the frequency span BEFORE performing a calibration.

Resolution Formula

You can calculate the ability of the PNA to resolve two closely spaced responses (in meters) using the following formula:

Resolution = (Vf * c * (points-1)) / (2 * Fspan * Ns)

Where:

Vf = Velocity Factor
c = velocity of light in a vacuum (2.99796x10^8 m/sec)
points = number of measurement points
Fspan = measurement frequency span
Ns = Sampling factor (**128** for 101 points; **256** for 201 points; **512** for 401 points)

Measurement Range and Alias Responses

Measurement range is the length in time in which true time domain responses can be seen. The measurement range should be large enough to see the entire test device response without encountering a repetition (alias) of the response. An alias response can hide a true time domain response.

To increase measurement range in both modes, change either of these settings:

- Increase the number of points
- Decrease the frequency span

Notes:

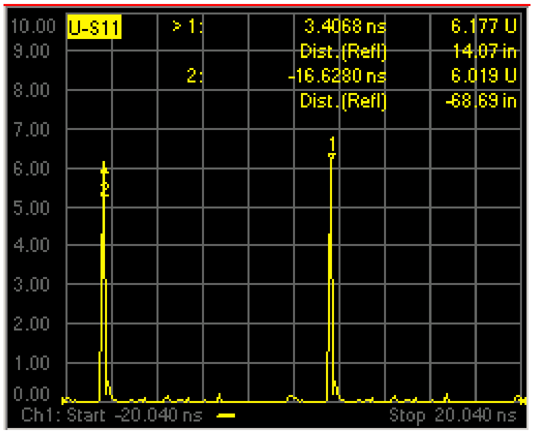
- After making these settings, you may need to adjust the transform time settings to see the new measurement range.

- Decreasing the frequency span degrades measurement resolution.
- Make frequency span and number of points settings BEFORE calibrating.
- Maximum range also depends on loss through the test device. If the returning signal is too small to measure, the range is limited regardless of the frequency span.

Alias Responses

An alias response is not a true device response. An alias response repeats because each time domain waveform has many periods and repeats with time (see How the PNA Measures in the Time Domain). Alias responses occur at time intervals that are equal to 1/ frequency step size.

The PNA adjusts the transform time settings so that you should only see one alias free range on either side (positive and negative) of zero time. However, these settings are updated only when one of the toolbar settings are changed.



To determine if a response is true, put a marker on the response and change the frequency span. A true device response will not move in time. An alias response will move.

For example, in the above graphic, the marker 1 response occurs at 14.07 inches. When the frequency span is changed, this response remains at 14.07 inches. The marker 2 response moves.

Range Formula

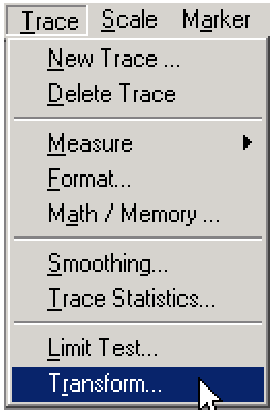
You can calculate the alias-free measurement range (in meters) of the PNA using the following formula:

Range = Vf * c * (points-1) / (2 * Fspan)

Where:

Vf = Velocity Factor
c = velocity of light in a vacuum (2.99796×10^8 m/sec)
points = number of measurement points
Fspan = measurement frequency span

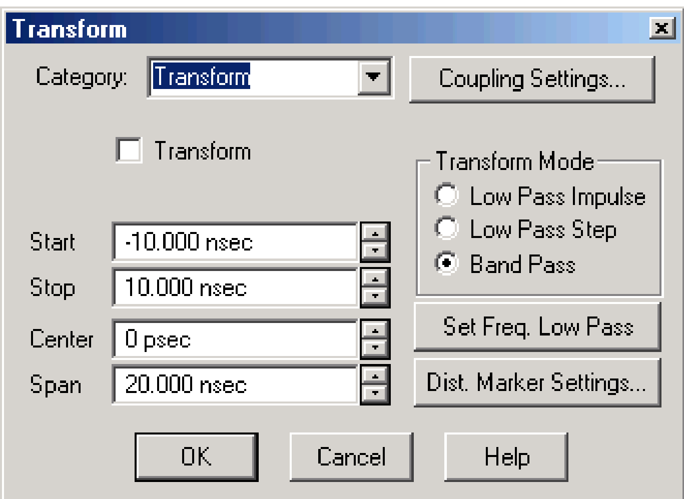
How to make Time Domain Settings



This launches the Time Domain toolbar

On the toolbar, click **More...** to launch the Time Domain dialog box

Learn more about using the front panel interface



Transform dialog box help

Category Select Transform, Window, or Gating

Transform Turns time domain transform ON and OFF.

Coupling Settings Launches the Trace Coupling Settings dialog box.

Time Settings

The following settings adjust the **display resolution**, allowing you to zoom IN or OUT on a response. They do NOT adjust measurement range or measurement resolution.

These settings automatically update (when one of these values are updated) to limit the display to one alias-free response on either side of zero time.

Start Sets the transform start time that is displayed on the PNA screen.

Note: Zero (0) seconds is always the measurement reference plane. Negative values are useful if moving the reference plane.

Stop Sets the transform stop time that is displayed on the PNA screen.

Center Sets the transform center time that is displayed in the center of the PNA screen.

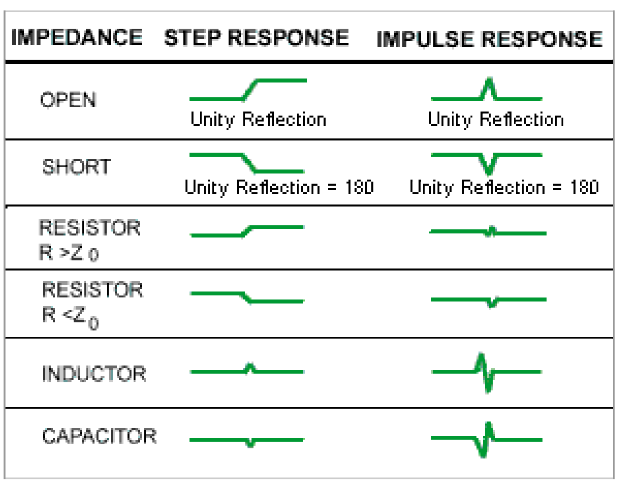
Span Sets the transform span time that is split on either side of the Center value.

Transform Mode

Transform modes are three variations on how the time domain transform algorithm is applied to the frequency domain measurement. Each method has a unique application.

Mode	Benefit - application	Limitation
Low pass Impulse	Highest resolution. Most useful for seeing small responses in devices that pass low frequencies, such as cables.	In both Low pass modes, frequencies down to DC and negative frequencies are extrapolated. Therefore, the Start frequency is adjusted when you click Set Freq.Low Pass Because this will affect calibration accuracy, be sure to calibrate AFTER completely setting up your time domain measurement.
Low pass Step	Easiest to identify inductive and capacitive discontinuities in devices that pass low frequencies, such as cables.	
Band pass Impulse	Easiest method - can be used with any frequency sweep. Most useful for measuring band limited devices such as filters and DC blocked cables.	Does NOT show capacitive and inductive reactance For the same frequency span and number of points, band pass mode has twice the impulse width, which hides closely spaced responses degrading the response resolution.

The following chart shows how to interpret results from various discontinuity impedances using Low pass Step and either Low pass or Band pass Impulse modes.



Effect on Measurement Range

Band pass mode - measurement range is inversely proportional to frequency step size.

Low pass mode - measurement range is inversely proportional to the fundamental (start) frequency AFTER clicking Set Freq. Low Pass.

Set Freq. Low Pass USE ONLY IN LOW PASS MODES

Recomputes the start frequency and step frequencies to be harmonics of the start frequency. Start frequency is computed by the following formula: **Low Pass Start Frequency = Stop Frequency / Number of points.**

The computed value must always be greater than or equal to the analyzer's minimum frequency.

Note: The number of points or stop frequency may be changed in order to compute this value.

Distance Marker Settings Launches the Distance Marker Settings dialog box.

Gating

Perhaps the most beneficial feature of time domain transform is the Gating function. When viewing the time domain response of a device, the gating function can be used to "virtually" remove undesired responses. You can then simultaneously view a frequency domain trace as if the undesired response did not exist.. This allows you to characterize devices without the effects of external devices such as connectors or adapters.

Note: When a discontinuity in a test device reflects energy, that energy will not reach subsequent discontinuities. This can "**MASK**", or hide, the true response which would have occurred if the previous discontinuity were not present. The PNA Gating feature does NOT compensate for this.

The following measurements images show a practical example how to use and perform gating. The test device is a 10inch cable, then a 6 dB attenuator, terminated with a short. The following four discontinuities are evident in window 2, from left to right:

1. A discontinuity in the test system cable which appeared after calibration. It is identified by marker 2 at -10.74 inches (behind the reference plane).
2. A discontinuity in the 10 inch device cable shortly after the reference plane.
3. The largest discontinuity is the attenuator and short shown by marker 1 at -12.67 dB (6 dB loss in both forward and reverse direction).
4. The last discontinuity is a re-reflection from the device cable.

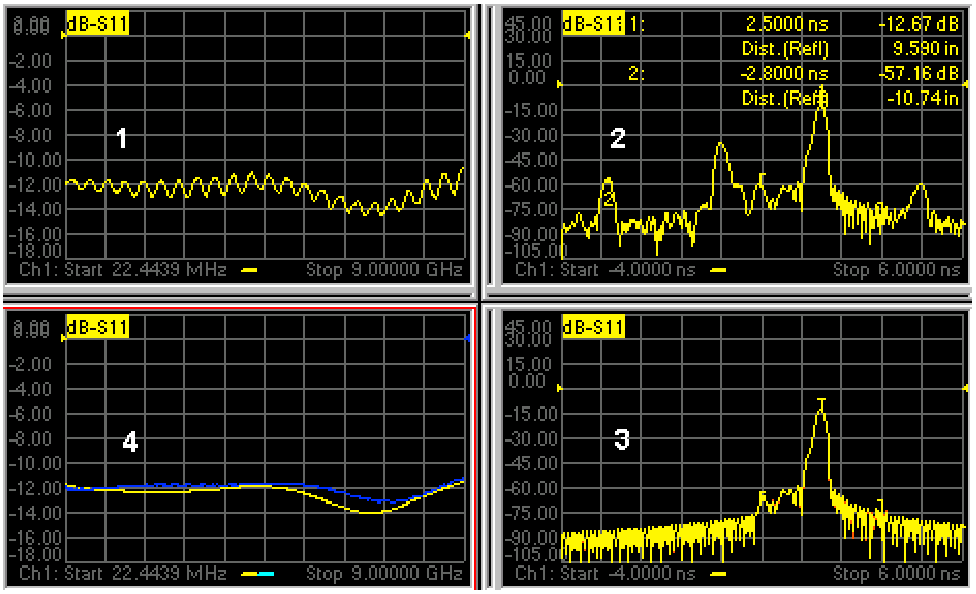
We will gate IN the attenuator response. All other responses will be gated OUT.

Window 1. Create original S11 frequency domain trace. Shows ripple from all of the reflections.

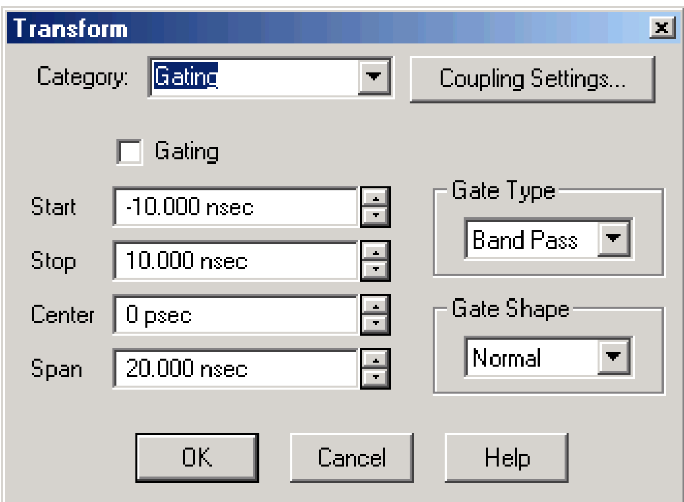
Window 2. Create a new S11 trace - same channel; new window. Turn Transform ON.

Window 3. On the transformed trace, turn gating ON. Center the gate on the large discontinuity (2.500ns). Adjust gate span to completely cover the discontinuity. Select Bandpass gating type.

Window 4. On the original frequency measurement, turn Gating ON (Transform remains OFF). View the measurement without the effects of the two unwanted discontinuities. The blue trace is a measurement of the 6 dB attenuator with the unwanted discontinuities PHYSICALLY removed. The difference between the two traces in window 4 is the effect of "masking".



Learn how to launch the Transform dialog box



Transform Gating dialog box help

Gating Turns Gating ON and OFF.

Coupling Settings Launches the Setup Trace Coupling dialog box.

Start Specifies the start time for the gate.

Stop Specifies the stop time for the gate.

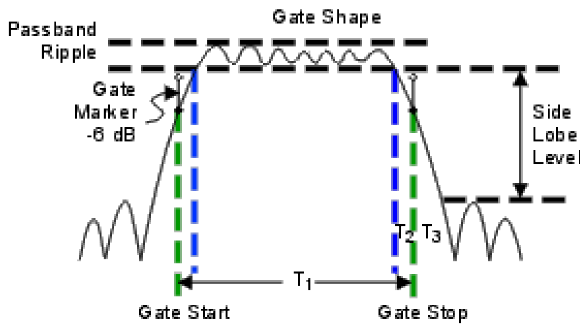
Center Specifies the value at the center of the area that is affected by the gating function. This value can be anywhere in the analyzer range.

Span Specifies the range to either side of the center value of area that is affected by the gating function.

Gate Type Defines the type of filtering that will be performed for the gating function. The gate start and stop flags on the display point toward the part of the trace you want to keep.

- **Bandpass -** KEEPS the responses within the gate span.
- **Notch -** REMOVES the responses with the gate span.

Gate Shape Defines the filter characteristics of the gate function. Choose from Minimum, Normal, Wide, Maximum



Gate Shape	Passband Ripple	Sidelobe Levels	Cutoff Time	Minimum Gate Span
Minimum	±0.1 dB	-48 dB	1.4/Freq Span	2.8/Freq Span
Normal	±0.1 dB	-68 dB	2.8/Freq Span	5.6/Freq Span
Wide	±0.1 dB	-57 dB	4.4/Freq Span	8.8/Freq Span
Maximum	±0.01 dB	-70 dB	12.7/Freq Span	25.4/Freq Span

Cutoff time -- is the time between the stop time (-6 dB on the filter skirt) and the peak of the first sidelobe. The diagram below shows the overall gate shape and lists the characteristics for each gate shape.

- T_1 is the gate span, which is equal to the stop time minus the start time.
- T_2 is the time between the edge of the passband and the 6 dB point, representing the cutoff rate of the filter.
- T_3 is the time between the 6 dB point and the edge of the gate stopband.
- For all filter shapes T_2 is equal to T_3, and the filter is the same on both sides of the center time.

Minimum gate span -- is twice the cutoff time. Each gate shape has a minimum recommended gate span for proper operation. This is a consequence of the finite cutoff rate of the gate. If you specify a gate span that is smaller that the minimum span, the response will show the following effects:

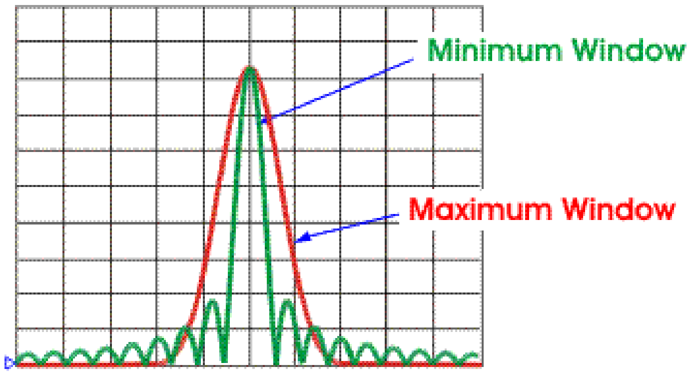
- distorted gate shape that has no passband
- distorted shape
- incorrect indications of start and stop times
- may have increased sidelobe levels

Window Settings

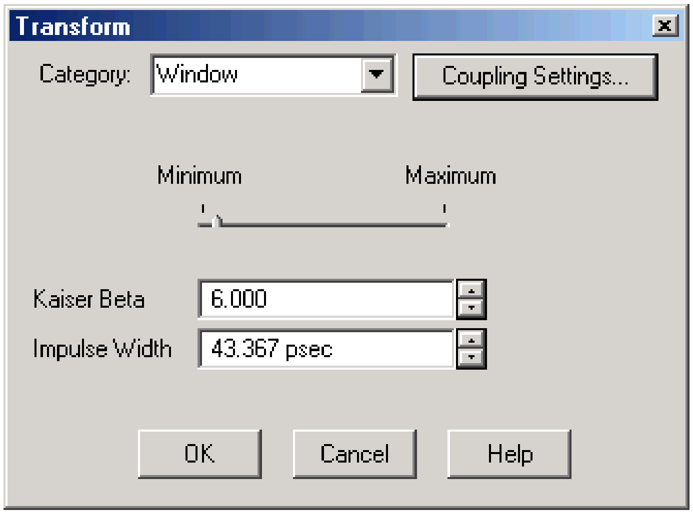
There are abrupt transitions in a frequency domain measurement at the start and stop frequencies, causing overshoot and ringing in a time domain response. The window feature is helpful in lessening the abruptness of the frequency domain transitions. This causes you to make a tradeoff in the time domain response. Choose between the following:

- **Minimum Window = Better Response Resolution** - the ability resolve between two closely spaced responses.

- **Maximum Window = Dynamic Range -** the ability to measure low-level responses.



Learn how to launch the Transform dialog box



Transform - Window dialog box help

Coupling Settings Launches the Setup Trace Coupling dialog box.

The window settings balance response resolution versus dynamic range.

- Minimum Window = Best Response Resolution
- Maximum Window = Best Dynamic Range

The following three methods all the set window size. For best results, view the time domain response while making these settings.

- **Minimum - Maximum** Move the slider with a mouse to change the window size
- **Kaiser Beta** Changes window size using a Kaiser Beta value
- **Impulse Width** Changes window size using an Impulse Width value

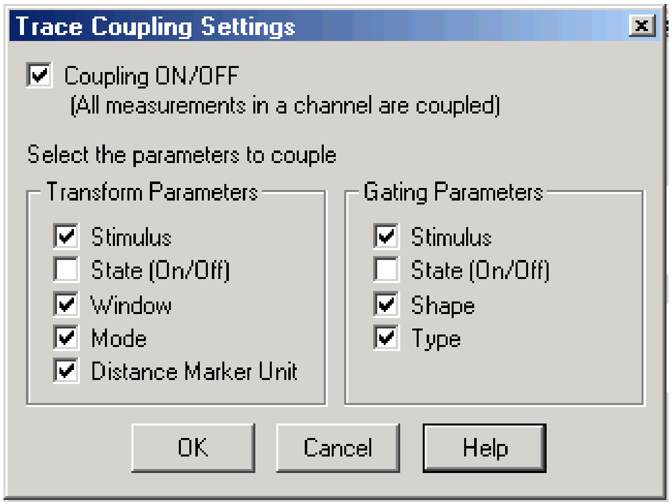
Learn more about Windowing (top)

How to make Trace Coupling Settings

You can launch the **Trace Coupling Settings** dialog box from any of the following dialog boxes:

- Transform
- Gating
- Window

Learn more about using the front panel interface



Trace Coupling Settings dialog box help

Trace coupling allows you to change time domain parameters on a measurement, and have the same changes occur for all other measurements in the channel.

For example:

If you are simultaneously viewing a frequency domain measurement and time domain measurement,

and **Coupling** is enabled in this dialog box,

and ALL **Gating Parameters** are checked in this dialog box,

and on the time domain measurement you change the **Gate Span** parameter,

Then the frequency domain measurement will automatically change to reflect the time domain gated span.

Coupling ON/OFF Check to enable coupling. All of the measurements in the active channel are coupled.

The following parameters are available for coupling:

Transform Parameters

Stimulus Start, Stop, Center, and Span TIME settings.

State (On/Off) Transform ON and OFF

Window Kaiser Beta / Impulse Width

Mode Low Pass Impulse, Low Pass Step, Band Pass

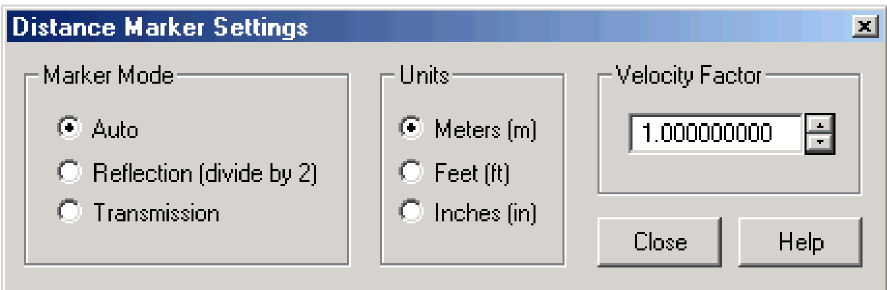
Gating Parameters

Stimulus Start, Stop, Center, and Span TIME settings.

State (On/Off) Gating ON and OFF

Shape Minimum, Normal, Wide, and Maximum

Type Bandpass and Notch



Distance Marker Settings dialog box help

To launch this dialog box, click **Dist. Marker Settings** on the Transform dialog box.

When markers are present on a time domain measurement, distance is automatically displayed on the marker readout, marker table, and print copy. To learn how to create markers on your measurement see marker settings.

This dialog box allows you to customize the time domain distance marker readings.

These settings affect the display of ALL markers for only the ACTIVE measurement (unless **Distance Marker Unit** is coupled on the Trace Coupling dialog box.

Marker Mode Specifies the measurement type in order to determine the correct marker distance.

- Select **Auto** for S-Parameter measurements.
- Select **Reflection** or **Transmission** for arbitrary ratio or unratioed measurements.

Auto If the active measurement is an S-Parameter, automatically chooses reflection or transmission. If the active measurement is a non S-Parameter, reflection is chosen.

Reflection Displays the distance from the source to the receiver and back divided by two (to compensate for the return trip.)

Transmission Displays the distance from the source to the receiver.

Units Specifies the unit of measure for the display of marker distance values.

Velocity Factor Specifies the numerical value related the speed of energy through transmission lines with different dielectrics. This is useful in Time Domain for accurate display of time and distance markers. Some common values are:

- 0.66 - Polyethylene dielectric cable
- 0.7 - Teflon dielectric
- 1.0 - The speed of light in a vacuum

This setting can also be made from the Electrical Delay and Port Extensions dialog boxes.

Frequency Offset

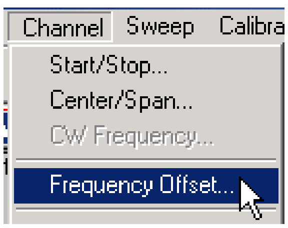
The PNA Option 080 provides you with the hardware and basic software to make Frequency Offset Measurements. This topic discusses the PNA settings that are relevant to making these types of measurements. See Frequency Converting Device Measurements for more information on making specific device measurements.

Note: The Frequency Converter Application Option 083 simplifies the task of making extremely accurate frequency offset measurements on MOST frequency converting devices.

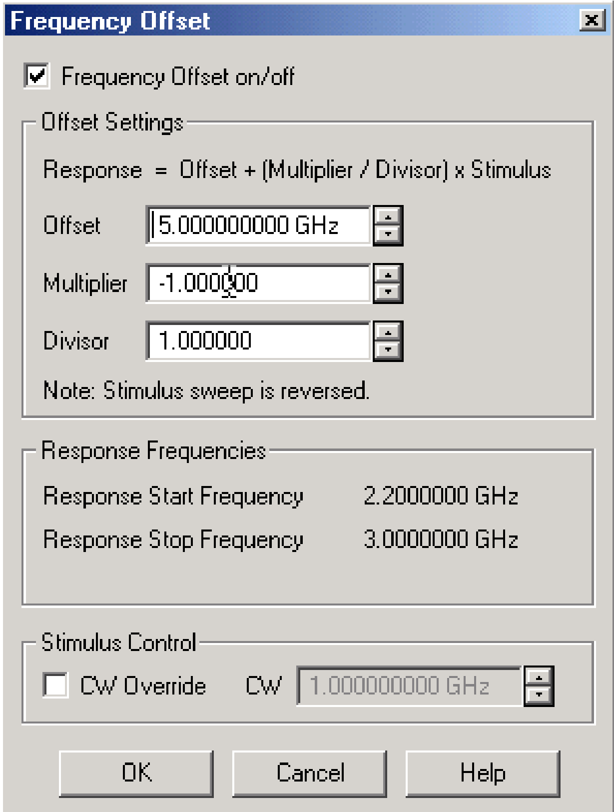
- **Frequency Offset Dialog Box**
- **Setup Examples**
- **Test Set (Reference Switch) Dialog Box**

Other Frequency Offset topics

How to access the Frequency Offset dialog



Learn more about using the front panel interface



Frequency Offset dialog box help

Note: With basic Frequency Offset measurements, Port 1 or Port 2 can be configured as either the source (stimulus) or receiver (response). This is NOT true with Option 083.

Frequency Offset on/off Enables Frequency Offset Mode on ALL measurements that are present in the active channel. This immediately causes the source and receiver to tune to separate frequencies. The receiver frequencies are specified in the remaining settings in this dialog box. To make the stimulus settings, on the **Channel** menu click **Start / Stop,** then **OK.**

Tip: To avoid unnecessary errors, first make other settings on this dialog box, then click **Frequency Offset** ON.

Offset Settings

The receivers tune to the frequency specified by the following formula:

Response = Offset +(Multiplier / Divisor) x Stimulus (or source)

Where:

Offset Specifies an absolute offset frequency in Hz. For mixer measurements, this would be the LO frequency. Range is +/- 1000 GHz. Offsets can be positive or negative. See an example.

Multiplier Specifies (along with the divisor) the value to multiply by the stimulus. Range is +/- 1000. Negative multipliers cause the stimulus to sweep in decreasing direction. For mixer measurements, this would be for setups requiring the Input frequency to be less than LO frequency. See an example..

Divisor Specifies (along with the multiplier) the value to multiply the stimulus. Range is 1 to 1000

Stimulus CW Override

Use this setting to establish a fixed (CW) stimulus frequency while measuring the Response over a swept frequency range. For example, the PNA stimulus may be applied to the RF input of a mixer whose local oscillator (LO) is being swept. Because the IF output of the mixer will be swept, the PNA receivers must also be swept. See an example.

To set up such an instrument state:

1. Set the Stimulus Sweep Type (Linear, Log, or Segment Sweep) and frequency settings as well as the Offset, Multiplier and Divisor to cause the desired Response Start and Stop frequencies.
2. Turn ON Frequency Offset - the receivers will now sweep over the desired frequencies.
3. Turn ON CW Override and set the desired CW frequency for the source.

Notes:

When Frequency Offset is enabled, ALL receivers on the channel tune to the offset frequency, including the reference receiver. Therefore S21 (conversion loss) measurements will not display the response frequency relative to the input frequency.

If you want to measure and display measurements at both the stimulus and response frequencies, you must use two channels. These types of ratioed measurements must be computed external to the PNA.

To calibrate for frequency offset measurements, see..Frequency Offset Calibration.

Using Power Sweep for Testing Mixers

To measure the gain compression of a mixer, you need to sweep the input power to the mixer. The input and output frequencies are fixed but offset from one another. To set the input and output frequencies of the mixer under test:

1. On the PNA menu, click **Sweep**, then **Sweep Type**. Select **Power Sweep**. Also, set the CW (input) frequency of the mixer under test.
2. Click **Channel**, then **Frequency Offset**. Set **Offset** to reflect the appropriate LO and response of the mixer under test.

Setup Examples

1. **Swept Stimulus** (1000 MHz to 1100 MHz)

   Swept Response (2500 MHz to 2700 MHz)

   Make the following settings:

   On the **Channel** menu click **Start / Stop**

   Start Frequency = 1000 MHz

   Stop Frequency = 1100 MHz

   In the **Frequency Offset** dialog:

   Offset = 500 MHz

   Multiplier = 2

   Divisor = 1

   Frequency Offset = ON

   The above settings cause the following response frequencies:

   Response = Offset + (Multiplier / Divisor) x Stimulus

   Response Start Frequency = 500 + [(2 /1) * 1000]

   = 2500 MHz

   Response Stop Frequency = 500 + [(2 /1) * 1100]

   = 2700 MHz

2. **Fixed Stimulus** (1000 MHz)

Swept Response (2500 MHz to 2700 MHz)

Make the following settings:

On the **Channel** menu click **Start / Stop**

Start Frequency = 2500 MHz

Stop Frequency = 2700 MHz

In the **Frequency Offset** dialog:

Stimulus CW Override: ON

CW Frequency: 1000 MHz

Offset = 0 MHz

Multiplier = 1

Divisor = 1

Frequency Offset = ON

---

3. **Swept Stimulus** (1000 MHz to 1100 MHz)

Fixed Response (2500 MHz)

Make the following settings:

On the **Channel** menu click **Start / Stop**

Start Frequency = 1000 MHz

Stop Frequency = 1100 MHz

In the **Frequency Offset** dialog:

Offset = 2500 MHz

Multiplier = 0

Divisor = 1

Frequency Offset = ON

The above settings cause the following response frequencies:

Response = Offset + (Multiplier / Divisor) x Stimulus

Response Start Frequency = 2500 + [(0 /1) * 1000]

= 2500 MHz

Response Stop Frequency = 2500 + [(0 /1) * 1100]

= 2500 MHz

---

4. **Swept Stimulus** (2500 MHz to 2600 MHz)

Swept Response (1000 MHz to 1100 MHz)

External LO (not supplied): 1500 MHz

Start Frequency = 2500 MHz

Stop Frequency = 2600 MHz

In the **Frequency Offset** dialog:

Offset = -1500 MHz

Multiplier = 1

Divisor = 1

Frequency Offset = ON

The above settings cause the following response frequencies:

Response = Offset + (Multiplier / Divisor) x Stimulus

Response Start Frequency = -1500 + [(1 /1) * 2500]

= 1000 MHz

Response Stop Frequency = -1500 + [(1 /1) * 2600]

= 1100 MHz

---

5. **Swept DECREASING Stimulus** (1100 MHz to 1000 MHz)

Swept INCREASING Response (1400 MHz to 1500 MHz)

External LO (not supplied): 2500 MHz

For some frequency-translating devices, as the input frequency increases, the output frequency decreases. The PNA can only display the response (output) as increasing. Therefore, the same effect can be obtained by sweeping the stimulus decreasing from higher to lower frequency. This causes an increasing response to be measured and displayed.

To accomplish this, set the multiplier to a negative value. This causes the PNA source to automatically sweep in reverse; from the higher frequency to the lower frequency.

Make the following settings:

On the **Channel** menu click **Start / Stop**

Start Frequency = 1000 MHz

Stop Frequency = 1100 MHz

In the **Frequency Offset** dialog:

Offset = 2500 MHz

Multiplier = -1

Divisor = 1

Frequency Offset = ON

The above settings cause the following response frequencies:

Response = Offset + (Multiplier / Divisor) x Stimulus

Response Start Frequency = 2500 + [(-1 /1) * 1000]

= 1500 MHz

Response Stop Frequency = 2500 + [(-1 /1) * 1100]

= 1400 MHz

The response cannot sweep backwards, so the PNA reverses the stimulus frequencies.

Stimulus Start Frequency = 1100 MHz

Stimulus Stop Frequency = 1000 MHz

Response Start Frequency = 1400 MHz

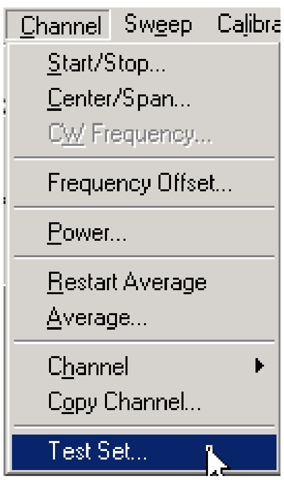
Response Stop Frequency = 1500 MHz

Test Set Reference Switch

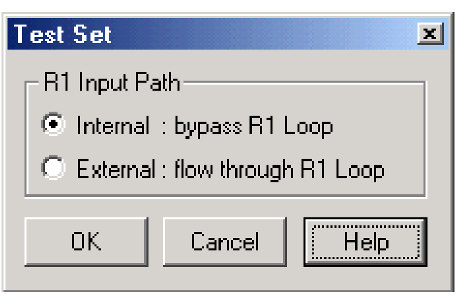
PNA models with option 081 have a switch in the test set that allows you to bypass the port 1 reference receiver through the front panel Reference 1 connectors. This switch lets you easily switch between standard S-Parameter measurements and measurements using a reference mixer. You could use this feature to make standard S11 measurements and converter transmission measurements relative to a reference ("golden") mixer.

Note: The Frequency Converter Application Option 083 simplifies the task of making extremely accurate phase measurements on MOST frequency converting devices.

How to access the Test Set dialog box



Learn more about using the front panel interface



Test Set dialog box help

Note: This feature is only available on PNA models with Option 081 - external reference switch.

R1 Input Path

Internal: bypass R1 Loop Connects the port 1 source directly to the R1 receiver.

External: flow through R1 Loop Allows direct access to the R1 receiver through the Reference 1 front-panel connectors.

See block diagram of reference switch.

Frequency Converting Device Measurements

Many frequency offset measurements can be made using the PNA with option 080. The following is a list of some of those measurements and how they are made.

- **Conversion Loss**
- **Conversion Compression**
- **Return Loss and VSWR**
- **Isolation**
- **Harmonic Distortion**

See Also: Frequency Offset Measurement Accuracy

Frequency Offset Measurement Accuracy

This topic discuss methods that can be used to make accurate frequency offset measurements.

- **Calibrations**
- **Mismatch Errors**
- **Accurate and Stable LO**

See other Mixer Measurement topics

Calibrations

With Frequency Offset measurements, the stimulus and response frequencies are different. Standard calibration error terms are calculated using reference measurements. Therefore, traditional calibration methods such as full 2-port SOLT cannot be used with frequency offset.

Frequency Converter Application (option 083) offers fully calibrated scalar and phase frequency offset measurements.

If you do not have option 083, Source and Receiver Power calibrations can be used to calibrate your Frequency Offset measurements.

Source Power calibration:

- Sets accurate power level at stimulus frequencies regardless of the receiver that will be used in the measurement.
- Can be copied to other channels with copy channels feature.
- Can be interpolated.

Receiver Power Cal:

- Requires a source cal to have already been performed and applied.
- Cannot be copied to other channels.

Therefore:

- Start by performing a source power cal over the combined stimulus and response frequencies.
- Copy the channel to other needed channels and the source power cal is copied.
- Change the frequency range of the copied channel to response frequency
- Perform a receiver cal at the response frequencies on individual channels.
- Change the frequency range to stimulus frequency and switch frequency offset ON.
- On Status Bar, ensure that source and receiver cals are ON (source cal will be interpolated).

See Conversion Loss Measurements for an example.

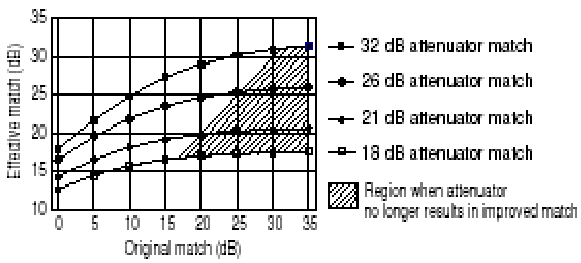
Learn more about source and receiver power calibrations.

Mismatch Errors

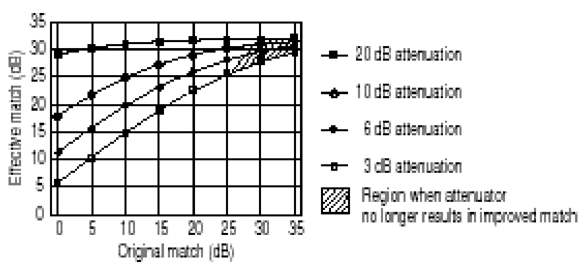
Mismatch errors result when there is a connection between two ports that have different impedances. With S-parameter measurements, these mismatches are measured and mathematically removed during a full 2-port calibration. This is much more difficult with frequency offset measurements. A much easier solution is to use high-quality attenuators on the input and output of the mixer.

By adding a high-quality attenuator to a port, the effective port match can be improved by up to twice the value of the attenuation. For example, a 10-dB attenuator, with a port match of 32 dB, can transform an original port match of 10 dB into an effective match of 25 dB. However, as the match of the attenuator approaches the match of the original source, the improvement diminishes.

Note: The Frequency Converter Application (option 083) uses calibration techniques that correct for mismatch errors.



The larger the attenuation, the more nearly the resulting match approaches that of the attenuator, as shown in the following graphic. However, excessive attenuation is not desired because that will decrease the dynamic range of the measurement system.



Accurate and Stable LO

When using frequency offset mode, if the LO signal is not accurate and stable, the output signal will not be at the expected response frequency. As a result, the output signal can fall on the skirts of the PNA receiver IF filter, or fall completely outside of the receiver filter passband.

Also, the LO power level is critical in mixer measurements. Be sure to monitor these power levels closely.

Conversion Loss (or Gain)

What is Conversion Loss?

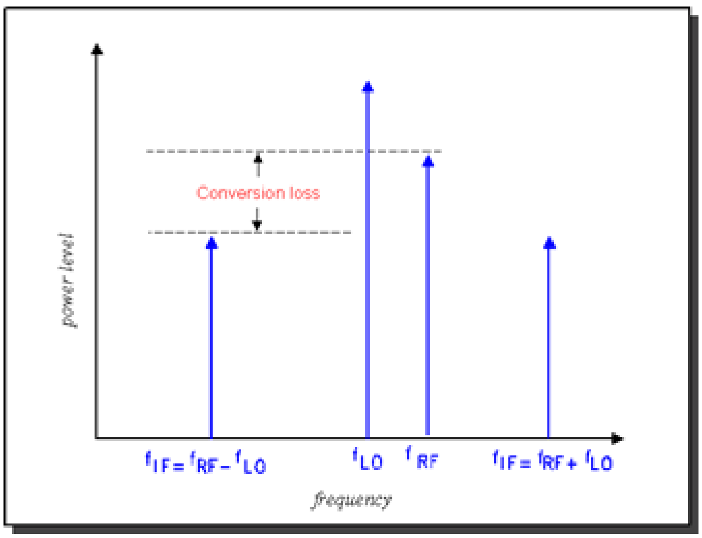
Why Measure Conversion Loss?

How to Measure Conversion Loss

See other Frequency Converting Device Measurements

What is Conversion Loss?

Conversion loss is defined as the ratio of the power at the output frequency to the power at the input frequency with a given LO (local oscillator) power. This is illustrated in the graphic below. A specified LO power is necessary because conversion loss varies with the level of the LO, as the impedance of the mixer diode changes.



Why Measure Conversion Loss?

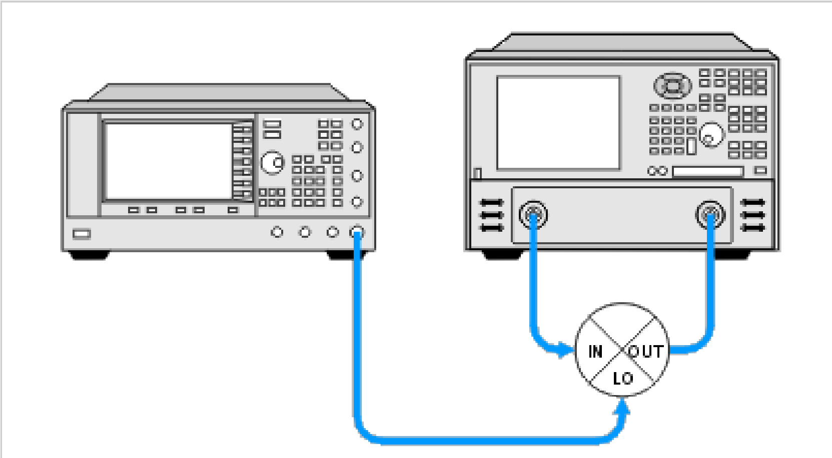
Conversion loss (or gain in the case of many converters and tuners) is a measure of how efficiently a mixer converts energy from the input frequency to the output frequency. If the conversion loss response of a mixer or converter is not flat over the frequency span of intended operation, valuable information may be lost from the resulting output signal.

How to Measure Conversion Loss

Conversion loss is a transmission measurement. It is measured by applying an input signal (stimulus) and an LO signal at specific known power levels, and measuring the resulting output signal level. Because the output frequency is different from the input frequency, frequency offset mode (option 080) must be used for this measurement.

Note: This measurement is made much easier if your PNA has the Frequency Converter Application

Equipment Setup



Example: A calibrated Conversion Loss (Down-converter) measurement

Swept Input with Fixed LO = Swept Output

- RF Input: 3.1 - 3.3 GHz
- LO: 2.2 GHz
- IF Output: 900 - 1100 MHz

PNA setup and calibrate on channel 1

- On channel 1 create an unratioed R measurement over the ENTIRE input and output frequency span (.9 - 3.3 GHz).   This will be the base source power cal that will be copied to the R and B channel measurements.
- Perform a source calibration using a power meter. This makes the power level at the input of the mixer very accurate.

Setup Reference measurement on channel 2

- Copy channel 1 to channel 2 which will display the reference input to the mixer. The channel 1 source power cal is copied with the other channel settings.
- Change measurement to R1 unratioed.
- Change RF Input frequency to 3.1 - 3.3 GHz. The source power cal becomes interpolated.
- Perform receiver power cal. Do not need to make physical connections. The PNA source is internally connected to the R1 receiver. Makes the R receiver read the source power level.

Setup B measurement on channel 3

- Copy channel 1 to channel 3. This channel will display the output of the mixer. The channel 1 source power cal is copied with the other channel settings.
- Change measurement to B unratioed.
- Change IF Output frequency to .9 - 1.1 GHz. This causes the source power cal becomes interpolated.

- Connect thru line from port 1 to port 2.
- Perform receiver power cal. This makes the B receiver read the source power at the IF Output frequencies.
- Turn OFF receiver power cal. This prevents an error when changing to input frequencies (next step).
- Change RF Input frequency to 3.1 - 3.3 GHz. This changes the channel back to the mixer RF Input frequencies.
- Enable Frequency Offset.
- Change Offset to (-2.2 GHz). This tunes the B receiver to the IF Output frequencies .9 to 1.1 GHz. **Note:** The minus sign indicates a down-converter measurement.
- Turn ON receiver power cal.

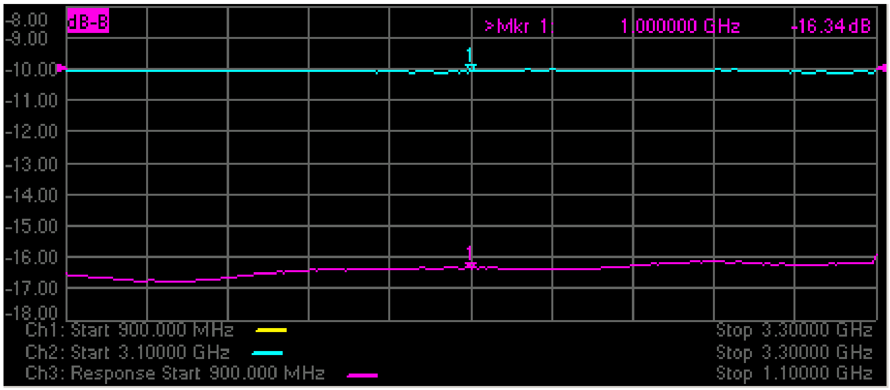
Measure the Mixer

- Connect the mixer.
- Adjust scaling to suit your needs.
- Enable markers to read power levels for each trace.

The display below shows:

- Ch3 B receiver (bottom trace) absolute output power.
- Ch2 R1 receiver measurement (top trace) absolute input power to the mixer.

With this method, the conversion loss math (B/R1) must be performed manually. Markers are set individually for each channel, and trace math is not supported between channels. Also, the B/R1 ratio measurement is not supported with receiver power Cal turned on. However, conversion loss (C21) measurements can be made directly and are much easier using the Frequency Converter Application, FCA (Opt 083).



Conversion Compression

What is Conversion Compression?

Why Measure Conversion Compression?

How to Measure Conversion Compression

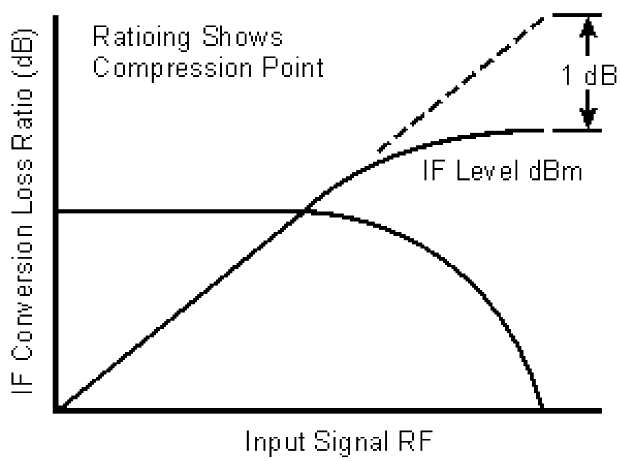
Measurement Accuracy Considerations

See other Frequency Converting Device Measurements

What is Conversion Compression?

Conversion compression is a measure of the maximum input signal level for which a mixer will produce linear operation. It is very similar to the gain compression experienced in amplifiers.

To understand conversion compression, you must first understand conversion loss. This is the ratio of the mixer output level to the mixer input level. This value remains constant over a specified input power range. When the input power level exceeds a certain maximum level, the constant ratio between input and output power levels begins to change. The point at which the ratio has decreased 1 dB is called the 1-dB compression point. This is illustrated in the graphic below.



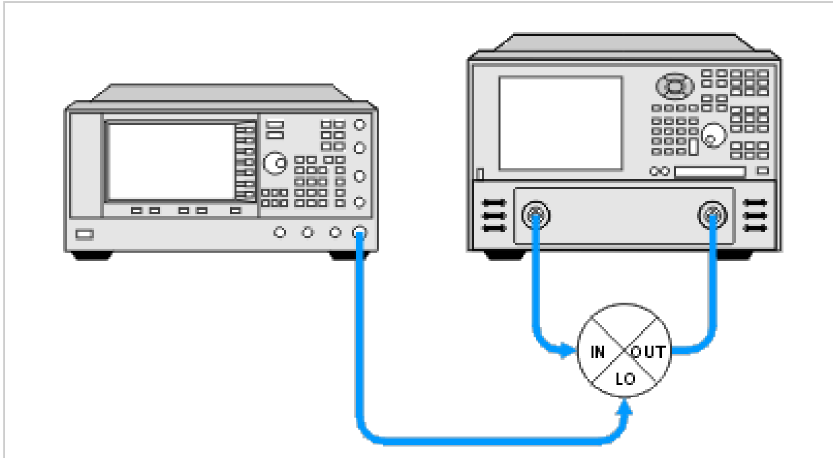
Why Measure Conversion Compression?

Conversion compression is an indicator of the dynamic range of a device. Dynamic range is generally defined as the difference between the noise floor and the 1-dB compression point.

How to Measure Conversion Compression

The equipment and setup used to measure conversion compression are essentially the same as for measuring conversion loss and is illustrated in the following graphic.

The PNA performs a power sweep using frequency-offset mode and the resulting display shows the mixer's output power as a function of its input power. The 1-dB compression point (or others such as 3-dB) can be determined using markers.



Measurement Accuracy Considerations

Equipment Setup Considerations

- The couplers in the PNA have very good directivity. If the return loss of the DUT is bad, the reflected signal gets sampled by the PNA and can result in errors. This relates to error in DUT gain. To increase the accuracy, an attenuator can be added between the PNA's source port and the DUT's input port. Normally a 6- to 10-dB attenuator is sufficient. Addition of this attenuator, however, decreases the available drive to the DUT.
- With high drive levels the PNA can be driven into compression resulting in measurement error. With excessive drive levels, the PNA can be damaged. Add an attenuator between the output of the DUT and the receiver input of the PNA to avoid these problems.

Calibration Considerations

- Source power calibration can be used to provide a high level of accuracy for this measurement.
- If your PNA has the Frequency Converter Application (option 083), you can perform a Scalar Mixer Calibration to obtain a more accurate measurement.

Isolation Measurements of Frequency Converting Devices

[What is Isolation?](#)

[Why Measure Isolation?](#)

[How to Measure Isolation](#)

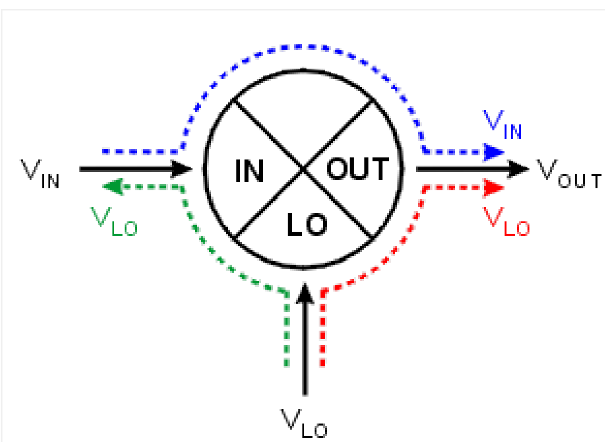
See other Frequency Converting Device Measurements

What is Isolation?

Isolation is a measure of the leakage, or feedthrough, from one port to another. The more isolation a mixer provides, the lower the amount of feedthrough. Isolation is measured at the same frequency as the stimulus, not the converted or shifted frequency. Therefore, Frequency Offset capability is not necessary for these measurements.

Three main isolation terms are of interest for mixer measurements:

- LO-to-OUT isolation ($\mathbf{V_{LO}}$)
- LO-to-IN isolation ($\mathbf{V_{LO}}$)
- IN-to-OUT feedthrough ($\mathbf{V_{IN}}$)



Why Measure Isolation?

Any unwanted signal "leaking" through the device will mix with the desired output signal creating intermodulation products, adding to intermodulation distortion. These unwanted signals may be difficult to filter out.

How to Measure Isolation

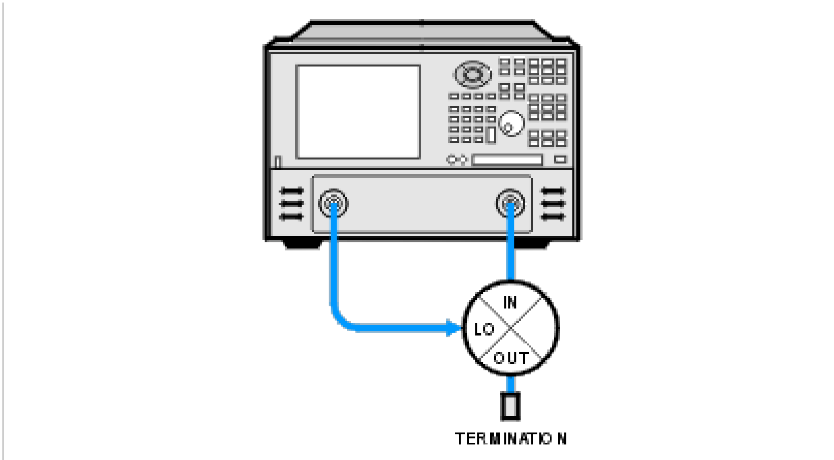
Use the following setups to measure the isolation of a mixer:

Note the following:

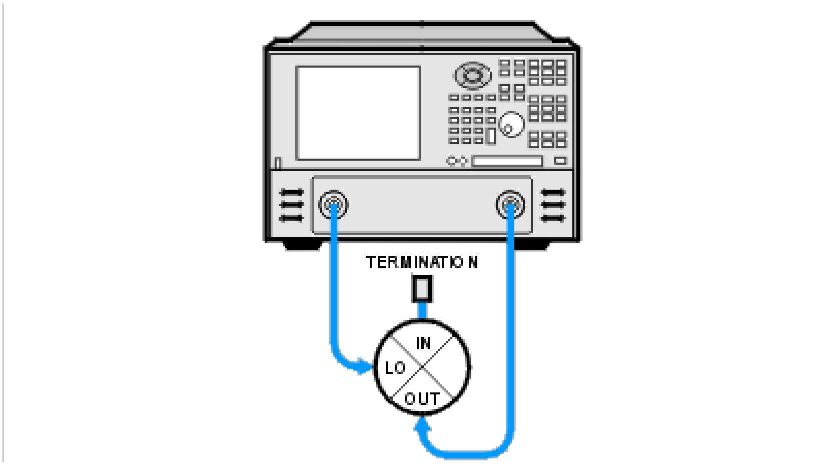
- The Input to Output isolation is very dependent on the LO power level. Isolation should be measured with the LO power at its normal operating level.

- Each of the ports not being tested should be terminated with an impedance typical of actual operation. This may not always be the characteristic impedance, Z0 (usually 50 or 75 ohms). For example, if the OUT port of a mixer is intended to be directly connected to a filter, then this filter should be used when measuring the LO-to-IN feedthrough.

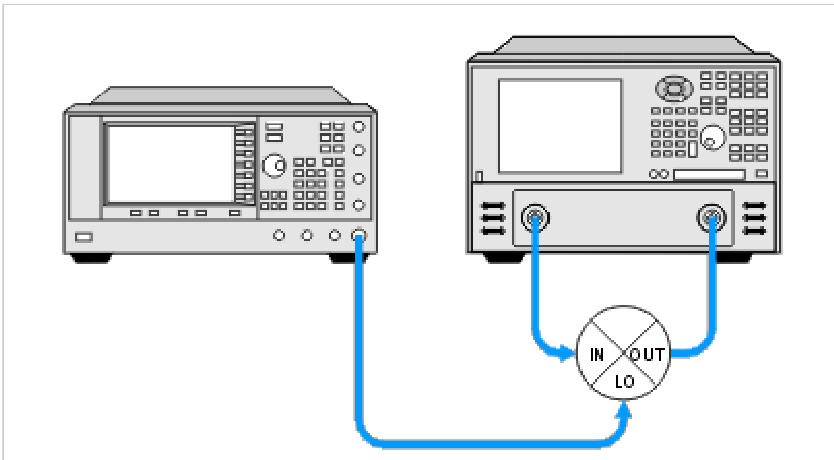
LO-TO-IN ISOLATION



LO-TO-OUT ISOLATION



IN-TO-OUT ISOLATION



Measuring Converters vs. Mixers

Measuring IN-to-OUT feedthrough of a converter is identical to that of a mixer. The IN-to-OUT feedthrough is generally very small for a converter due to the inclusion of an IF filter in the device. Because of this, the measurement may require the PNA to have increased dynamic range.

Measuring LO leakage (LO-to-OUT and LO-to-IN) of a converter requires a different technique because the LO port is typically not accessible:

- The PNA can be tuned to the frequency of the LO signal and either the OUT or IN port connected to the PNA receiver port. The PNA source port is not connected.
- A spectrum analyzer can be connected to either the OUT or IN port and tuned to the frequency of the LO signal.

Harmonic Distortion

What is Harmonic Distortion?

Why Measure Harmonic Distortion?

How to Measure Harmonic Distortion

Measurement and Accuracy Considerations

See other Frequency Converting Device Measurements

What is Harmonic Distortion?

Harmonics are multiples of any signal appearing at the mixer input and also multiples of the LO input. The distortion of the mixer's output characteristics caused by these harmonics is referred to as harmonic distortion. Harmonic distortion is caused by non-linearities in the device.

Harmonics are NOT signals created by two or more signals interacting (mixing); these signals are known as intermodulation products, which result in intermodulation distortion.

Why Measure Harmonic Distortion?

- It can degrade the performance of devices connected to the output of the mixer.
- The harmonics can also mix with other signals present in the mixer, adding to the intermodulation distortion of the mixer.

How to measure Harmonic Distortion

The harmonics can be measured using the PNA with Frequency Offset (option 80). The frequency of the LO to the mixer is set to zero and multiplier of the RF input is used to set the IF frequency (the harmonic). The equipment setup is shown below.

Since harmonics are specified in dBc, the fundamental RF and both the second and third harmonics are measured and the differences calculated. Multiple channels can be used to do this.

1. Connect the equipment.
2. Setup the measurement for calibration. See also Measurement and Accuracy Considerations.

   Use three channels and frequency offset mode:

   Channel 1 = F1 to F2

   Channel 2 = F1 to 2F2 (frequency offset mode, multiplier = 1)

   Channel 3 = F1 to 3F2 (frequency offset mode, multiplier = 1)

- Perform a source power calibration and receiver power calibration over the entire frequency range. See Measurement and Accuracy Considerations.

- Reduce the frequency span and increase the frequency offset multiplier on Channels 2 and 3:

  Channel 2 = F1 to F2 (frequency offset mode, multiplier = 2)

  Channel 3 = F1 to F2 (frequency offset mode, multiplier = 3)

  Note: Because the frequency span has been changed from that used for calibration, the source and receiver calibrations will be interpolated.

- Connect the DUT, make the measurement, and calculate the harmonic response:

  Set up markers on Channels 1, 2 and 3, and determine the difference between the marker values to get the dBc value of each harmonic.

  Channel 1 - Channel 2 = 2nd harmonic (dBc)

  Channel 1 - Channel 3 = 3rd harmonic (dBc)

  Note: Be sure to set the markers to the appropriate stimulus. Channel 2 markers should be set to twice the frequency of Channel 1 markers. Channel 3 markers should be set to three times the frequency of Channel 1 markers.

Measurement and Accuracy Considerations

Equipment Setup Considerations

- A filter must be used at the input of the mixer to remove the PNA source harmonics.

Calibrations

- If your PNA has the Frequency Converter Application (FCA), you can perform a Scalar Mixer Calibration to obtain a more accurate measurement.

Return Loss and VSWR

What are Return Loss and VSWR?

Why Measure Return Loss and VSWR?

How to Measure Return Loss and VSWR

See other Frequency Converting Device Measurements

What is Return Loss and VSWR?

Return loss and VSWR are both linear reflection measurements, even when testing frequency conversion devices, because the reflected frequency is not converted. These measurements are essentially the same as for filters and amplifiers. Learn more about Reflection Measurements.

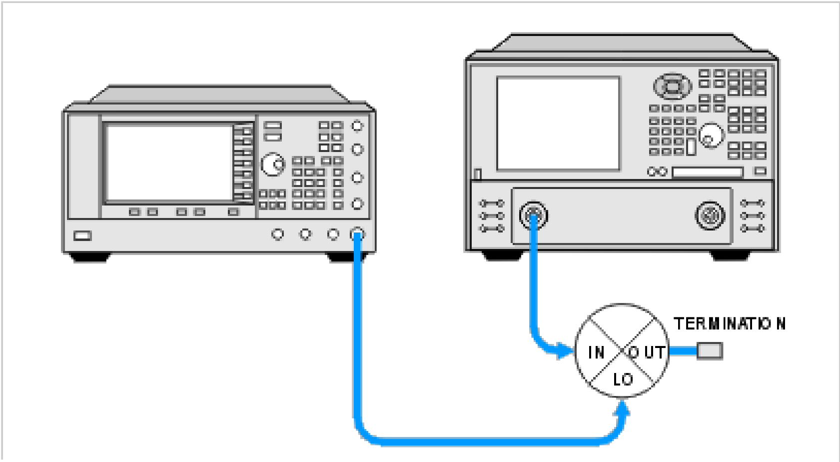
Why Measure Return Loss and VSWR?

Devices which have poor return loss and VSWR result in loss of signal power or degradation of signal information.

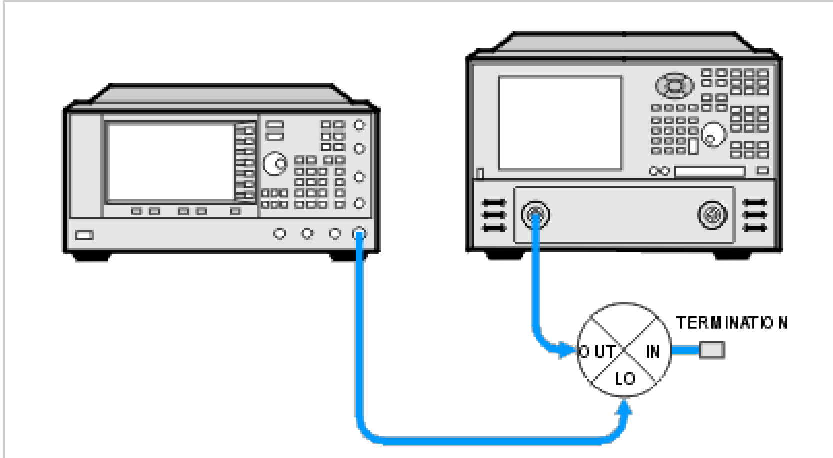
How to Measure Return Loss and VSWR

Setup the PNA measure return loss and VSWR as you would any two-port device. Connect your frequency converting device as shown in the following diagrams:

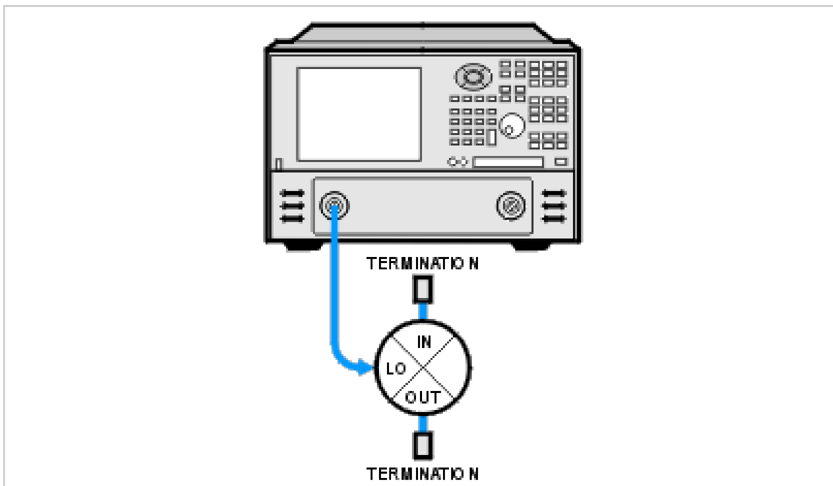
RETURN LOSS AND VSWR OF MIXER INPUT PORT



RETURN LOSS AND VSWR OF MIXER OUTPUT PORT



RETURN LOSS AND VSWR OF MIXER LO PORT



Frequency Converter Application Known Issues

To see the current list of known FCA issues, please visit http://na.tm.agilent.com/fca/ and click the known FCA issues link.

Frequency Converter Application

The Frequency Converter Application (Option 083) simplifies testing of frequency converting devices. It includes the following:

- Advanced calibration techniques that provide exceptional amplitude and phase accuracy.
- Control of external signal sources for use as local oscillators.
- A graphical set-up dialog box that lets you:
  - quickly set up the PNA for single or dual conversion devices.
  - calculate and choose where mixing and image products will fall.

For more information, see the following topics:

- **Using the FCA**
- **Configure Your Mixer**
- **FCA Calibrations**
- **Configure an External LO Source**

For a detailed understanding of FCA, visit www.Agilent.com and search for AN 1408-1, AN1408-2, and 1408-3.

Note: Please submit FCA issues that you find, as well as enhancement requests, to fca_support@agilent.com
See Known Issues with the FCA

Using the Frequency Converter Application (Option 083)

- Overview
- Create a Measurement
- Measurements Offered
- Scalar Measurement Setup
- Vector Measurement Setup
- FCA Measurement Settings
  - Change a Measurement
  - Speed Up Fixed Output SMC Measurements
  - Use Nominal Incident Power
  - Select X-axis Display
  - Save Trace Data
  - Avoid Spurs

Note: Please submit FCA issues that you find, as well as enhancement requests, to fca_support@agilent.com (See Known FCA Issues.)

Not sure if your analyzer is equipped with Option 083? Here's how to identify your analyzer.

Other Frequency Converter Application topics

Overview

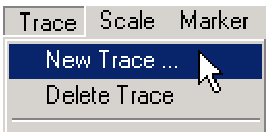
The following is an overview of how to make an FCA measurement:

1. DECIDE to make either a Scalar measurement or Vector measurement. The calibration method is unique to each of these. See a comparison of these two measurement types.
2. CREATE a default Conversion loss measurements: Learn how.
3. CHANGE the measurement if you want to measure something other than S21. Learn the various FCA measurements that are offered and how to CHANGE an FCA measurement.
4. CALIBRATE your Scalar or Vector measurement.

Note: An FCA measurement and a non-FCA measurement can NOT reside on the same channel.

How to CREATE a Frequency Converter Application Measurement

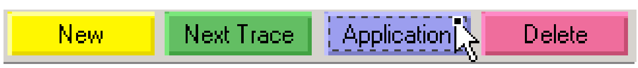
Use any of the following methods to open the Choose Application Measurement dialog box:

- 

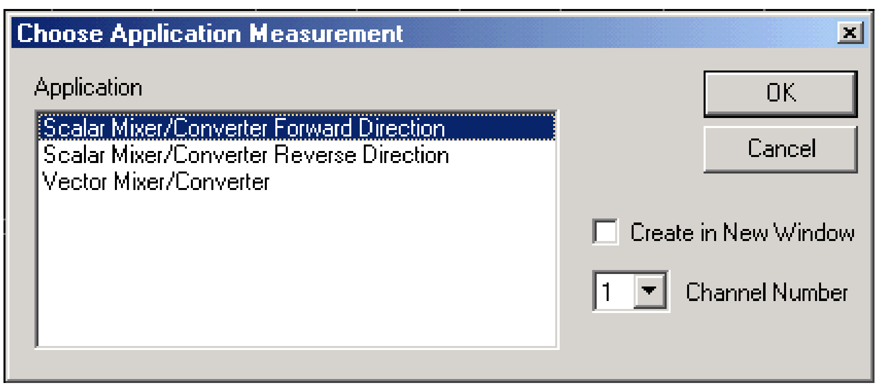
then 

OR

- 

+ 

Then select one of the following default conversion loss measurements:



Note: Each of these three measurements must be created in a separate channel. They can NOT share the same channel.

Learn to make this selection using the front panel interface

FCA Measurements that are Offered

Learn how to change from the default measurement.

Note: With PNA revision 4.0 you can do **FIXED OUTPUT** measurements.

Learn how to speed up your fixed output measurements.

Scalar Mixer/Converter (Forward Direction)

- **SC21 Conversion Loss/Gain (default)**
- **S11** at Input frequency

- **S22** at Output frequency
- **Incident Power** Calibrated Incident power to the DUT. If not calibrated, then absolute R1 power at Input frequency - stimulus at Port 1
- **Output Power** Calibrated Output power from the DUT. If not calibrated, then absolute B receiver power at Output frequency - stimulus at Port 1

Scalar Reverse Direction

- **SC12 Conversion Loss/Gain (default)**
- **S11** Output frequency
- **S22** Input frequency
- **Incident Power** Calibrated Incident power to the DUT. If not calibrated, then absolute R2 power at Input frequency - stimulus at Port 2
- **Output Power** Calibrated Output power from the DUT. If not calibrated, then absolute A receiver power at Output frequency - stimulus at Port 2

Vector (Forward Direction Only because of the use of a reference mixer)

- **VC21 Conversion Loss/Gain (default)**
- **S11**
- **S22**
- **R1 Power** Absolute R1 power at Output frequency going through the reference mixer path - stimulus at Port 1. Not calibrated.
- **B Power** Absolute B receiver power at Output frequency - stimulus at Port 1. Not calibrated.

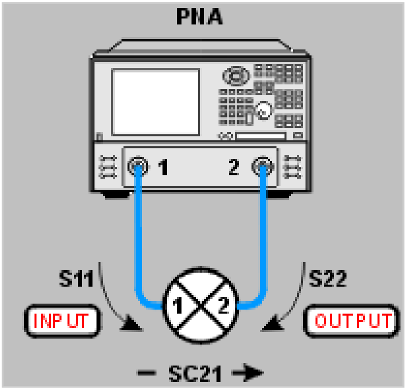
Understanding Scalar FORWARD and REVERSE Mixer measurements

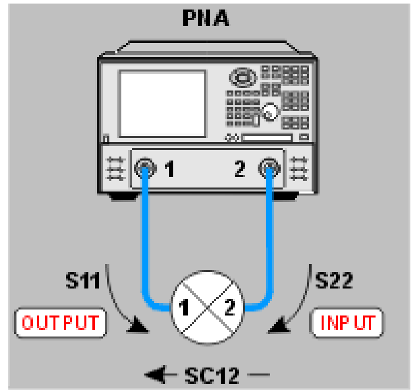
Mixers are typically reciprocal devices (S21 = S12) and are therefore used and measured in both directions. To help avoid confusion, the following diagrams do not use the traditional mixer labels RF and IF. They are labeled ports 1 and 2. The LO port is not relevant to this discussion.

During Calibration you specify **Input, Output,** or **Both** directions. Learn more.

Input and **Output** correspond to frequency settings in the Mixer setup dialog box.

FORWARD **REVERSE**

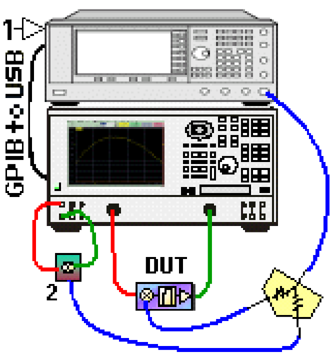




Note:

A and R1 are port 1 receivers
B and R2 are port 2 receivers

Vector Measurement Setup



1: External source (fixed LO signal)

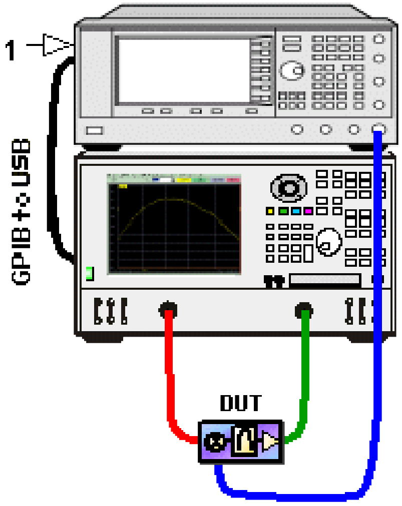
2: Reference mixer

Learn how to calibrate a Vector Mixer measurement.

Connect GPIB equipment to the PNA using either of the following:

- the Agilent 82357A USB/GPIB Interface
- the GPIB Interface. To do this, first put the PNA in System Controller mode. While in this mode, the PNA cannot be a talker / listener over GPIB.

Scalar Measurement Setup



1: External source (fixed LO signal)

Learn how to calibrate a Scalar Mixer measurement

Connect GPIB equipment to the PNA using either of the following:

- the Agilent 82357A USB/GPIB Interface
- the GPIB Interface. To do this, first put the PNA in System Controller mode. While in this mode, the PNA cannot be a talker / listener over GPIB.

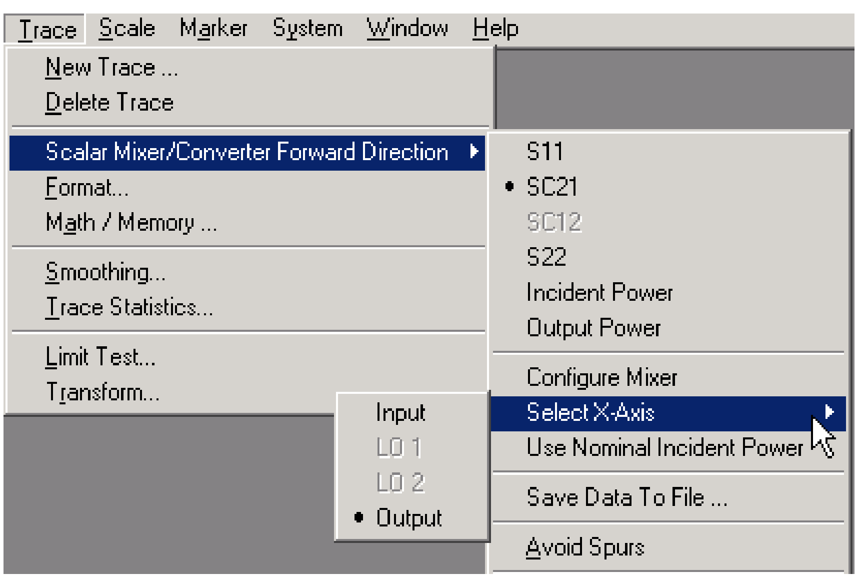
FCA Measurement Settings

Most of the FCA measurement settings in the remainder of this topic are made using the following menu selection. The choice will be slightly different depending on the active FCA measurement.

How to select several FCA measurement settings

First create an FCA measurement. Then:

(Click on links to learn more)

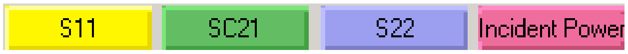


Learn to make this selection using the front panel interface

How to CHANGE a Frequency Converter Application Measurement

Select an FCA measurement. Then:

- 

+ 

- or use the previous **Trace** menu selection

Learn about FCA Measurements that are offered.

Speeding Up Fixed Output Measurements

Fixed output measurements require that an external LO source step in frequency. This can be extremely slow depending on your measurement setup. The following features can significantly speed up your fixed output measurement:

- BNC External LO trigger method
- Cal Set Application Choice - Select **SMC Rsp** (SMC Only)
- Use Nominal Incident Power - (SMC Only)

Learn how to configure Fixed Output measurements.

Use Nominal Incident Power

Each data sweep of a fully corrected SC21 measurement actually requires THREE data sweeps. Two of the sweeps are not displayed. When you select Use Nominal Incident Power, the reference receiver (R1 or R2) does not measure incident power. Instead, the incident power is assumed to be at the level that was set with the Source Power Calibration that is done as part of every SMC measurement. The degradation in accuracy is very negligible.

This selection eliminates a sweep ONLY when either:

- Output Power is measured or
- SMC Response Cal is applied.

This selection applies to all SMC measurements.

See how to select **Use Nominal Incident Power.**

Select X-axis Display for FCA Measurements

FCA measurements typically have more than one swept parameter. You can choose to view the response (output) of the measurement on the Y-axis while displaying any of the swept parameters (Input, LO1, LO2, Output) on the X-axis of the PNA display.

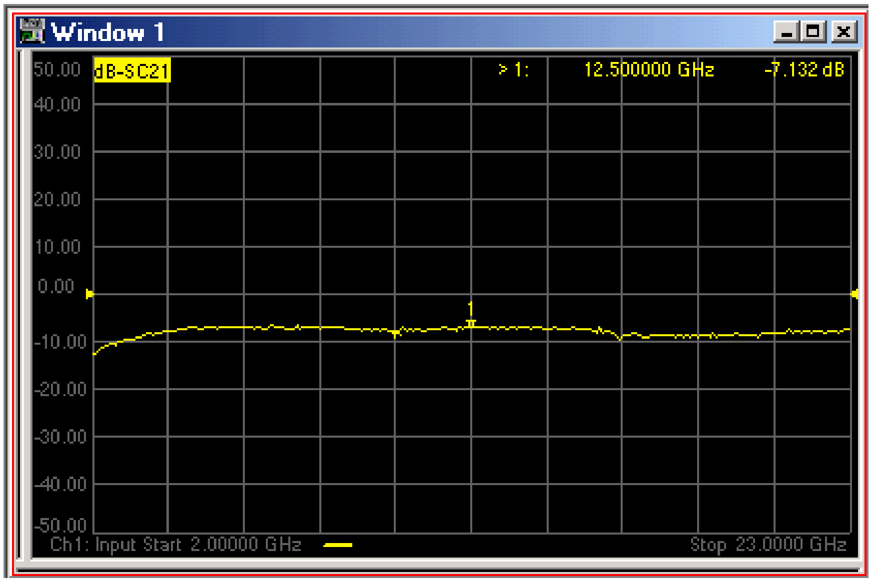
For example, the following image shows an SMC Forward Fixed Output response versus the swept Input.

Output: 100 MHz (data trace)

Input: 2 GHz to 23 GHz (X-axis)

LO: 1.9 GHz to 22.9 GHz (not shown)

Marker annotation shows Output power at Input frequency.



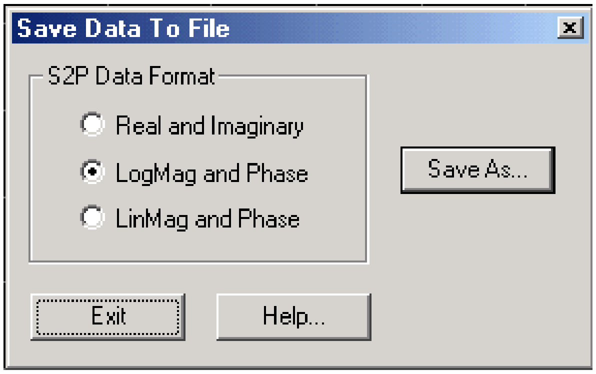
See How to Select X-axis Display

Save Trace Data

You can save your Frequency Converter measurement data in S2P format to disk.

Note: This is the only method to save Frequency Converter .S2P files.
Do NOT click **File, Save As**... to save these S2P data files.

See How to Save Trace Data



Save Data to File dialog box help

Allows you to save Frequency Converter measurement data to an S2P file. The data is saved in S2P format much like standard PNA data. Learn more about .S2P files.

Note: This is the only method to save Frequency Converter .S2P files.
Do NOT click **File, Save As**... to save these S2P data files.

S2P Data Format Select the data format. This selection is independent of the PNA display.

Save As Click to specify a file name and location for the saved data.

Exit Closes the dialog box without saving the data. To save the data, you must click on the Save As button before clicking the Exit button.

Notes:

Each record contains 1 stimulus value and 4 parameters (total of 9 values) as follows:

Stim Real(p1) Imag(p1) Real(p2) Imag(p2) Real(p3) Imag(p3) Real(p4) Imag(p4)

where **pX** is the parameter depending on measurement type:

Measurement Type	p1	p2	p3	p4
Scalar	S11	SC21 (FWD)	SC12 (REV)	S22
Vector	S11	VC21	VC12	S22
Mixer Characterization	Directivity	Source Match	Reflection Tracking	M21

- If correction is OFF, data is only saved for the active parameter. Zeros are saved for all other parameters.
- If correction is ON, data is saved for all of the parameters.

All files contain the following Header Information: Brackets [ ] contain parameters.

```
!Agilent [Instrument Model Number]: [version]
!Mixer S2P File: [Mixer Measurement Type]
!Parameters: [Parameter List]
!Calibration State: [On/Off]

!# Begin Mixer Setup
![Mixer Setup parameters listed here]
![Mixer Parameter 1]
.
.
![Mixer Parameter n]
!# End Mixer Setup

# [S2P data here]
```

Avoid Spurs

The Avoid Spurs feature of the Frequency Converter Application attempts to prevent unwanted mixing products from appearing on the PNA screen. The Avoid Spurs feature does not significantly impact measurement speed.

Note: The Avoid Spurs feature is OFF by default for FCA calibrations. For highest accuracy, make measurements with the Avoid Spurs feature at the same state (ON or OFF) as was used when calibrating.

See how to Avoid Spurs OR Check **Avoid Spurs** on the Mixer Setup dialog box.

Description

A spur, or spurious signal, is a term used to describe the unwanted product of two signals mixing together. When you configure the mixer setup dialog box for a desired Output, the PNA computes the frequencies of potential unwanted signals. By manipulating internal PNA hardware, these signals are avoided and do not appear on the PNA display. This means you do not need to use external filters to prevent spurious signals from appearing on the PNA display.

The time required for the PNA to compute the frequencies of unwanted spurious signals MAY be noticeable depending on the number of data points in your measurement. However, once computed, the time required for the PNA to avoid the spurs is usually insignificant.

Limitations

The Avoid Spurs utility cannot avoid every spur. However, when there is a choice of spurs to avoid, it will avoid the largest spur.

The Computation of Avoided Spurs

The Avoid Spur computer avoids the following spurs:

- LO, and its interaction with internal PNA components, and 16 of its harmonics.
- Input frequencies and 16 of its harmonics.
- Undesired Image frequencies (Sum or Difference) and 16 of its harmonics.

Frequency Converter Application (Option 083) Calibrations

Frequency Converter Application (Option 083) offers two advanced calibration choices for mixer or converter measurements that provide exceptional amplitude and phase accuracy.

- **Comparison of Scalar and Vector Mixer Cals**
- **SMC Setup and Overview**
- **VMC Setup and Overview**
- **FCA Calibration Process Diagram**
- **How to Perform an FCA Calibration**
- **Apply an FCA Cal Set**

Not sure if your analyzer is equipped with Option 083? Here's how to identify your analyzer.

To learn more, see www.Agilent.com and search for "Mixer Transmission Measurements Using The Frequency Converter Application (AN 1408-1)"

Please submit FCA issues that you find, as well as enhancement requests, to fca_support@agilent.com (See Known FCA Issues.)

Other Frequency Converter Application topics

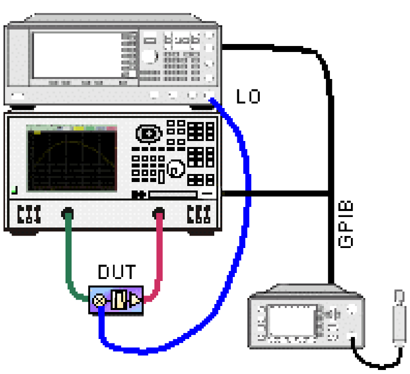
Comparison of Scalar and Vector Mixer Cals

	Scalar Mixer Calibration	**Vector Mixer Calibration**
Overview	Provides highest Scalar (amplitude only) accuracy for measurements of conversion loss/gain. Combines SOLT and power-meter calibration. Simpler setup than Vector Mixer Calibration.	Provides unparalleled accuracy for measurements of relative phase and absolute group delay. Uses combination of SOLT standards and a reciprocal mixer/filter pair during calibration. More complicated setup and calibration procedure than Scalar Mixer Calibration. After calibration, both reciprocal and non-reciprocal mixers and converters can easily be measured.
Types of Transmission Measurements	Can test in both forward (SC_{21}) and reverse (SC_{12}) directions.	Amplitude response VC_{21} Phase response Group delay

Options and Equipment Required	Option 014 Option 080 Option 083 Mechanical cal kit or ECal module Power meter and sensor Agilent 82357A USB/GPIB Interface (included with Frequency Converter Application, Option 083)	Option 014 Option 080 Option 081 Option 083 Mechanical cal kit or ECal module Calibration mixer/filter combination (must be reciprocal $S_{21} = S_{12}$.) Reference mixer/filter combination External source Agilent 82357A USB/GPIB Interface (included with Frequency Converter Application, Option 083)

Note: Find more information comparing these two mixer calibration types at http://www.tm.agilent.com. Click "Library" and use the search function to find the white-paper titled "Comparison of Mixer Characterization Using New Vector Characterization Techniques".

SMC Calibration Setup



Connect external source and power meter to the PNA GPIB using either:

- The Agilent 82357A USB/GPIB Interface - **highly recommended -** allows for the use of a remote PC to control the PNA.
- The standard GPIB Interface - with the following limitations:
  - The PNA cannot be controlled remotely as talker / listener over GPIB. First put the PNA in System Controller mode. Learn how.
  - Available only on PNA releases 4.2 and later.

Learn how to Configure an External LO Source

Overview of the Scalar Mixer Calibration.

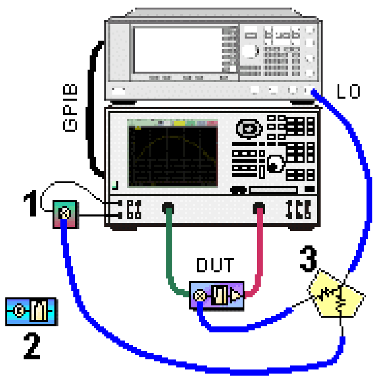
The Calibration Wizard guides you through this process.

1. Perform two 2-port calibrations: one over the INPUT frequencies and one over the OUTPUT frequencies of the DUT. (If your DUT is a linear device, the calibration uses only the INPUT frequency range.) Use either a mechanical calibration kit or an ECal module.

2. Connect a power meter sensor to PNA Port 1. At each step of the input and output frequency, the PNA measures:

    - input match of the power sensor
    - source power of the PNA

VMC Calibration Setup



1. Reference mixer whose signal is synchronously converted with the device under test. This means the reference mixer and the mixer or converter-under-test use LO signals in a constant frequency and phase relationship.

- Connect the Reference Mixer INPUT to PNA **Ref 1 Source out**
- Connect the Reference Mixer OUTPUT to PNA **Rcvr R1 In**

2. Calibration mixer/filter combination that meets the following requirements:

- The mixer must be reciprocal. This means that it has the same magnitude and phase response in the up-converting and down-converting directions ($C_{21} = C_{12}$).
- If the Input and Output frequency ranges are overlapping, the mixer must have Input to Output Isolation greater than 10 dB more than the conversion loss in the overlapping range.
- The filter must reject the undesired mixing product at the output of the cal mixer. For example, if you have selected the **Input + LO** mixing product in the cal mixer setup, you must use a high-frequency bandpass filter to reject the **Input - LO** mixing product when characterizing the cal mixer. If this is not practical, see further details at www.Agilent.com and search for "Mixer Transmission Measurements Using The Frequency Converter Application (AN 1408-1)".

3. Power splitter

4. LO Source(s) Connect external sources to the PNA GPIB using either:

- The Agilent 82357A USB/GPIB Interface - **highly recommended -** allows for the use of a remote PC to control the PNA.
- The standard GPIB Interface - with the following limitations:

- The PNA cannot be controlled remotely as talker / listener over GPIB. First put the PNA in System Controller mode. Learn how.
- Available only on PNA releases 4.2 and later.

Learn how to Configure an External LO Source

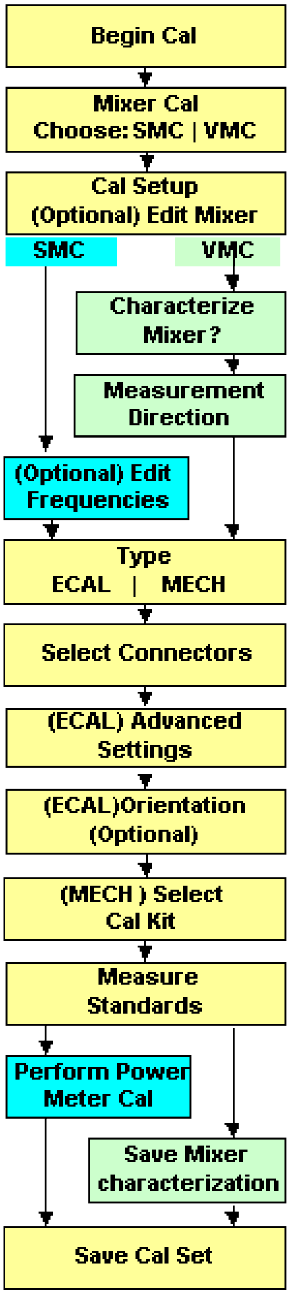
Overview of the Vector Mixer Calibration

The Calibration Wizard guides you through this process. The first three steps characterize the calibration mixer that is used as the THRU standard during the calibration process.

1. Perform two 2-port calibrations over the INPUT and OUTPUT frequency ranges of the DUT. Use either a mechanical calibration kit or an ECal module.
2. Characterize the input and output match of the calibration mixer/filter combination with the external LO connected and the output terminated with an open, short, and load.
3. Connect a reference mixer/filter combination between the Source Out and Rcvr R1 front-panel connectors. Connect the output port of the calibration mixer/filter combination to PNA Port 2 (or at the end of the cable attached to the port).
4. With the PNA in frequency offset mode, measure the calibration mixer/filter combination as the THRU calibration standard.
5. The PNA calculates the error terms necessary to make corrected phase measurements of your mixer/converter under test.

The FCA Calibration Process

Click a box to learn about that step.

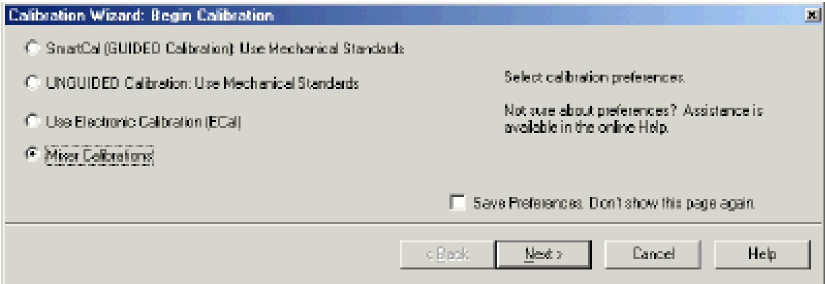


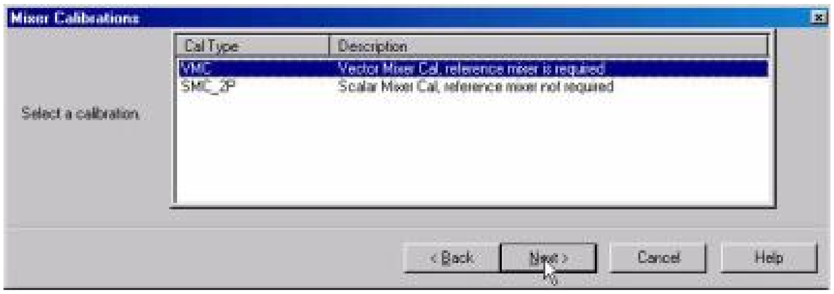
Note: In the above diagram and following procedure, the color coding indicates that the step is performed either in **SMC ONLY** or **VMC ONLY**. The **yellow** blocks indicate the steps that are common to both calibration methods.

How to Perform an FCA Calibration

Begin an FCA calibration as you would any PNA calibration:

1. Click **Calibration**, then **Calibration Wizard**.
2. Then select **Mixer Calibrations.**





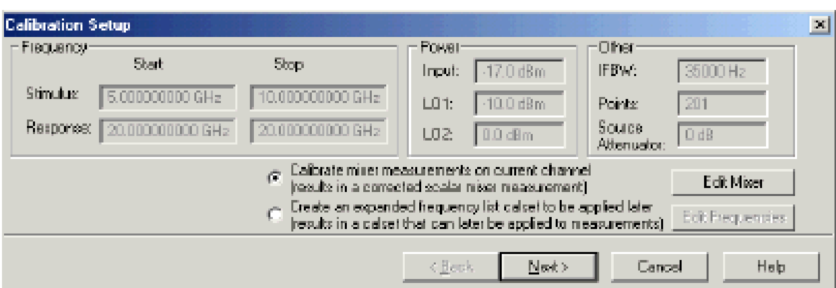
Mixer Calibrations dialog box help

See this step in the FCA cal process

Allows you to select a mixer/converter calibration method:

VMC Vector Mixer Cal Selects the Vector Mixer Cal.

SMC_2P Scalar Mixer Cal Selects the Scalar Mixer Cal.



Calibration Setup dialog box help

Allows you to review and change the settings for your FCA calibration.

- To change **Frequency** and **Power** settings click **Edit Mixer**
- To change **Other** settings (IFBW, Points, Attenuator) scroll or type in a new value.

Edit Mixer Displays the Configure Mixer dialog box.

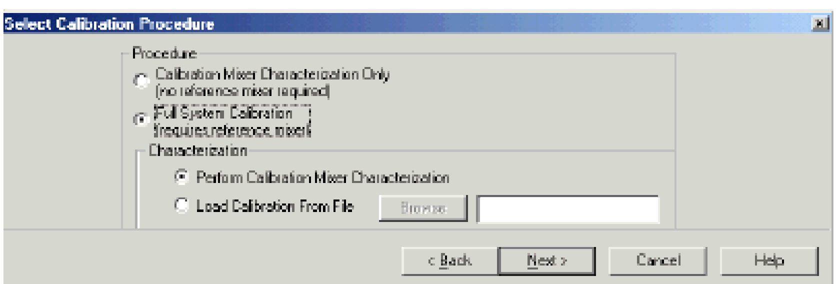
SMC ONLY

Calibrate mixer measurements on current channel (results in a corrected scalar mixer measurement). Allows you to make corrected scalar measurements of your DUT on the current channel. When the calibration is completed, a Cal Set is saved and the PNA automatically creates an SC_{21} measurement.

Create an expanded frequency list calset to be applied later (results in a calset that can later be applied to measurements).
Allows you to select a calibration frequency range that is wider than the frequency range of your DUT. When the calibration is completed, a Cal Set is saved but the PNA does NOT automatically create a measurement. To make a measurement, you must apply the Cal Set to the desired channel and select a measurement.

Edit Frequencies Enabled when **Create an expanded...** is selected. Displays the Frequencies dialog box where you can expand the frequency range of the calibration beyond the range of your DUT. Learn more.

VMC ONLY



Select Calibration Procedure dialog box help

Allows you to perform calibration mixer characterization only or perform a full system calibration.

What is Calibration Mixer Characterization?
For a brief explanation, see Calibration Mixer (Step 2).
For a detailed explanation, see www.Agilent.com and search for "Mixer Transmission Measurements Using The Frequency Converter Application (AN 1408-1)"

Procedure

Calibration Mixer Characterization Only Performs a 2-port calibration and characterizes the calibration mixer. Does not require a reference mixer. Choose this if you want to save the mixer characterization data to a file that will be downloaded later as part of a full system calibration.

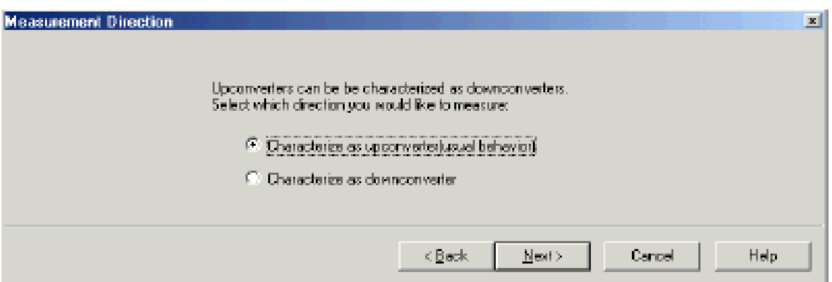
Full System Calibration Performs a full system calibration. Requires a reference mixer.

Characterization - Only available if performing a Full System Calibration

Perform calibration mixer characterization Characterizes the calibration mixer. Choose this selection if you do NOT have the mixer characterization data on file.

Load characterization from file Loads calibration mixer characterization data from your file. Use the **Browse** function to locate the file.

VMC ONLY



Measurement Direction dialog box help

This dialog box appears ONLY if your settings in the Mixer Setup dialog box indicate that your DUT is being tested as an upconverter (input < output). It allows you to characterize the Calibration Mixer as a downconverter (input > output) or an upconverter.

Characterize as upconverter (usual behavior) The disadvantage of characterizing as an upconverter is that you must use an expensive high-frequency filter at the output of the calibration mixer.

Characterize as downconverter Characterizes the calibration mixer as a downconverter which allows you to use a more economical low-frequency filter.

Expanded SMC Calibration

An SMC calibration does not require that the calibration input and output frequencies match the current mixer IF frequency calculations.

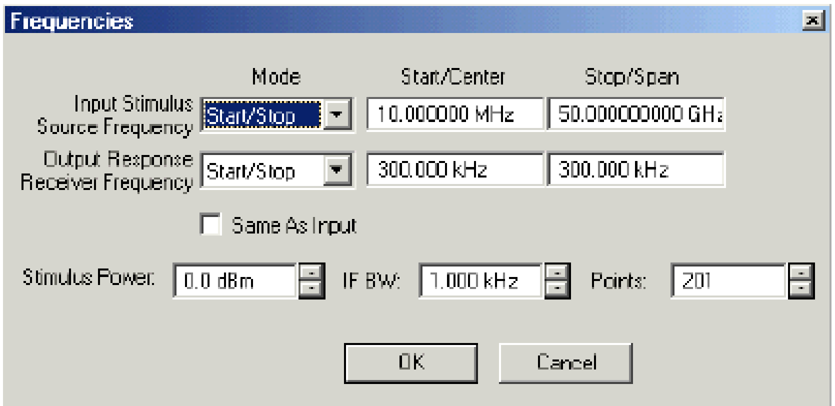
For example, you can perform an SMC calibration over the following frequencies:

- Input frequency range: 10 MHz to 20 GHz
- Output frequency range: 10 MHz to 20 GHz.

Once this calibration is performed, you can save and apply it to any SMC measurement with Input and Output frequencies between 10 MHz and 20 GHz that share the same physical measurement setup. Learn how to apply an SMC calibration.

To perform an expanded SMC calibration, in the Calibration Setup dialog box, select **Create an expanded frequency list calset to be applied later.**

For high measurement accuracy, use a high number of data points. Learn more about Interpolation Accuracy.



Frequencies dialog box help

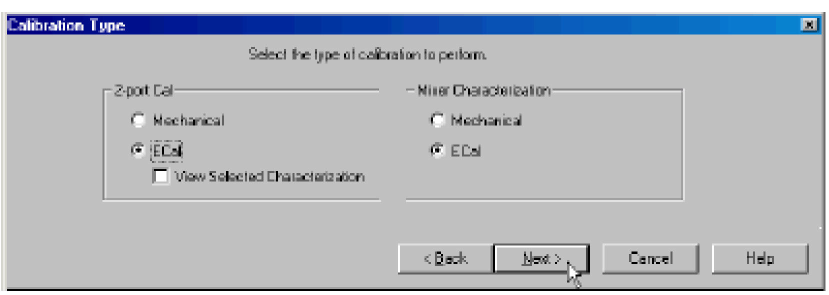
Allows you to edit your calibration frequency range so that it is wider than the frequency range of your DUT. Learn more...

Mode Selects the format to specify the frequency information in your test setup. Frequency information can be specified using start/stop or center/span formats.

Start/Center Allows you to enter the start frequency (for Start/Stop mode) or the center frequency (for Center/Span mode).

Stop/Span Allows you to enter the stop frequency (for Start/Stop mode) or the frequency span (for Center/Span mode).

Same As Input Select this checkbox to set the output response frequency the same as the input stimulus frequency.



Calibration Type dialog box help
Allows you to specify either ECal or mechanical calibration. 2-port Cal **Mechanical** Specifies mechanical calibration. **ECal** Specifies ECal calibration. **View Selected Characterization** Check to invoke a dialog box to View the Selected Characterization (user or factory) for performing the ECAL. Learn more about User Characterization. If the frequency range of the active channel is greater than that of the selected characterization, a **Caution: Additional Standards Required** message is displayed. To correct the problem, perform one of the following solutions: • Reduce the frequency range of the active channel. • Select a different characterization that covers the required frequency range. • Re-characterize the module with an increased frequency range. **VMC ONLY:** Mixer Characterization **Mechanical** Specifies mechanical calibration. **ECal** Specifies ECal calibration. Learn more about Mixer Characterization

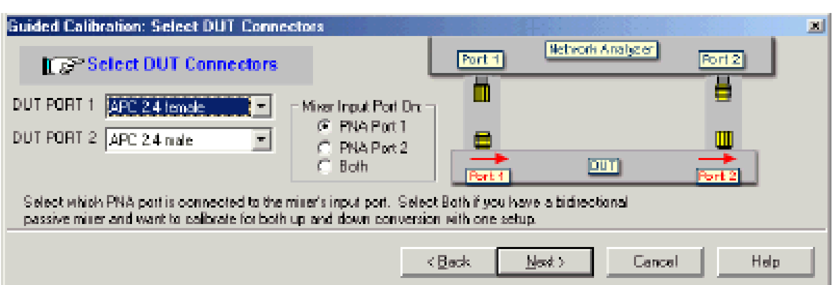
If using ECal:

In the Advanced Settings dialog box, choose the settings for your ECal calibration.

If Adapter Removal calibration was selected, the Select DUT Connectors dialog box is displayed. Select the connectors for port 1 and port 2.

In the Electronic Calibration Steps dialog box, follow the instructions for using the ECal module.

If using Mechanical Cal Kit



Select DUT Connectors dialog box help

Allows you to specify the connector type of each DUT port.

DUT Port 1 Specify the connector type of DUT port 1.

DUT Port 2 Specify the connector type of DUT port 2.

Note: If your DUT connectors are:

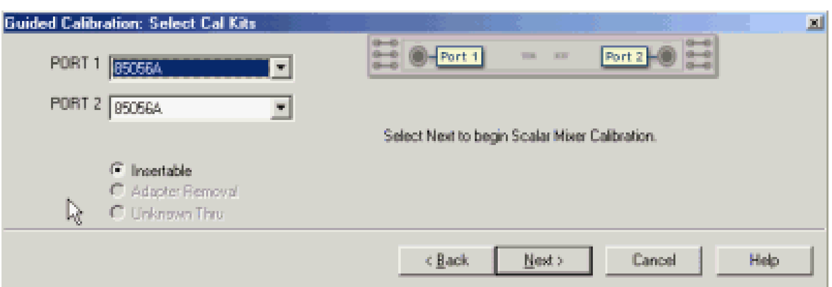
- **Waveguide** Change the system impedance to 1 ohm before performing a calibration. See Setting System Impedance.
- **Not listed** (male and female) Select **Type A** as the connector type. Type A requires a calibration kit file containing the electrical properties of the standards used for calibration (see Calibration kits).
- **Unspecified** (like a packaged device) Select **Type B** as the connector type. Type B requires a calibration kit file containing the electrical properties of the standards used for calibration (see Calibration kits).
- **Non-insertable** Guided Cal automatically performs an Adapter Removal calibration. This requires that you have an adapter that has the same connectors as the DUT.

SMC ONLY:

PNA Port 1 Make this selection to calibrate for ONLY forward direction measurements. Input frequencies are associated with the DUT port that is connected to PNA port 1.

PNA Port 2 Make this selection to calibrate for ONLY Reverse direction measurements. Input frequencies are associated with the DUT port that is connected to PNA port 2.

Both Provides Scalar calibration for both Forward and Reverse SMC measurements. Two cal sets are saved. You apply the appropriate calibration to the measurement. Learn more about SMC Forward and Reverse measurements.

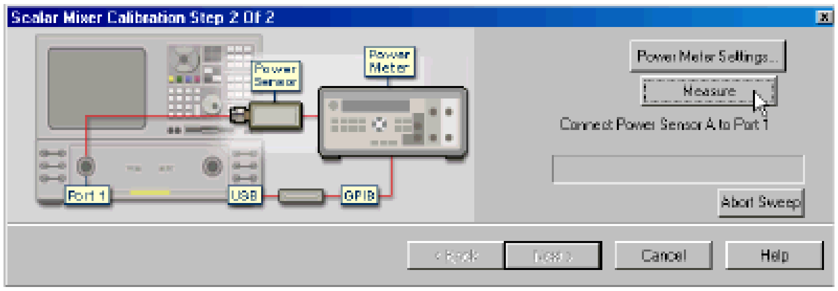


Select Cal Kits dialog box help
Allows you to select the calibration kit used with each measurement port. The list for each port displays kits having the same connector type as the DUT. **PORT 1** Lists the calibration kits available with port 1 **PORT 2** Lists the calibration kits available with port 2 **Insertable** Select if your DUT is an insertable device. **Adapter Removal** Select if your DUT is a noninsertable device and you want to perform an Adapter Removal calibration. **Unknown Thru** Select if your DUT is a noninsertable device and you want to perform an Unknown Thru calibration.

Measure Calibration Standards dialog box help
Illustrates how to connect the calibration standard specified in the prompt. The illustration will vary depending on which calibration standard you are measuring. The illustration will also vary if: • you are using an ECal module • you are performing a calibration-mixer characterization or a full system calibration • your test setup includes a single LO or a dual LO • you selected "Characterize as downconverter" in the Measurement Direction dialog box. In this case, the illustration shows the mixer configured backwards for reflection measurements. The mixer's output is connected to the PNA's test port 2, and the calibration standard (or ECal module) is connected to the mixer's input. **Measure & Next** Measures the mechanical standard and continue to the next calibration step. **Back** Returns to the previous dialog box. **Next** Appears only after clicking Back. Allows you to proceed to the next required step. Does NOT make a measurement. **ReMeasure** Appears only after clicking Back. Allows you to remeasure a standard.

SMC ONLY

For both ECal and Mechanical calibration:



Scalar Mixer Calibration Power Cal dialog box help.

Prompts you to perform the power-meter section of the calibration.

Note: SMC calibration performs 10 averages at the beginning and at the end of the power cal step to ratio the difference between normal and offset R1 measurements in the calibration band of frequencies. The averaging is done to remove a reasonable amount of noise from the ratio measurement.

- Connect your power sensor to the port shown in the diagram. The diagram will vary depending if port 1 was designated as "output" "input" or "both" in the Select DUT Connectors dialog box.
- Begins the power meter calibration on Port 1, 2, or both.

Power Meter Settings Invokes the Power Meter Settings dialog box. See a list of supported power meters.

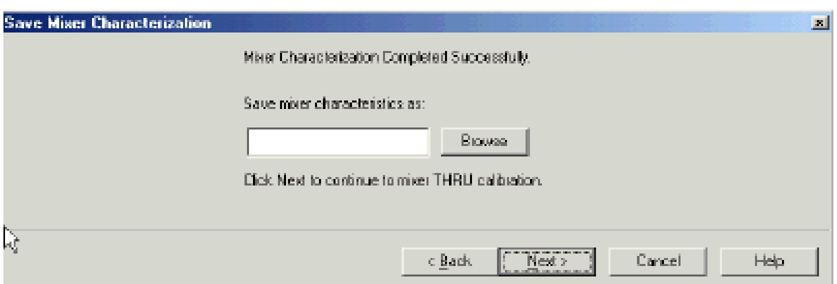
Note: From Power Meter Settings, you can use the Power Loss Compensation Table to compensate for an adapter used to connect the power meter sensor.

Measure Begins the power meter measurements and then continues to the next step.

Back Returns to the previous dialog box.

Next Continues to the next calibration step. Does NOT make a measurement.

VMC ONLY



Save Mixer Characterization dialog box help

Allows you to save the characterization data of your calibration mixer.

Browse Navigate to the location where you want to save the characterization data of your calibration mixer. Either use the default file name or enter a custom file name.

Next Saves the mixer characterization file and continues with the next step in the full system calibration routine.

Finish Replaces **Next** if you are only characterizing the calibration mixer instead of performing a full system calibration. Saves the mixer characterization file and exits the mixer characterization routine.

Save Calibration

If performing a full system calibration:

In the Calibration Completed dialog box, click **No. Finish now** to exit the Calibration Wizard.

Create and Apply an FCA Cal Set

You can create an FCA measurement and apply an existing Cal Set as you can with any PNA measurement. Learn about Cal Sets.

When you apply an FCA Cal Set, be sure to apply the appropriate Cal Type (see table below) to the FCA measurement. The following two notes highlight subtle differences for applying SMC Cal Sets. VMC calibrations are only performed in the forward direction.

- SMC Cals save separate Cal Sets for the Forward and Reverse directions. An SMC Forward Cal Set cannot be applied to an SMC Reverse measurement and vice versa. Learn more about Forward and Reverse Scalar measurements.
- A single SMC Cal Set can contain the following SMC Cal Types, allowing you to optimize Fixed Output measurements for speed versus accuracy. Learn other ways to speed up your fixed output measurements.

FCA Cal Type	FCA Measurement
SMC_2P: (Response + Input + Output) Most accurate **SMC Rsp**: no Input or Output match - Fastest for fixed output, but not most accurate. **SMC Rsp + In**: no Output match - Best combined accuracy and speed for fixed Output. **SMC Rsp + Out**: no Input match	SC21 SC12 S11 S22 Incident Power Output Power
VMC (source is Port 1)	VC21 S11 S22

Configure a Mixer

- How to Start the Mixer Setup dialog box
- Learning the Mixer Setup Dialog Box
- Rules for Configuring a Mixer
- Using Power Sweep for Testing Mixers
- Input > LO Example
- Configure Fixed Output Measurements
- Fractional Multiplier Examples

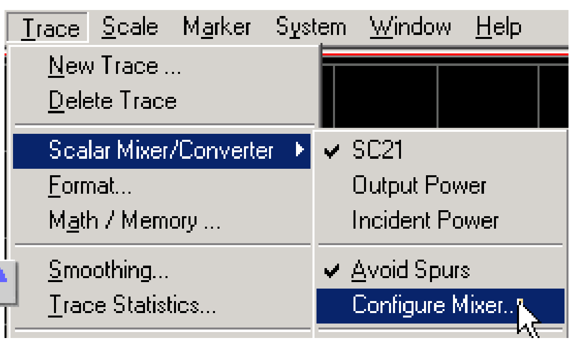
Note: Please submit FCA issues that you find, as well as enhancement requests, to fca_support@agilent.com (See Known FCA Issues.)

Other Frequency Converter Application topics

How to start the Mixer Setup dialog box

Choose any of the following methods:

Create an FCA measurement. Then



OR

Create an FCA measurement. Then press one of these buttons:

or 

or 

OR

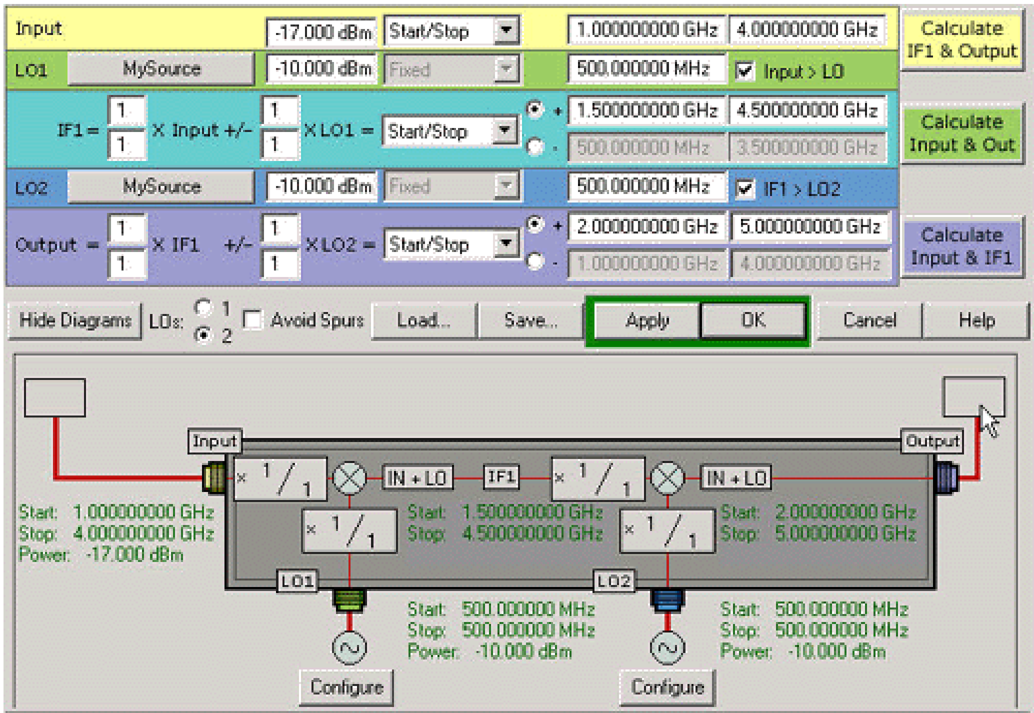
Create an FCA measurement. Then

Click **Calibration**, then **Calibration Wizard**
Click **Mixer Calibration**
Click **Edit Mixer**

Learn more about using the front panel interface

Mixer Setup dialog box

Click on sections of the image to learn about a setting.



Note: This image shows two LOs.

See Fractional Multiplier Examples (below)

Mixer Setup dialog box help

Rules for Using the Mixer Setup Dialog Box

Red **Apply** and **OK** buttons indicate that one or more of the following settings are invalid.

1. The INPUT start frequency can NOT exceed the stop frequency.(The OUTPUT start frequency CAN exceed the stop frequency.)
2. INPUT or OUTPUT frequencies cannot be outside the range of the PNA
3. Any combination of Input and LO which results in an Output that sweeps through zero Hz is NOT allowed.
4. The range for the numerator and denominator of a fractional multiplier is from +1 to +10. Negative values are not allowed.

Note: Throughout the dialog box, the Mixer / converter ports are color coded (Input, LO1, IF1, LO2, Output)

Power Sets the power level of the input signal, and both LO signals.

Frequency Format Selects the format to specify the frequency information for each signal in your test setup. The Input, LO1, LO2, IF, and Output frequency information can be specified using start/stop or center/span formats.

Only LO1, LO2, IF, and Output formats can be set to Fixed.

When you select a swept LO, you can also select the information you want to display on the X-axis.

Return to **Mixer Setup** image

Source Configuration Buttons Performs the same function as the configuration buttons on the lower diagram. The current source is displayed on the button label.

Resulting Frequencies Either sets or calculates the frequency values for each of the signals in your test setup. For example, if you enter the Input frequency range and press the Calculate button adjacent to the Input, the PNA will calculate and display the Output frequencies.

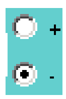
Return to **Mixer Setup** image

Input > LO

These check boxes are used to remove ambiguity when using the Calculate button to determine the INPUT frequency.

	Check if the **INPUT** is **GREATER THAN** the **LO** Clear if the **INPUT** is **LESS THAN** the **LO**
	Check if the **IF1** is **GREATER THAN** the **LO2** Clear if the **IF1** is **LESS THAN** the **LO2**

These boxes are only used when all 3 of the following conditions are TRUE:
(If ALL 3 are NOT true, the PNA does not read these boxes).

1. Difference (Low) sideband  is selected for the corresponding Calculate button AND
2. Output frequency is less than the LO frequency AND
3. The Green or Blue Calculate button is used to calculate the Input frequency.

To learn more see this example.

Fractional Multiplier

The combination of (numerator / denominator) forms a fractional value that is multiplied by the input and LO frequency ranges (also the IF and LO2 frequency ranges for a test setup with two LOs). These values are used to Calculate the response frequency of the PNA receiver. Use the fractional multipliers to:

- replicate the action of harmonic mixers
- replicate the action of multipliers and dividers that may exist in your test setup
- tune the PNA receiver frequency to a harmonic of the mixer/converter

See Fractional Multiplier examples.

Return to **Mixer Setup** image

Mixer-Product Selector Determines whether the receivers will tune to the Sum (+) or the Difference (**-**) of the Input and LO frequencies. Click the adjacent Calculate button after your selection.

Calculate buttons Calculates frequency information based on your other mixer settings. The mixer port settings next to the Calculate button you press remain fixed. For example, in a 1-LO scenario, specify the Input and LO frequencies, specify **+** (sum), then click the **Calculate** button next to the Input. The input remains fixed and the output frequency range is calculated for you.

Return to **Mixer Setup** image

Hide / Show Diagrams Hides and displays the test setup diagram. Your measurement trace is displayed when the diagram is hidden.

LOs Click **1** or **2** to select the number of external LO sources in your test setup. When you select 2 LOs, the IF1 frequencies are set for you.

Measurements of a DUT with an embedded LO are not supported at this time.

Return to **Mixer Setup** image

Avoid Spurs Check to invoke the Avoid Spurs feature.

Load Loads a previously-configured mixer attributes file (.mxr).

Note: A .mxr file includes an LO source name. However, It does NOT include the LO Source configuration. Therefore, when using a .mxr file that was created on a different PNA, the PNA will display an error if does not find the LO Source configuration using EXACTLY the same LO source name.

Save Saves the settings for your mixer/converter test setup to a mixer attributes file (.mxr).

Apply Applies the settings for your mixer/converter test setup to the measurement. The mixer setup dialog box remains OPEN. If shaded red, see rules.
Return to **Mixer Setup** image

OK Applies the settings for your mixer/converter test setup to the measurement. The mixer setup dialog box CLOSES. If shaded red, see rules.

Cancel Closes the mixer setup dialog box and does NOT apply the settings.

Frequency Diagram: Provides a display of the frequency information for the signals in the test setup.

Configure Source Displays the External Source Configuration dialog box. To configure your external source with this dialog box, the source must be connected to the PNA with the Agilent 82357 USB/GPIB Interface provided with the Frequency Converter Application (Option 083).
Return to **Mixer Setup** image

Using Power Sweep for Testing Mixers

To measure the gain compression of a mixer, you need to sweep the input power to the mixer. The input and output frequencies are fixed but offset from one another. To set Power Sweep and the input and output frequencies of the mixer under test:

1. On the mixer dialog box, set the LO frequency, identical input start and stop frequencies, and identical output start and stop frequencies. These selections create fixed input and output frequencies.
2. On the PNA menu, click **Sweep**, then **Sweep Type**. Select **Power Sweep**. Do NOT change the CW frequency on the Power Sweep dialog box. The mixer dialog box settings will not be automatically updated.

For more information, see Conversion Compression.

Input > LO Example

For the following single stage mixer:

- Output = 2 GHz
- LO = 3 GHz
- Diff (-) selected

Clicking **Calculate Input** could yield two Input frequencies:

Formula for **Diff**:

Input - LO = Output

Substitute our example values in the formula:

Input - 3GHz = 2 GHz

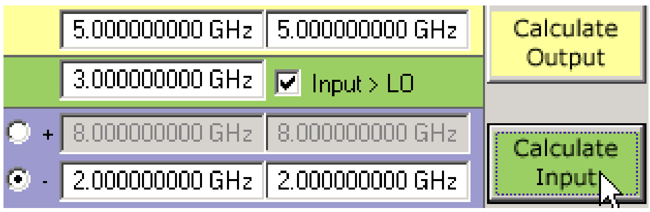
Solving the formula can yield either:

Input = 5 GHz

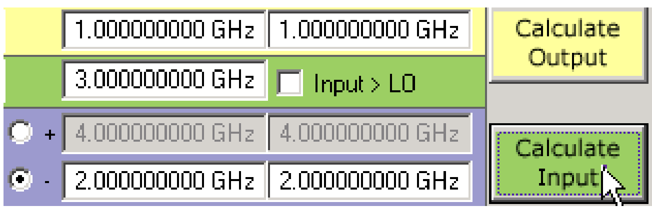
OR

Input = 1 GHz

(Although 1-3 = - 2 GHz, the analyzer displays the absolute value of the frequency.)



Check - use the Input frequency (5 GHz) that is greater than LO (3 GHz)



Clear - use the Input frequency (1 GHz) that is less than LO (3 GHz)

Configure Fixed Output Measurements

The Frequency Converter Application can make calibrated Fixed Output **SMC** and **VMC** measurements. This requires that one or both LO (external sources) sweep in a synchronized manner with the Input.

Do the following to make a Fixed Output measurement with 1 LO:

1. Set Input to **Start/Stop**.
2. Set Input start and stop frequencies.
3. Set Output to **Fixed -** LO is automatically set to **Start/Stop.**
4. Set the Output frequency.
5. Click **Calculate LO** button
6. Click the LO **Not controlled** button to Configure the External LO Source. Although the PNA will allow you to leave LO as **Not Controlled**, this configuration will usually result in meaningless Fixed Output measurements.

Notes:

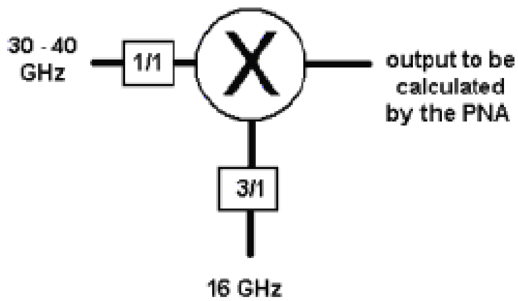
- In VMC fixed-output measurements, the phase response is offset as a function of the Calibration Mixer. Using a different Calibration Mixer will result in a different phase offset, though the relative phase responses (ripple, deviation from linear phase) will remain calibrated.

- Do NOT use the Group Delay format with VMC fixed-output as the phase response contains only a single point for each LO frequency. Therefore, the calculation of group delay produces an invalid result.
- Because your Fixed Output measurement is controlling an external source, you may find that the measurement is very slow. Learn how to speed up your Fixed Output measurement.

Fractional Multiplier Examples

Example 1

Use the LO fractional multiplier to replicate the action of the third-harmonic mixer so the PNA can accurately calculate the receiver frequency. The input and LO frequencies are known.



Enter these settings in the **Mixer Setup** dialog box:

- **Input Start Freq: 30 GHz**
- **Input Stop Freq: 40 GHz**
- **LO Fixed Freq: 16 GHz**
- Mixer-Product Selector: - (difference)
- LOs: 1
- LO fractional multiplier: 3/1
- INPUT fractional multiplier: 1/1

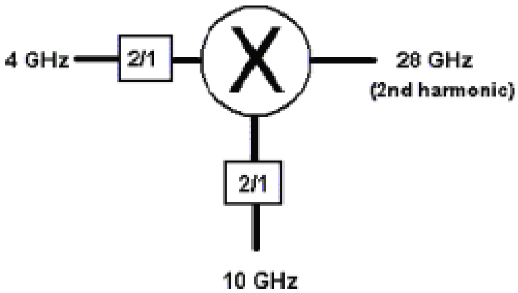
Click **Calculate Output**

Results:

- **Output Start Freq: 18 GHz**
- **Output Stop Freq: 8 GHz**

Example 2

Use the fractional multipliers to tune the PNA receiver frequency to the second harmonic of the mixer's 14 GHz fundamental output. The input, LO, and output frequencies are known.



Enter these settings in the **Mixer Setup** dialog box:

- **Input Start Freq: 4 GHz**
- **Input Stop Freq: 4 GHz**
- **LO Fixed Freq: 10 GHz**
- Mixer-Product Selector: + (Sum) of the input and LO signals
- LOs: 1
- INPUT fractional multiplier = 2/1
- LO fractional multiplier = 2/1

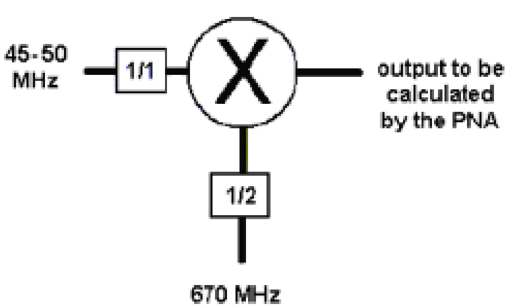
Click **Calculate Output**

Results:

- **Output Start Freq: 28 GHz**
- **Output Stop Freq: 28 GHz**

Example 3

Use the LO fractional multiplier to replicate the action of the divide-by-two mechanism inside the mixer package. Having done this, the PNA can accurately calculate the receiver frequency. The input and LO frequencies are known.



Enter these settings in the **Mixer Setup** dialog box:

- **Input Start Freq: 45 MHz**
- **Input Stop Freq: 50 MHz**

- **LO Fixed Freq: 670 MHz**
- Mixer-Product Selector: **+** (Sum) of the input and LO signals
- LOs: 1
- INPUT fractional multiplier = 1/1
- LO fractional multiplier = 1/2

Click **Calculate Output**

Results:

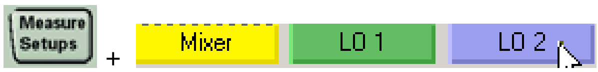
- **Output Start Freq: 380 MHz**
- **Output Stop Freq: 385 MHz**

Configure an External LO Source

You can configure a LO source to be used when making either SCALAR or VECTOR Frequency Converter Application (option 083) measurements.

How to Configure an External LO Source

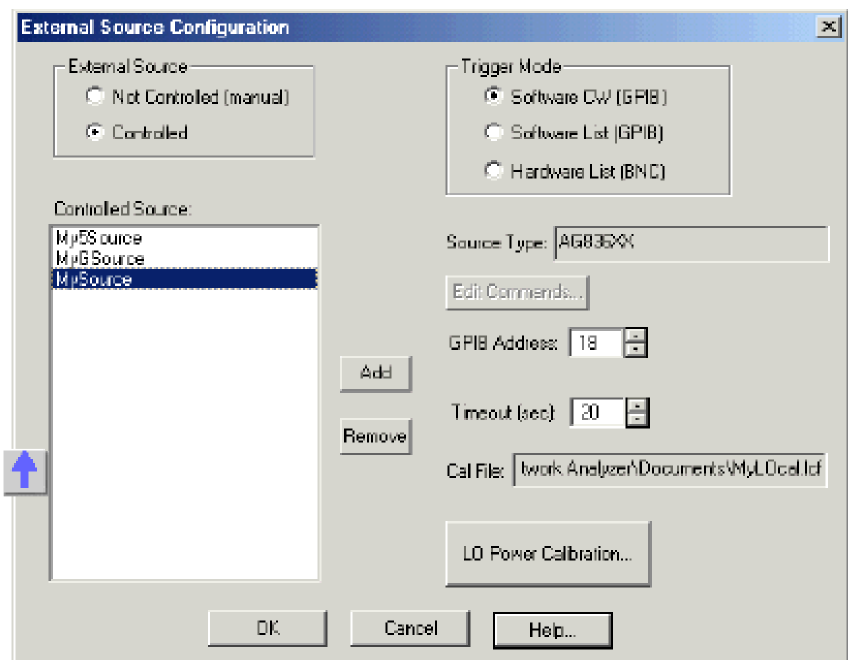
1. Create an FCA measurement
2. Then either:

- 

- or click **LO Control** in the Configure Mixer dialog box.

Learn more about using the front panel interface

Note: If an **External Source Not Found** error occurs, the Agilent I/O Library may no longer be running. To check, look in the Windows task bar of the PNA for the **IO** icon. If not present, restart the IO library. Click Start, Programs, Agilent I/O Libraries, IO Control.



External Source Configuration dialog box help

This dialog box is ALSO used to configure external sources for Millimeter Wave Test Heads. Learn more.

There are currently no PNA programming commands to configure the external source.

Connect external sources to the PNA GPIB using either:

- The Agilent 82357A USB/GPIB Interface - **highly recommended -** allows for the use of a remote PC to control the PNA.
- The standard GPIB Interface - with the following limitations:

- The PNA cannot be controlled remotely as talker / listener over GPIB. First put the PNA in System Controller mode. Learn how.
- Available only on PNA releases 4.2 and later.

External Source Configuration

Not Controlled (manual) The PNA does NOT control the external source. When this selection is made, all other settings in this dialog box do NOT apply.

Controlled The PNA controls the external sources using GPIB.

Trigger Source ONLY used when the external source is stepped (FCA Fixed output measurements).

Note: The PNA Trigger Source setting must NOT be set to EXTERNAL for any of the following selections.

See SCPI and COM examples of an SMC fixed output measurement.

Software CW (GPIB) Slowest method. The external source receives the CW frequency from the PNA over GPIB This is the only selection available for generic (not listed), and Agilent 837X sources.

Software List (GPIB) The external source receives a list of CW frequencies from the PNA, then receives only trigger signals as required, over GPIB. Used ONLY for Agilent 836X sources.

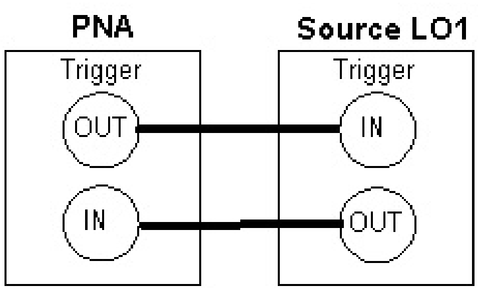
Hardware List (BNC) Fastest method. The external source receives a list of CW frequencies from the PNA, then receives trigger signals as required from the PNA rear-panel BNC Trigger connectors.

Note: Hardware List is ALWAYS used for 2 External Sources driving Millimeter Wave Heads.

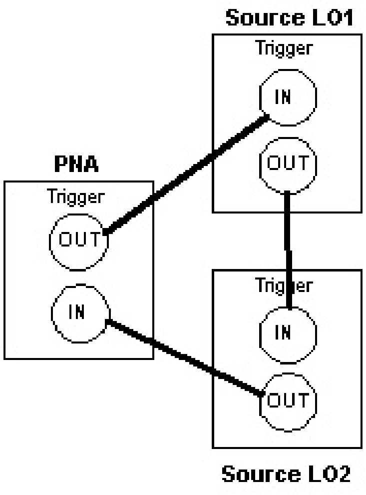
The PNA will NOT respond to Manual Triggers in **Hardware List (BNC)** mode. The PNA trigger source must be set to INTERNAL. Then use the PNA channel trigger settings "Group" or "Single" to specify how the FCA measurement responds to internal triggers.

The sources must be connected in the following manner:

1 External Source



2 External Sources



Note: The PNA checks communication with the sources the at the time of the first measurement sweep. If an error occurs, the PNA puts the channel in Hold trigger mode. You must fix the problem, then put the channel into either continuous or single sweep triggering to restart the measurement.

For more information, see:

- Speeding Up Fixed Output SMC Measurements

- PNA Trigger model

Controlled Source

Add Displays the Add New Source dialog box.

Remove Removes an external source from your setup.

Source Type Shows the model number of the external source that is selected in the displayed list.

Edit Commands Only available with generic (non-Agilent) sources. Invokes the Edit Commands dialog box.

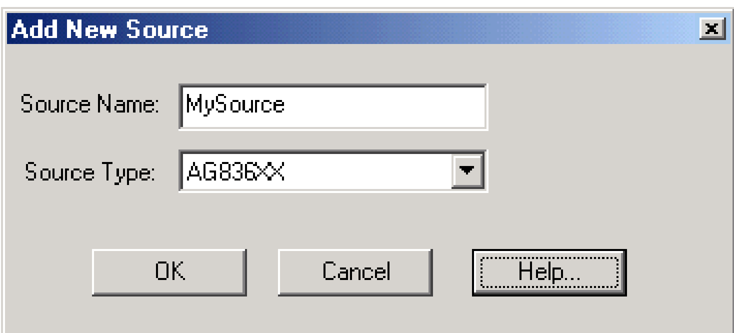
GPIB Address Sets the GPIB address of the selected external source.

Timeout (sec) Sets a time limit for the source to make contact with the PNA. If this time limit is exceeded, the PNA stops the measurement procedure and displays a diagnostic-type error message. If this occurs, check the connections of your PNA and source.

Cal File: Displays the calibration file that is in use with the selected external source. After configuring a source, a calibration must be performed to see the file. Enabling **Slope Offset** constitutes a calibration.

LO Power Calibration Displays the LO Power Calibration dialog box. A warning message appears stating that the changes you have made will be saved.

Cancel Closes the dialog box without saving changes (unless the **LO Power Calibration** button was pressed).



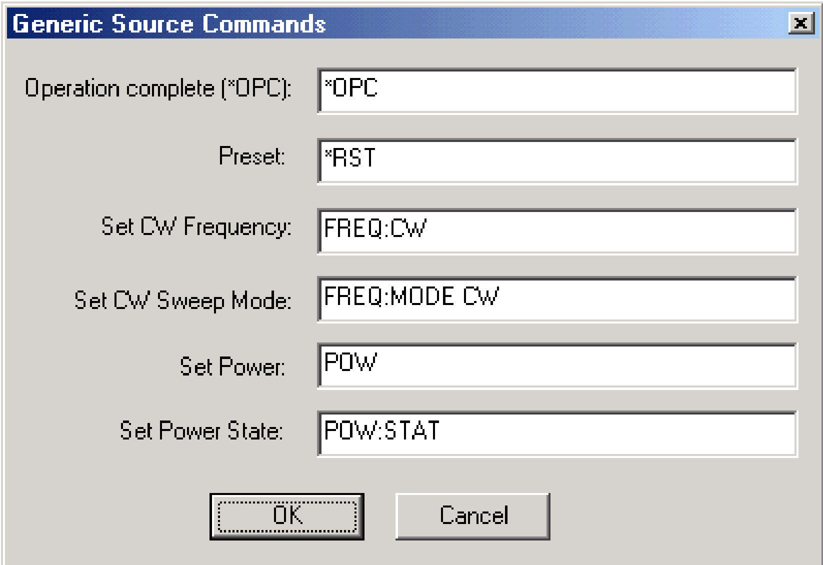
Add New Source dialog box help

Allows you to add an external source. The new source appears in the list of sources displayed in the External Source Configuration dialog box.

Source Name Allows you to enter a name for your source.

Source Type Allows you to select a source type from the scrolling list.

To use an Generic external source (not listed) Select **AGGeneric.** This invokes the following **Generic Source Commands** dialog box**.**



Generic Source Commands dialog box help

Load the following SCPI commands that control the functions on your Generic (not listed) source.

Operation Complete (*OPC) .

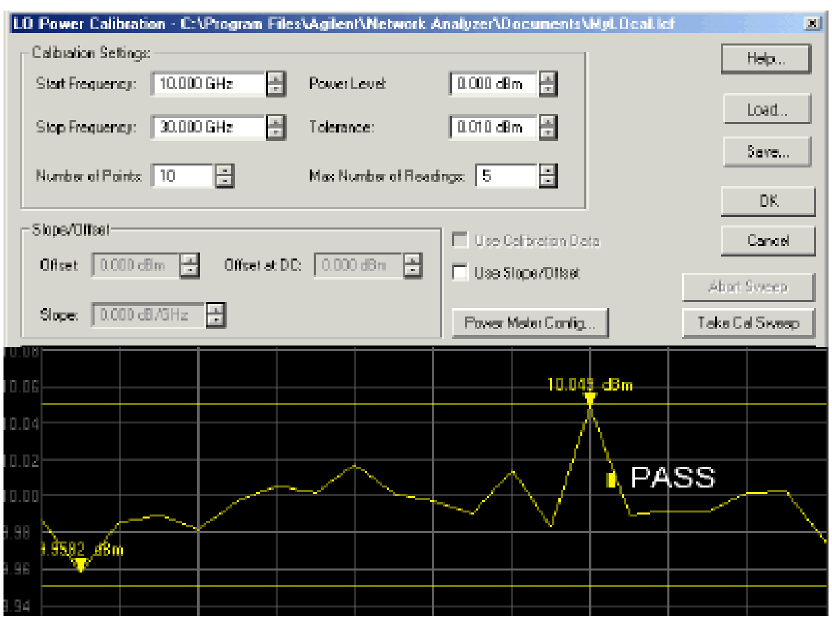
Preset Presets the source

Set CW Frequency Sets CW Frequency to value in Mixer Setup dialog

Set CW Sweep Mode Sets source sweep mode

Set Power Sets source power to value in Mixer Setup dialog

Set Power State Turns Power ON or OFF



LO Calibration dialog box help

Allows you to select, load, and save, calibration settings for your external source. Also, you can perform an LO calibration, set the slope/offset, and configure your power meter.

Note: The LO cal sweep, along with "tolerance" limit lines, is visible below the dialog box. Right click on the screen to access display settings.

Calibration Settings

Start Frequency Sets the start frequency for the LO calibration.

Stop Frequency Sets the stop frequency for the LO calibration.

Number of Points Sets the number of data points that the power meter measures in the frequency range of the LO calibration.

Power Level Sets the power level for the LO calibration.

Tolerance Sets the maximum permissible deviation from the specified power level.

Max Number of Readings Sets the maximum number of iterative routines the PNA will perform to implement the power setting.

Slope/Offset

Offset Sets the amount of offset power applied to each measurement point of the LO calibration frequency range. This power is offset from the power setting of the Slope feature.

Offset at DC Sets the amount of offset power applied at DC.

Slope Allows you to set the power slope (dB)/GHz. Compensates for cable and test fixture power loss by increasing the External LO output power over frequency.

Use Calibration Data Check to use the calibration data to correct your external source. The calibration data may come from a source calibration file (.lcf) that you have loaded, or from a new LO calibration.

Use Slope/Offset Check to apply the slope/offset settings to your external source. Using the slope/offset settings is a faster, but less thorough way to improve LO accuracy than performing an LO calibration. These settings can be used in addition to an LO calibration. For example, if you add an adapter in your LO path, you can use the slope/offset settings to compensate for the effects of the adapter rather than repeat the LO calibration.

Power Meter Config Displays the Power Meter Settings dialog box.

Abort Sweep Immediately stops the calibration measurement.

Take Cal Sweep Starts the calibration measurement after a prompt to connect the power meter to the external source right at the DUT LO port.

Load Loads a source calibration file (.lcf) of your choice.

Save Saves your LO calibration data in a source calibration file (.lcf).

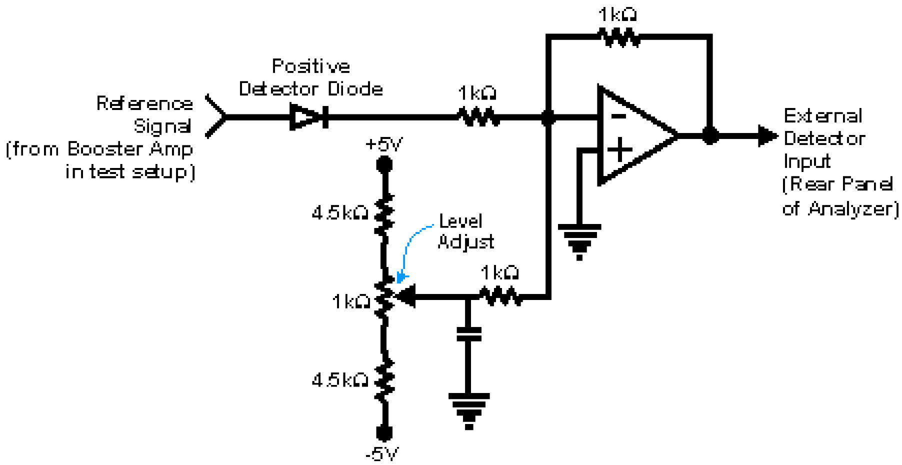
Power Meter Settings See Source Power Calibration

Sensors See Source Power Calibration

External Leveling Circuitry that Uses a DC Preamplifier

Note: This circuitry will not work with E836x models; they do not have an External Leveling input on the rear panel.

The graphic below is a schematic diagram of **external leveling circuitry** (provided by the user) that uses a DC preamplifier. A preamplifier is recommended when the input power-level to the detector diode is less than   dBm.



- The external leveling circuitry consists primarily of
  - A detector diode
  - An operational amplifier
- Changes in the power measured by the diode are detected by the operational amplifier circuit. This circuit regulates the voltage at the analyzer's External Detector Input, adjusting the analyzer's source power to achieve a flat response at the DUT on a real-time basis.
- For external leveling to work correctly, the voltage at the analyzer's External Detector Input must be between -0.002 mVDC and -6 VDC, depending on frequency and power level.
- Making too great a change in the reference-signal power can cause the external-leveling function to become ineffective due to the limited power-control range of the analyzer's source.

High-Power Amplifier Measurements Using a PNA

High-power amplifiers have one or both of the following attributes:

- They require more input power than a standard analyzer can provide.
- They deliver more output power than a standard analyzer can measure.

Click on a solution to learn to make measurements of high-power amplifiers:

Challenge	Solution
Which test-set configuration does my PNA have?	PNA test-set configurations
DUT input-power is too low	Booster amplifiers in the test setup
E835xA, E880xA, and N338xA only: Booster amplifier response is not flat	External leveling for booster amplifiers
Phase-lock lost when using a booster amplifier	Disable auto-frequency calibration
Which measurement calibration do I use?	Measurement calibration choices
What calibration power levels do I use?	Power-level recommendations for calibration
DUT output power is too high	Components that reduce DUT output power
Will my calibration standards withstand high power?	Calibration standards that withstand high power
Thermal variations occur in my test setup	Thermal requirements for DUTs and loads
DUT has an Automatic Gain Control (AGC) loop	Stimulus-sweep recommendation for a DUT with an AGC loop
What procedure do I use to measure high-power amplifiers?	Procedure for Measuring High-Power Amplifiers

Note: Many of the external components shown in the test set-up configurations of this tutorial are available from Agilent Technologies. For example, Agilent sells couplers, power splitters, and attenuators. Go to **http://www.agilent.com/find/accessories** and use the search function to locate information on components you may want to purchase. You can also get Agilent technical support by telephone or fax.

PNA Test-Set Configurations

Here's how to identify your PNA.

Configuration graphics of PNA test sets:

PNA without Configurable Test Set	PNA with Configurable Test Set
E835xA	E835xA, Option 015
E880xA	E880xA, Option 014
N338xA	N338xA, Option 014
E836xA/B	E836xA/B, Option 014
N5230A	N5230A, Option 225, or 425, or 525
	N5250A

Booster Amplifiers in the Test Setup

If your DUT requires more input power to be tested accurately than the analyzer can provide, you need to provide a booster amplifier in your test setup. (See your analyzer's power range in the Technical Specifications.) However, using a booster amp may limit the types of testing and calibration you can perform. Also, your booster amplifier may not provide a consistent power level to your DUT over the frequency range of interest. If your booster amplifier requires you to provide a leveling loop, see External Leveling for Booster Amplifiers, available only on models **E835xA, E880xA, and N338xA.**

Configuration graphics for using booster amp circuitry:

Configuration	PNA without Configurable Test Set[1]	PNA with Configurable Test Set
One Booster Amp	E835xA[2] E880xA[2] N338xA[2] E836xA/B[2] N5230A	E835xA, Option 015[3] E880xA, Option 014[3] N338xA, Option 014[3] E836xA/B, Option 014[3] N5230, Option 225, or 425, or 525 N5250A
One Booster Amp & One Reverse Coupler	E835xA[3]	
Two Booster Amps	E835xA[4]	E835xA, Option 015 E880xA, Option 014 N338xA, Option 014[5] E836xA/B, Option 014 N5230A, Option 225, or 425, or 525 N5250A

1 You can choose from other booster amplifier configurations by purchasing the Configurable Test Set upgrade. For E835xA analyzers, order Option 015. For all other analyzers, order Option 014. **See PNA Options.**

Limitations (footnotes 2 - 5)**:**

2 • Allows only a Thru Response Cal for boosted forward transmission (S_{21}) measurements.

• Allows only a 1-Port Reflection Cal or Open/Short Response Cal for nonboosted reverse reflection (S_{22}) measurements.

• Reverse *boosted-power* measurements are not possible unless the booster amplifier is relocated.

• Reflection measurements on Port 1 or reverse transmission measurements are not possible. The signal is blocked by the booster amplifier.

3 Reverse *boosted-power* measurements are not possible unless the booster amplifier is relocated.

4 None.

5 **N338xA Option 014 only:** Reverse boosted-power measurements are not possible from Port 3. This is because Port 3 does not have Source Out and Coupler Thru channels.

Disable Automatic Frequency Calibration when using Booster Amps.

When using a booster amplifier, or whenever removing a front-panel jumper cable to provide a signal to the internal reference receiver, you may see a phase-lock lost indication. This can occur if you use a narrow-band filter or coupler in your test setup.

If phase-lock lost indication occurs, the PNA will automatically re-calibrate the tuning curve of the internal source. To prevent the PNA from re-calibrating itself, disable the Automatic Frequency Calibration while the PNA is still phase locking correctly.

To do this, Click **System**, then point to **Configure**, then click **SICL/GPIB**. At the prompt, type **DIAG:AUTOCAL OFF**.

The PNA may still lose phase-lock outside of the pass band of your coupler or filter, but the PNA will not try to re-calibrate itself.

Note: For best performance, the reference receiver power must be 22dB (+/-5dB) below the power at the test port.

External Leveling for Booster Amplifiers

Note: External leveling is not possible with the E8361A, E8362A/B, E8363A/B, E8364A/B, N5230, N5250.

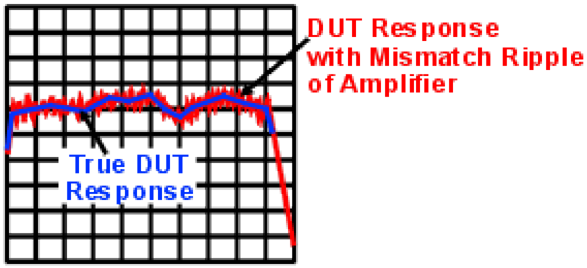
If the boosted response of your DUT is not as flat as you expect, it may be attributed to the output of your external booster amp. If the output has ripple across the frequency band of interest, your DUT will amplify that ripple. You can add your own external leveling circuitry to compensate for poor flatness of an external booster amplifier in your test setup.

How do I Determine If I Need to Use External Leveling with my Booster Amplifier?

Use your analyzer to measure the peak-to-peak ripple of your booster amplifier. If the level of peak-to-peak ripple is unacceptable for the measurement accuracy you require, use external leveling.

Note: Be sure to calibrate before measuring the peak-to-peak ripple of your booster amplifier. Calibration will prevent noise (if present) from varying the look of your measurement during each sweep of the analyzer.

As shown in the following graphic, a booster-amplifier without external leveling may have ripple that causes DUT response ripple.



Some causes of booster-amplifier ripple:

- Low quality booster amplifier (poor match or flatness)
- Noisy power-supply voltage for booster amplifier
- Saturated booster amplifier

What Configurations Do I Use For External Leveling?

Note: External leveling is not possible with the E8361A, E8362A/B, E8363A/B, E8364A/B.

Configuration graphics for using external leveling circuitry:

External Leveling Configuration	Standard Analyzers	Option 014 and 015 Analyzers
One Booster Amplifier	E835xA E880xA N338xA	E835xA, Option 015 E880xA, Option 014 N338xA, Option 014
One Booster Amplifier and a Reverse Coupler	E835xA	
Two Booster Amplifiers and Two Reverse Couplers	E835xA	
Two Booster Amplifiers		E835xA, Option 015 E880xA, Option 014 N338xA, Option 014

Power Levels for External Leveling

The analyzer's test port power should be maintained between **+10 dBm and -15 dBm** for proper phase lock and ALC (automatic level control) operation. If no external DC preamplifier is used at the DC output of the detector diode, the power applied to the input of the detector diode should be maintained between **+10 dBm and -20 dBm**. This power range is necessary to achieve an adequate DC output from the detector diode for proper ALC operation. A negative-output detector diode should be used, such as an Agilent 3330C (you can find information about this detector diode at **http://www.agilent.com/find/accessories**.) The DC output of the detector is connected directly (using a coaxial cable) to the analyzer's rear panel "Ext Detector Input."

More information about the external leveling circuitry:

- The detector diode's output voltage, applied to the analyzer's External Detector Input, adjusts the source power to achieve a flat response at the DUT on a real-time basis.
- For external leveling to work correctly, the voltage at the analyzer's External Detector Input must be between -0.002 mVDC and -6 VDC, depending on frequency and power level.

- Making too great a change in the reference-signal power can cause the external leveling function to become ineffective due to the limited power-control range of the analyzer's source.

View external leveling circuitry that uses a DC preamplifier (recommended when the input power-level to the detector diode is less than -20 dBm).

Note: When performing a calibration, keep the source-leveling process active, just as it will be during the measurement.

Which Analyzer Menu Choices Do I Select for External Leveling?

Note: External leveling is not possible with the E8361A, E8362A/B, E8363A/B, E8364A/B.

E835xA:

1. Select the **Sweep** menu.
2. Select **Sweep Setup**. Select the **External ALC** checkbox.
3. Select the **Channel** menu.
4. Select **Power**. In the **Attenuator Control** section of the dialog box, clear the **Auto** checkbox.

E880xA and N338xA only:

1. Select the **Sweep** menu.
2. Select **Sweep Setup**. Select the **External ALC** checkbox.

Find more information about using external leveling at **http://www.tm.agilent.com**. Click "Library" and use the search function to search for "ALC."

Measurement Calibration Choices

Note: Calibration should be performed with all hardware in place. For example, include any booster amplifiers, cables, attenuators, or couplers that will be used when measuring the DUT. For more calibration information, refer to Measurement Calibration Choices.

CAUTION: When using an open or a short during calibration, be sure that the incident power level is low enough so that the reflected signal doesn't exceed the maximum input to the analyzer's receivers.

Calibration choices are determined by the measurement configuration you use.

The following are configuration graphics and calibration choices for high-power measurements:

Measurement Direction	PNA without Configurable Test Set	PNA with Configurable Test Set	Calibration Choices
Forward	E835xA	E835xA, Opt 015 E880xA, Opt 014 N338xA, Opt 014 E836xA/B, Opt 014 N5230A, Opt 225, or 425, or 525 N5250A	2-Port Calibration[1] 1-Port Calibration[2]
Forward (single-port cal only)	E835xA E880xA N338xA E836xA/B N5230A		Thru Response[3] 1-Port Reflection[4] Open/Short Response[4]
Forward and reverse	E835xA	E835xA, Opt 015 E880xA, Opt 014 N338xA, Opt 014 E836xA/B, Opt 014 N5230A, Opt 225, or 425, or 525 N5250A	2-Port Calibration[5] 1-Port Calibration[6]
Forward and reverse (non-boosted measurements)	E835xA E880xA N338xA E836xA/B N5230A		2-Port Calibration[7] 1-Port Calibration[8]

1 (TRL or SOLT) For forward high-power transmission and reflection measurements. Also for reverse non-boosted transmission and reflection measurements.

2 (Thru Response, 1-Port Reflection, or Open/Short Response) For forward high-power transmission or reflection measurements. Or, for reverse non-boosted transmission or reflection measurements.

3 For forward high-power transmission measurements.

4 For reverse non-boosted reflection measurements.

5 (TRL or SOLT) For forward and reverse high-power transmission and reflection measurements.

6 (Thru Response, 1-Port Reflection, or Open/Short Response) For forward or reverse high-power transmission or reflection measurements.

7 (TRL or SOLT) For forward and reverse non-boosted transmission and reflection measurements.

8 (Thru Response, 1-Port Reflection, or Open/Short Response) For forward or reverse non-boosted transmission or reflection measurements.

Power Level Recommendations for Calibration

For greatest measurement accuracy, calibrate at the highest possible power below the onset of receiver compression. This power level depends on the frequency of the response being measured. (See your analyzer's Technical Specifications for the maximum input power to a receiver.)

Analyzers with front-panel jumpers only: Receiver power levels can be measured with a power meter where your external circuitry connects to:

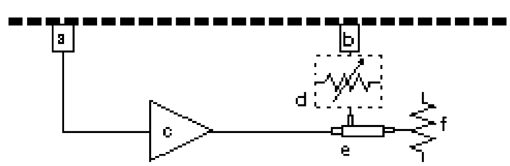
- Rcvr A In channel or (for E835xA) A In channel
- Rcvr A In channel or (for E835xA) B In channel

During calibration (when the DUT is removed) adjust the power to get a power level at the receivers that safely minimizes receiver compression. During measurements (when the DUT is installed) adjust the power to maintain the safe power level at the receivers.

Components that Reduce DUT Output Power

Maximum input power at the analyzer's receivers must not be exceeded. (See your analyzer's Technical Specifications for the maximum input power to a receiver.) If your DUT has a high output-power level, it must be reduced to a level that can be safely handled by the analyzer's receivers. As shown in the following graphic, when testing a CDMA base-station amplifier, some or all of the following components are configured to reduce amplifier output power:

- An external high-power **coupler** can be installed at the output of the amplifier. A special **load** that can dissipate high power is used to terminate the through-arm of the coupler.
- An external high-power **attenuator** can be attached to the analyzer's input port.



Item	Description	Item	Description
a	PORT 1	**e**	Directional coupler, customer supplied
b	PORT 2	**f**	Load, customer supplied
c	CDMA base station amplifier, customer supplied		
d	Variable attenuator (optional), customer supplied. (If your E835x PNA is equipped with Option 015, or any other PNA is equipped with Option 016, it has receiver attenuation and does not require external attenuation. See Analyzer Options.)		

To protect your analyzer's receiver from too much input power, follow these steps to determine values of an external attenuator (or an internal receiver attenuator, if equipped with Option 015 or 016) or coupler and load (or both).

1. Determine the maximum power out of the amplifier [Ex: 40 dBm).
2. Subtract 15 dB (the coupling-factor value of the directional coupler inside the analyzer) [Ex: 40 dBm - 15dBm = 25 dBm).
3. Subtract the power level required at the receiver [Ex: 25 dBm - (-10 dBm)].
4. The result is the appropriate value of the external attenuator (or an internal receiver attenuator, if equipped with Option 015 or 016) or coupler and load or both [Ex: 35 dB].

Note: Refer to the specifications for the coupler, load, and attenuator to ensure they have adequate power-handling capability.

Find more information about testing CDMA amplifiers at **http://www.tm.agilent.com**. Click "Library" and use the search function to search for CDMA amplifiers.

Calibration Standards that Withstand High Power

- The line, open, short, and thru standards are not a problem with high-power levels since they do not dissipate any energy.
- The load dissipates energy. To prevent causing damage to the load, read its specifications to verify it can handle the power level.

CAUTION: When using an open or a short during calibration, be sure that the incident power level is low enough so that the reflected signal does not exceed the maximum input to the analyzer's receiver. See your analyzer's Technical Specifications for the damage level of a receiver.

If you are using an Electronic Calibration (ECal) Module, refer to its specifications for maximum input power.

Thermal Requirements for DUTs and Loads

Thermal conditions during the high-power measurement should also be considered for accurate measurements. For example:

- The high-power DUT may respond very differently at various temperatures. The tests should be done when the amplifier is at the desired temperature.
- The load should be allowed to temperature-stabilize if its impedance characteristics change significantly versus temperature.

Stimulus-Sweep Recommendation for a DUT with an AGC Loop

It is recommended that you set the analyzer to sweep across only one frequency band if you are testing an amplifier with an AGC loop. This will require a separate measurement across each of the required frequency bands, but it will ensure the safety of your amplifier.

Turning off the amplifier's source power is a problem for amplifiers (DUTs) containing an AGC (Automatic-Gain Control) loop. The problem occurs when the analyzer's source power sweeps through its band-crossings. The analyzer keeps power constant while sweeping or retracing if only **one** frequency band is used. If switching frequency bands, the analyzer blanks, or turns off, the source power as it sweeps or retraces.

When the analyzer's source power is blanked, the AGC loop of the amplifier tries to compensate by increasing its gain to keep the output power level constant. When blanking stops and the sweep begins again, the analyzer restores its signal, and suddenly there is power again at the input of the amplifier, which has ramped-up its gain. If the AGC loop cannot respond quickly enough, the momentary high output power can damage or destroy the amplifier under test and the analyzer's receiver.

View your analyzer's frequency band crossings.

Procedure for Measuring High-Power Amplifiers

Caution: Avoid expensive repairs to your PNA. Read Electrostatic Discharge Protection.

1. Select the correct analyzer configuration for your DUT and your measurement requirements. Do not connect the DUT yet.
2. Preset the analyzer.
3. If necessary, adjust the analyzer's source power:
    - Set the analyzer's source power to be in the linear region of the amplifier's output response (typically 10 dB below the 1 dB compression point).
    - Select an external attenuator (if needed) so the amplifier's output power will be sufficiently attenuated to avoid causing receiver compression or damage to the analyzer.
4. Select the desired measurement.
5. Connect the DUT.
6. Select the settings for the DUT.
7. Remove the DUT and perform a measurement calibration. Be sure to include any booster amplifiers, cables, attenuators, or couplers that will be used when measuring the DUT.
8. Save the instrument state to memory.
9. Reconnect the DUT.
10. Scale the displayed measurement for optimum viewing and use markers to read-out measurement results.

11. Print the data or save it to a disk.

Diagram: (E835x) Basic Configuration

Explore the graphic with your mouse.

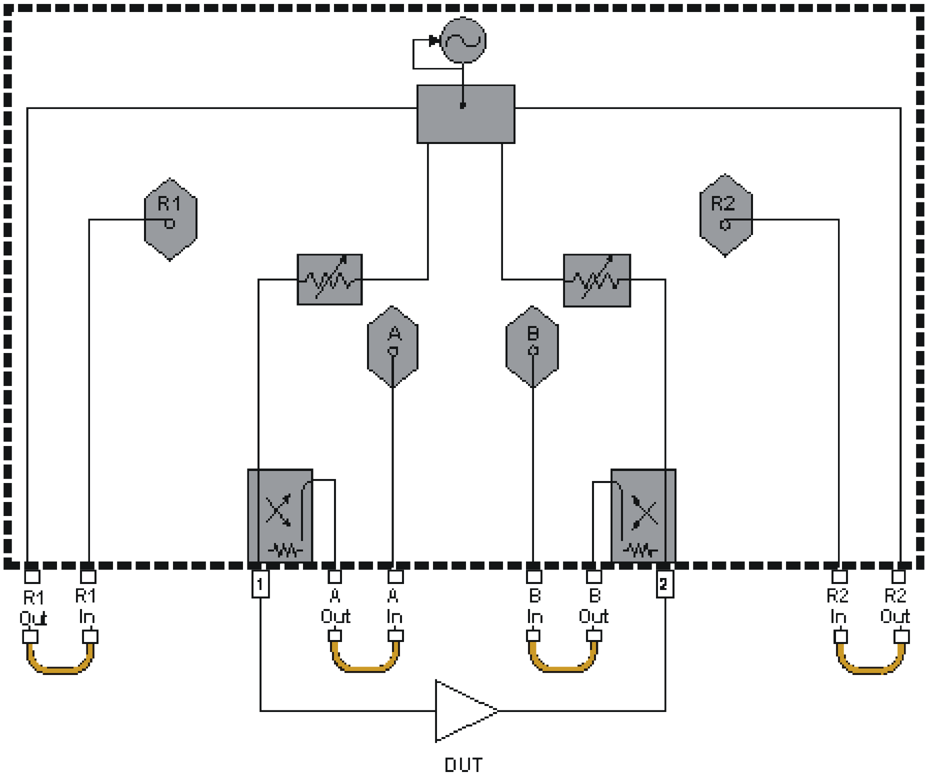


Diagram: (E835x) Configuration for High-Power Measurements in the Forward Direction (Single-Port Cal Only)

Explore the graphic with your mouse.

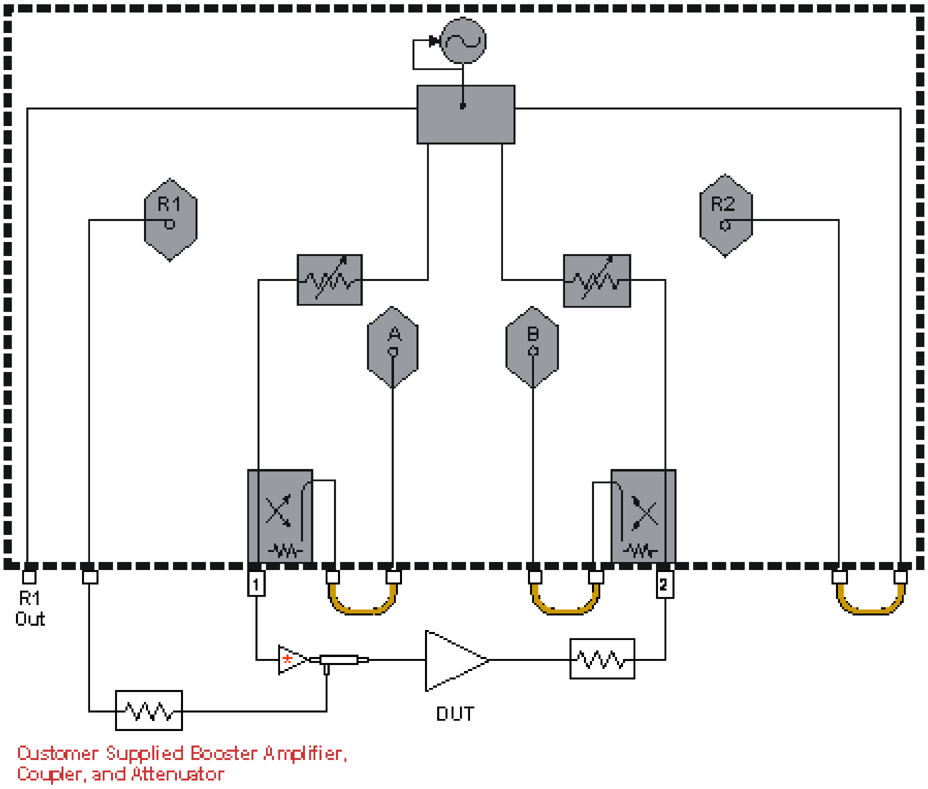


Diagram: (E835x) Configuration for High-Power Measurements in the Forward Direction (2-Port Cal)

Explore the graphic with your mouse.

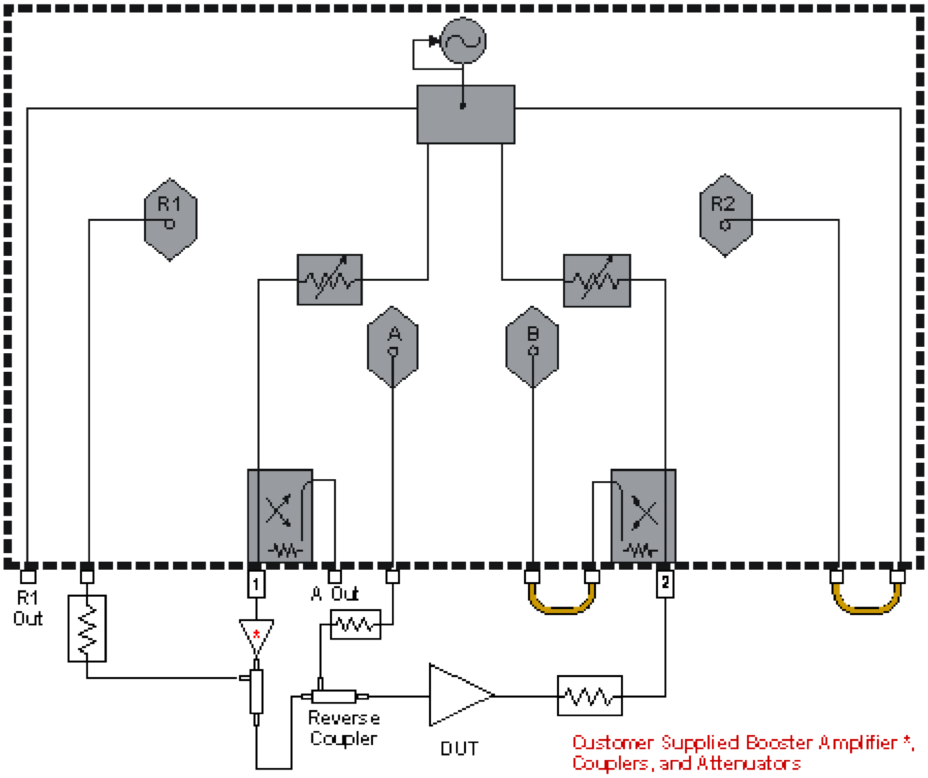


Diagram: (E835x) Configuration for Non-Boosted High-Power Measurements in both Forward & Reverse Directions

Explore the graphic with your mouse.

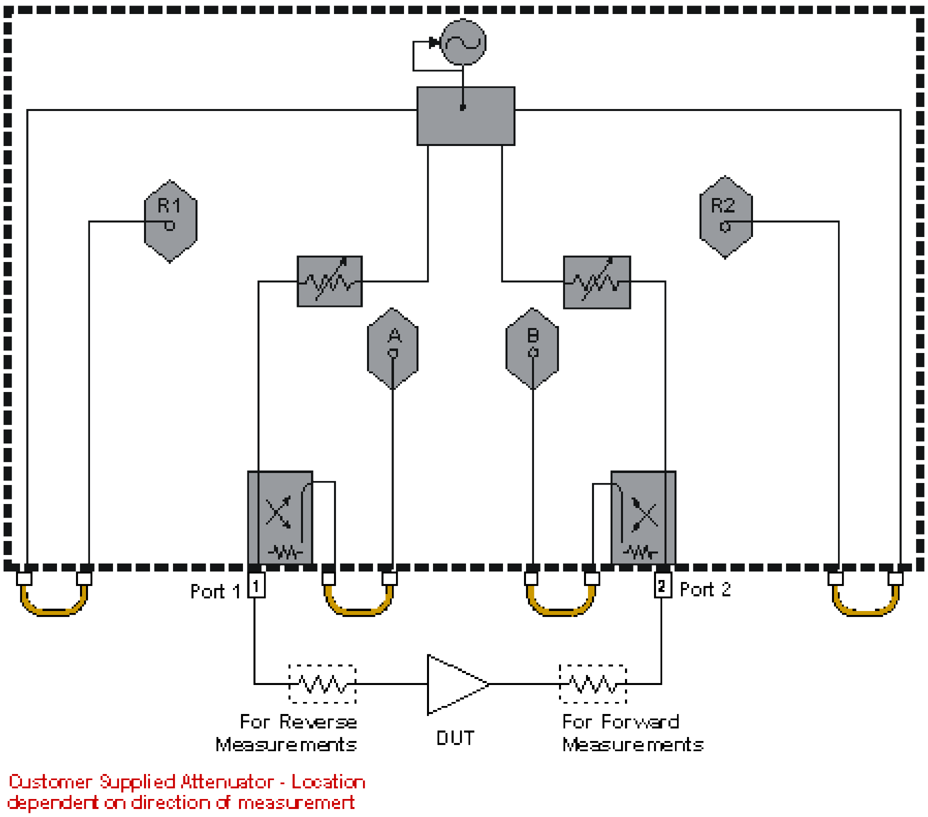


Diagram: (E835x) Configuration for High-Power Measurements in both the Forward & Reverse Directions (2-Port Cal)

Explore the graphic with your mouse.

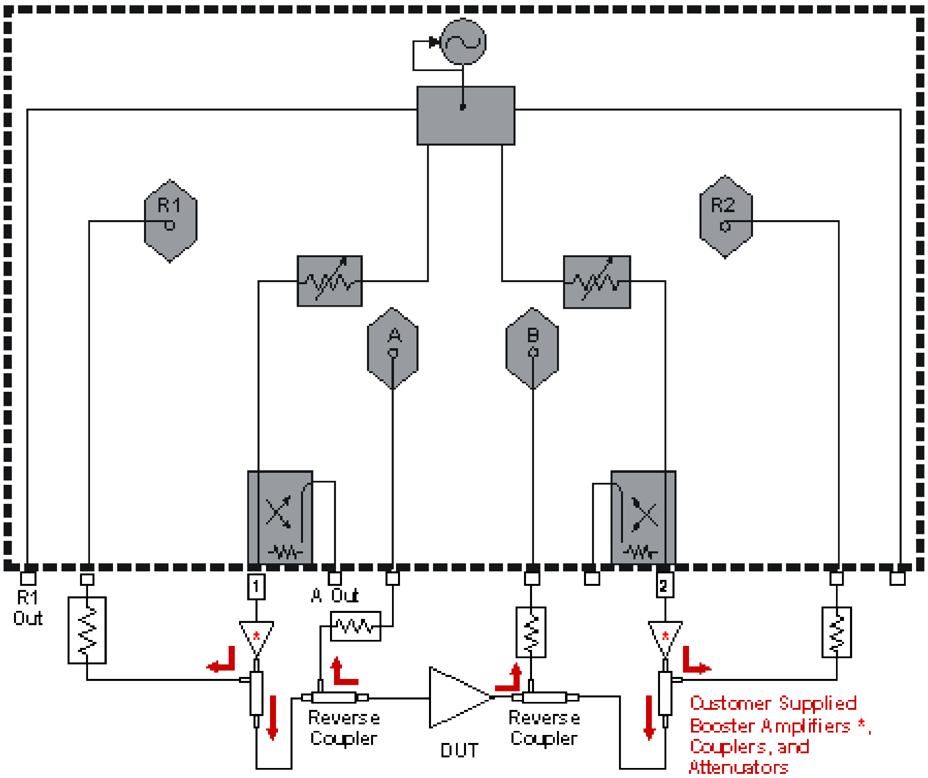


Diagram: (E835x) Configuration using One Booster Amplifier

Explore the graphic with your mouse.

Note: the circuitry inside the yellow lines contains the following components supplied by you - a booster amplifier, a forward coupler, and an attenuator.

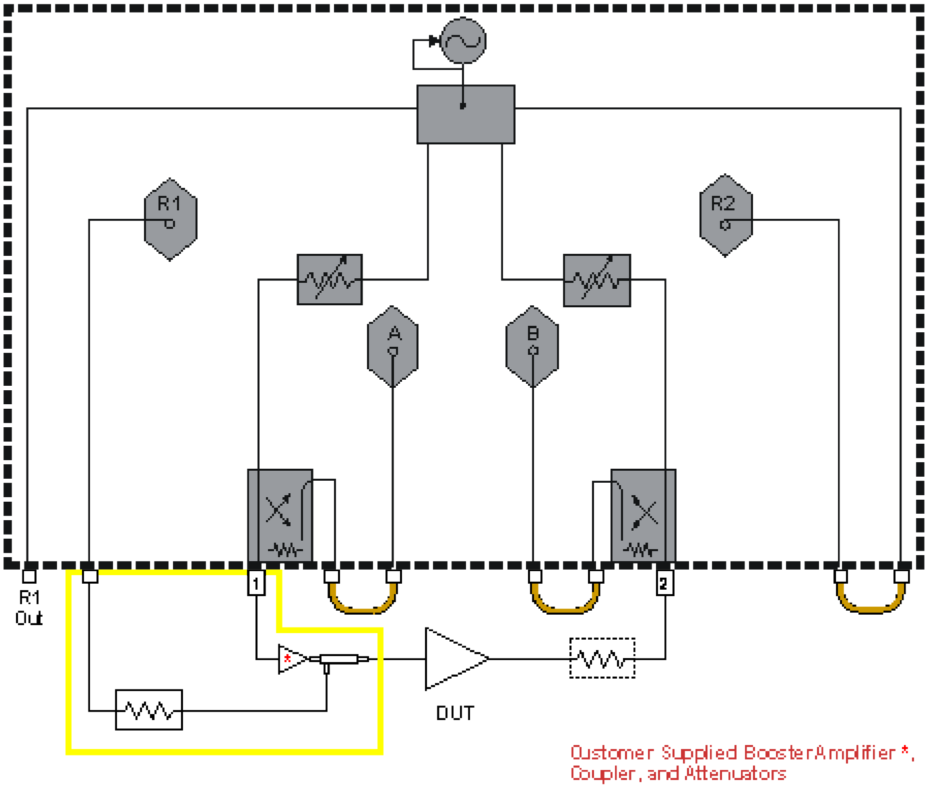


Diagram: (E835x) Configuration using One Booster Amplifier and a Reverse Coupler

Explore the graphic with your mouse.

Note: the circuitry inside the yellow lines contains the following components supplied by you - a booster amp, a forward coupler, a reverse coupler (allows S11 measurements), and two attenuators.

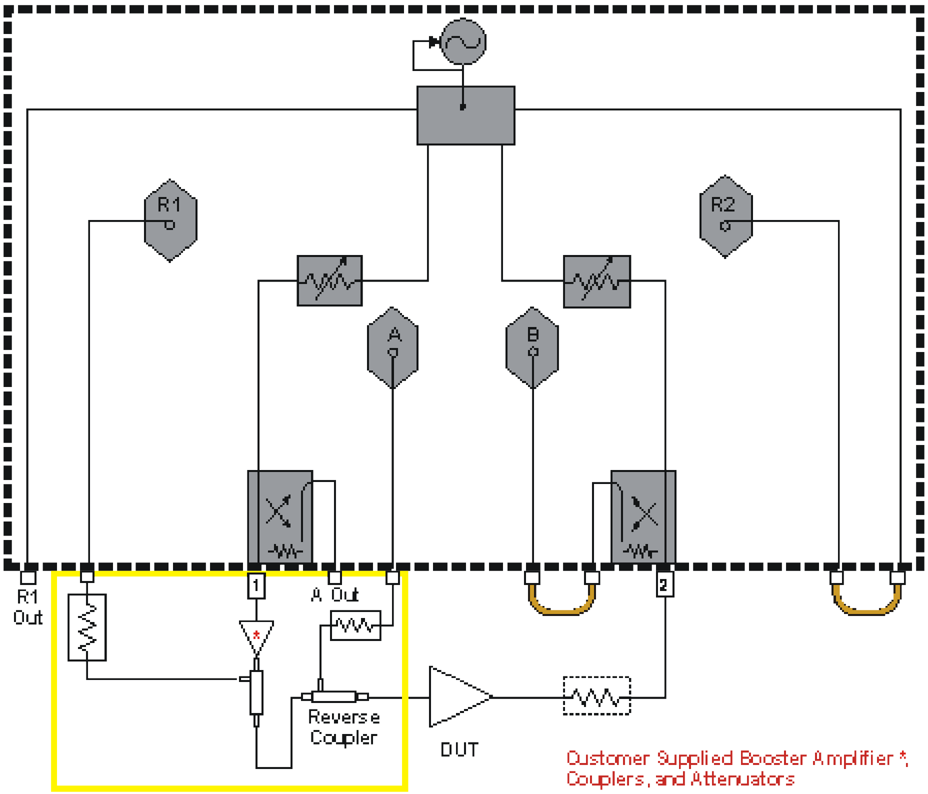


Diagram: (E835x) Configuration using Two Booster Amplifiers

Explore the graphic with your mouse.

Note: the circuitry inside the yellow lines contains the following components supplied by you - two booster amps, 2 forward couplers, 2 reverse couplers (allows S11 and S22 measurements), and 4 attenuators.

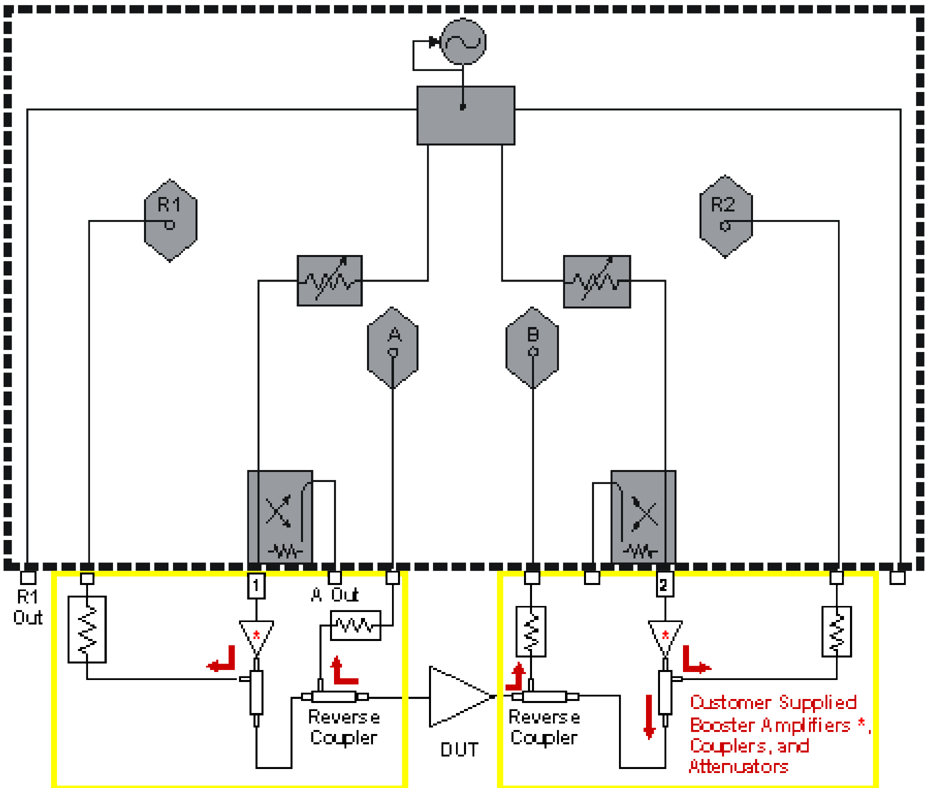


Diagram: (E835x) External Leveling using One Booster Amplifier

Explore the graphic with your mouse.

Note: the circuitry inside the yellow lines contains the following components supplied by you - a booster amplifier, forward coupler, power splitter, two attenuators, and a negative detector diode.

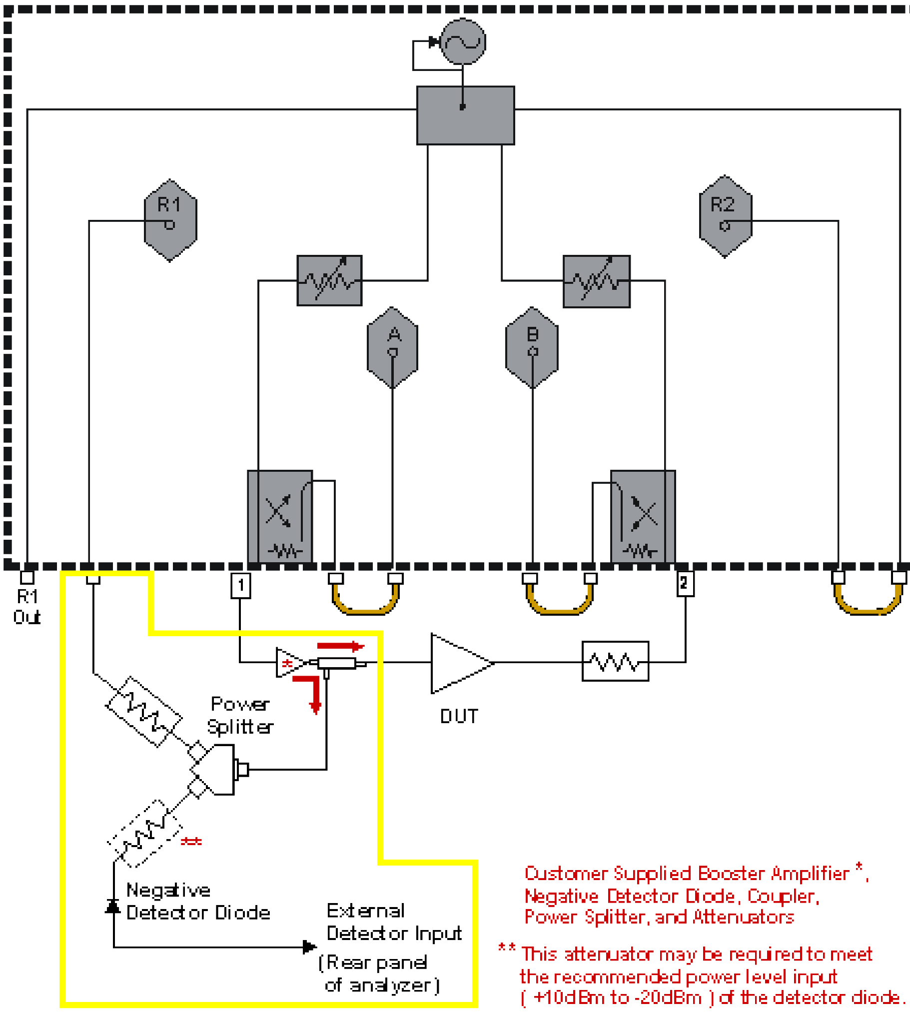


Diagram: (E835x) External Leveling using One Booster Amplifier and a Reverse Coupler

Explore the graphic with your mouse.

Note: the circuitry inside the yellow lines contains the following components supplied by you - a booster amp, forward coupler, reverse coupler (allows S11 measurements), power splitter, three attenuators, and a negative detector diode.

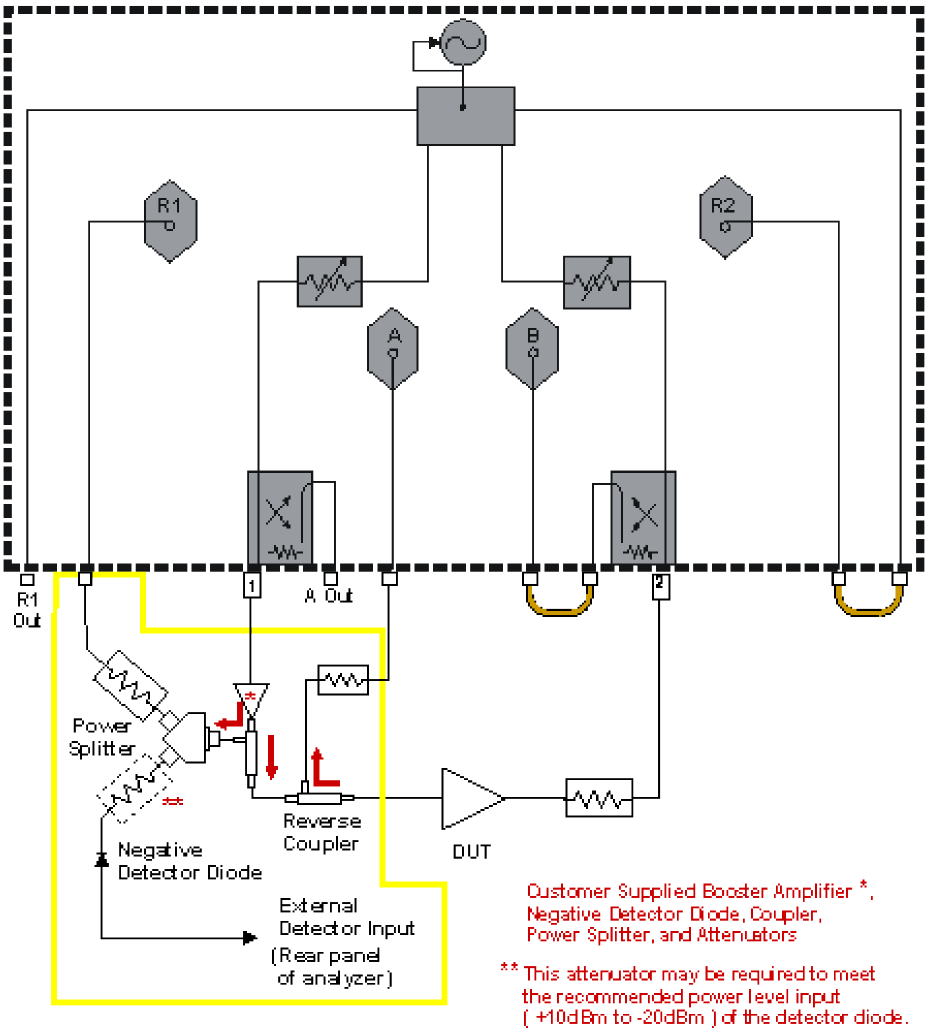


Diagram: (E835x) External Leveling using Two Booster Amplifiers and Two Reverse Couplers

Explore the graphic with your mouse.

Note: the circuitry inside the yellow lines contains the following components supplied by you - 2 booster amps, 2 forward couplers, 2 reverse couplers (allow S11 and S22 measurements), 2 power splitters, 6 attenuators, and a negative detector diode.

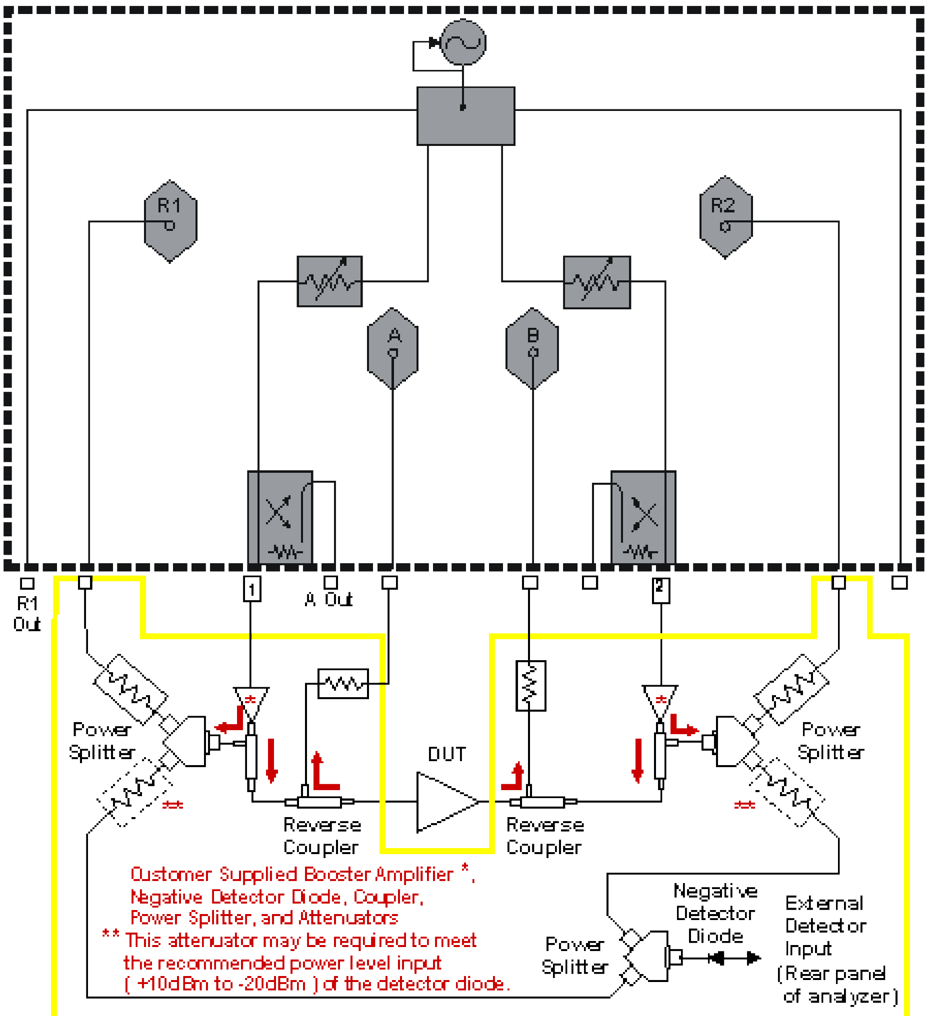
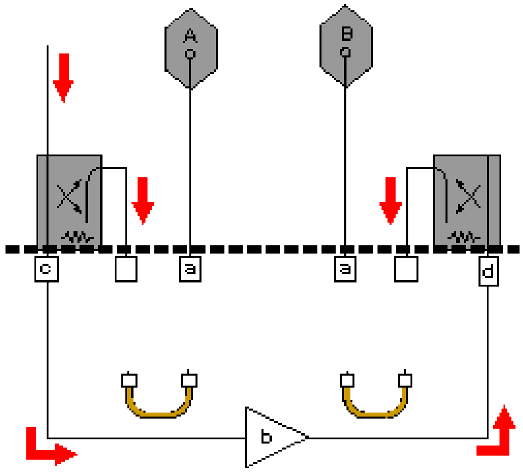


Diagram: (E835x) Measuring Calibration Power Levels



Item	Description
a	Measure the receiver input power where your external circuitry connects here
b	DUT, customer supplied
c	PORT 1
d	PORT 2

Diagram: (E835x, Option 015) Basic Configuration

Explore the graphic with your mouse.

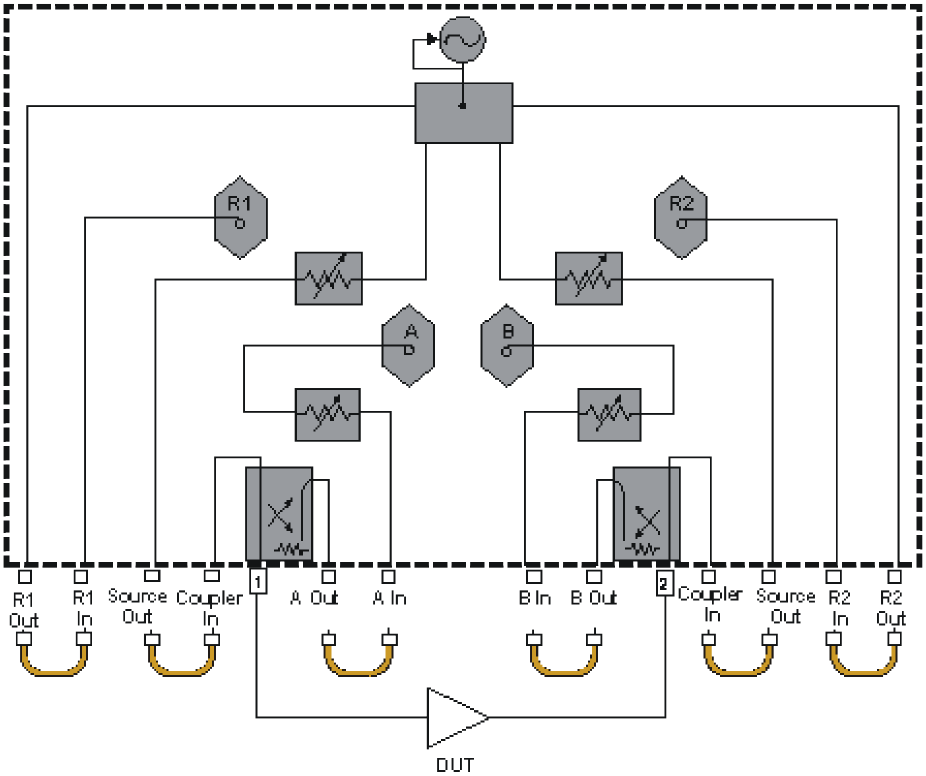


Diagram: (E835xA, Option 015) Configuration for High-Power Measurements in the Forward Direction

Explore the graphic with your mouse.

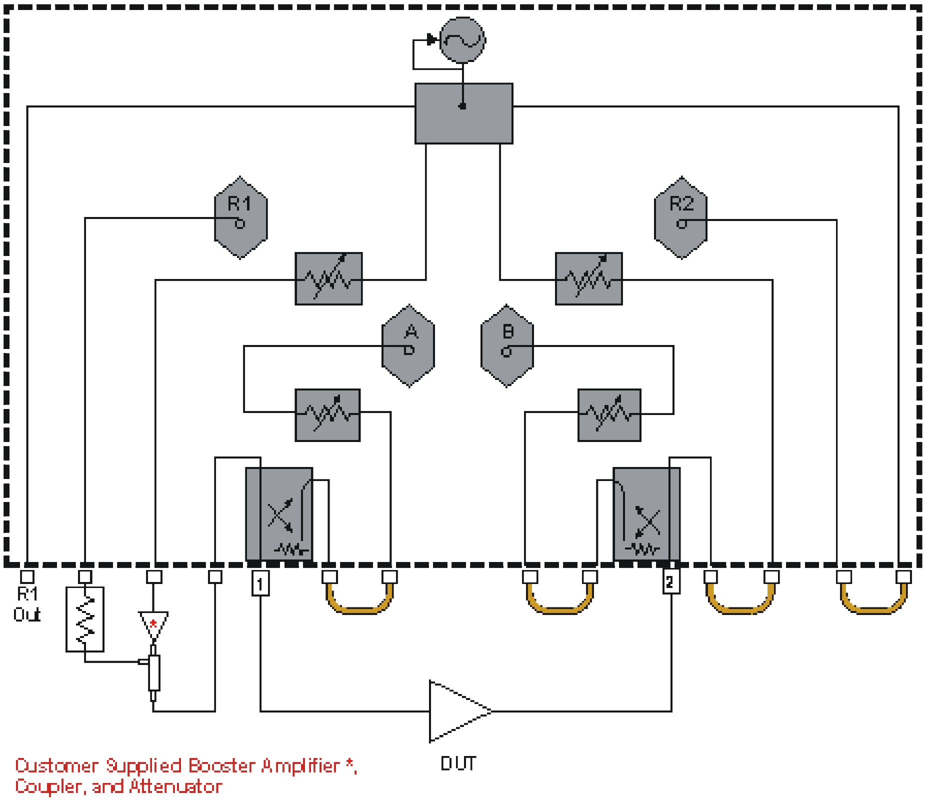


Diagram: (E835xA, Option 015) Configuration (with an Isolator) for High-Power Measurements in the Forward Direction

Explore the graphic with your mouse.

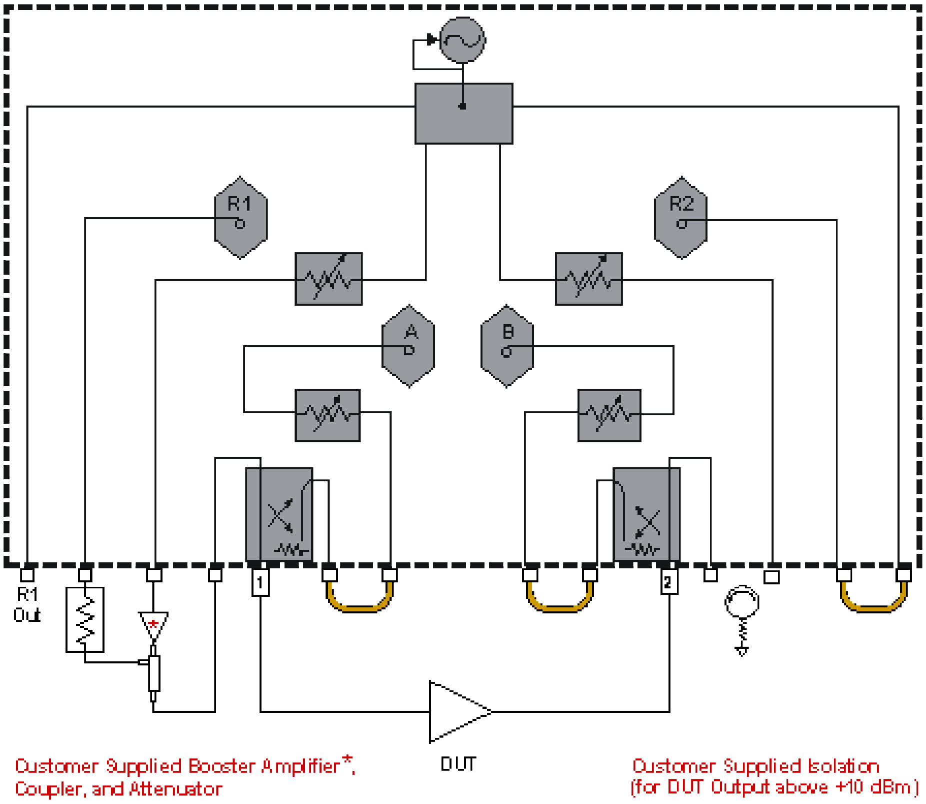


Diagram: (E835xA, Option 015) Configuration for High-Power Measurements in both Forward & Reverse Directions

Explore the graphic with your mouse.

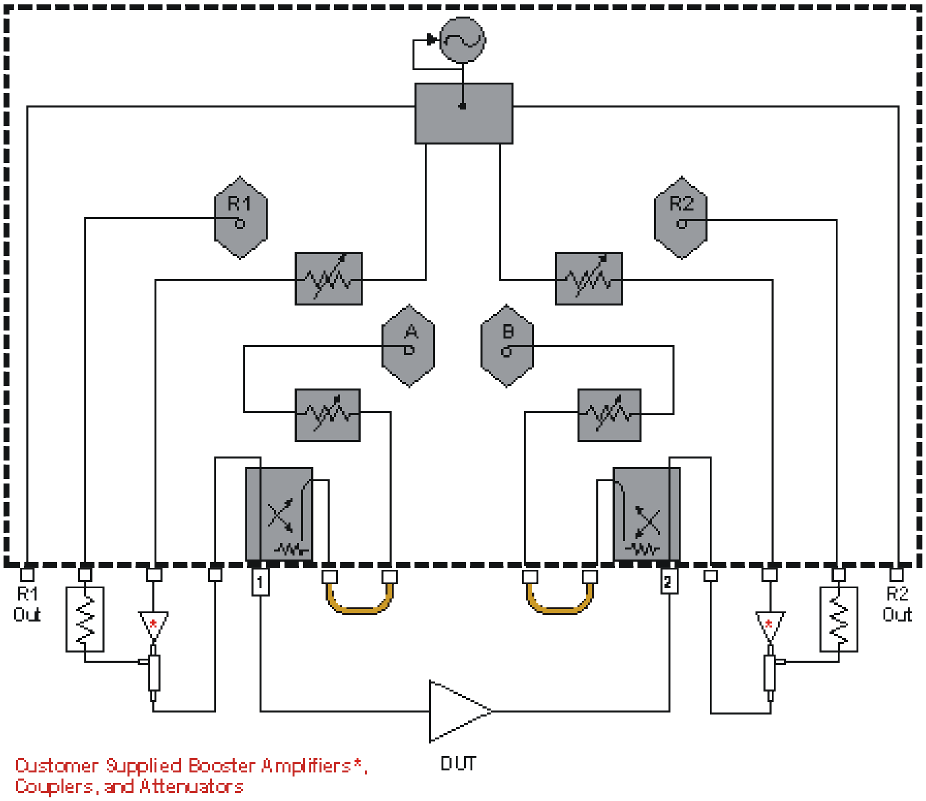


Diagram: (E835x, Option 015) Configuration using One Booster Amplifier

Explore the graphic with your mouse.

Note: the circuitry inside the yellow lines contains the following components supplied by you: a booster amplifier, a forward coupler, and an attenuator.

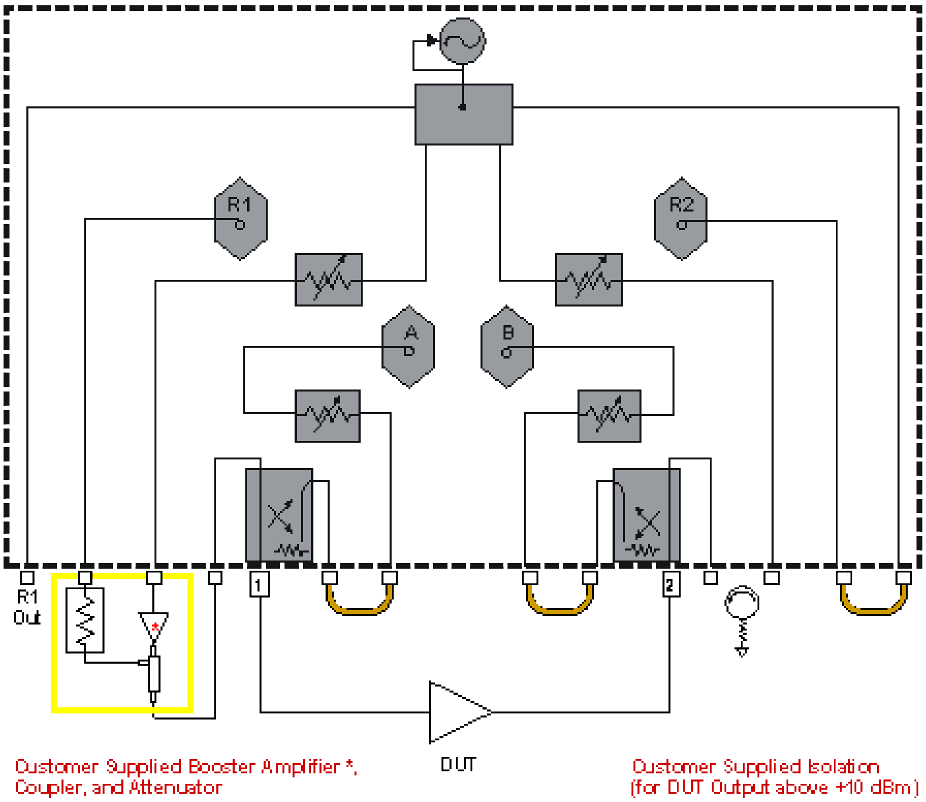


Diagram: (E835xA, Option 015) Configuration Using Two Booster Amplifiers

Explore the graphic with your mouse.

Note: the circuitry inside the yellow lines contains the following components supplied by you: two booster amps, two forward couplers, and two attenuators.

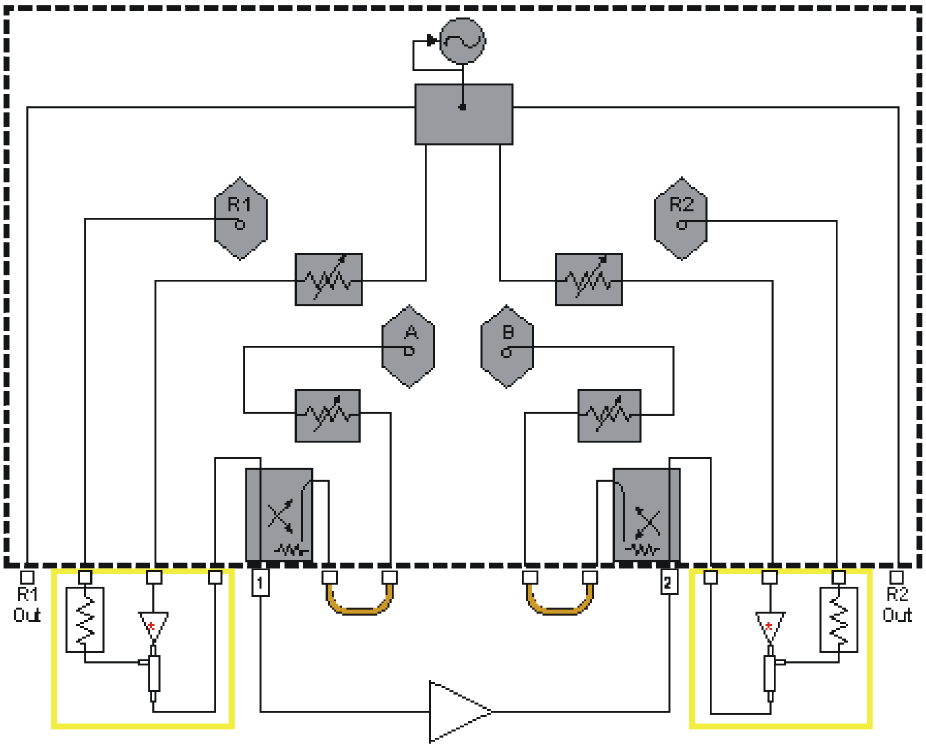


Diagram: (E835x, Option 015) External Leveling using One Booster Amplifier

Explore the graphic with your mouse.

Note: the circuitry inside the yellow lines contains the following components supplied by you - a booster amplifier, a forward coupler, power splitter, attenuators, and a negative detector diode.

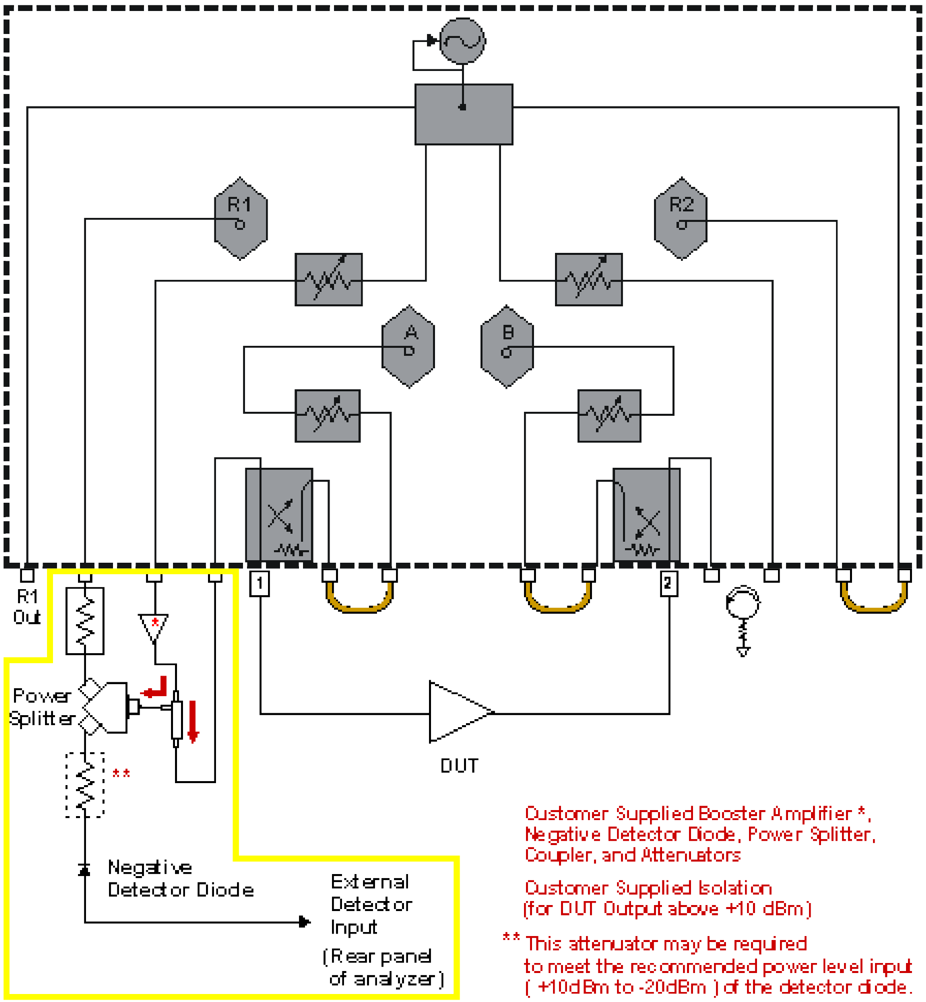
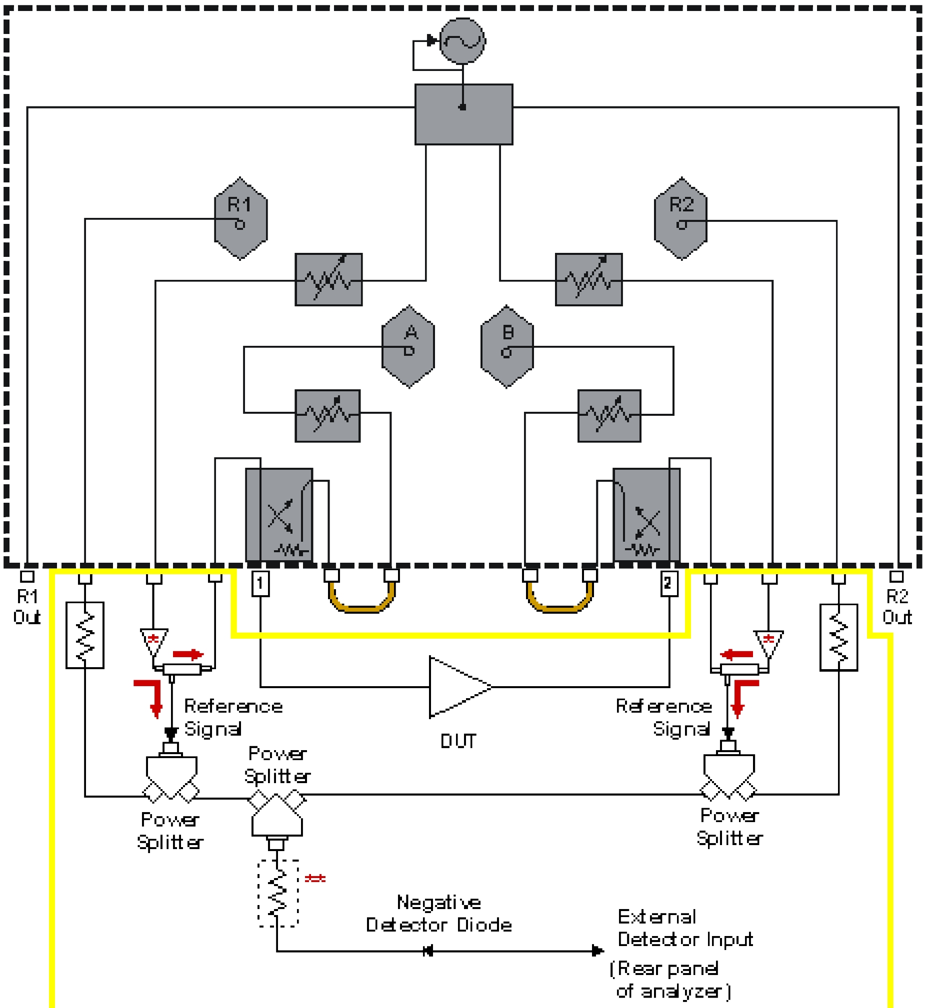


Diagram: (E835xA, Option 015) External Leveling using Two Booster Amplifiers

Explore the graphic with your mouse.

Note: the circuitry inside the yellow lines contains the following components supplied by you: two booster amps, two forward couplers, attenuators, three power splitters, and a negative detector diode.



Customer Supplied Booster Amplifiers *, Negative Detector Diode, Couplers, Power Splitters and Attenuators

** This attenuator may be required to meet the recommended power level input (+10dBm to -20dBm) of the detector diode.

Diagram: (E835xA, Option 015) Measuring Calibration Power Levels

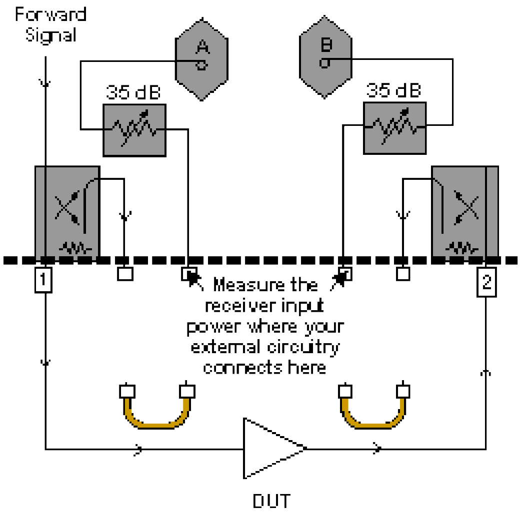
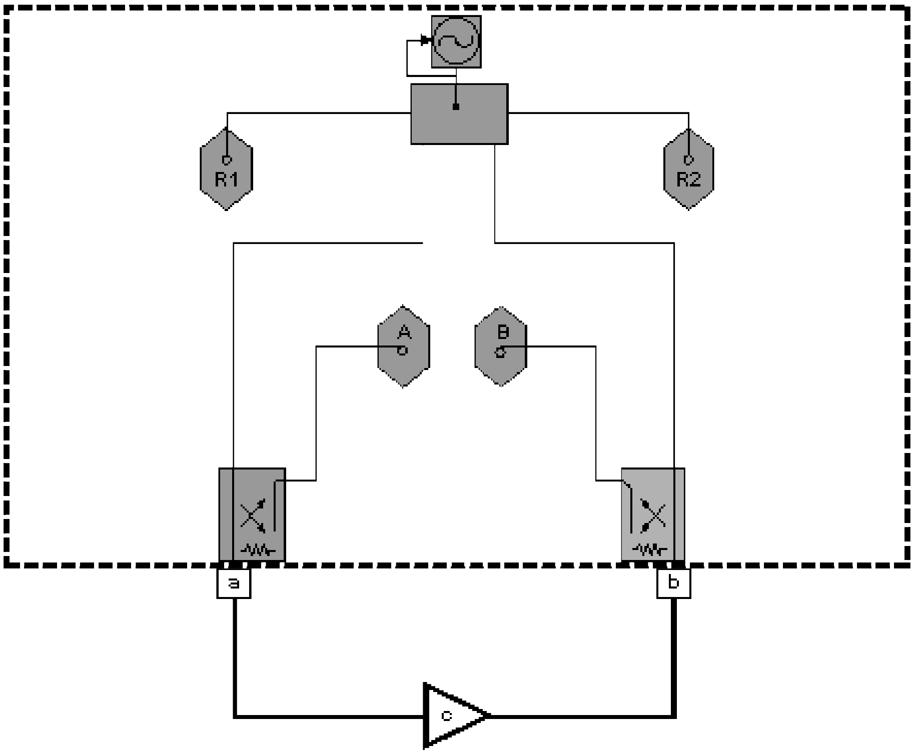


Diagram: Basic Configuration

- **E8361A. E8362A/B, E8363A/B, E8364A/B**
- **N5230A**

Explore the graphic with your mouse.

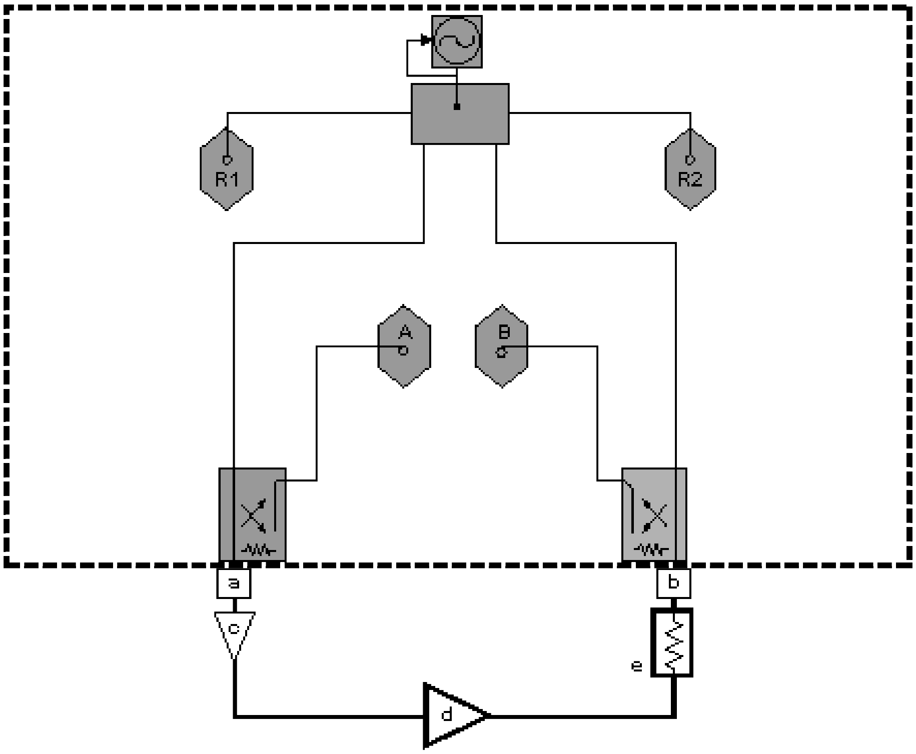


Item	Description
a	PORT 1
b	PORT 2
c	DUT, customer supplied

Diagram: Configuration for High-Power Measurements in the Forward Direction (Single-Port Cal Only)

- **E8361A, E8362A/B, E8363A/B, E8364A/B**
- **N5230A**

Explore the graphic with your mouse.

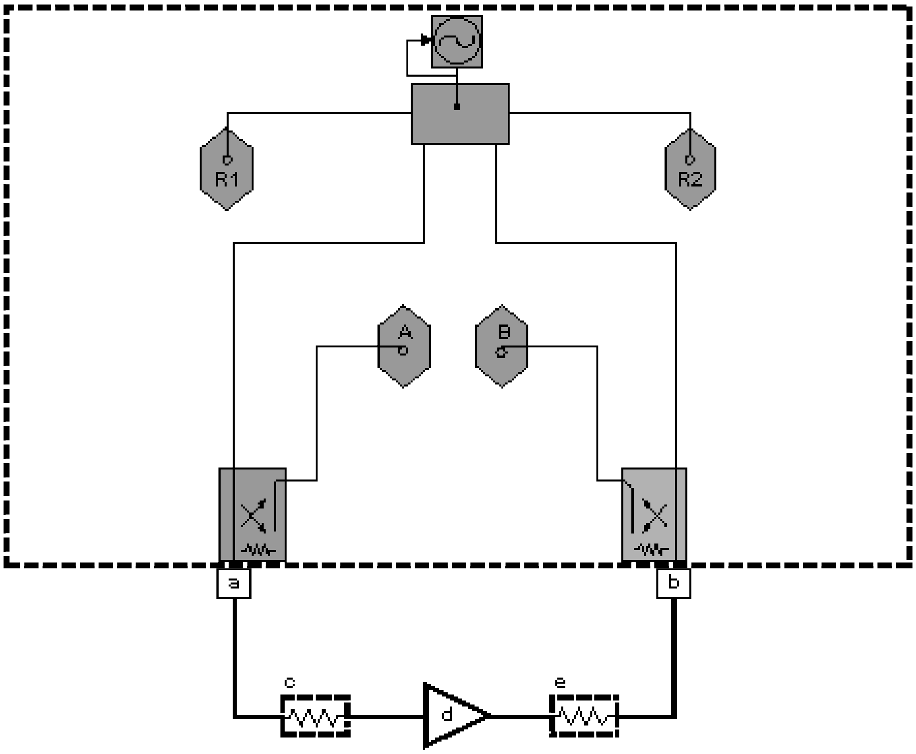


Item	Description	Item	Description
a	PORT 1	**e**	Attenuator (for forward measurements), customer supplied
b	PORT 2		
c	Booster amplifier, customer supplied		
d	DUT, customer supplied		

Diagram: Configuration for Non-Boosted High-Power Measurements in both Forward and Reverse Directions

- **E8361A, E8362A/B, E8363A/B, E8364A/B**
- **N5230A**

Explore the graphic with your mouse.



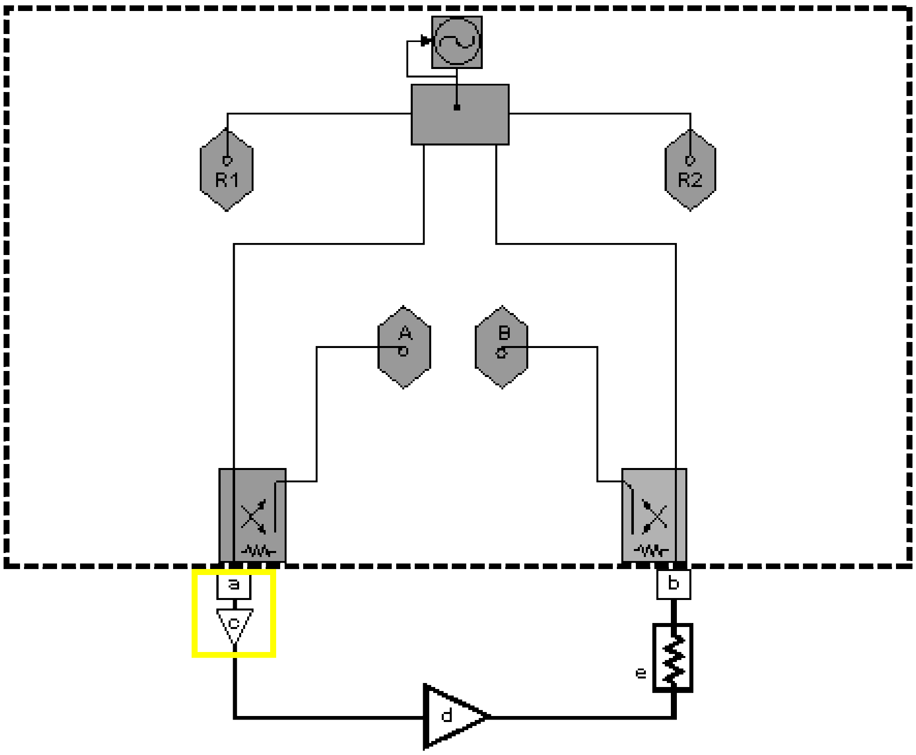
Item	Description	Item	Description
a	PORT 1	**e**	Attenuator (for forward measurements), customer supplied
b	PORT 2		
c	Attenuator (for reverse measurements), customer supplied		
d	DUT, customer supplied		

Diagram: Configuration using a Booster Amp

- **E8361A, E8362A/B, E8363A/B, E8364A/B, N5230A**
- **N5230A**

Explore the graphic with your mouse.

Note: the circuitry inside the yellow lines contains a booster amplifier supplied by you.

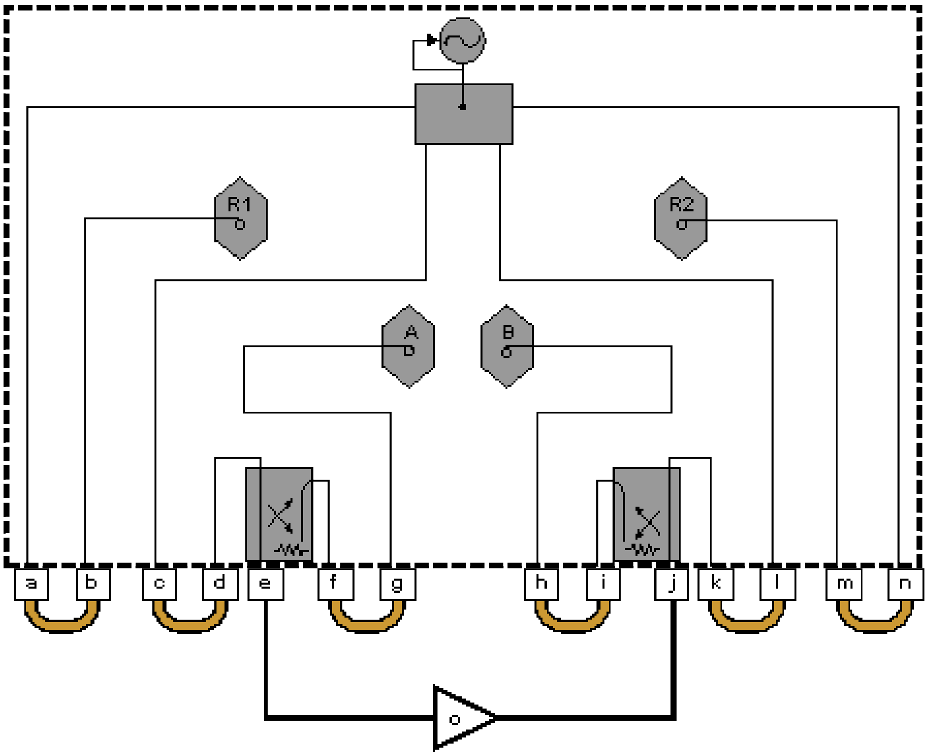


Item	Description	Item	Description
a	PORT 1	**e**	Attenuator (for forward measurements), customer supplied
b	PORT 2		
c	Booster amplifier, customer supplied		
d	DUT, customer supplied		

Diagram: Basic Configuration

- **E8361A, E8362A/B, E8363A/B, E8364A/B, Option 014**
- **N5230A Option 225, or 425, or 525**
- **N5250A**

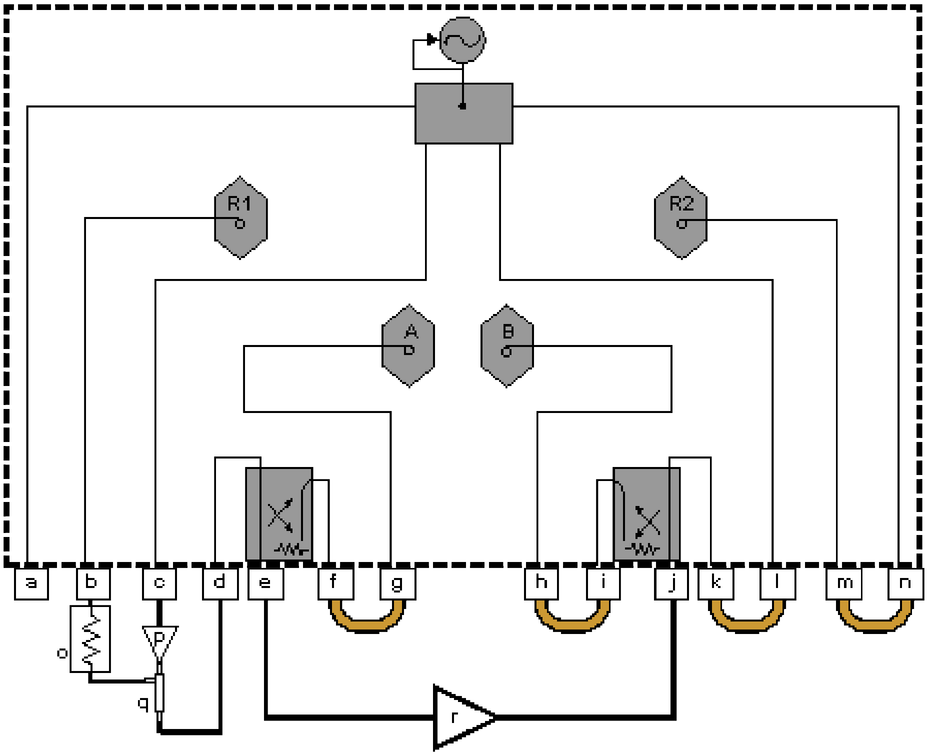
Explore the graphic with your mouse.



Item	Description	Item	Description	Item	Description
a	SOURCE OUT	**h**	RCVR B IN	**o**	DUT, customer supplied
b	RCVR R1 IN	**i**	CPLR ARM		
c	SOURCE OUT	**j**	PORT 2		
d	CPLR THRU	**k**	CPLR THRU		
e	PORT 1	**l**	SOURCE OUT		
f	CPLR ARM	**m**	RCVR R2 IN		
g	RCVR A IN	**n**	SOURCE OUT		

Diagram: (E8361A, E8362A/B, E8363A/B, E8364A/B, Option 014) Configuration for High-Power Measurements in the Forward Direction

Explore the graphic with your mouse.

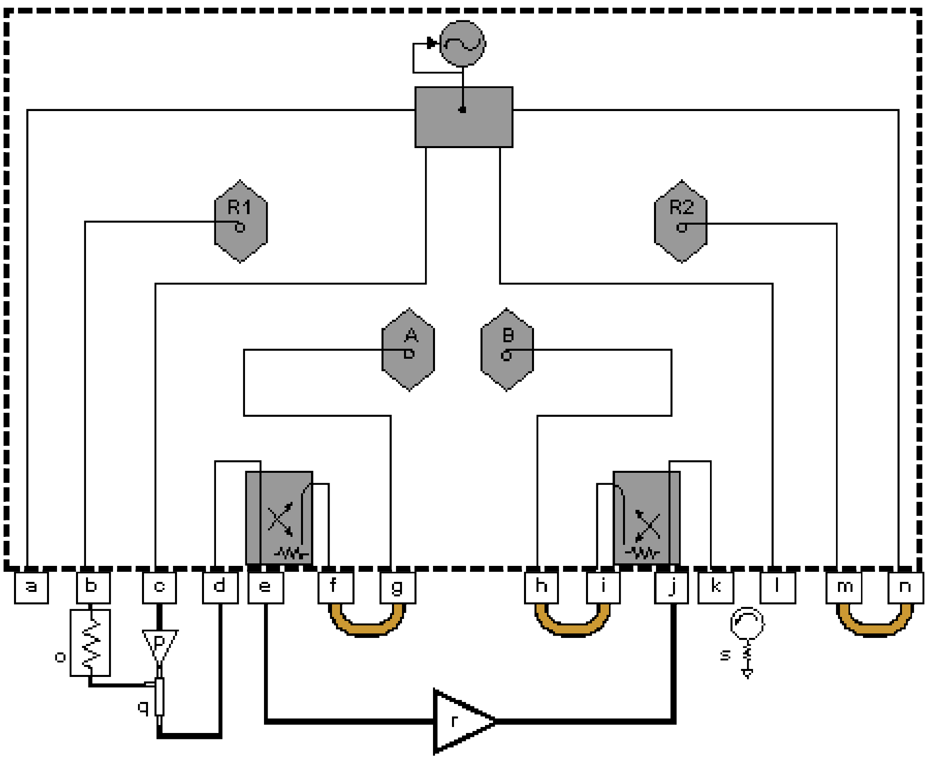


Item	Description	Item	Description	Item	Description
a	SOURCE OUT	**h**	RCVR B IN	**o**	Attenuator, customer supplied
b	RCVR R1 IN	**i**	CPLR ARM	**p**	Booster amplifier, customer supplied
c	SOURCE OUT	**j**	PORT 2	**q**	Directional coupler, customer supplied
d	CPLR THRU	**k**	CPLR THRU	**r**	DUT, customer supplied
e	PORT 1	**l**	SOURCE OUT		
f	CPLR ARM	**m**	RCVR R2 IN		
g	RCVR A IN	**n**	SOURCE OUT		

Diagram: Configuration (with an Isolator) for High-Power Measurements in the Forward Direction

- **E8361A, E8362A/B, E8363A/B, E8364A/B, Option 014**
- **N5230A, Option 225, or 425, or 525**
- **N5250A**

Explore the graphic with your mouse.



Item	Description	Item	Description	Item	Description
a	SOURCE OUT	**h**	RCVR B IN	**o**	Attenuator, customer supplied
b	RCVR R1 IN	**i**	CPLR ARM	**p**	Booster amplifier, customer supplied
c	SOURCE OUT	**j**	PORT 2	**q**	Directional coupler, customer supplied
d	CPLR THRU	**k**	CPLR THRU	**r**	DUT, customer supplied
e	PORT 1	**l**	SOURCE OUT	**s**	Isolator, customer supplied (for DUT output above +10 dBm)
f	CPLR ARM	**m**	RCVR R2 IN		

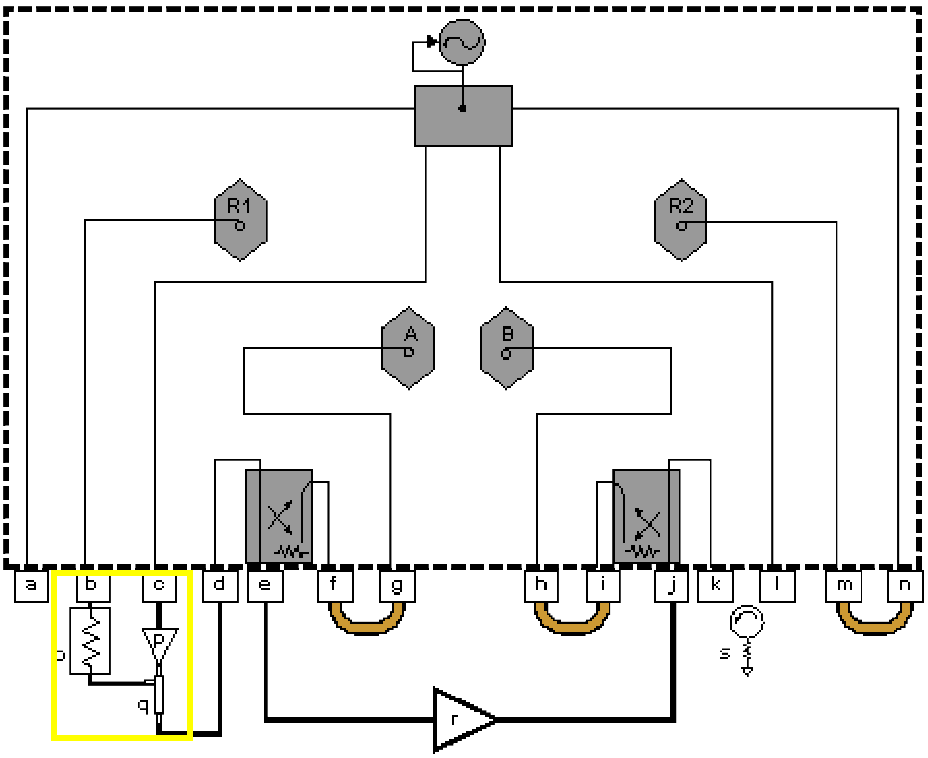
g	RCVR A IN	**n**	SOURCE OUT

Diagram: Configuration using One Booster Amplifier

- **E8361A, E8362A/B, E8363A/B, E8364A/B, Option 014**
- **N5230A, Option 225, or 425, or 525**
- **N5250A**

Explore the graphic with your mouse.

Note: the circuitry inside the yellow lines contains the following components supplied by you: a booster amplifier, a forward coupler, and attenuators.



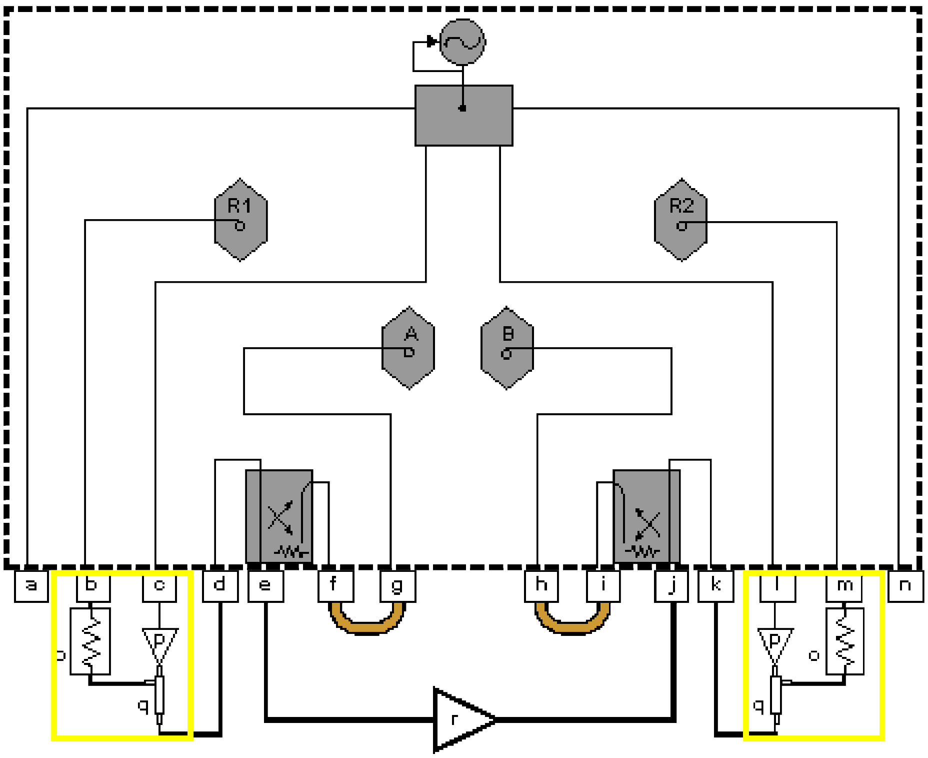
Item	Description	Item	Description	Item	Description
a	SOURCE OUT	h	RCVR B IN	o	Attenuator, customer supplied
b	RCVR R1 IN	i	CPLR ARM	p	Booster amplifier, customer supplied
c	SOURCE OUT	j	PORT 2	q	Directional coupler, customer supplied
d	CPLR THRU	k	CPLR THRU	r	DUT, customer supplied
e	PORT 1	l	SOURCE OUT	s	Isolator, customer supplied (for DUT output above +10 dBm)
f	CPLR ARM	m	RCVR R2 IN		
g	RCVR A IN	n	SOURCE OUT		

Diagram: Configuration Using Two Booster Amplifiers

- **E8361A, E8362A/B, E8363A/B, E8364A/B, Option 014**
- **N5230A, Option 225, or 425, or 525**
- **N5250A**

Explore the graphic with your mouse.

Note: the circuitry inside the yellow lines contains the following components supplied by you: two booster amps, two forward couplers, and attenuators.

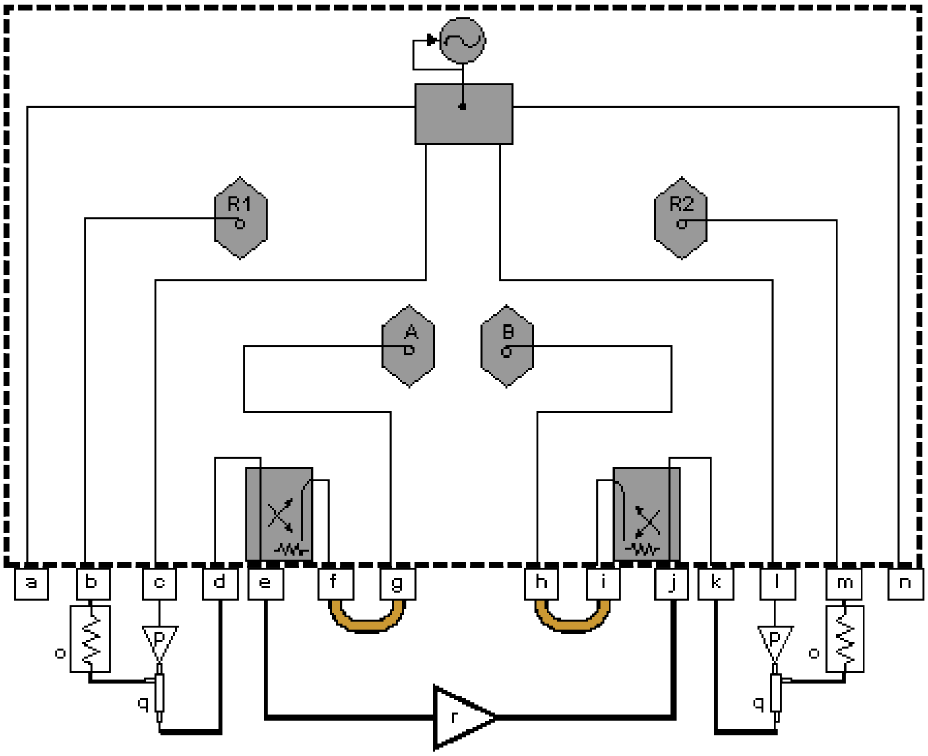


Item	Description	Item	Description	Item	Description
a	SOURCE OUT	h	RCVR B IN	o	Attenuator, customer supplied
b	RCVR R1 IN	i	CPLR ARM	p	Booster amplifier, customer supplied
c	SOURCE OUT	j	PORT 2	q	Directional coupler, customer supplied
d	CPLR THRU	k	CPLR THRU	r	DUT, customer supplied
e	PORT 1	l	SOURCE OUT		
f	CPLR ARM	m	RCVR R2 IN		
g	RCVR A IN	n	SOURCE OUT		

Diagram: Configuration for High-Power Measurements in both Forward & Reverse Directions

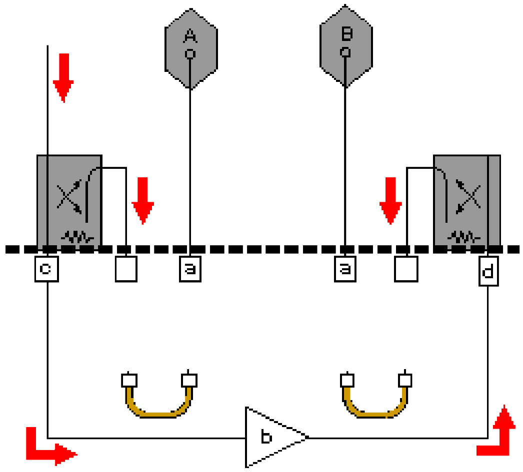
- **E8361A, E8362A/B, E8363A/B, E8364A/B, Option 014**
- **N5230A, Option 225, or 425, or 525**
- **N5250A**

Explore the graphic with your mouse.



Item	Description	Item	Description	Item	Description
a	SOURCE OUT	**h**	RCVR B IN	**o**	Attenuator, customer supplied
b	RCVR R1 IN	**i**	CPLR ARM	**p**	Booster amplifier, customer supplied
c	SOURCE OUT	**j**	PORT 2	**q**	Directional coupler, customer supplied
d	CPLR THRU	**k**	CPLR THRU	**r**	DUT, customer supplied
e	PORT 1	**l**	SOURCE OUT		
f	CPLR ARM	**m**	RCVR R2 IN		
g	RCVR A IN	**n**	SOURCE OUT		

Diagram: Measuring Calibration Power Levels



Item	Description
a	Measure the receiver input power where your external circuitry connects here
b	DUT, customer supplied
c	PORT 1
d	PORT 2

Diagram: (E880xA) Basic Configuration

Explore the graphic with your mouse.

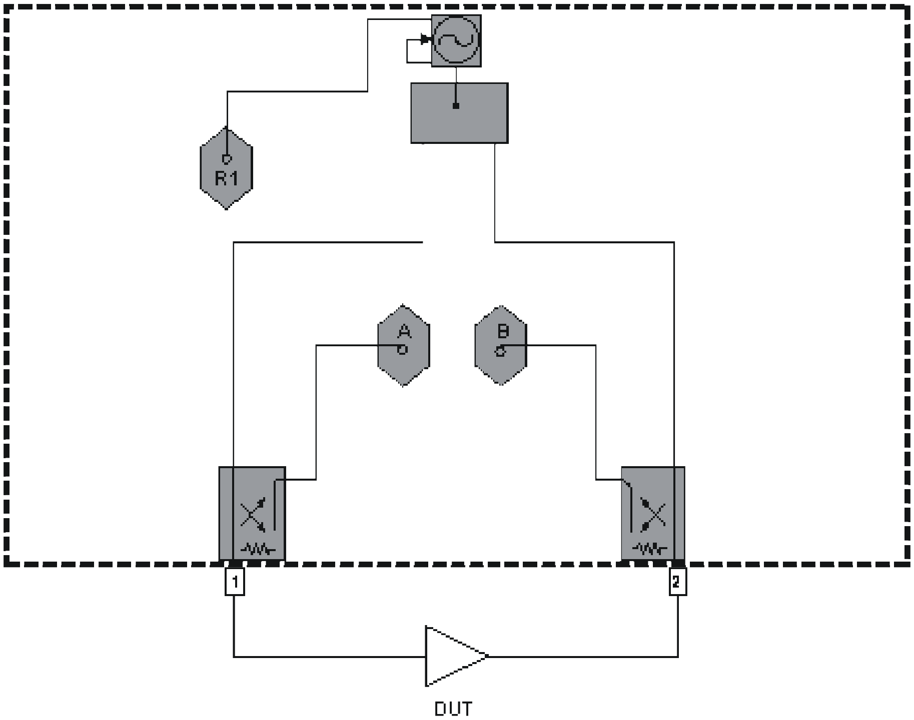


Diagram: (E880xA) Configuration for High-Power Measurements in the Forward Direction (Single-Port Cal Only)

Explore the graphic with your mouse.

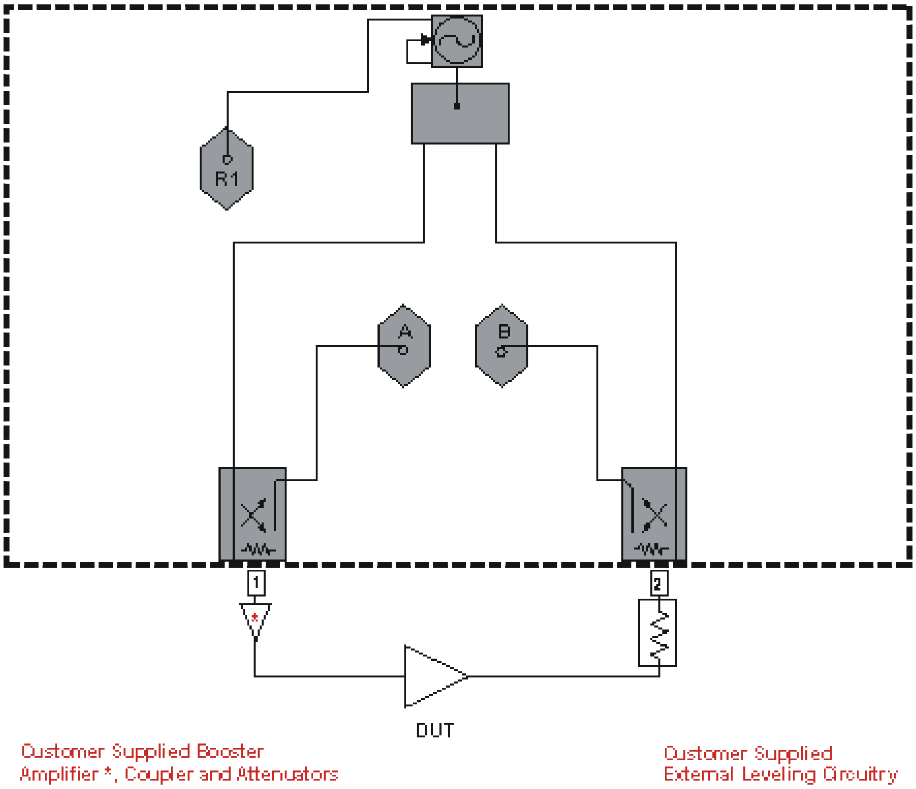


Diagram: (E880xA) Configuration for Non-Boosted High-Power Measurements in both Forward and Reverse Directions

Explore the graphic with your mouse.

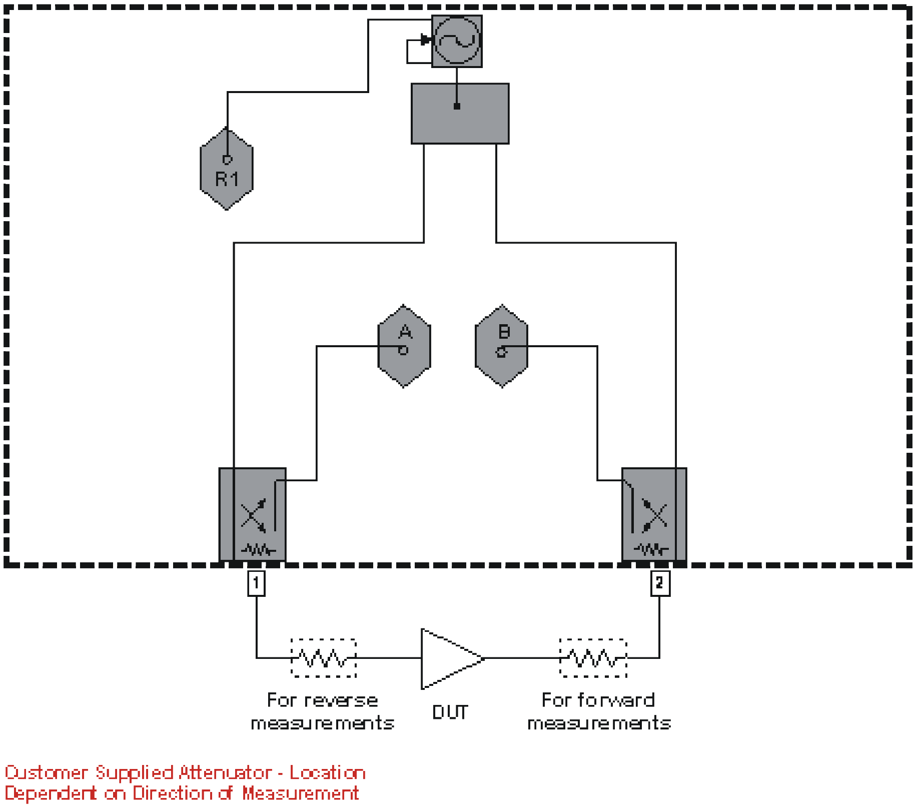


Diagram: (E880xA) Configuration using a Booster Amp

Explore the graphic with your mouse.

Note: the circuitry inside the yellow lines contains a booster amplifier supplied by you.

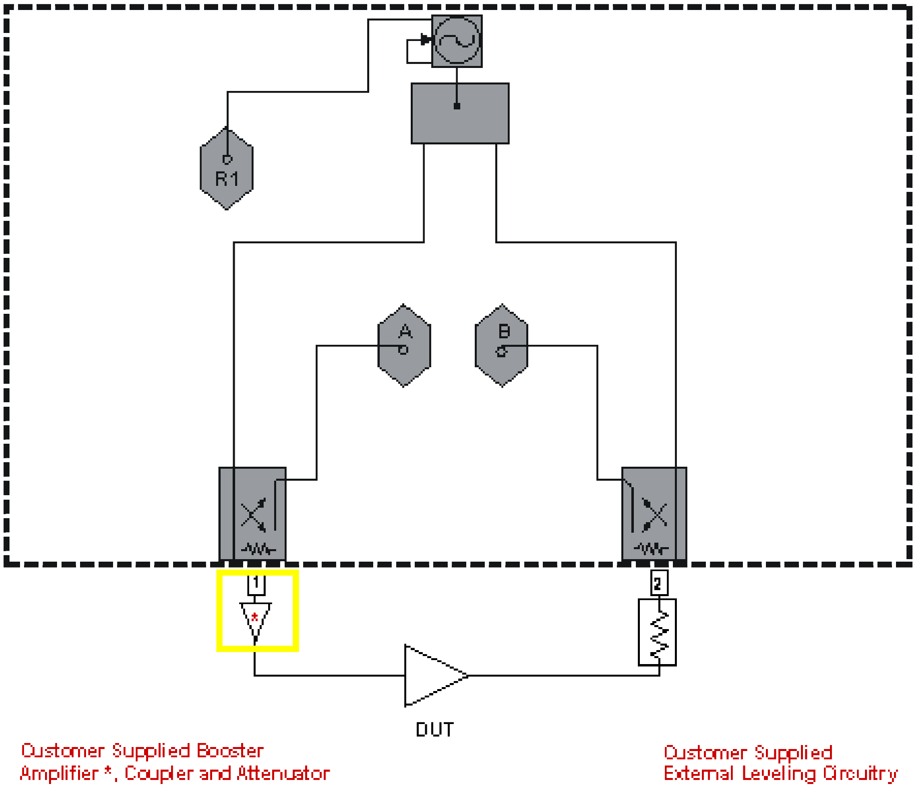


Diagram: (E880xA) External Leveling Configuration

Explore the graphic with your mouse.

Note: the circuitry inside the yellow lines contains the following components supplied by you - a booster amplifier, forward coupler, attenuator, and negative detector diode.

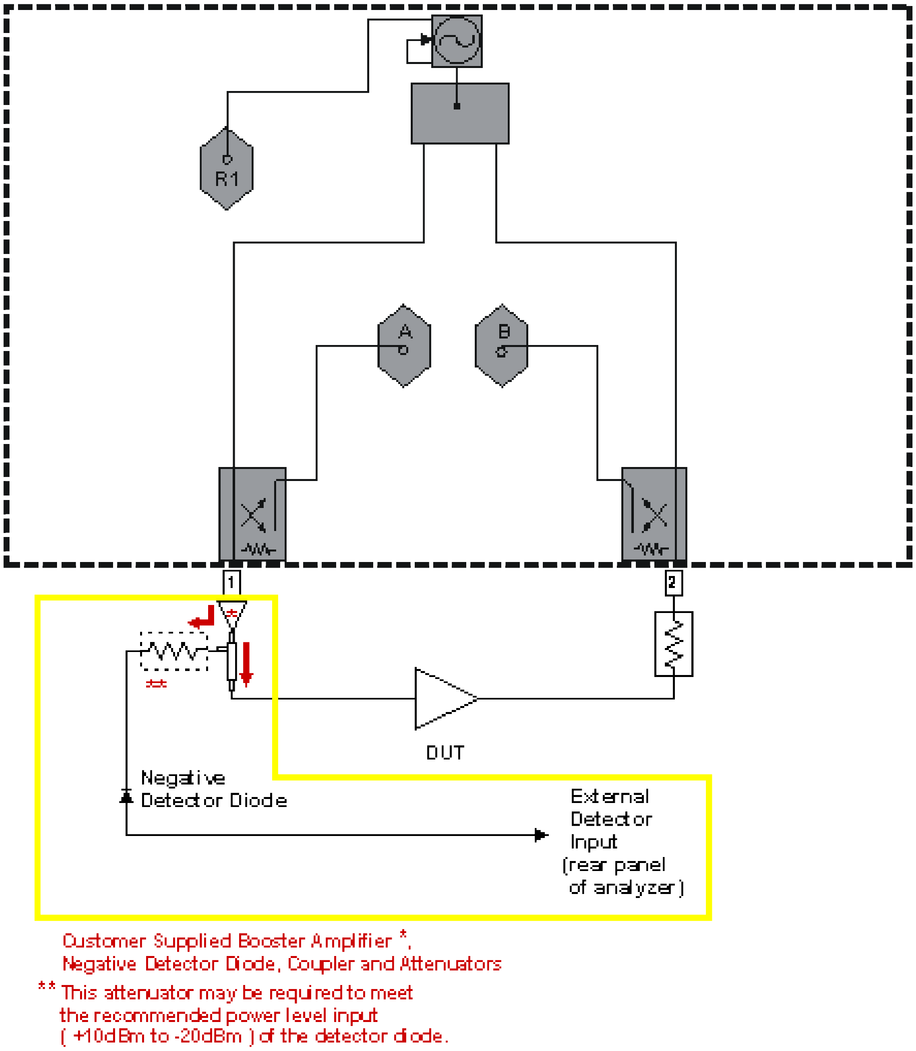


Diagram: (E880xA, Option 014) Basic Configuration

Explore the graphic with your mouse.

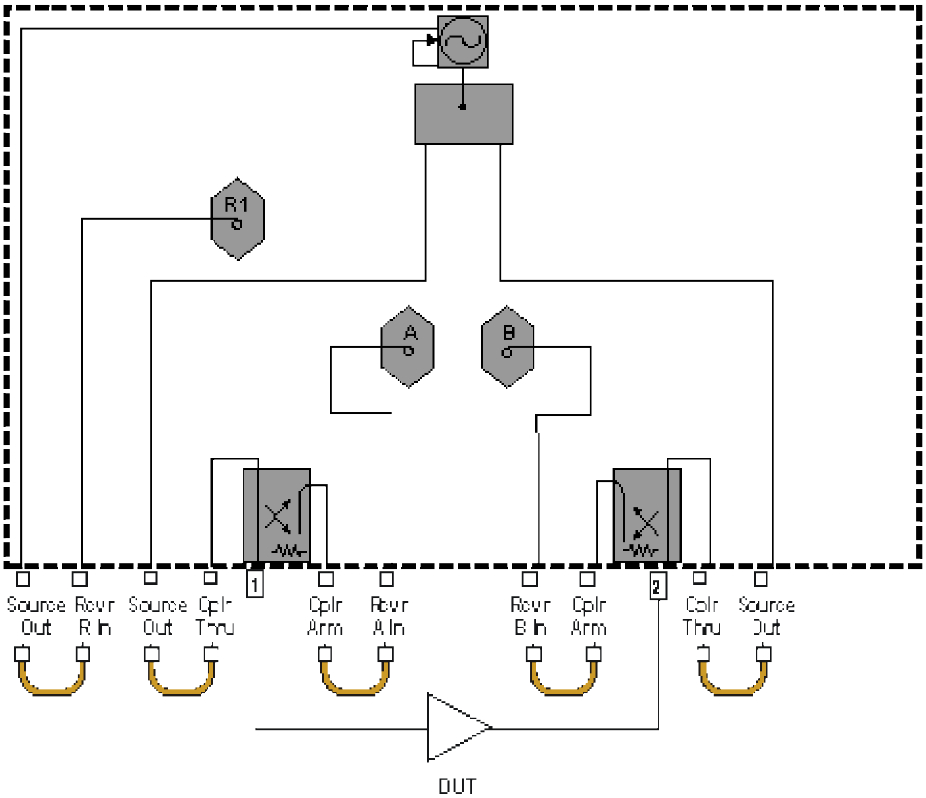


Diagram: (E880xA, Option 014) Configuration for High-Power Measurements in the Forward Direction

Explore the graphic with your mouse.

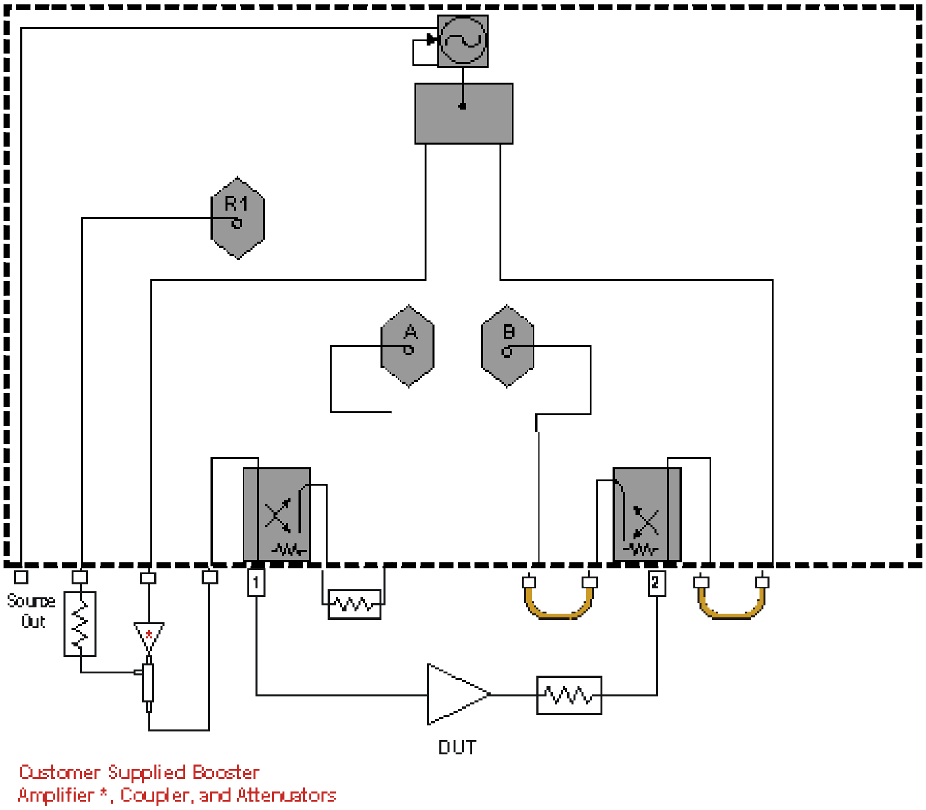


Diagram: (E880xA, Option 014) Configuration (with an Isolator) for High-Power Measurements in the Forward Direction

Explore the graphic with your mouse.

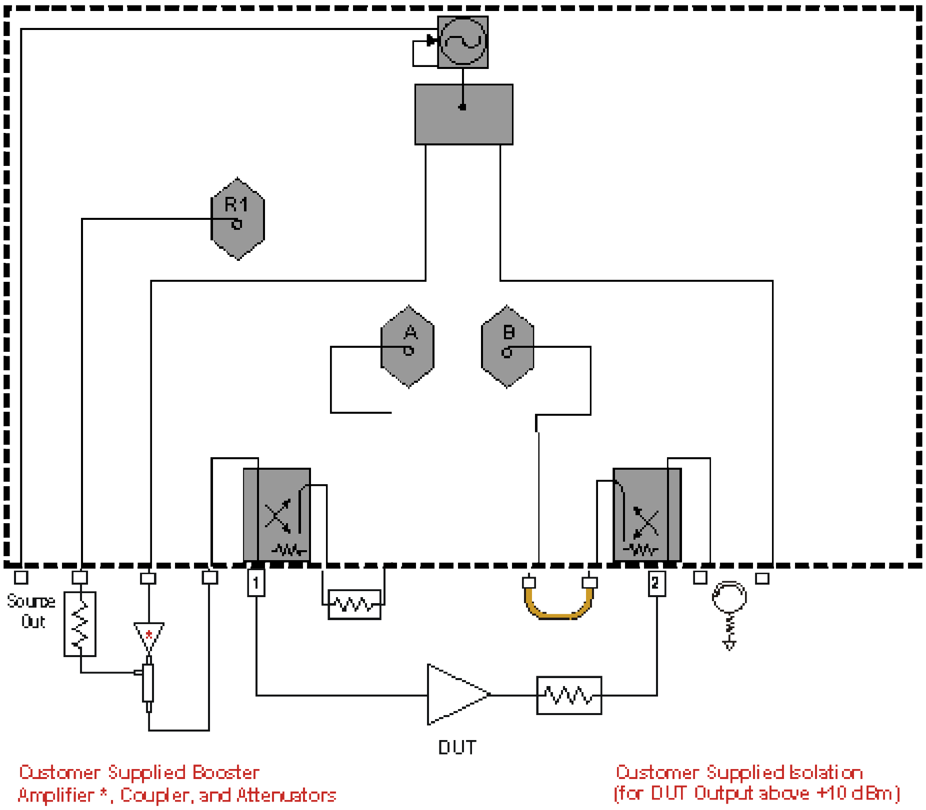


Diagram: (E880xA, Option 014) Configuration for High-Power Measurements in both Forward and Reverse Directions

Explore the graphic with your mouse.

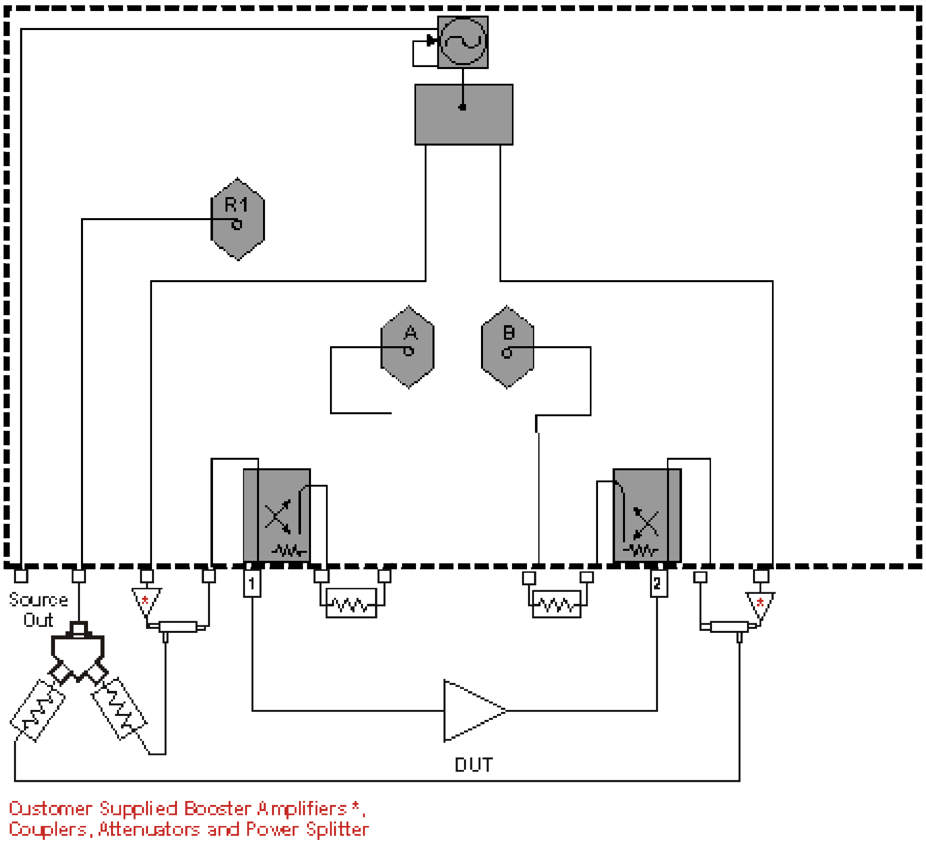


Diagram: (E880xA, Option 014) Configuration Using One Booster Amplifier

Explore the graphic with your mouse.

Note: the circuitry inside the yellow lines contains the following components supplied by you - a booster amplifier, a forward coupler, an isolator, and attenuators.

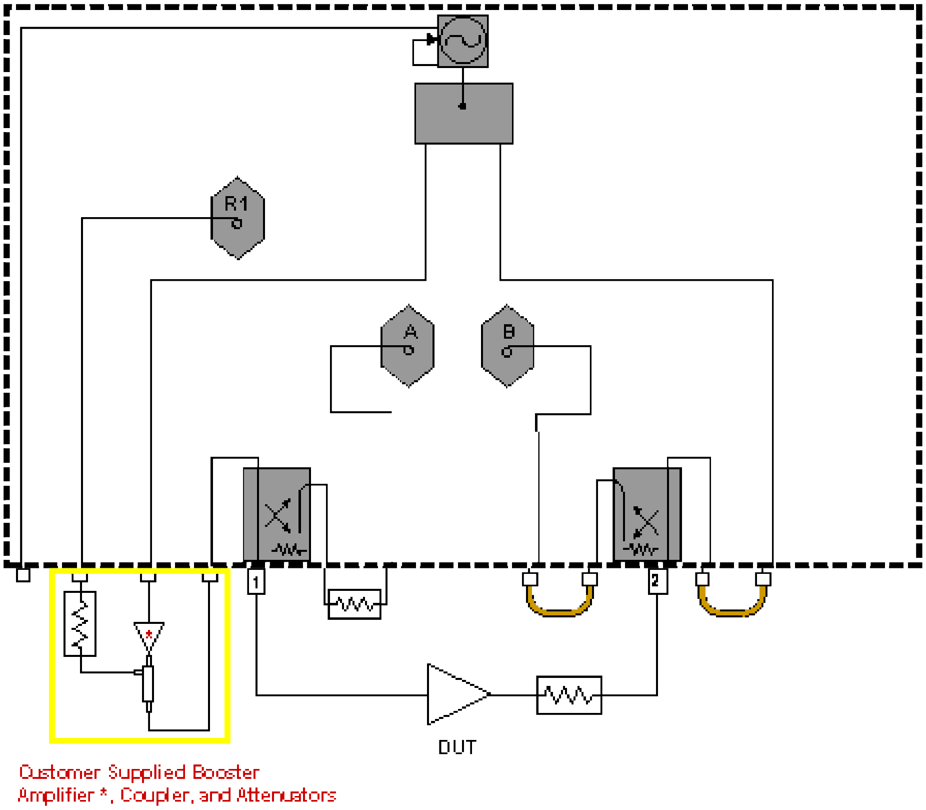


Diagram: (E880xA, Option 014) Configuration Using Two Booster Amplifiers

Explore the graphic with your mouse.

Note: the circuitry inside the yellow lines contains the following components supplied by you - two booster amplifiers, two forward couplers, two attenuators, and a power splitter.

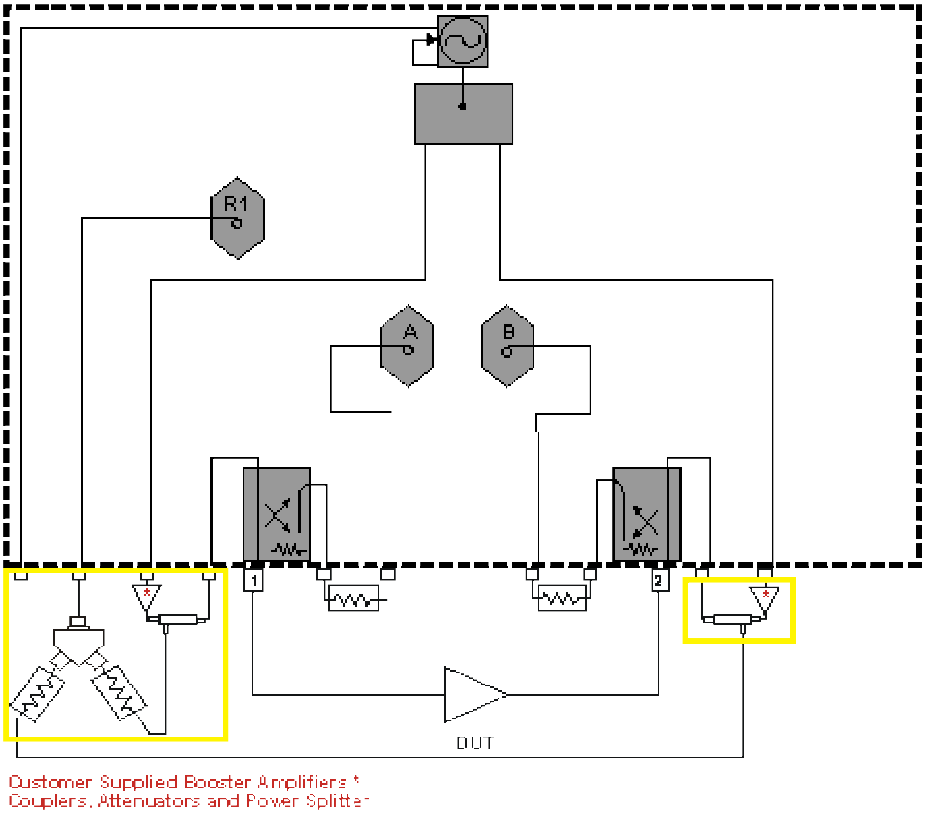


Diagram: (E880xA, Option 014) External Leveling using One Booster Amplifier

Explore the graphic with your mouse.

Note: the circuitry inside the yellow lines contains the following components supplied by you - a booster amplifier, a forward coupler, two attenuators, a power splitter, and a negative detector diode.

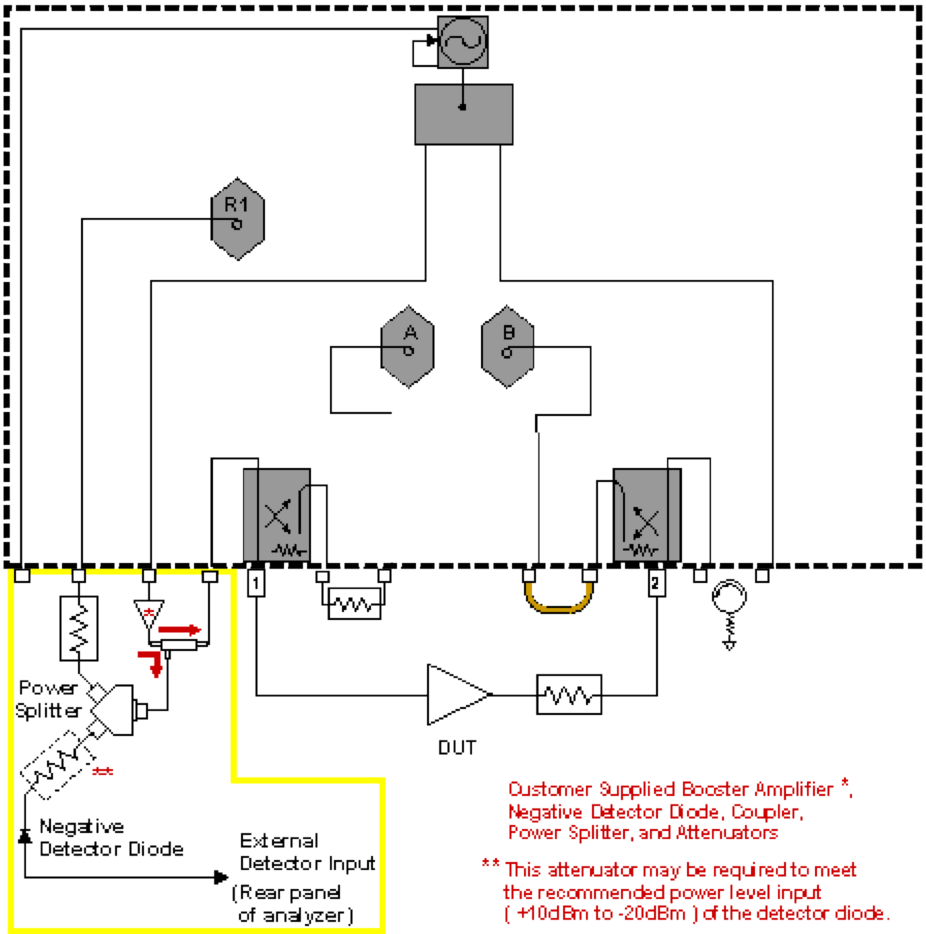


Diagram: (E880xA, Option 014) External Leveling using Two Booster Amplifiers

Explore the graphic with your mouse.

Note: the circuitry inside the yellow lines contains the following components supplied by you - two booster amplifiers, two forward couplers, two attenuators, four power splitters, and a negative detector diode.

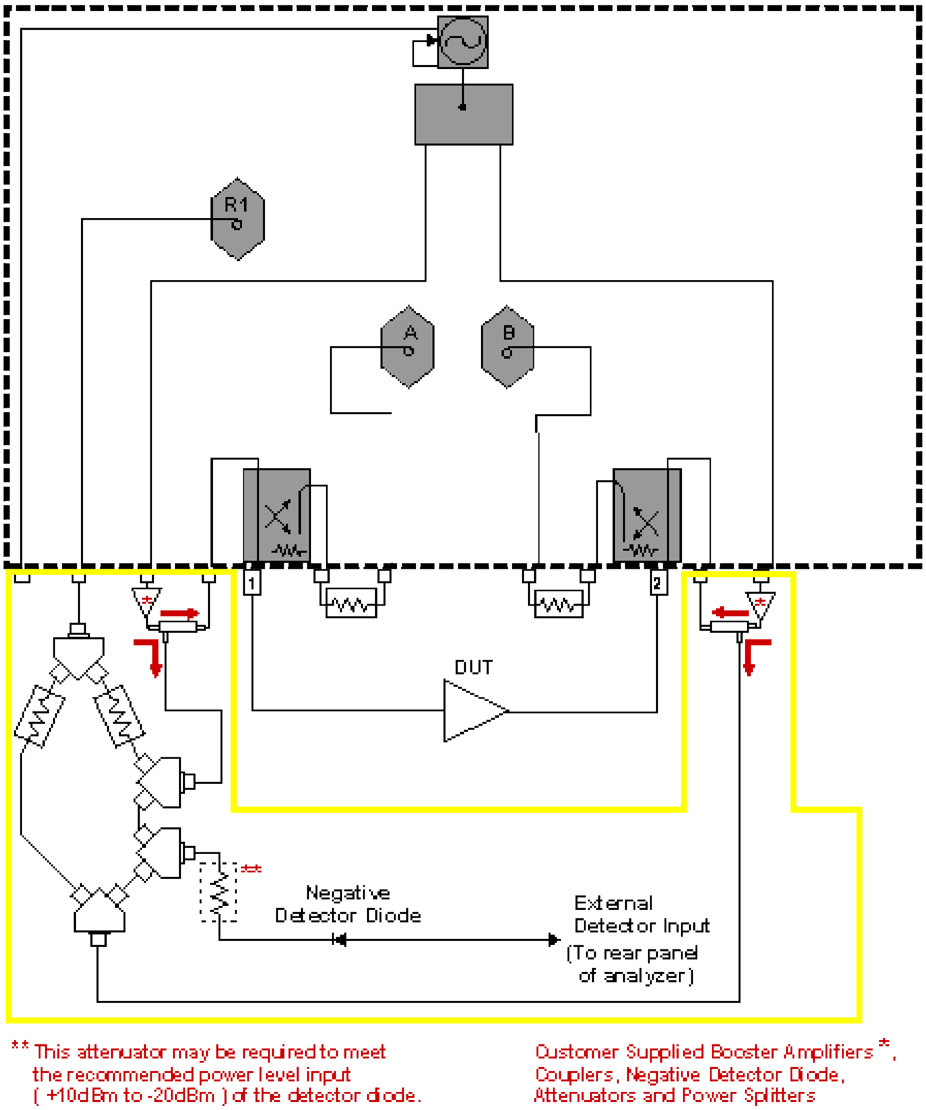
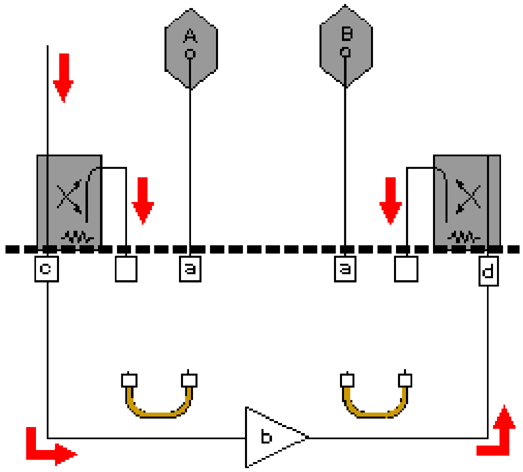


Diagram: (E880xA, Option 014) Measuring Calibration Power Levels



Item	Description
a	Measure the receiver input power where your external circuitry connects here
b	DUT, customer supplied
c	PORT 1
d	PORT 2

Diagram: (N338x) Basic Configuration

Explore the graphic with your mouse.

Note: Either Port 2 or Port 3 can be used for testing a 2-Port DUT.

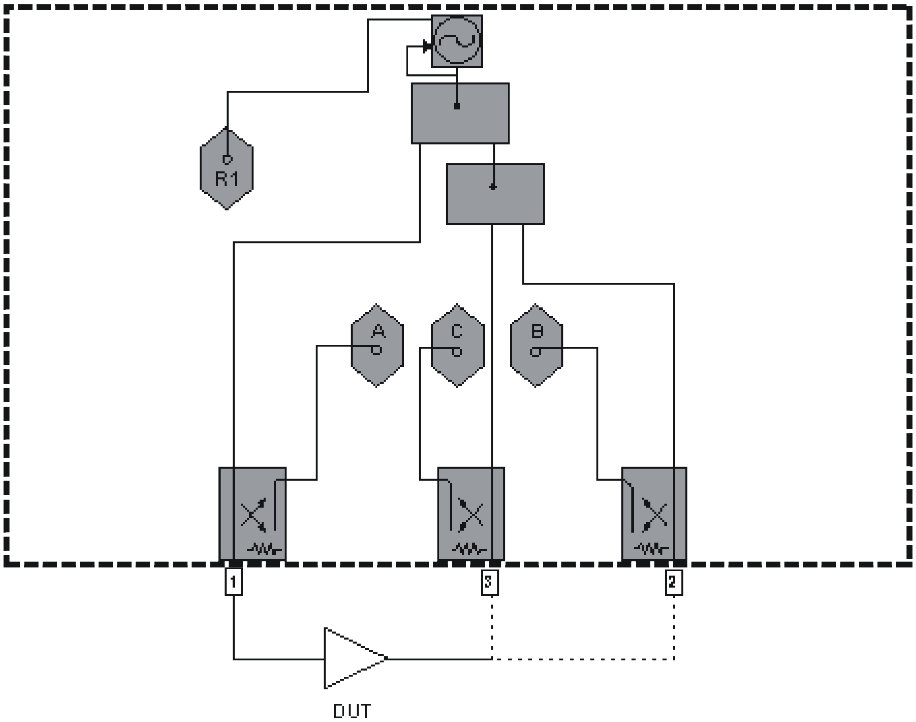


Diagram: (N338x) Configuration for High-Power Measurements in the Forward Direction (Single-Port Cal Only)

Explore the graphic with your mouse.

Note: Either Port 2 or Port 3 can be used for testing a 2-Port DUT.

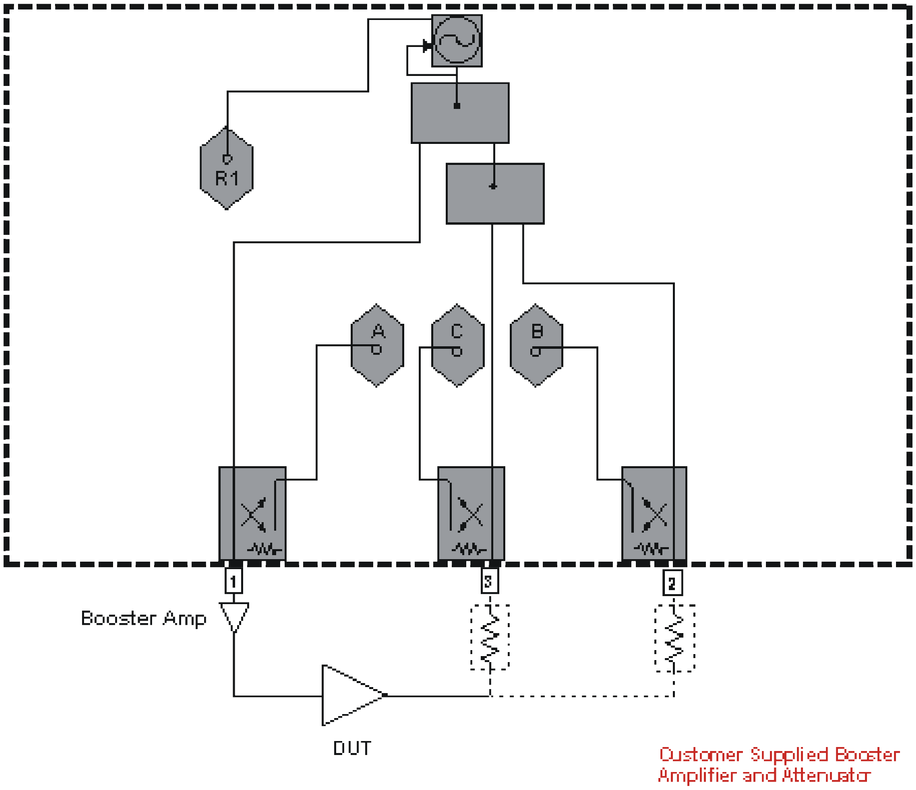


Diagram: (N338x) Configuration for Non-Boosted High-Power Measurements in both Forward and Reverse Directions

Explore the graphic with your mouse.

Note: Either Port 2 or Port 3 can be used for testing a 2-Port DUT.

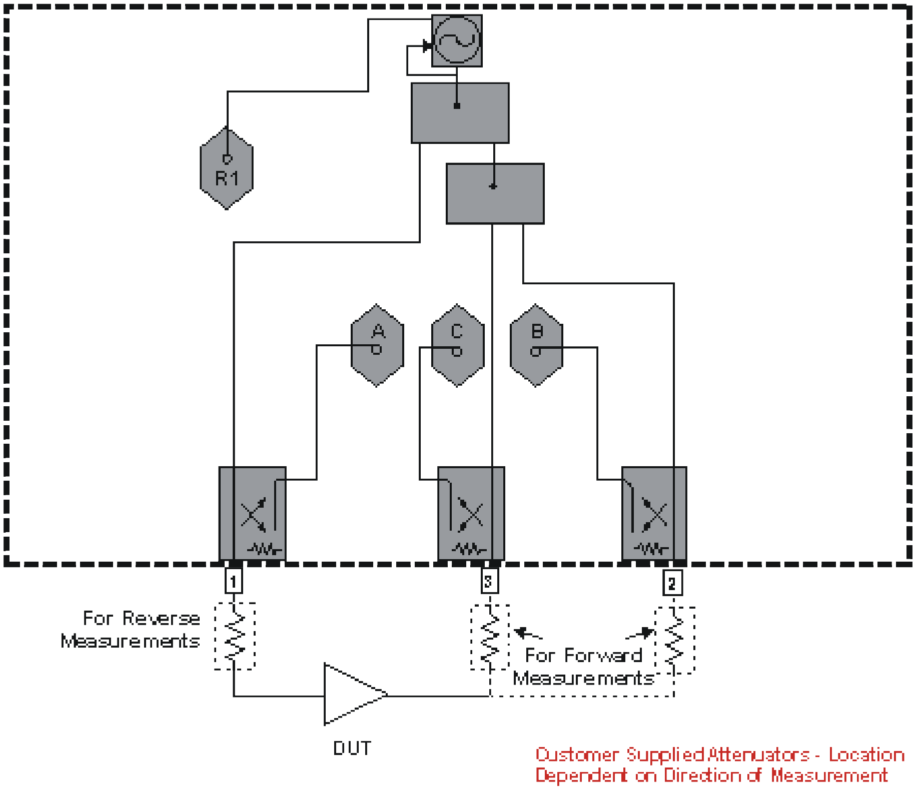


Diagram: (N338x) Configuration using a Booster Amplifier

Explore the graphic with your mouse.

Note: Either Port 2 or Port 3 can be used for testing a 2-Port DUT. The circuitry inside the yellow lines contains a booster amplifier supplied by you.

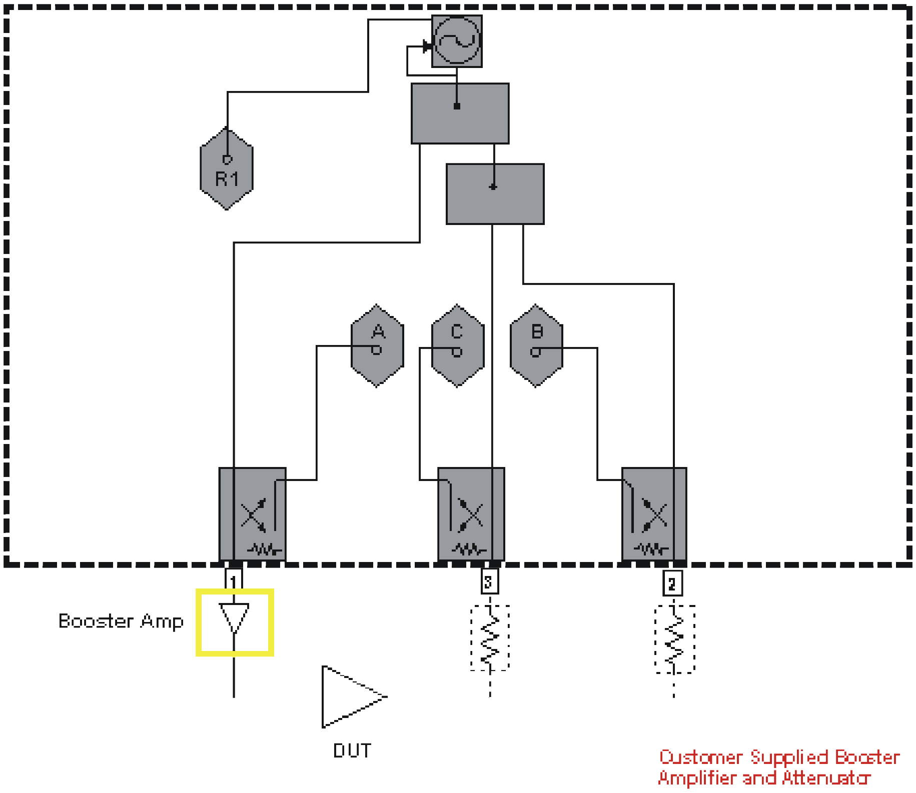


Diagram: (N338x) External Leveling Configuration

Explore the graphic with your mouse.

Note: Either Port 2 or Port 3 can be used for testing a 2-Port DUT. The circuitry inside the yellow lines contains the following components supplied by you - a booster amplifier, a forward coupler, an attenuator, and a negative detector diode.

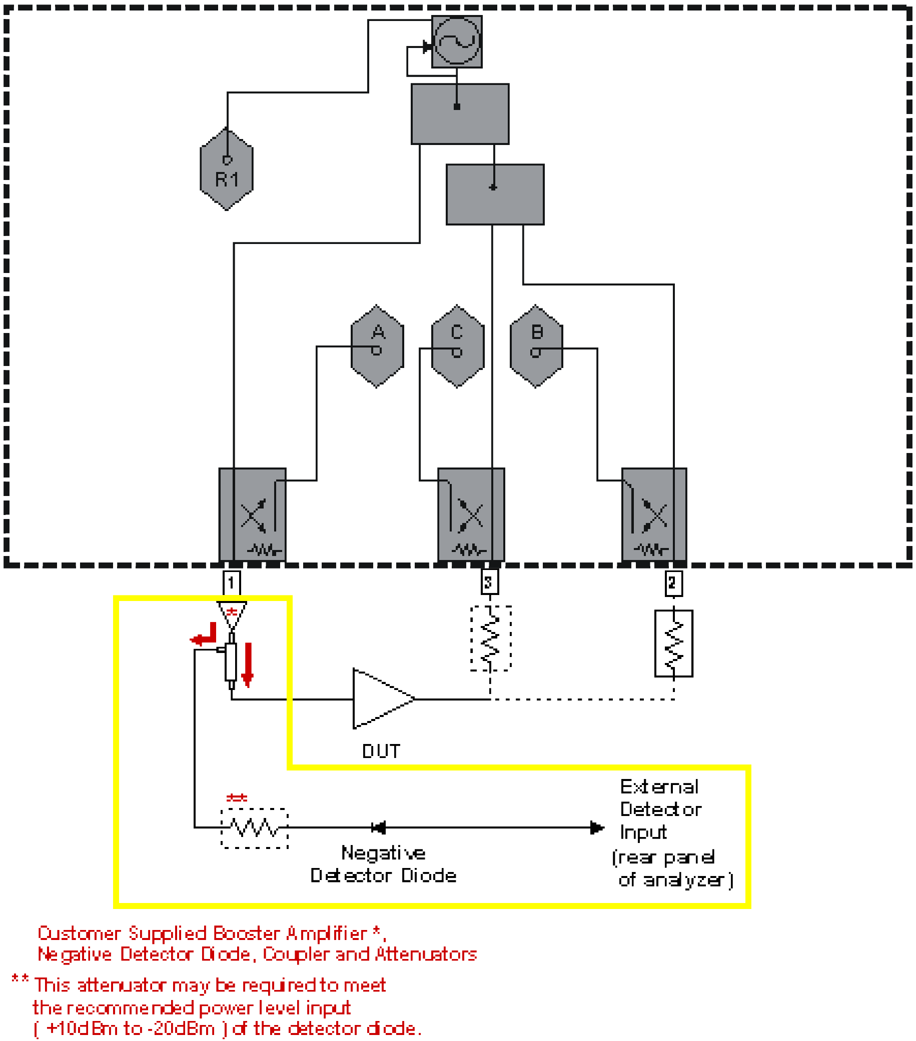


Diagram: (N338xA, Option 014) Basic Configuration

Explore the graphic with your mouse.

Note: Either Port 2 or Port 3 can be used for testing a 2-Port DUT.

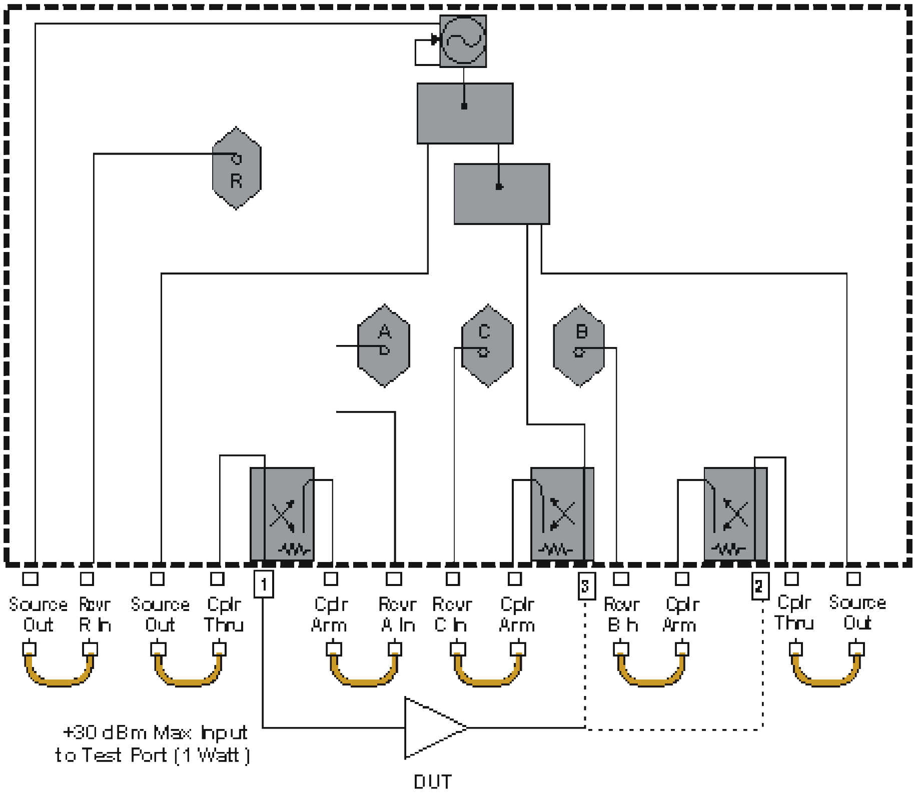


Diagram: (N338xA, Option 014) Configuration for High-Power Measurements in the Forward Direction

Explore the graphic with your mouse.

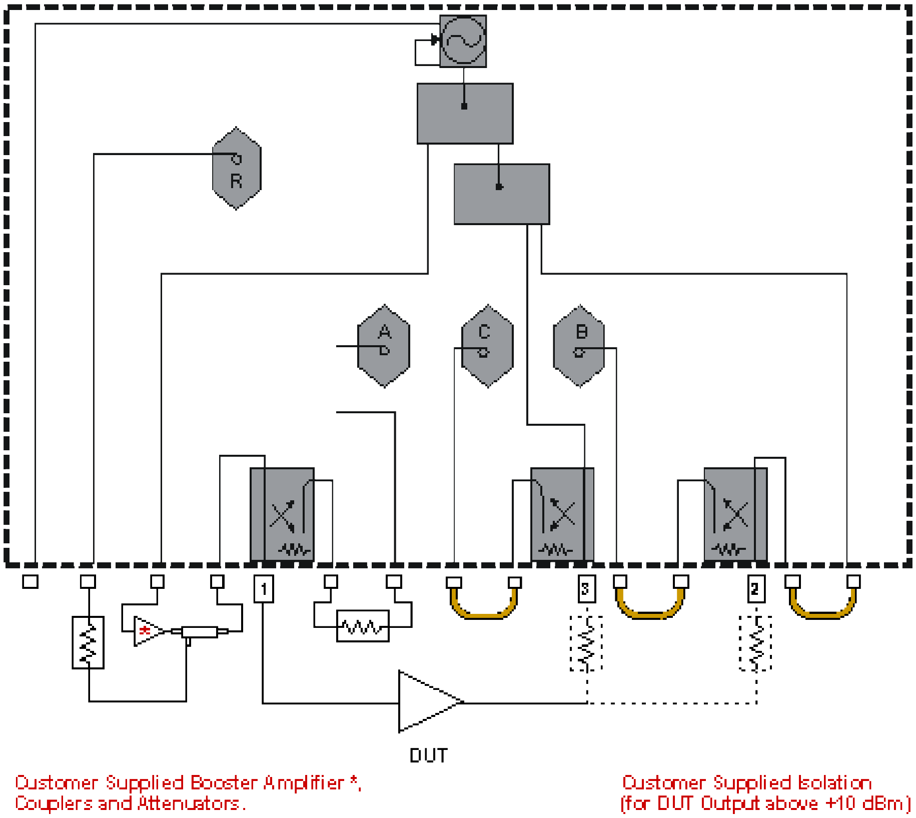


Diagram: (N338xA, Option 014) Configuration (with an Isolator) for High-Power Measurements in the Forward Direction

Explore the graphic with your mouse.

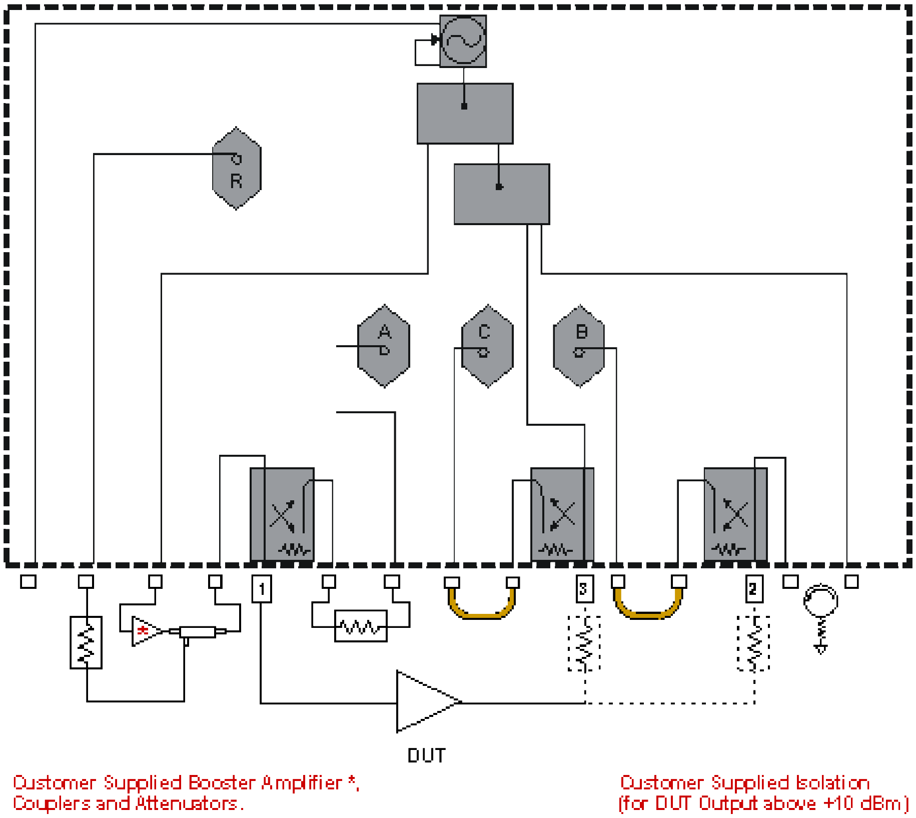


Diagram: (N338xA, Option 014) Configuration for High-Power Measurements in both Forward and Reverse Directions

Explore the graphic with your mouse.

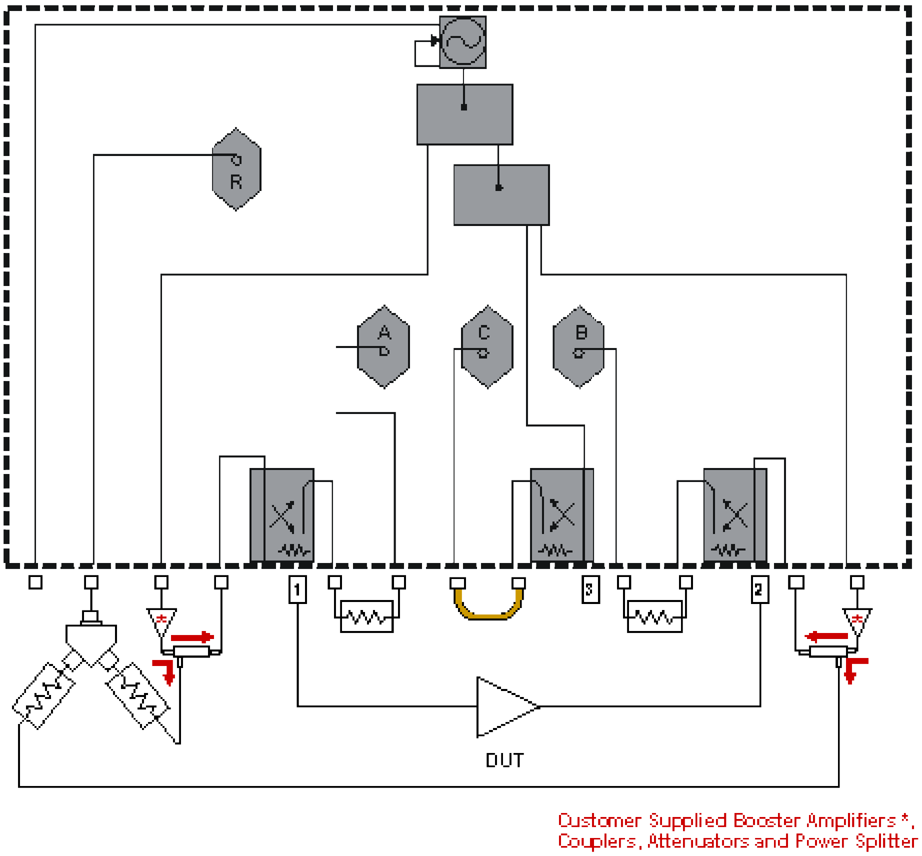


Diagram: (N338xA, Option 014) Configuration Using One Booster Amplifier

Explore the graphic with your mouse.

Note: Either Port 2 or Port 3 can be used for testing a 2-Port DUT. The circuitry inside the yellow lines contains the following components supplied by you - a booster amplifier, a forward coupler, and an attenuator.

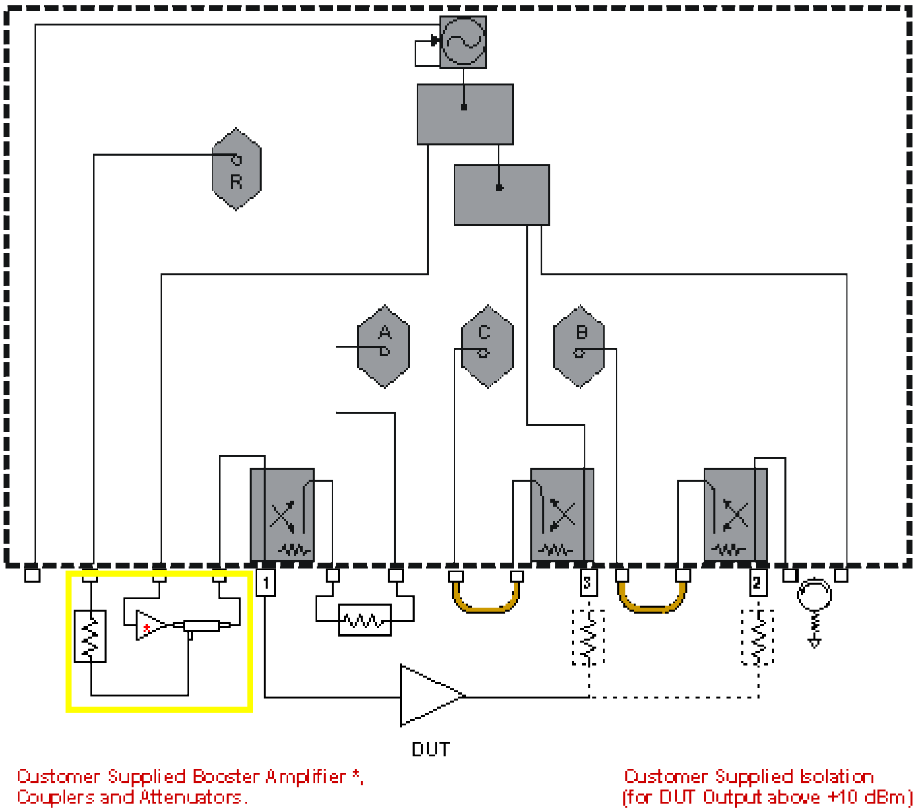


Diagram: (N338xA, Option 014) Configuration Using Two Booster Amps

Explore the graphic with your mouse.

Note: the circuitry inside the yellow lines contains the following components supplied by you - two booster amplifiers, two forward couplers, two attenuators and a power splitter.

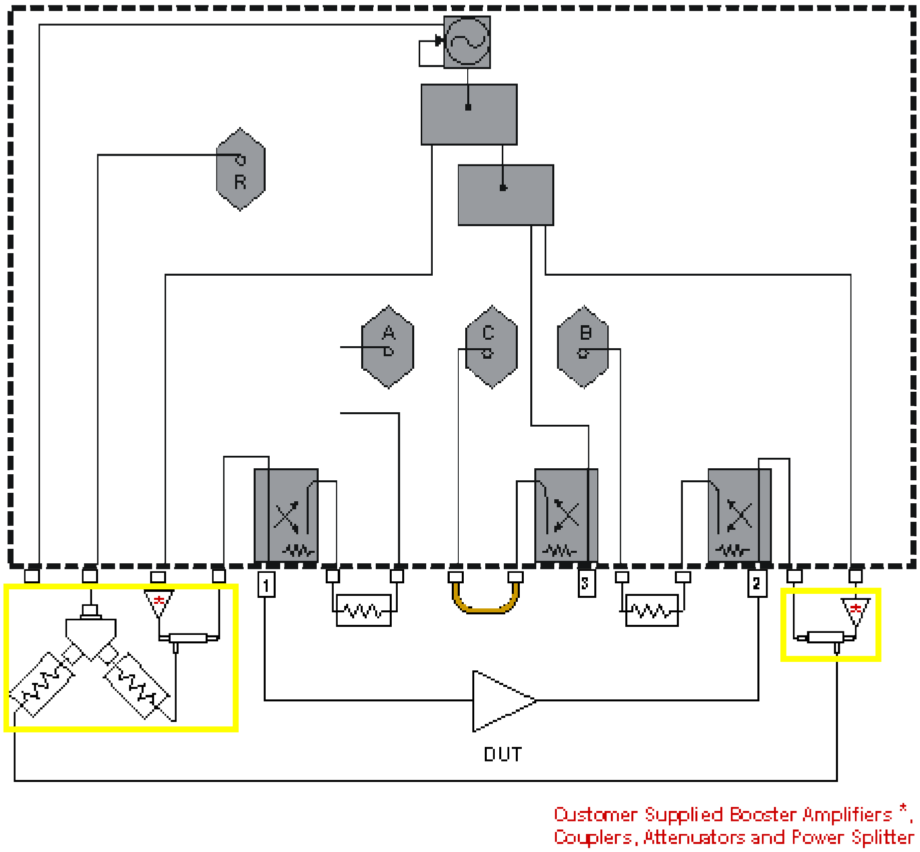


Diagram: (N338xA, Option 014) External Leveling using One Booster Amplifier

Explore the graphic with your mouse.

Note: Either Port 2 or Port 3 can be used for testing a 2-Port DUT. The circuitry inside the yellow lines contains the following components supplied by you - a booster amplifier, a forward coupler, two attenuators, a power splitter and a negative detector diode.

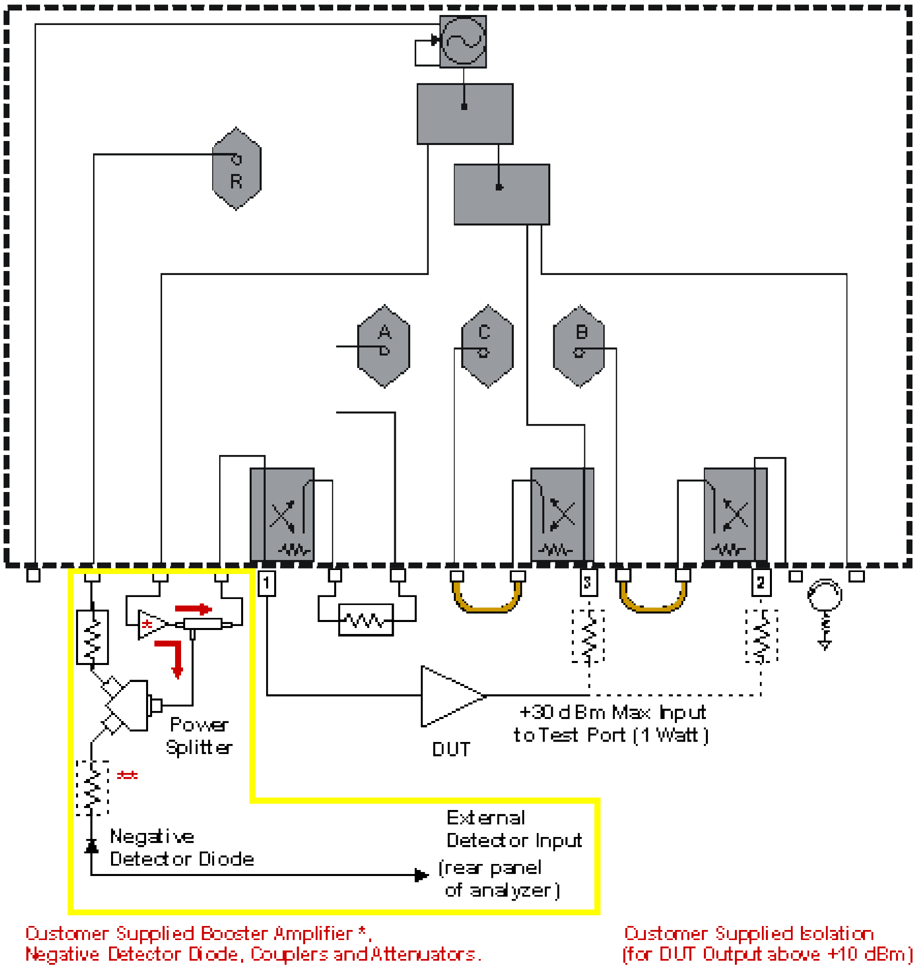


Diagram: (N338xA, Option 014) External Leveling using Two Booster Amplifiers

Explore the graphic with your mouse.

Note: the circuitry inside the yellow lines contains the following components supplied by you - two booster amplifiers, two forward couplers, two attenuators, three power splitters, and a negative detector diode.

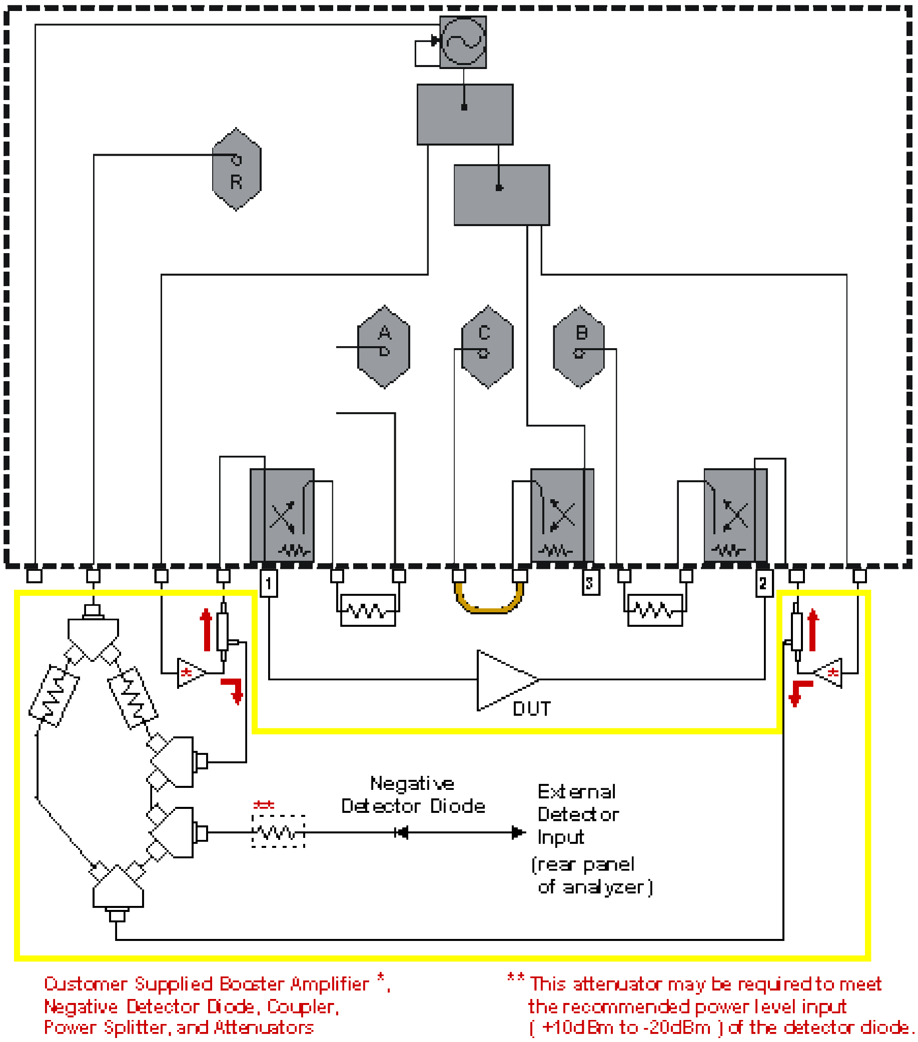
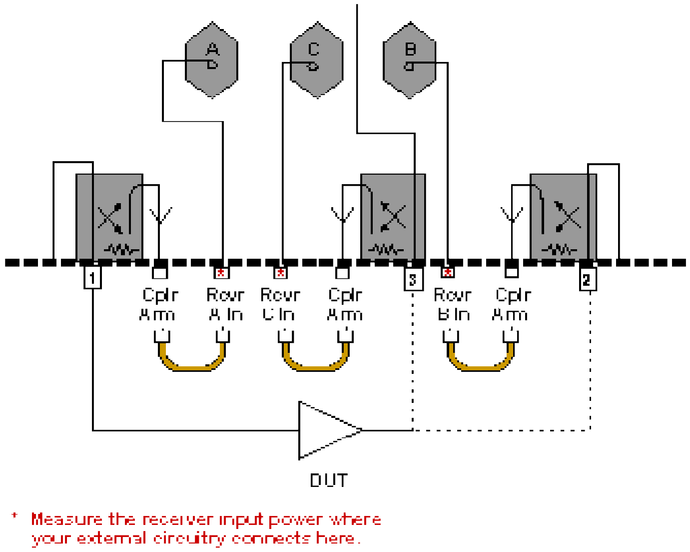


Diagram: (N338xA, Option 014) Measuring Calibration Power Levels

Note: Either Port 2 or Port 3 can be used for testing a 2-Port DUT.



Easy versus Secure Configuration

When upgrading Firmware on the PNA, you encounter a **Choose Configuration** dialog box. This is used to determine the level of security set for the DCOM interface on the PNA. For more detailed information on the security settings for the DCOM interface, including a procedure for making these settings manually, see Configure for COM-DCOM Programming.

Comparison of the "Easy and More Secure" settings are as follows:

Easy Connection:

- No configuration of the PNA required for remote access to connect.
- Anyone on the local subnet can access the PNA remotely.
- People from other NT domains can connect to the PNA.

More Secure:

- Requires creating users on the PNA or adding the PNA to a domain
- An administrator of the PNA can specify users or groups that are allowed remote access to the PNA application

Changing Network Client

If your PC network uses Novell NetWare servers, a change must be made to the PNA setup before it can operate on your network. If you are unsure, ask your local IT department.

Note: Do NOT **Uninstall** "Client for Microsoft Networks". This will prevent proper operation of the PNA..

To remove "Client for Microsoft Networks" (Remove is different from Uninstall):

1. From the PNA Desktop, right-click **My Network Places**
2. Click **Properties**
3. Right-click **Local Area Connection**
4. Click **Properties**
5. Click (remove the check from) **Client for Microsoft Networks**

To install "Client Service for NetWare".

1. Click **Install**
2. In **Select Network Component Type**, make sure **Client** is selected
3. Click **Add**
4. In **Select Network Client**, make sure **Client Service for NetWare** is selected
5. Click **OK**.

Troubleshooting the PNA

By running a few checks, you can identify if the analyzer is at fault. Before calling Agilent Technologies or returning the instrument for service, please make the following checks.

- Check the Basics
- PNA Application Terminates Unexpectedly
- Check Error Terms
- Check the Service Guide

Other Support Topics

Check the Basics

A problem can often be solved by repeating the procedure you were following when the problem occurred. Before calling Agilent Technologies or returning the instrument for service, please make the following checks:

1. Is there power at the power socket? Is the instrument plugged in?
2. Is the instrument turned on? Check to see if the front panel line switch and at least one of the LED rings around the test ports glows green. This indicates the power supply is on.
3. If you are experiencing difficulty with the front-panel keypad or peripherals, the USB bus may be overloaded. Remove the USB devices, restart the PNA, and reconnect the USB devices. See Power-up.
4. If other equipment, cables, and connectors are being used with the instrument, make sure they are connected properly and operating correctly.
5. Review the procedure for the measurement being performed when the problem appeared. Are all the settings correct?
6. If the instrument is not functioning as expected, return the unit to a known state by pressing the **Preset** key.
7. Is the measurement being performed, and the results that are expected, within the specifications and capabilities of the instrument?
8. If the problem is thought to be due to firmware, check to see if the instrument has the latest firmware before starting the troubleshooting procedure.
9. Check that the measurement calibration is valid. See Accurate Measurement Calibrations for more information.
10. If the necessary test equipment is available, perform the operator's check and system verification in Chapter 2 of the PNA Service Guide, "System Tests, Verifications, and Adjustments".You can download a copy of the Service Guide from our Web site at http://www.agilent.com/find/pna

PNA Application Terminates Unexpectedly

If an unexpected and irrecoverable error occurs, Agilent would like to know about it. The PNA attempts to save pertinent information about the state of the system. **The PNA does NOT send this information to Agilent**.

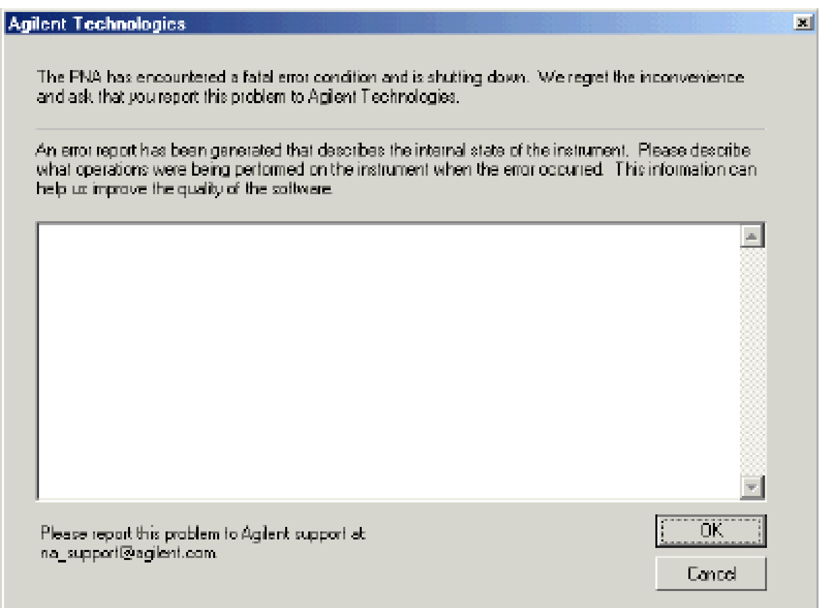
We respect the privacy of our customers. However, access to information that helps us improve the PNA is a benefit to both Agilent and you. Please take the time to contact us or email the saved information to **na_support@agilent.com.**

The following procedure shows how to do this:

1. A message box immediately appears on the screen containing the location of a directory. Please record this message. If you miss the message, you can find the directory location using the Windows Event Log: On the PNA, click Start, Settings, Control Panel, Administrative Tools, Event Viewer. Double-click the top line (most recent event). The location of the directory is seen in the Description.
2. A dialog box may appear on the screen (shown below) allowing you to add comments to help us replicate the crash.
3. Find the directory (described in Step 1) which contains the following files:

- 835x.dmp which is the 835x.exe capturing the context in which the program crashed.
- 835x.xml which reports some very basic information (exception code, OS version, and the list of modules loaded at the time of the crash and their respective version numbers).
- 835xCrashLog.txt: The text file with your comments (described in Step 2), if submitted.

4. If your PNA is not connected to LAN or is not configured to send email, copy the files to a PC. Then, please email the files to na_support@agilent.com



Check Error Terms

If you print the error terms at set intervals (weekly, monthly, and so forth), you can compare current error terms to these records. A stable, repeatable system should generate repeatable error terms over long time intervals, for example, six months. If a subtle failure or mild performance problem is suspected, the magnitude of the error terms should be compared against values generated previously with the same instrument and calibration kit. See the procedure for monitoring error terms.

- A long-term trend often reflects drift, connector and cable wear, or gradual degradation, indicating the need for further investigation and preventative maintenance. Yet, the system may still conform to specifications. The cure is often as simple as cleaning and gaging connectors or inspecting cables.

- A sudden shift in error terms reflects a sudden shift in systematic errors, and may indicate the need for further troubleshooting.

Consider the following while troubleshooting:

- All parts of the system, including cables and calibration devices, can contribute to systematic errors and impact the error terms.
- Connectors must be clean and gauged, and within specification for error term analysis to be meaningful. See the Chapter 2 in the PNA Service Guide for information on cleaning and gaging connectors.
  - Avoid unnecessary bending and flexing of the cables following measurement calibration, thus minimizing cable instability errors.
  - Use good connection techniques during the measurement calibration. The connector interface must be repeatable. See the PNA Service Guide for information on connection techniques.
- It is often worthwhile to perform the procedure twice (using two distinct measurement calibrations) to establish the degree of repeatability. If the results do not seem repeatable, check all connectors and cables.
- Use error-term analysis to troubleshoot minor, subtle performance problems. See Chapter 3, "Troubleshooting," in the PNA Service Guide if a blatant failure or gross measurement error is evident.

Check the Service Guide

Check the PNA Service Guide for specific troubleshooting procedures to help identify problems. It is included on the CD-ROM that was shipped with your analyzer. You can download a copy of the Service Guide from our Web site at http://www.agilent.com/find/pna

PNA Error Messages

- 500 - 750 Calibrate
- 770 - 1000 Hardware
- 1000 - 1200 Measure
- 1281 - 1535 Parser
- 1536 - 1650 Display
- 1700 - 2000 Channel
- 2048 - 2200 General
- Standard SCPI Errors

Note: The **EventID**'s listed below are provided for COM programming. For more information, see Working with PNA Events

For more information on PNA error messages (see Error Messages).

Cal Errors

Message: 512

"A secondary parameter (power, IFBW, sweep time, step mode) of the calibrated state has changed."

Severity: Informational

Further explanation: The calibration is questionable when any of these secondary parameters change after the calibration is performed.

Suggestions: If you require an accurate measurement with the new settings, repeat the calibration.

EventID: 68020200 (hex)

Message: 513

"Calibration cannot be completed until you have measured all the necessary standards for your selected Cal Type."

Severity: Informational

Further explanation: You probably received this message because you attempted to turn correction on without first measuring all of the calibration standards

Suggestions: Finish measuring the cal standards

EventID: 68020201 (hex)

Message: 514

"Calibration set has been recalled using a file previously saved on an analyzer that had a different hardware configuration."

Severity: Informational

Further explanation:

Suggestions:

EventID: 68020202 (hex)

Message: 515

"Calibration is required before correction can be turned on. Channel number is <x>, Measurement is <x>."

Severity: Informational

Further explanation: There are no error correction terms to apply for the specified channel and measurement.

Suggestions:Perform or recall a calibration

EventID: 68020203 (hex)

Message: 516

"Critical parameters in your current instrument state do not match the parameters for the calibration set, therefore correction has been turned off. The critical instrument state parameters are sweep type, start frequency, frequency span, and number of points."

Severity: Informational

Further explanation: None

Suggestions: You can either re-calibrate using the new settings or change back to the original setting that was used when the calibration was performed.

EventID: 68020204 (hex)

Message: 517

"Interpolation is turned off and you have changed the stimulus settings of the original calibration, so correction has been turned off."

Severity: Informational

Further explanation: The most accurate calibration is maintained only when the original stimulus settings are used.

Suggestions: If reduced accuracy is OK, set interpolation ON to allow stimulus setting changes.

EventID: 68020205 (hex)

Message: 518

"Interpolation is turned off and you have selected correction ON. Correction has been restored with the previous stimulus settings."

Severity: Informational

Further explanation: None

Suggestions: None

EventID: 68020206 (hex)

Message: 519

"Stimulus settings for your current instrument state exceeded the parameters of the original calibration, so correction has been turned off."

Severity: Informational

Further explanation: Correction data outside the stimulus settings does not exist.

Suggestions: Perform a broadband calibration, with increased numbers of points with interpolation ON, to maintain calibration over the widest possible stimulus frequency settings.

EventID: 68020207(hex)

Message: 520

"Cal Type is set to NONE for Channel <x>, Measurement <x>; please select Calibration menu or press Cal hard key."

Severity: Informational

Further explanation: A cal operation can not proceed until a calibration exists or the cal type is selected. This error can occur if the calibration can not be found. Also this error can happen if a calibration type is not specified before attempting to programmatically execute cal acquisitions.

Suggestions To find a calibration, select a Cal Set that contains the calibration needed for the current measurements. OR specify the cal type before beginning a calibration procedure.

EventID: 68020208 (hex)

Message: 521

"The measurement you set up does not have a corresponding calibration type, so correction has been turned off or is not permitted."

Severity: Informational

Further explanation: The calibration for the channel may apply only to certain S-Parameters. For example, a 1-Port calibration for S11 can not be applied to a 1-Port calibration applied to S22.

Suggestions: Select a calibration type, such as full 2-Port cal, that can be applied to all the measurements to be selected.

EventID: 68020209 (hex)

Message: 522

"The calibration type you selected cannot be set up."

Severity: Informational

Further explanation: **"**Please use the SCPI command ROUTe:PATH:DEFine:PORT <num>,<num> for full 2 port type port assignment."

Suggestions:

EventID: 6802020A (hex)

Message: 523

"The calibration path you selected cannot be set up because it is not valid for the current measurement."

Severity: Informational

Further explanation: "Please use the SCPI command ROUTe:PATH:DEFine:PORT <num>,<num> for full 2 port type port assignment related to your current measurement."

Suggestions:

EventID: 6802020B (hex)

Message: 524

"The source power calibration is complete."

Severity: Informational

Further explanation:

Suggestions:

EventID: 6802020C (hex)

Message: 525

"You have specified more than 7 standards for one or more calibration classes."

Severity: Informational

Further explanation: These have been truncated to 7 selections.

EventID: 6802020D (hex)

Message: 526

"No user calibration found for this channel."

Severity: Informational

Further explanation: A cal operation can not proceed until a calibration exists.

Suggestions: To find a calibration, you can select a Cal Set that contains the calibration needed for the current measurement.

EventID: 6802020E (hex)

Message: 527

"You do not need to acquire this standard for this calibration type."

Severity: Informational

Further explanation: This error can happen as a result of PROGRAMMATICALLY requesting the measurement of an un-needed calibration standard during a calibration procedure.

Suggestions: Check the specified cal type or eliminate the request for the measurement of the standard.

EventID: 6802020F (hex)

Message: 528

"Could not configure the Electronic Calibration system. Check to see if the module is plugged into the proper connector."

Severity: Informational

Further explanation: During an ECal operation, communication could not be established with the ECal module. The calibration will not be initiated until the presence of the ECal module is verified.

Suggestions: Verify the USB cable is connected properly. Disconnect and re-connect the cable to ensure the analyzer recognizes the module.

EventID: 68020210 (hex)

Message: 529

"DATA OUT OF RANGE: Design Limits Exceeded"

Severity: Error

Further explanation:

Suggestions:

EventID: E8020211(hex)

Message: 530

"EXECUTION ERROR: Could not open ECal module memory backup file"

Severity: Error

Further explanation:

Suggestions:

EventID: E8020212 (hex)

Message: 531

"EXECUTION ERROR: Access to ECal module memory backup file was denied"

Severity: Error

Further explanation:

Suggestions:

EventID: E8020213 (hex)

Message: 532

"EXECUTION ERROR: Failure in writing to ECal module memory backup file"

Severity: Error

Further explanation:

Suggestions:

EventID: E8020214 (hex)

Message: 533

"EXECUTION ERROR: Failure in reading from ECal module memory backup file"

Severity: Error

Further explanation:

Suggestions:

EventID: E8020215 (hex)

Message: 534

"EXECUTION ERROR: Array index out of range"

Severity: Error

Further explanation:

Suggestions:

EventID: E8020216 (hex)

Message: 535

"EXECUTION ERROR: Arrays wrong rank"

Severity: Error

Further explanation:

Suggestions:

EventID: E8020217 (hex)

Message: 536

"EXECUTION ERROR: CPU"

Severity: Error

Further explanation:

Suggestions:

EventID: E8020218 (hex)

Message: 537

"EXECUTION ERROR: Cannot ERASE module"

Severity: Error

Further explanation:

Suggestions:

EventID: E8020219 (hex)

Message: 538

"EXECUTION ERROR: Cannot WRITE module"

Severity: Error

Further explanation:

Suggestions:

EventID: E802021A (hex)

Message: 539

"EXECUTION ERROR: Entry Not Found"

Severity: Error

Further explanation:

Suggestions:

EventID: E802021B (hex)

Message: 540

"EXECUTION ERROR: Invalid command while system is busy"

Severity: Error

Further explanation:

Suggestions:

EventID: E802021C (hex)

Message: 541

"Electronic Cal: Unable to orient ECal module. Please ensure the module is connected to the necessary measurement ports."

Severity: Error

Further explanation: There is no RF connection to the ECal module during a calibration step. An ECal orientation measurement has been attempted but the signal was not found.

Suggestions: Connect the ECal module RF connections to ports specified for the calibration step. The ECal module typically requires at least -18dBm for measurements. If your measurement requires the power level to be less than that, clear the **Do orientation** checkbox to bypass the automatic detection step.

EventID: E802021D (hex)

Message: 542

"EXECUTION ERROR: NO SPACE for NEW CAL, DELETE A CAL"

Severity: Error

Further explanation:

Suggestions:

EventID: E802021E (hex)

Message: 543

"EXECUTION ERROR: No More Room"

Severity: Error

Further explanation:

Suggestions:

EventID: E802021F (hex)

Message: 544

"EXECUTION ERROR: Other array error"

Severity: Error

Further explanation:

Suggestions:

EventID: E8020220 (hex)

Message: 545

"EXECUTION ERROR: Ranks not equal"

Severity: Error

Further explanation:

Suggestions:

EventID: E8020221 (hex)

Message: 546

"EXECUTION ERROR: Too few CONSTANT ranks"

Severity: Error

EventID: E8020222 (hex)

Message: 547

"EXECUTION ERROR: Too few VARYing ranks"

Severity: Error

EventID: E8020223 (hex)

Message: 548

"EXECUTION ERROR: Unknown error"

Severity: Error

EventID: E8020224 (hex)

Message: 549

"EXECUTION ERROR: ecaldrvr.dll bug or invalid module #"

Severity: Error

EventID: E8020225 (hex)

Message: 550

"EXECUTION ERROR: unexpected error code from ecal driver"

Severity: Error

EventID: E8020226 (hex)

Message: 551

"EXECUTION ERROR: unexpected internal driver error"

Severity: Error

EventID: E8020227 (hex)

Message: 552

"HARDWARE ERROR: Can't access ECal Interface Module"

Severity: Error

EventID: E8020228 (hex)

Message: 553

"HARDWARE ERROR: Can't release LPT port, reboot"

Severity: Error

EventID: E8020229 (hex)

Message: 554

"HARDWARE ERROR: VNA Error"

Severity: Error

EventID: E802022A (hex)

Message: 555

"HARDWARE ERROR: not enough data read from ECal module"

Severity: Error

EventID: E802022B (hex)

Message: 556

"OPERATION ABORTED BY HOST COMPUTER"

Severity: Error

EventID: E802022C (hex)

Message: 557

"OPERATION ABORTED BY USER"

Severity: Error

EventID: E802022D (hex)

Message: 558

"OUT OF MEMORY"

Severity: Error

EventID: E802022E (hex)

Message: 559

"QUERY INTERRUPTED:Message(s Abandoned"

Severity: Error

EventID: E802022F (hex)

Message: 560

"QUERY UNTERMINATED: INCOMPLETE PROGRAM Message"

Severity: Error

Further explanation:

Suggestions:

EventID: E8020230 (hex)

Message: 561

"QUERY UNTERMINATED: NOTHING TO SAY"

Severity: Error

Further explanation:

Suggestions:

EventID: E8020231 (hex)

Message: 562

"QUEUE OVERFLOW"

Severity: Error

EventID: E8020232 (hex)

Message: 563

"SETTINGS CONFLICT: ADDITIONAL STANDARDS ARE NEEDED"

Severity: Error

EventID: E8020233 (hex)

Message: 564

"SETTINGS CONFLICT: Adapter Cal is NOT possible"

Severity: Error

EventID: E8020234 (hex)

Message: 565

"SETTINGS CONFLICT: COMMAND OUT OF SEQUENCE"

Severity: Error

EventID: E8020235 (hex)

Message: 566

"SETTINGS CONFLICT: Cal STOPPED - VNA SETUP CHANGED"

Severity: Error

EventID: E8020236 (hex)

Message: 567

"SETTINGS CONFLICT: Calibration is NOT in progress"

Severity: Error

EventID: E8020237 (hex)

Message: 568

"SETTINGS CONFLICT: Can't find specified GPIB board"

Severity: Error

EventID: E8020238 (hex)

Message: 569

"SETTINGS CONFLICT: Can't find/load gpib32.dll"

Severity: Error

EventID: E8020239 (hex)

Message: 570

"SETTINGS CONFLICT: Can't find/load sicl32.dll"

Severity: Error

EventID: E802023A (hex)

Message: 571

"SETTINGS CONFLICT: Can't initialize VNA (bad address?"

Severity: Error

EventID: E802023B (hex)

Message: 572

"SETTINGS CONFLICT: Can't load LPT port driver or USB driver DLL"

Severity: Error

EventID: E802023C (hex)

Message: 573

"SETTINGS CONFLICT: Invalid Calibration Sweep Mode."

Severity: Error

EventID: E802023D (hex)

Message: 574

"SETTINGS CONFLICT: Invalid Calibration Type"

Severity: Error

EventID: E802023E (hex)

Message: 575

"SETTINGS CONFLICT: Invalid Calibration"

Severity: Error

EventID: E802023F (hex)

Message: 576

"SETTINGS CONFLICT: Invalid GPIB board number specified"

Severity: Error

EventID: E8020240 (hex)

Message: 577

"SETTINGS CONFLICT: Invalid GPIB board type specified"

Severity: Error

EventID: E8020241 (hex)

Message: 578

"SETTINGS CONFLICT: Invalid Module Status"

Severity: Error

EventID: E8020242 (hex)

Message: 579

"SETTINGS CONFLICT: Invalid States"

Severity: Error

EventID: E8020243 (hex)

Message: 580

"SETTINGS CONFLICT: LPT port must be between 1 and 4"

Severity: Error

EventID: E8020244 (hex)

Message: 581

"Could not configure the Electronic Calibration system. Check to see if the module is properly connected."

Severity: Error

EventID: E8020245 (hex)

Message: 582

"SETTINGS CONFLICT: Specified LPT port does not exist"

Severity: Error

EventID: E8020246 (hex)

Message: 583

"SETTINGS CONFLICT: Use frequency domain for cal"

Severity: Error

EventID: E8020247 (hex)

Message: 584

"SETTINGS CONFLICT: Use step sweep type for cal."

Severity: Error

EventID: E8020248 (hex)

Message: 585

"SETTINGS CONFLICT: VNA address must be between 0 and 30"

Severity: Error

EventID: E8020249 (hex)

Message: 586

"SETTINGS CONFLICT: Wrong LPT port driver or USB driver DLL"

Severity: Error

EventID: E802024A (hex)

Message: 587

"SYNTAX ERROR: ECAL:DELAY command must have 2 numbers"

Severity: Error

EventID: E802024B (hex)

Message: 588

"SYNTAX ERROR: INCORRECT SYNTAX"

Severity: Error

EventID: E802024C (hex)

Message: 589

"SYNTAX ERROR: UNKNOWN COMMAND"

Severity: Error

EventID: E802024D (hex)

Message: 590

"Wrong port of module in RF path"

Severity: Error

EventID: E802024E (hex)

Message: 591

"User characterization not found in module"

Severity: Error

EventID: E802024F (hex)

Message: 592

Severity: Informational

"No source power calibration found for the channel and source port of the current measurement."

Further explanation: You tried to turn on source power cal but there is no source power cal data.

Suggestions: Perform a source power calibration

EventID: 68020250 (hex)

Message: 593

Severity: Informational

"A source power calibration sweep was not performed, so there is no correction for the channel and source port of the current measurement."

Further explanation: You tried to turn on source power cal but there is incomplete source cal data.

Suggestions:Perform a complete source power calibration

EventID: 68020251 (hex)

Message: 594

Severity: Informational

"A new trace could not be added to the active window for viewing the source power cal sweep, because it would have exceeded the limit on number of traces/window. Please remove a trace from the window before proceeding with source power cal."

Further explanation: The source power cal attempts to add a data trace to the active window. The active window already contains four traces.

Suggestions:Make the active window contain less than four traces.

EventID: 68020252 (hex)

Message: 595

Severity: Informational

"A new measurement could not be added for performing the source power cal sweep, because the limit on number of measurements has been reached. Please remove a measurement before proceeding with source power cal."

Further explanation: The source power cal attempts to add a measurement. The PNA already has the maximum number of measurements.

Suggestions: Delete a measurement.

EventID: 68020253 (hex)

Message: 596

Severity: Informational

"The calibration power value associated with the source power calibration of Port %1 on Channel %2 was changed with the calibration on. The calibration was not turned off, but the power value might no longer represent the calibration."

Further explanation: The source power cal accuracy is questionable.

Suggestions: If high accuracy is required, perform another source power calibration.

EventID: 68020254 (hex)

Message: 597

Severity: Informational

- Message that is passed from the power meter driver for a source power calibration. -

Further explanation: This error is generated by the power meter driver and passed through the PNA.

EventID: 68020255 (hex)

Message: 598

"During the acquisition of the sliding load standard, the slide was not properly moved to perform a circle fit. The standard's raw impedance was used to determine the directivity for one or more points."

Severity: Informational

Further Explanation: To accurately characterize the standard, the sliding load must be move sufficiently to ensure enough samples around the complex circle or Smith Chart. Under-sampling will cause an inaccurate result.

Suggestions: For best results when using a sliding load, be sure to use multiple slide positions that cover the full range of movement from front to back of the slot.

EventID: 68020256 (hex)

Message: 599

"This feature requires an unused channel, but could not find one. Please free up a channel and try again."

Severity: Informational

Further Explanation: You attempted to view an item within a calset. However, the calset viewer requires that the result be displayed in a channel that is not currently in use. All the channels are currently used. The view can not display the requested item.

Suggestions: You must delete at least one channel that is currently in use.

EventID: 68020257 (hex)

Message: 600

"Interpolation of the original calibration is not allowed since it was performed using Segment Sweep. Correction has been turned off."

Severity: Informational

EventID: 68020258 (hex)

Message: 601

"Cal preferences saved. Cal preference settings can be changed from the 'Cal Preferences' drop down Cal menu."

Severity: Informational

Further explanation: See Cal Preferences

EventID: 68020259 (hex)

Message: 608

"CalType not set."

Severity: Error

Further explanation: A cal operation can not proceed until a calibration exists or the proper cal type is selected.

Suggestions: This error can happen if the calibration can't be found. To find a calibration, you can select a Cal Set that contains the calibration needed for the current measurements. This error can also happen if a calibration type is not specified before attempting to programmatically execute cal acquisitions. Specify the cal type before beginning a calibration procedure.

EventID: E8020260 (hex)

Message: 609

"The Calibration feature requested is not implemented."

Further explanation: The specified cal type can be one of many choices. For example, response calibrations require single standards, 1-Port calibrations require 3 standards, and 2-Port calibrations require up to 12 standards.

Suggestions: Be sure to measure only the standards needed for the specified cal type.

EventID: E8020261 (hex)

Message: 610

"The Calibration Class Acquisition requested is not valid for the selected Calibration Type. Please select a different acquisition or a different Calibration Type."

EventID: E8020262 (hex)

Message: 611

"The Calibration Standard data required for the selected caltype was not found."

Severity: Error

Further explanation: An unsuccessful attempt was made to retrieve a specified standard from the raw measurement buffer. The buffer should contain the raw measurements of cal standards stored during a calibration procedure.

Suggestions: Be sure the requested standard is required for the current cal type. Not all standards are needed for all cal types.

EventID: E8020263 (hex)

Message: 612

" The Error Term data required for the selected caltype was not found."

Severity: Error

Further explanation: An unsuccessful attempt was made to retrieve a specified error term from the error correction buffer. The buffer should contain the error correction arrays for the current calibration.

Suggestions: Be sure the requested error term is required for the current cal type. Not all error terms are needed for all cal types.

EventID: E8020264 (hex)

Message: 613

The Calibration data set was not found.

Severity: Error

Further explanation: An unsuccessful attempt to access a cal set has been made. This may indicate a calset has been deleted or has been corrupted.

Suggestions: Try again or select another cal set. If the cal set appears in the cal set list, it may need to be deleted.

EventID: E8020265 (hex)

Message: 614

"The specified measurement does not have a calibration valid for Confidence Check. Please select a different measurement, or recall or perform a different Calibration Type."

Severity: Error

Further explanation: The measurement choice is prevented so that calibration will not be turned off. Not all cal types support all measurements. For example, an 1-Port cal on S11 can not be used to calibrate an S12 measurement. When a measurement is selected that does not have a calibration which can be applied, an informational message is displayed and calibration is turned off.

Suggestions: Use a full 2-Port calibration to be compatible with any S-Parameter.

EventID: E8020266 (hex)

Message: 615

" New calset created."

Severity: Informational message.

Further explanation: The newly created cal set will be automatically named and time stamped. If this is the beginning of a calibration procedure, the cal set will not be stored to memory until the calibration has completed successfully. The new cal set will be deleted if the calibration is canceled or does not otherwise complete successfully.

Suggestions: Informational

EventID: 68020267

Message: 617

The calset file: <x> appears to be corrupted and cannot be removed. Exit the application, remove the file, and restart.

Severity: Error

Suggestions: The cal set file is stored in the application home directory C:\Program Files\Agilent\Network Analyzer\PNACalSets.dat. Remove this file, then restart the application.

EventID: E8020269 (hex)

Message: 634

"The calset file: <x> load failed."

Severity: Error

Further explanation: The calset file contains a collection of calsets. The file resides on the hard drive.

Suggestions: Try restarting the application. If the failure persists, you may have to delete the cal set data file and restart the application. The cal set file is stored in the application home directory. C:\Program Files\Agilent\Network Analyzer\PNACalSets.dat. Remove this file, then restart the application.

EventID: E802027A (hex)

Message: 635

"The calset file: <x> save failed."

Severity: Error

Further explanation: The file operation detected an error. The save operation was aborted.

Suggestions: Retry.

EventID: E802027B (hex)

Message: 636

"A calset was deleted."

Severity: Informational

Further explanation: One of the calsets has been successfully deleted from the collection of calsets available. This can happen as the result of a user request or intentional operation.

Suggestions: None

EventID: 6802027C (hex)

Message: 637

"The version of the calset file: <x> is not compatible with the current instrument."

Severity: Error

Further explanation: A versioning error can prevent a calset from being used. This can happen as a result of instrument firmware upgrades.

Suggestions: If the versioning error is the result of firmware upgrade, you will have to re-install the old version of firmware to re-use the calset file. Or you can re-create the calsets with the current version of firmware.

The cal set file is stored in the application home directory C:\Program Files\Agilent\Network Analyzer\PNACalSets.dat. Remove this file, then restart the application.

EventID: E802027D (hex)

Message: 638

"Incompatible CalSets found: <x> of <y> stored calsets have been loaded."

Severity: Error

Further explanation: Errors were found on some of the calsets stored in the calset file. The errors may have been caused by versioning issues that may have corrupted the various calset keys.

Suggestions: Use the calset viewer to look at the contents of calset files. Delete the files that are corrupted.

EventID: 6802027E (hex)

Message: 639

"The Calset file: <x> was not found. A new file has been created."

Severity: Informational

Further explanation: The calset file should be stored on the hard drive. When the application is started, a search is done and the file is loaded if it can be found. If the file is not found, the analyzer will create a new file and display this message.

Suggestions: None

EventID: 6802027F (hex)

Message: 640

"The Calset specified is currently in use."

Severity: Error

Further explanation: This may indicate a conflict between multiple calset users attempting calibration tasks.

Suggestions: Save the instrument state. Preset the analyzer and recall the instrument state. This may abort any processes that may be in progress.

EventID: E8020280 (hex)

Message: 641

"The calset specified has not been opened."

Severity: Error

Further explanation: Multiple users may be attempting to access the calset.

Suggestions: Close multiple calset users so that only one user will access the calset.

EventID: E8020281 (hex)

Message: 642

"The maximum number of cal sets has been reached. Delete old or unused cal sets before attempting to create new ones."

Severity: Error

Suggestions: You may also delete the calsets data file.

The cal set file is stored in the application home directory. C:\Program Files\Agilent\Network_Analyzer\PNACalSets.dat. Remove this file, then restart the application.

EventID: E8020282 (hex)

Message: 643

The requested power loss table segment was not found.

Severity: Error

EventID: E8020283 (hex)

Message: 644

"A valid calibration is required before correction can be turned on."

Severity: Error

Further explanation: This usually indicates a calibration procedure has not run to completion or that the selected measurement does not have a valid calibration available from within the currently selected cal set.

Suggestions: To find a calibration, you can select a Cal Set that contains the calibration needed for the current measurements. This error can happen if a calibration type is not specified before attempting to programmatically execute cal acquisitions. Specify the cal type before beginning a calibration procedure.

EventID: E8020284 (hex)

Message: 645

The cal data for <x> is incompatible and was not restored. Please recalibrate."

Severity: Warning

Further explanation: None

Suggestions: None

EventID: A8020285 (hex)

Message: 646

"CalSet not loaded, version is too new."

Severity: Error

Further explanation: An old version of firmware is attempting to run with a new calset version. The version is incompatible.

Suggestions: The calset can be removed. You may also delete the calsets data file if you are migrating between various firmware revisions often and you would like to avoid this error. The cal set file is stored in the application home directory. C:\Program Files\Agilent\Network Analyzer\PNACalSets.dat. Remove this file, then restart the application.

EventID: E8020286 (hex)

Message: 647

"Custom cal type not found."

Severity: Error

Further explanation:

Suggestions:

EventID: E8020287 (hex)

Message: 648

"Custom correction algorithm defers to the client for interpolation."

Severity: Informational

EventID: 68020288 (hex)

Message: 649

"Custom cal dll threw an exception."

Severity: Error

EventID: E8020289 (hex)

Message: 650

"Could not load the ecal.dll library"

Severity: Error

EventID: E802028A (hex)

Message: 656

"The argument specified is not a valid cal type."

Severity: Error

EventID: E8020290 (hex)

Message: 657

"The function found existing interpolated data"

Severity: Informational

EventID: 68020291 (hex)

Message: 658

"The function computed new interpolation values."

Severity: Informational

EventID: 68020292 (hex)

Message: 659

"The source power measurement failed."

Severity: Error

Suggestions: Please check GPIB, power meter settings and sensor connections.

EventID: E8020293 (hex)

Message: 660

"Duplicate session found. Close session and retry."

Severity: Error

EventID: E8020294 (hex)

Message: 661

"The session does not exist. Open the session and try again."

Severity: Error

Further explanation:

EventID: E8020295 (hex)

Message: 662

"Attempt to launch a custom calibration failed."

Severity: Error

Further explanation:

EventID: E8020296 (hex)

Message: 663

"Request to measure a cal standard failed."

Severity: Error

Further explanation: Please ensure you are requesting to measure standards which are defined for this calibration.

EventID: E8020297 (hex)

Message: 664

"Since Electronic Calibration Kit is selected, Mechanical Cal Kit parameter cannot be changed."

Severity: Error

Further explanation:

EventID: E8020298 (hex)

Message: 665

"Frequencies of the active channel are below minimum or above maximum frequencies of the ECal module factory characterization."

Suggestions: Change the channel frequencies, or select another ECal module.

Severity: Error

EventID: E8020299 (hex)

Message: 666

"Calset chosen for characterizing the ECal Module Ports %1 does not contain a calibration for PNA Ports %2."

Severity: Error

Suggestions: Go back to select another calset or to perform another cal.

EventID: E802029A (hex)

Message: 667

"ECal module only has sufficient memory remaining to store a maximum of %1 points in User Characterization %2."

Severity: Error

Suggestions: Decrease your number of points, or choose to overwrite another user characterization.

EventID: E802029B (hex)

Message: 668

Input values are non-monotonic. Cannot interpolate.

Severity: Error

EventID: E802029C (hex)

Message: 669

Interpolation target is out of range. Cannot interpolate.

Severity: Error

EventID: E802029D (hex)

Message: 670

Guided Calibration Error: <>

Severity: Error

EventID: E802029E (hex)

Message: 671

The first call to the guided calibration interface must be Initialize.

Severity: Error

EventID: E802029F (hex)

Message: 672

The selected thru cal method was not recognized.

Severity: Error

EventID: E80202A0 (hex)

Message: 673

Could not generate the error terms.

Severity: Error

EventID: E80202A1 (hex)

Message: 674

Guided calibration must be performed on the active channel

Severity: Error

EventID: E80202A2 (hex)

Message: 675

You can not start using calibration steps until you have successfully called generate steps.

Severity: Error

EventID: E80202A3 (hex)

Message: 676

The step number given is out of range. Step numbers should be between 1 and the number of steps. 0 is not a valid step number.

Severity: Error

EventID: E80202A4 (hex)

Message: 677

A calset was selected for channel: <n> without restoring stimulus.

Severity: Informational

EventID: 680202A5 (hex)

Message: 678

A calset was selected for channel: <n> restoring stimulus.

Severity: Informational

EventID: 680202A6 (hex)

Message: 679

The selected calset stimulus could not be applied to the channel.

Severity: Informational

EventID: 680202A7 (hex)

Message: 680

You attempted to measure power at a frequency outside the frequency range defined for the specified power sensor. Select another sensor or adjust the range for this sensor.

Severity: Error

EventID: E80202A8 (hex)

Message: 681

Specified frequency is outside the frequency ranges currently defined for the power meter's sensors.

Severity: Error

EventID: E80202A9 (hex)

Message: 682

Additional Calibration Standards need to be acquired in order to calibrate over the entire frequency range currently being measured.

Severity: Informational

EventID: 680202AA (hex)

Message: 683

The PNA failed to convert cal kits for use by unguided calibrations. The recommended action is to restore Cal Kit defaults.

Severity: Error

EventID: E80202AB (hex)

Message: 684

The PNA failed to convert cal kits for use by unguided calibrations. CalKit defaults have been restored.

Severity: Error

EventID: E80202AC (hex)

Message: 685

Power meter is reserved by a source power cal acquisition already in progress.

Severity: Error

EventID: E80202AD (hex)

Message: 686

Source power calibration has not been performed or uploaded for the specified channel and source port.

Severity: Error

EventID: E80202AE (hex)

Message: 687

Source power calibration data array size for the specified channel and source port does not match it's associated stimulus number of points.

Severity: Error

EventID: E80202AF (hex)

Message: 688

Source power calibration of Port <n> on Channel <n> was turned off because the correction array no longer exists.

Severity: Error

EventID: E80202B0 (hex)

Message: 689

This command can only be used on a measurement created with a specified calibration loadport.

Severity: Error

EventID: E80202B1 (hex)

Message: 690

Interpolation is turned off and you have changed the stimulus settings of the original calibration, so correction has been turned off.

Severity: Error

EventID: E80202B2 (hex)

Message: 691

Stimulus settings for your current instrument state exceeded the parameters of the original calibration, so correction has been turned off.

Severity: Error

EventID: E80202B3 (hex)

Message: 692

Fixturing: the requested S2P file cannot be read. Possible formatting problem.

Severity: Error

EventID: E80202B4 (hex)

Message: 693

Fixturing: the requested S2P file cannot be opened.

Severity: Error

EventID: E80202B5 (hex)

Message: 694

Fixturing: the requested S2P file cannot be interpolated. This is usually because the frequency range in the file is a subset of the current channel frequency range.

Severity: Error

EventID: E80202B6 (hex)

Message: 696

Fixturing: cannot be enabled with Response Calibrations and has been turned off.

Severity: Error

EventID: E80202B8 (hex)

Message: 697

The selected calibration cannot be performed for this measurement.

Severity: Error

EventID: E80202B9 (hex)

Message: 698

Fitting: RemoveAllConnectors() should be called prior to calling AddConnector after a fit has been attempted.

Severity: Error

EventID: E80202BA (hex)

Message: 699

An attempt was made to acquire calibration data before the system was properly initialized.

Severity: Error

EventID: E80202BB (hex)

Message: 700

Use IGuidedCalibration for multiport calibration types.

Severity: Error

EventID: E80202BC (hex)

Message: 701

Guided calibration requires number of thru measurement paths be at least equal to the number of calibration ports minus 1.

Severity: Error

EventID: E80202BD (hex)

Message: 702

A thru path was specified that includes a port which the calibration was not specified to include.

Severity: Error

EventID: E80202BE (hex)

Message: 703

One or more of the ports to be calibrated was not found in the set of specified thru paths.

Severity: Error

EventID: E80202BF (hex)

Hardware Errors

Message: 770

Input power too high. Source power is off.

Severity: Warning

EventID: A8030302 (hex)

Message: 771

Source power restored.

Severity: Informational

EventID: 68030303 (hex)

Message: 772

"The spampnp.sys driver is not working. Check system hardware. ! Data will be simulated. !"

Severity: Error

Further explanation: The Network Analyzer application cannot locate the DSP board. Hardware or a driver may be malfunctioning. This is also common when attempting to run the Network Analyzer on a workstation.

EventID: E8030304 (hex)

Message: 773

"Instrument Serial Bus Not Working."

Severity: Error

Further explanation: The instrument EEPROM appears to contain either all ones or all zeros. A serial bus hardware failure prevents reading the EEPROM.

EventID: E8030305 (hex)

Message: 848

"Phase lock lost"

Severity: Error

Further explanation: The instrument source was not able to lock properly. This can be the result of broken hardware, poor calibration, or bad EEPROM values.

Suggestions: Perform source calibration. Click System / Service / Adjustments / Source Calibration

EventID: E8030350 (hex)

Message: 849

Phaselock restored.

Severity: Success

EventID: 0x28030351 (hex)

Message: 850

"Unknown hardware error."

Severity: Error

Further explanation: Hardware malfunctioned prevents communication with the DSP.

EventID: E8030352 (hex)

Message: 851

DSP communication lost.

Severity: Error

EventID: E8030353 (hex)

Message: 852

RF power off.

Severity: Error

EventID: E8030354 (hex)

Message: 853

RF power on.

Severity: Success

EventID: 28030355 (hex)

Message: 854

Hardware OK.

Severity: Success

EventID: 28030356 (hex)

Message: 855

"Source unleveled."

Severity: Error

Further explanation: The source was unable to properly level at the requested power. The indicated power may not be accurate.

Suggestions: Try a different power level. Recalibrate source, if problem persists.

EventID: E8030357 (hex)

Message: 856

Source leveled.

Severity: Success

EventID: 28030358 (hex)

Message: 857

Input overloaded.

Severity: Error

EventID: E8030359 (hex)

Message: 858

Input no longer overloaded.

Severity: Success

EventID: 2803035A (hex)

Message: 859

"Yig calibration failed."

Severity: Error

Further explanation: Internal self-calibration of YIG oscillator tuning failed.

EventID: E803035B (hex)

Message: 860

Yig calibrated.

Severity: Success

EventID: 2803035C (hex)

Message: 861

"Analog ramp calibration failed."

Severity: Error

Further explanation: Internal analog sweep ramp calibration has failed.

EventID: E803035D (hex)

Message: 862

Analog ramp calibrated.

Severity: Success

EventID: 2803035E (hex)

Message: 863

Source temperature high.

Severity: Error

EventID: E803035F (hex)

Message: 864

Source temperature OK.

Severity: Success

EventID: 28030360 (hex)

Message: 865

"EEPROM write failed."

Severity: Error

Further explanation: Attempt to store calibration data to EEPROM has failed. There is a possible hardware failure.

EventID: E8030361 (hex)

Message: 866

EEPROM write succeeded.

Severity: Success

EventID: 28030362 (hex)

Message: 867

Attempted I/O write while port set to read only.

Severity: Error

Further explanation: Attempt to write to an I/O data port while the port set to input/read only.

Suggestions: Set data port to write/output before attempting to write to port.

EventID: E8030363 (hex)

Message: 868

" Attempted I/O read from write only port.

Severity: Error

Further explanation: Attempt to read from an I/O data port while the port set to output/write only.

Suggestions: Set data port to read/input before attempting to read from port.

EventID: E8030364 (hex)

Message: 869

Invalid hardware element identifier.

Severity: Error

EventID: E8030365 (hex)

Message: 870

Invalid gain level setting.

Severity: Error

EventID: E8030366 (hex)

Message: 871

Device driver was unable to allocate enough memory. Please try rebooting.

Severity: Error

EventID: E8030367 (hex)

Message: 872

DSP Error. Please Contact Agilent Support. Technical Information: DSP Type 1

Severity: Error

EventID: E8030368 (hex)

Message: 873

DSP Error. Please Contact Agilent Support. Technical Information: DSP Type 2

Severity: Error

EventID: E8030369 (hex)

Message: 874

DSP Error. Please Contact Agilent Support. Technical Information: DSP Type 3

Severity: Error

EventID: E803036A (hex)

Message: 875

DSP Error. Please Contact Agilent Support. Technical Information: DSP Type 4

Severity: Error

EventID: E803036B (hex)

Message: 876

DSP Error. Please Contact Agilent Support. Technical Information: DSP Type 5

Severity: Error

EventID: E803036C (hex)

Message: 910

The trigger connection argument was not recognized as valid by the firmware.

Severity: Error

EventID: 0xE803038E (hex)

Message: 911

The trigger connection specified does not support this trigger behavior

Severity: Error

EventID: E803038F (hex)

Message: 912

The trigger behavior specified was not recognized as valid by the firmware.

Severity: Error

EventID: E8030390 (hex)

Message: 913

The trigger connection specified does not physically exist on this network analyzer

Severity: Error

EventID: E8030391 (hex)

Message: 914

Cannot set "Accept Trigger Before Armed", since this hardware configuration does not support edge triggering.

Severity: Error

EventID: E8030392 (hex)

Message: 915

Cannot set "Trigger Output Enabled", since this hardware configuration does not support BNC2.

Severity: Error

EventID: E8030393 (hex)

Message: 916

Exceeded maximum trigger delay.

Severity: Error

EventID: E8030394 (hex)

Message: 917

Exceeded minimum trigger delay.

Severity: Error

EventID: E8030395 (hex)

Measure Errors

Message: 1024

If you are going to display or otherwise use a memory trace, you must first store a data trace to memory.

Severity: Warning

EventID: A8040400 (hex)

Message: 1025

"The measurement failed to shut down properly. The application is in a corrupt state and should be shut down and restarted."

Severity: Error

Further explanation: This message is displayed if the PNA application becomes corrupt. If you continue to get this error, please call customer service

EventID: E8040401 (hex)

Message: 1026

The measurement failed to shut down properly. The update thread failed to exit properly.

Severity: Warning

EventID: A8040402 (hex)

Message: 1027

"Group Delay format with CW Time or Power sweeps produces invalid data."

Severity: Warning

Further explanation: Group Delay format is incompatible with single-frequency sweeps. Invalid data is produced.

Suggestions: Ignore the data or choose a different format or sweep type.

EventID: A8040403 (hex)

Message: 1028

Severity: Informational

"MSG_LIMIT_FAILED"

Further explanation: Limit line test failed.

EventID: 68040404 (hex)

Message: 1029

Severity: Informational

"MSG_LIMIT_PASSED"

Further explanation: Limit line test passed.

EventID: 68040405 (hex)

Message: 1030

"Exceeded the maximum number of measurements allowed."

Severity: Warning

Further explanation: See Traces, Channels, and Windows on the PNA for learn about maximum measurements.

EventID: A8040406 (hex)

Message: 1031

"Network Analyzer Internal Error. Unexpected error in AddNewMeasurement."

Severity: Warning

Further explanation: If you continue to get this message, contact product support.

EventID: A8040407 (hex)

Message: 1032

"No measurement was found to perform the selected operation. Operation not completed."

Severity: Warning

Further explanation: None

Suggestions:Create a measurement before performing this operation.

EventID: A8040408 (hex)

Message: 1033

The Markers All Off command failed.

Severity: Warning

EventID: A8040409 (hex)

Message: 1034

"A memory trace has not been saved for the selected trace. Save a memory trace before attempting trace math operations."

Severity: Warning

Further explanation: Must have a memory trace when trying to do Trace Math,

EventID: A804040A (hex)

Message: 1035

"MSG_SET_AVERAGE_COMPLETE"

Severity: Informational

Further explanation: Informational for COM programming. Averaging factor has been reached.

EventID: 6804040B (hex)

Message: 1036

"MSG_CLEAR_AVERAGE_COMPLETE"

Further explanation: Informational for COM programming. Averaging factor has NOT been reached.

EventID: 6804040C (hex)

Message: 1037

"Time Domain transform requires at least 3 input points. The transform has been deactivated."

Severity: Informational

Further explanation: None

Suggestions:Increase the number of points.

EventID: 6804040D (hex)

Message: 1038

Smoothing requires a scalar format, and has been deactivated.

Severity: Informational

EventID: 6804040E (hex)

Message: 1039

A receiver power calibration in this instrument state file cannot be recalled into this firmware version.

Severity: Warning

EventID: A804040F (hex)

Message: 1104

"Exceeded limit on number of measurements."

Severity: Error

Further explanation: See Traces, Channels, and Windows on the PNA for measurement limits.

EventID: E8040450 (hex)

Message: 1105

"Parameter not valid."

Severity: Error

Further explanation: A measurement parameter that was entered programmatically is not valid.

EventID: E8040451 (hex)

Message: 1106

"Measurement not found."

Severity: Error

Further explanation: Any of these could be the cause:

Trying to calibrate but already have maximum measurements.

Trying to do a confidence check but there is not a measurement.

Trying to create, activate, or alter a measurement through COM that has been deleted through the front panel.

Trying to use a trace name through programming that is not unique.

EventID: E8040452 (hex)

Message: 1107

"No valid memory trace."

Severity: Error

Further explanation: Must have a memory trace when trying to do Trace Math,

Suggestions: Store a memory trace.

EventID: E8040453 (hex)

Message: 1108

"The reference marker was not found."

Severity: Error

Further explanation: Attempted to create a delta marker without first creating a reference marker (COM only).

EventID: E8040454 (hex)

Message: 1109

"Data and Memory traces are no longer compatible. Trace Math has been turned off."

Severity: Error

Further explanation: Warning - channel setting has changed while doing trace math.

Suggestions:Store another memory trace and turn trace math back on.

EventID: A8040455 (hex)

Message: 1110

"Data and Memory traces are not compatible. For valid trace math operations, memory and data traces must have similar measurement conditions."

Severity: Error

Further explanation: Tried to do trace math without compatible data and memory traces.

Suggestions: Store another memory trace.

EventID: E8040456 (hex)

Message: 1111

"Marker Bandwidth not found."

Severity: Error

Further explanation: Could not find a portion of trace that meets the specified bandwidth criteria.

EventID: E8040457 (hex)

Message: 1112

"The peak was not found."

Severity: Error

Further explanation: Could not find portion of trace that meets peak criteria.

Suggestions: See Marker Peak criteria.

EventID: E8040458 (hex)

Message: 1113

"The target search value was not found."

Severity: Error

Further explanation: Could not find interpolated data point that meets search value.

EventID: E8040459 (hex)

Message: 1114

"Reflection measurement, such as S11, must supply an auxiliary port to disambiguate 2-port measurements on multiport instruments."

Severity: Error

Further explanation:

EventID: E804045A (hex)

Message: 1115

"The receiver power calibration has been turned off because the type of measurement or source port has changed, so the calibration is no longer valid."

Severity: Warning

Further explanation:

EventID: A804045B (hex)

Message: 1116

"Receiver power cal requires the active measurement to be of unratioed power."

Severity: Warning

Further explanation:

EventID: A804045C (hex)

Message: 1117

"There is currently no source power calibration associated with the channel and source port of the active measurement. A source power cal should be performed or recalled before performing a receiver power calibration."

Severity: Warning

Further explanation:

EventID: A804045D (hex)

Message: 1118

"The attempted operation can only be performed on a standard measurement type."

Severity: Error

Further explanation:

EventID: E804045E (hex)

Message: 1119

"The custom measurement cannot be loaded because it is not compatible with the Network Analyzer hardware."

Severity: Error

Further explanation:

Suggestions:

EventID: E804045F (hex)

Message: 1120

"The custom measurement cannot be loaded because it is not compatible with the Network Analyzer software."

Severity: Error

Further explanation:

EventID: E8040460 (hex)

Message: 1121

"The custom measurement load operation failed for an unspecified reason."

Severity: Error

Further explanation:

EventID: E8040461 (hex)

Message: 1122

"The custom measurement data processing has generated an unhandled exception, and will be terminated. The PNA software may be in an unstable state and it is recommended that the PNA software be shutdown and restarted."

Severity: Error

Further explanation:

EventID: E8040462 (hex)

Message: 1123

"The attempted operation can only be performed on a custom measurement type."

Severity: Error

Further explanation:

EventID: E8040463 (hex)

Message: 1124

"The requested custom measurement is not available."

Severity: Error

Further explanation:

EventID: E8040464 (hex)

Message: 1125

"The requested custom algorithm was not found."

Severity: Error

Further explanation:

EventID: E8040465 (hex)

Message: 1126

"Normalization cannot be turned on because the measurement does not have a valid divisor buffer."

Severity: Error

Further explanation:

EventID: E8040466 (hex)

Message: 1127

"The Raw Data requested by the measurement could not be provided."

Severity: Warning

Further explanation:

EventID: A8040467 (hex)

Message: 1128

"The selected Sweep Type does not allow Transform and Gating. Transform and Gating disabled. "

Severity: Error

Further explanation:

EventID: E8040468 (hex)

Message: 1129

Memory trace can not be applied to this measurement

Severity: Error

EventID: E8040469 (hex)

Message: 1130

Normalization can not be applied to this measurement

Severity: Error

EventID: E804046A (hex)

Message: 1131

The data provided has an invalid number of points. It could not be stored

Severity: Error

EventID: E804046B (hex)

Message: 1132

The measurement stored in the save/recall state has an invalid version. It could not be loaded

Severity: Error

EventID: E804046C (hex)

Message: 1133

This data format argument for this operation must be "naDataFormat_Polar"

Severity: Error

EventID: E804046D (hex)

Message: 1134

This data format argument for this operation must be a scalar data format

Severity: Error

EventID: E804046E (hex)

Message: 1135

The memory trace is not valid for the current measurement setup.

Severity: Error

EventID: E804046F (hex)

Message: 1136

This measurement is incompatible with existing measurements in this channel. Choose another channel.

Severity: Error

EventID: E8040470 (hex)

Message: 1137

Port extension correction is not available for offset frequency measurements. Port extension correction has been disabled.

Severity: Error

EventID: E8040471 (hex)

Message: 1138

Physical port number assignments for logical port mappings must be unique.

Severity: Error

EventID: E8040472 (hex)

Parser Errors

Message: 1281

"You have sent a read command to the analyzer without first requesting data with an appropriate output command. The analyzer has no data in the output queue to satisfy the request."

Severity: Error

EventID: 68050501 (hex)

Message: 1282

"You must remove the active controller from the bus or the controller must relinquish the bus before the analyzer can assume the system controller mode."

Severity: Error

EventID: E8050502(hex)

Message: 1283

"The analyzer did not receive a complete data transmission. This is usually caused by an interruption of the bus transaction."

Severity: Error

EventID: E8050503 (hex)

Message: 1284

"The instrument status byte has changed."

Severity: Informational

EventID: 68050504 (hex)

Message: 1285

"The SCPI command received has caused error number %1: "%2"."

Severity: Informational

EventID: 68050505 (hex)

Message: 1286

"The INET LAN server has been started as process number %1."

Severity: Informational

EventID: 68050506 (hex)

Message: 1360

"Execution of the SCPI command has failed"

Severity: Error

EventID: E8050550 (hex)

Message: 1361

" The INET/LAN device is not accessible."

Severity: Error

EventID: E8050551 (hex)

Message: 1362

"The INET/LAN driver was not found. "

Severity: Error

EventID: E8050552 (hex)

Message: 1363

"The INET/LAN driver was not found."

Severity: Error

EventID: E8050553 (hex)

Message: 1364

"The INET/LAN device is unable to acquire the necessary resources. "

Severity: Error

EventID: E8050554 (hex)

Message: 1365

"The INET/LAN device generated a generic system error. "

Severity: Error

EventID: E8050555 (hex)

Message: 1366

"Invalid address for the INET/LAN device."

Severity: Error

EventID: E8050556 (hex)

Message: 1367

"The INET I/O library was not found. "

Severity: Error

EventID: E8050557 (hex)

Message: 1368

"An error occured in the INET system. "

Severity: Error

EventID: E8050558 (hex)

Message: 1369

"Access to the INET/LAN driver was denied. "

Severity: Error

EventID: E8050559 (hex)

Message: 1370

"Could not load error system message dll."

Severity: Error

EventID: E805055A (hex)

Message: 1371

"ErrorSystemMessage.dll does not export the right function."

Severity: Error

EventID: E805055B (hex)

Message: 1372

"Custom scpi library was not able to be knitted"

Severity: Error

EventID: E805055C (hex)

Message: 1373

"Could not knit the scpi error messages from the ErrorSystemMessage lib"

Severity: Error

EventID: E805055D (hex)

Message: 1374

Command is obsolete with this software version.

Severity: Error

EventID: E808055E (hex)

Message: 1375

CALC measurement selection set to none. Use Calc:Par:Sel

Severity: Error

EventID: E808055F (hex)

Message: 1535

"Parser got command: %1."

Severity: Informational

EventID: 680505FF (hex)

Display Errors 1536 - 1621

Message: 1536

"Exceeded the maximum of 4 traces in each window. The trace for <x> will not be added to window <x>."

Severity: Warning

Further explanation: None

Suggestions: Create the trace in another window. See the PNA window limits.

EventID: A8060600 (hex)

Message: 1537

"Exceeded the maximum of 16 data windows. New window will not be created."

Severity: Warning

Further explanation: None

Suggestions: Create the trace in an existing window. See the PNA window limits.

EventID: A8060601 (hex)

Message: 1538

"No Data Windows are present. Unable to complete operation."

Severity: Warning

Further explanation: Your remote SCPI operation tried to create a new measurement while there were no windows present

Suggestions: Create a new window before creating the measurement. See example Create a measurement using SCPI

EventID: A8060602 (hex)

Message: 1539

"No data traces are present in the selected window. Operation not completed."

Severity: Warning

Further explanation: None

EventID: A8060603 (hex)

Message: 1540

"Cannot complete request to arrange existing measurements in <x> windows due to the limit of <x> traces per window."

Severity: Informational

Further explanation: The arrange window feature cannot put the existing traces into the number of windows you requested because only 4 traces per window are allowed. See Arranging Existing Measurements

Suggestions:Either create more windows or delete some traces.

EventID: 68060604 (hex)

Message: 1541

"Unable to establish a connection with the specified printer."

Severity: Warning

Further explanation: None

Suggestions: Refer to Printer Help

EventID: A8060605 (hex)

Message: 1542

"Printout canceled."

Severity: Informational

EventID: 68060606 (hex)

Message: 1616

"Window not found."

Severity: Error

Further explanation: A window was specified in your program which does not exist.

Suggestions: Query the name of your window before specifying.

EventID: E8060650 (hex)

Message: 1617

"Duplicate window ID specified."

Severity: Error

Further explanation: None

EventID: E8060651 (hex)

Message: 1618

"Exceeded limit on number of windows."

Severity: Error

Further explanation: There is a limit of 4 windows per screen.

EventID: E8060652 (hex)

Message: 1619

"Exceeded limit on number of traces/window."

Severity: Error

Further explanation: There is a limit of 4 traces per window. See the Traces, Channels, and Windows on the PNA.

Suggestions: Create the trace in another window

EventID: E8060653 (hex)

Message: 1620

"Trace not found."

Severity: Error

Further explanation: Your program tried to communicate with a non-existing trace.

Suggestions:Query the trace ID before writing to it.

EventID: E8060654 (hex)

Message: 1621

"The operating system does not recognize this printer."

Severity: Warning

EventID: A8060655 (hex)

Message: 1622

Duplicate trace ID specified.

Severity: Error

EventID: E8060656 (hex)

Channel Errors 1792 -1878

Message: 1792

"Sweep Complete."

Severity: Informational

Further explanation: None

Suggestions:None

EventID: 68070700 (hex)

Message: 1793

"All triggerable acquisitions have completed."

Severity: Informational

Further explanation:

EventID: 68070701 (hex)

Message: 1794

"The last trigger produced an aborted sweep."

Severity: Informational

Further explanation:

EventID: 68070702 (hex)

Message: 1795

"The segment list must be adjusted to have at least one active segment with more than 0 points to use segment sweep."

Severity: Informational

Further explanation: You attempted to change **Sweep type** to Segment sweep, but there is either no segments defined or no sweep points in the defined segments

Suggestions: Define at least one segment with at least one measurement point. See Segment sweep for more information

EventID: 68070703 (hex)

Message: 1796

"MSG_SET_CHANNEL_DIRTY"

Severity: Informational

Further explanation: This informational message occurs when a channel setting has changed but the channel still has data that was taken with the previous setting. The following CLEAR message occurs when new channel data is taken.

EventID: 68070704 (hex)

Message: 1797

"MSG_CLEAR_CHANNEL_DIRTY"

Severity: Informational

Further explanation: The previous SET message occurs when a channel setting has changed but the channel still has data that was taken with the previous setting. This CLEAR message occurs when new channel data is taken.

EventID: 68070705 (hex)

Message: 1798

Sweep time has changed from Auto to Manual mode. If desired to return to Auto mode, enter sweep time value of 0.

Severity: Informational

EventID: 68070706 (hex)

Message: 1799

"Set Sweep Completed"

Severity: Informational

Further explanation: This event occurs when a sweep and it's associated sweep calculations finish. This is typically when all sweeps on a channel complete.

EventID: 68070707 (hex)

Message: 1800

"Clear Sweep Completed"

Severity: Informational

Further explanation: This event occurs immediately after the SET SWEEP COMPLETED event. These two events set and clear the "Sweep Completed" bit (bit 4) on the SCPI Device Status register.

EventID: 68070708 (hex)

Message: 1801

"All Sweeps Completed and Processed"

Severity: Informational

Further explanation: This event occurs when all of the sweeps and sweep calculations are complete for a channel.

EventID: 68070709 (hex)

Message: 1802

Low Pass : Frequency limits have been changed.

Severity: Informational

EventID: 6807070A (hex)

Message: 1803

Low Pass : Number of points have been changed.

Severity: Informational

EventID: 6807070B (hex)

Message: 1804

Low Pass : Frequency limits and number of points have been changed.

Severity: Informational

EventID: 6807070C (hex)

Message: 1872

"Channel not found."

Severity: Error

Further explanation: A non-existent channel is being referenced under program control.

Suggestions: Query the channel number, then refer to it by number.

EventID: E8070750 (hex)

Message: 1873

"The requested sweep segment was not found."

Severity: Error

Further explanation: A non-existent sweep segment is being referenced under program control.

EventID: E8070751 (hex)

Message: 1874

"The sweep segment list is empty."

Severity: Error

Further explanation: Segment Sweep cannot be specified unless there is at least one defined segment. This error will only occur under remote control.

EventID: E8070752 (hex)

Message: 1875

"The number of points in active sweep segment list segments is 0."

Severity: Error

Further explanation: Segment Sweep cannot be specified unless there is at least data point specified in a segment. This error will only occur under remote control.

EventID: E8070753 (hex)

Message: 1876

"The specified source attenuator is not valid."

Severity: Error

Further explanation: You tried to set the Attenuator property on the Channel object on a PNA that doesn't have a source attenuator.

EventID: E8070754 (hex)

Message: 1877

"Log Frequency sweep cannot be selected with the current Number of Points. Please reduce Number of Points."

Severity: Error

Further explanation: The maximum number of points that can be used for Log sweep is 401.

EventID: E8070755 (hex)

Message: 1878

"The requested Number of Points is greater than can be selected for Log Frequency sweep."

Severity: Error

Further explanation: The maximum number of points that can be used for Log sweep is 401.

EventID: E8070756 (hex)

Message: 1879

"Response frequencies exceeded instrument range so Frequency Offset has been turned off."

Severity: Error

Further explanation: This error is returned whenever the instrument detects that the stimulus sweep setup and Frequency Offset settings result in computed response frequencies that exceed instrument limits. When this occurs, the instrument automatically turns off Frequency Offset to avoid the out-of-range conditions.

Suggestions: When this condition has occurred, change settings for either the stimulus frequencies or Frequency Offset so that the Response frequencies are within instrument bounds. Once this is done, Frequency Offset can once again be turned on.

EventID: E8070757 (hex)

Message: 1880

The total number of points for all the given segments exceeds the maximum number of points supported. The segments were not changed.

Severity: Error

EventID: E8070758 (hex)

Message: 1881

This instance of the Channels object was not used to place the channels in Hold, so no channels were resumed.

Severity: Error

EventID: E8070759 (hex)

Message: 1882

The port number was outside the range of allowed port numbers.

Severity: Error

EventID: E807075A (hex)

Message: 1883

More ports than are present are required for this operation.

Severity: Error

EventID: E807075B (hex)

General Errors

Message: 2048

"The function you requested requires a capability provided by an option to the standard analyzer. That option is not currently installed."

Severity: Error

Further explanation: None

Suggestions:To view the options on your analyzer, click **Help \ About Network Analyzer**. For more information see PNA Options

EventID: 68080800 (hex)

Message: 2049

"The feature you requested is not available on the current instrument."

Severity: Error

Further explanation: None

EventID: 68080801 (hex)

Message: 2050

"The feature you requested is incompatible with the current instrument state."

Severity: Error

Further explanation: None

Suggestions: None

EventID: 68080802 (hex)

Message: 2051

"File<x> has been saved."

Severity: Informational

Further explanation: None

EventID: 68080803 (hex)

Message: 2052

"Attempt to save <x> failed."

Severity: Error

Further explanation: None

Suggestions: If using a floppy disk, ensure it is inside the drive and the disk is not full. Check the filename for special characters.

EventID: E8080804 (hex)

Message: 2053

"Attempt to recall file failed because <x> was not found."

Severity: Error

Further explanation: None

EventID: E8080805 (hex)

Message: 2054

"<x> has a bad header."

Severity: Error

Further explanation: None

Suggestions: Recopy the file and / or delete the file.

EventID: E8080806 (hex)

Message: 2056

"Request to enter hibernate state."

Further explanation: None

EventID: 68080808 (hex)

Message: 2057

"Power up from automatic hibernate state. Program received PBT_APMRESUMEAUTOMATIC Message."

Further explanation: None

EventID: 68080809 (hex)

Message: 2058

"Power up from suspend hibernate state. Program received PBT_APMRESUMESUSPEND Message."

Further explanation: None

EventID: 6808080A (hex)

Message: 2059

"Power up from suspend hibernate state. Program received PBT_APMRESUMECRITICAL Message."

Severity: Warning

Further explanation: None

EventID: A808080B (hex)

Message: 2060

"Power up from unknown hibernate state UI recovery called. Program received no PBT_Message within the time allotted and is attempting recovery."

Severity: Warning

Further explanation: None

EventID: A808080C (hex)

Message: 2061

"<x> already exists. File is being overwritten."

Further explanation: Used only for remote applications

EventID: 6808080D (hex)

Message: 2062

"File has not been saved."

Severity: Error

Further explanation: Used only for remote applications

EventID: E808080E (hex)

Message: 2063

"File <x> has been recalled."

Further explanation: Used only for remote applications

EventID: 6808080F (hex)

Message: 2064

"State version in <x> is considered obsolete by this version of this code."

Severity: Error

Further explanation: You attempted to recall a file that is no longer valid.

Suggestions: You must recreate the file manually.

EventID: E8080810 (hex)

Message: 2065

"State version in <x> is newer than the latest version supported by this code."

Severity: Error

Further explanation: You attempted to recall a file that was created by a later version of the PNA application.

Suggestions: You must recreate the file manually.

EventID: E8080811 (hex)

Message: 2066

"Error occurred while reading file <x>"

Severity: Error

Further explanation: The file may be corrupt.

Suggestions: Try to recreate the file.

EventID: E8080812 (hex)

Message: 2067

"Windows shell error: <x>"

Severity: Error

Further explanation: None

EventID: E8080813 (hex)

Message: 2068

Send message timed out returning: <x>.

Severity: Error

Further explanation: None

EventID: E8080814 (hex)

Message: 2069

"Changing GPIB mode to System Controller."

Severity: Informational

Further explanation: None

EventID: 68080815 (hex)

Message: 2070

"Changing GPIB mode to Talker Listener."

Severity: Informational

Further explanation: None

EventID: 68080816 (hex)

Message: 2071

"The Network Analyzer can not be put in GPIB System Controller mode until the GPIB status is Local. Stop any remote GPIB programs which may be using the Network analyzer, press the Macro/Local key and try again. "

Severity: Informational

Further explanation: See **LCL and RMT Operation**

Suggestions: Press the Macro/Local key and try again.

EventID: 68080817 (hex)

Message: 2120

"This method can not be invoked through a late-bound COM call."

Severity: Error

Further explanation: None

Suggestions: Use the alternate method described in the COM programming documentation

EventID: E8080878 (hex)

Message: 2128

"The specified format is invalid."

Severity:Error

Further explanation: None

EventID: E8080850 (hex)

Message: 2129

"WINNT exception caught by Automation layer."

Severity: Error

Further explanation: None

EventID: E8080851 (hex)

Message: 2130

"Bad port specification."

Severity: Error

Further explanation: None

EventID: E8080852 (hex)

Message: 2131

"Failed to find a printer."

Severity: Error

Further explanation: None

Suggestions: See Connecting to a Printer

EventID: E8080853 (hex)

Message: 2132

"Manual trigger ignored."

Severity: Error

Further explanation: None

EventID: E8080854 (hex)

Message: 2133

"Attempt to set trigger failed."

Severity: Error

Further explanation: None

EventID: E8080855 (hex)

Message: 2134

"Macro execution failed."

Severity: Error

Further explanation: None

EventID: E8080856 (hex)

Message: 2135

"Specified macro definition is incomplete."

Severity: Error

Further explanation:

EventID: E8080857 (hex)

Message: 2137

"Block data length error."

Severity: Error

Further explanation: See Getting Data from the Analyzer

EventID: E8080859 (hex)

Message: 2139

"Requested data not found."

Severity: Error

Further explanation: None

EventID: E808085B (hex)

Message: 2142

"The parameter supplied was out of range, so was limited to a value in range before being applied to the instrument."

Severity: Success

Further explanation: None

Suggestions: View range limits before sending programming commands.

EventID: 2808085E (hex)

Message: 2143

The parameter supplied was out of range, so was limited to a value in range before being applied to the instrument.

Severity: Error

EventID: E808085F (hex)

Message: 2144

"Request failed. The required license was not found."

Severity: Error

Further explanation: None

EventID: E8080860 (hex)

Message: 2145

"A remote call to the front panel has returned hresult <x>"

Severity: Error

Further explanation: This may indicate a problem with the front panel

Suggestions: Contact Technical support

EventID: E8080861 (hex)

Message: 2146

The recall operation failed.

Severity: Error

Further explanation:

EventID: E8080862 (hex)

Message: 2147

Attempt to save file failed.

Severity: Error

Further explanation:

EventID: E8080863 (hex)

Message: 2148

Recall attempt failed because file was not found.

Severity: Error

Further explanation:

EventID: E8080864 (hex)

Message: 2149

Recall file has a bad header.

Severity: Error

Further explanation:

EventID: E8080865 (hex)

Message: 2150

Recall file version is obsolete and no longer compatible with this instrument.

Severity: Error

Further explanation:

EventID: E8080866 (hex)

Message: 2151

The recall file contains an istate version newer than this instrument. A remote call to the front panel has returned hresult %1

Severity: Error

Further explanation:

EventID: E8080867 (hex)

Message 2152

"Front Panel <x>

Severity: Error

Further explanation: None

EventID: E8080868 (hex)

Message 2153

"Front Panel message"

Severity: Informational

Further explanation: None

EventID: 68080869 (hex)

Message 2154

"Power Service <x>

Severity: Error

Further explanation: There is more than 1 instance of powerservice running. There should only be one running. This might happen after running install shield - especially when upgrading the CPU board.

Suggestions: Try rebooting. If this persists, please call Customer Support.

EventID: E808086A (hex)

Message 2155

"Power Service <x>

Severity: Informational

Further explanation: None

EventID: 6808086B (hex)

Message 2156

"The Agilent Technologies GPIB driver can not be loaded or unloaded."

Severity: Error

Further explanation: None

Suggestions: If the problem persists, from the PNA desktop, right-click on My Computer. Click Properties, Click Hardware Tab. Click Device Manager Button. Expand GPIB Devices. Right-click and click Uninstall all GPIB interfaces devices. Reboot the PNA.

EventID: E808086C (hex)

Message 2157

"The National Instruments GPIB driver can not be loaded or unloaded."

Severity: Error

Further explanation: None

Suggestions: If the problem persists, from the PNA desktop, right-click on My Computer. Click Properties, Click Hardware Tab. Click Device Manager Button. Expand GPIB Devices. Right-click and click Uninstall all GPIB interfaces devices. Reboot the PNA.

EventID: E808086D (hex)

Message 2158

"The Agilent GPIB driver is loaded but it can not start its parser."

Severity: Error

Further explanation: None

EventID: E808086E (hex)

Message: 2159

The front panel is in remote mode.

Severity: Warning

EventID: A808086F (hex)

Message: 2160

The Registry Key specified could not be found.

Severity: Error

EventID: E8080870 (hex)

Message: 2161

An overcurrent condition has been detected on a probe plugged into the front panel.

Severity: Warning

EventID: A8080871 (hex)

Message: 2162

The operation timed out.

Severity: Error

EventID: E8080872 (hex)

Message 2163

"The Network Analyzer executed a preset."

Severity:Informational

Further explanation: None

EventID: 68080873 (hex)

Message 2164

"Access to file denied."

Severity: Error

Further explanation: This means that the system can not open an output file for writing. Most likely because the file is write protected.

Suggestions: Pick another file name or file directory, check floppy disk hard disk write access.

EventID: E8080874 (hex)

Message 2165

"File type is structured storage."

Severity: Informational

Further explanation: None

EventID: 68080875 (hex)

Message 2166

"The trigger operation failed."

Severity: Error

Further explanation: None

EventID: E8080876 (hex)

Message 2167

"Argument out of range error."

Severity: Error

Further explanation: None

Suggestions: None

EventID: E8080877 (hex)

Message: 2169

The given COM object is not a custom application

Severity: Error

EventID: E8080879 (hex)

Message: 2170

The eventID supplied was not recognized as a valid PNA eventID

Severity: Error

EventID: E808087A (hex)

Message: 2171

The operation was canceled.

Severity: Error

EventID: E808087B (hex)

Message: 2172

High security level cannot be disabled directly. Only an instrument preset or recall of lower security instrument state will reset this security level.

Severity: Error

EventID: E808087C (hex)

Message: 2173

Local lockout mode is on. The PNA application will not accept input from front panel, keyboard or mouse until this mode is turned off from a remote interface.

Severity: Error

EventID: E808087D (hex)

Message: 2174

The SnP request is not valid for the selected measurement.

Severity: Error

EventID: E808087E (hex)

Message: 2175

Preset is not supported while this dialog or wizard is open. Close the dialog or wizard and then try again.

Severity: Error

EventID: E808087F (hex)

Message: 2176

The function you requested requires a capability provided by an option to the standard analyzer. That option is not currently installed.

Severity: Error

EventID: E8080880 (hex)

Message: 2177

Catastrophic error. Crash dump recorded at <n>

Severity: Error

EventID: E8080881 (hex)

Message: 2178

<n>

Severity: Error

EventID: E8080882 (hex)

Message: 2179

Failed to open gen.lic.

Severity: Error

EventID: E8080883 (hex)

About Error Messages

PNA errors and Operating System errors are displayed and logged in an error file. You can choose how to display PNA errors, or choose to not display PNA errors at all.

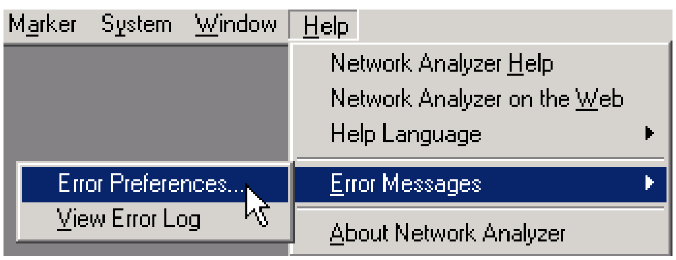
- Error Preferences
- View Error Log
- List of PNA Errors
- SCPI Errors

Other Support topics

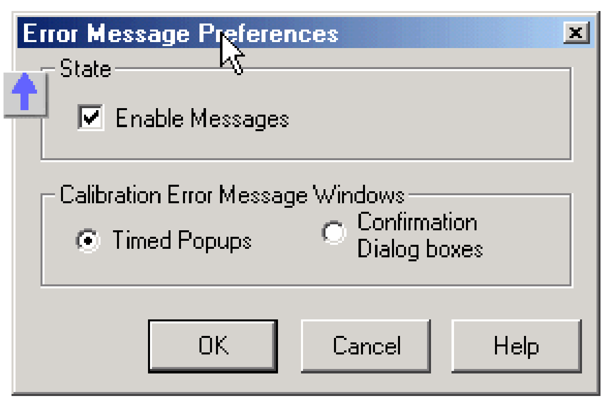
Error Preferences

By default, error messages appear on the PNA screen for a brief period. You can choose to have them stay on the screen until you click an OK button, or have them not appear at all. When they stay on the screen, a Help button is available to provide further assistance.

How to select Error Preferences



Learn more about using the front panel interface



Error Preferences dialog box help

Enable Messages Check to display all PNA error messages as they occur. Clear to suppress the display of PNA error messages. You can still view them in the error log.

Calibration Error Message Windows

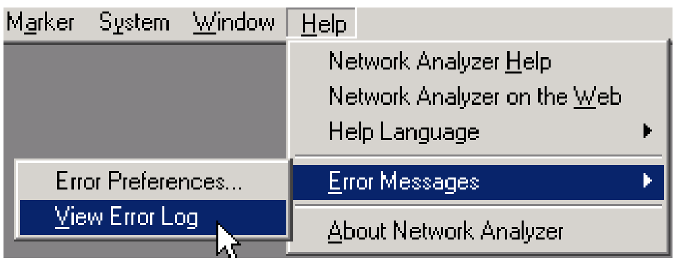
Timed Popups Displays error messages on the screen for a duration of time proportional to the length of the message. You can then view the message in the error log and get further assistance.

Confirmation Dialog boxes Displays error messages in a standard dialog box. You then choose OK or Cancel to close the dialog box, or press Help to get further information on the error message.

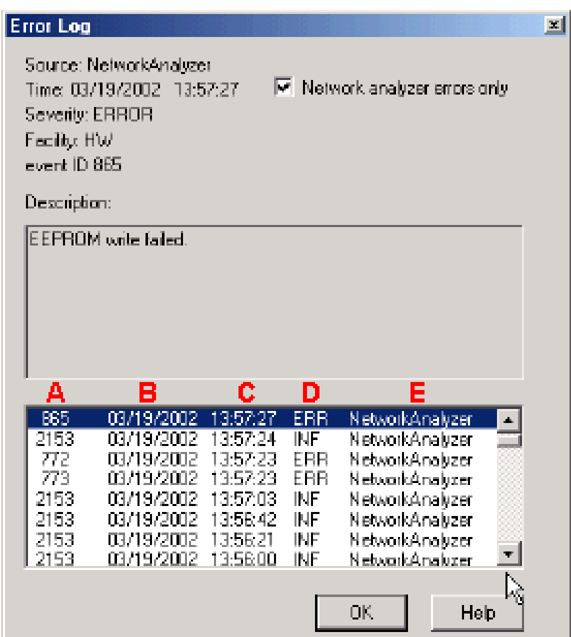
View Error Log

The PNA Error Log is a list of all events that have occurred. (Events are used in programming the PNA using COM.) PNA errors is a subset of PNA events. Only events with severity codes of ERROR are displayed on the PNA screen as they occur. From the error log, you can access further help with an error by selecting the error and clicking Help.

How to view the Error Log



Learn more about using the front panel interface



Error Log dialog box help

Network analyzer errors only Select to view only PNA errors. Clear to view all errors that occur on all applications of the computer.

Description Error message that appears on the PNA screen.

A - Event ID Error message number

B - Date the Error occurred

C - Time the Error occurred

D - Severity Code **-** All events have one of the following severity codes:

- SUCcess **-** the operation completed successfully
- INFormational - events that occur without impact on the measurement integrity
- WARning **-** events that occur with potential impact on measurement integrity
- ERRor - events that occur with serious impact on measurement integrity

E - Application in which the error occurred.

OK Closes the Dialog box

Help Provides further information on the selected Error message

To clear the Error Log:

1. From the **View** menu click **Minimize Application**
2. On the desktop, select **Start, Settings, Control Panel**
3. On the Control Panel, click **Administrative Tools**
4. On the Administrative Tools window, click **Event Viewer**
5. On the Event Viewer window, right-click **Application Log**
6. Select **Clear all Events**
7. If you want to save a file with the contents of the Event Log, click **Yes**. Otherwise, click **No**

To restore the PNA application, click on the PNA Analyzer taskbar button at the bottom of the screen

Analyzer Accessories

- **Coax Mechanical Calibration Kits**
- **Waveguide Mechanical Calibration Kits**
- **Electronic Calibration (ECal)**
- **Mechanical Verification Kits**
- **Adapter and Accessory Kits**
- **Test Port Cables**
- **USB Peripherals**
- **Connector Care and Cleaning Supplies**
- **ESD Protection**

Other Support topics

For product and order information:

- Call 1-800-452-4844 (8am-8pm EST)
- Visit www.agilent.com/find/accessories
  Use the search function to locate information about a particular accessory or view the entire RF and Microwave Test Accessories Catalog.

Accessories are available in these connector types:

- 50 ohm Type-N
- 75 ohm Type-N
- 3.5 mm
- 7 mm (APC-7)
- 7-16
- 2.92 mm
- 2.4 mm
- 1.85 mm
- 1 mm

Test port cables and a calibration kit are necessary for a complete measurement system.

A verification kit is used to verify corrected system performance.

See the connector type for each PNA model

Coax Mechanical Calibration Kits

Model	Connector Type	Frequency Upper Limit
85032B	Type-N (50 Ohm)	6 GHz
85032F	Type-N (50 Ohm)	9 GHz
85054B	Type-N (50 Ohm)	18 GHz
85036E	Type-N (75 Ohm)	3 GHz
85050B	7 mm	18 GHz
85033D	3.5 mm	6 GHz
85038A	7-16	7.5 GHz
85033E	3.5 mm	9 GHz
85052B	3.5 mm	26.5 GHz
85052C	3.5 mm TRL	26.5 GHz
85056K	2.92 mm	50 GHz
85056A	2.4 mm	50 GHz
85058B/E (data-based)	1.85 mm	67 GHz
85059A (data-based)	1.00 mm	110 GHz

Waveguide Mechanical Calibration Kits

Model	Connector Type	Frequency Range
X11644A	WR-90	8.2-12.4 GHz
P11644A	WR-62	12.4-18 GHz
K11644A	WR-42	18-26.5 GHz
R11644A	WR-28	26.5-40 GHz
Q11644A	WR-22	33-50 GHz
U11644A	WR-19	40-60 GHz
V11644A	WR-15	50-75 GHz

Electronic Calibration (ECal)

Model	Connector Type	Frequency Range
RF Two-Port		
85091C	7 mm (APC-7)	300 kHz-9 GHz
85092C	Type-N (50 ohm) Port B available with 3.5 mm or 7-16[a]	300 kHz-9 GHz
85093C	3.5 mm Port B available with Type-N (50 ohm) or 7-16[a]	300 kHz-9 GHz
85096C	Type-N (75 ohm)	300 kHz-3 GHz
85098C	7-16[a] Port B available with Type-N (50 ohm) or 3.5 mm	300 kHz-7.5 GHz
85099C	Type-F	300 kHz-3 GHz
RF Four-Port		
N4431A Option 010	3.5mm (f) (four-port)	300 kHz-9 GHz
N4431A Option 020	Type-N (f) (four-port)	300 kHz-9 GHz
Microwave Two-Port		

N4690A	Type-N (50 ohm)	10 MHz-18 GHz
N4691A	3.5 mm	10 MHz-26.5 GHz
N4692A	2.92 mm	10 MHz-40 GHz
N4693A	2.4 mm	10 MHz-50 GHz
N4694A	1.85 mm	10 MHz-67 GHz
N4696A	7 mm	10 MHz-18 GHz

a Limits ECal module high frequency to 7.5 GHz.

Verification Kits

Model	Connector Type	Frequency Range
85055A	Type-N (50 Ohm)	300 kHz-9 GHz
85053B	3.5 mm	300 kHz-26.5 GHz
85057B	2.4 mm	.045-50 GHz
R11645A	WR-28	26.5-40 GHz
Q11645A	WR-22	33-50 GHz

Adapters and Accessory Kits

Model	Description
11878A	Type-N to 3.5 mm Adapter Kit
11525A	Type-N (m) to 7 mm (APC-7)
11853A	Type-N Accessory Kit
11900B	2.4 mm (f) to 2.4 mm (f)
11900C	2.4 mm (f) to 2.4 mm (m)
85130G	Test Port Adapter Set, 2.4 mm (f) to 2.4 mm (m,f)
11901B	2.4 mm (f) to 3.5 mm (f)
11901D	2.4 mm (f) to 3.5 mm (m)

85130F	Test Port Adapter Set, 2.4 mm (f) to 3.5 mm (m,f)
11902B	2.4 mm (f) to 7 mm (APC-7)
11920A	1 mm (m) to 1 mm (m)
11920B	1 mm (f) to 1 mm (f)
11920C	1 mm (m) to 1 mm (f)
11921A	1 mm (m) to 1.85 mm (m)
11921B	1 mm (f) to 1.85 mm (f)
11921C	1 mm (m) to 1.85 mm (f)
11921D	1 mm (f) to 1.85 mm (m)
11922A	1 mm (m) to 2.4 mm (m)
11922B	1 mm (f) to 2.4 mm (f)
11922C	1 mm (m) to 2.4 mm (f)
11922D	1 mm (f) to 2.4 mm (m)

Test Port Cables

Model	Description
N4697E	1.85 mm (f) to 1.85 mm (rugged f) flexible (single)
N4697F	1.85 mm (rugged f, f) to 1.85 mm (rugged m, rugged f) flexible (set)
N6315A	Type-N (m) to Type-N (f), 16 in. (single)
N6314A	Type-N (m) to Type-N (m), 24 in. (single)
85133D	2.4 mm (f) to 2.4 mm (m,f) semi-rigid (set)
85133F	2.4 mm (f) to 2.4 mm (m,f) flexible (set)
85134D	2.4 mm (f) to 3.5 mm (m,f) semi-rigid (set)
85134F	2.4 mm (f) to 3.5 mm (m,f) flexible (set)

USB Peripherals

Model	Description
N4688A	**CD RW drive -** with USB cable.
N4689A	**USB 4-port hub** - for connecting additional USB peripherals.
82357A	**USB/GPIB Interface** - for controlling GPIB devices through USB. Learn more about using the 82357A with the PNA

Connector Care and Cleaning Supplies

Part Number	Description
9301-1243	Lint-Free Swabs (small), 100 ct.
8500-5344	IPA 99.5% alcohol, 30 ml. bottle
8500-6659	Compressed Air, 235 ml. can
5021-7607	Type-N Contact Removal Tool
1401-0225	Standard End-Cap, Type-N (m)
1401-0248	ESD Safe End-Cap, Type-N (m)
1401-0225	Standard End-Cap, Type-N (f)
1401-0247	ESD Safe End-Cap, Type-N (f)

ESD Supplies

Part Number	Description
9300-1367	Adjustable antistatic wrist strap
9300-0980	Antistatic wrist strap grounding cord (5 foot)
9300-0797	Static control table mat (2 foot x 4 foot) with earth ground wire
9300-1126	ESD heel strap

82357A USB / GPIB Interface

The Agilent 82357A is an adapter that creates a GPIB Interface from one of your unused PNA USB ports.

- **Applications**
- **Installing**
- **Configuring**
- **Connecting**
- **Communicating with other Equipment**

Applications

The 82357A can be used for the following PNA applications:

- **Frequency Converter Application** - The 82357A is included with the Frequency Converter Application (option 083). External sources MUST be connected to this Interface if controlling the PNA using an external PC. See connecting diagram below. To learn more, see Configure an external LO source.

Note: If the PNA is hibernated during an FCA measurement involving an external source under FCA control, and then the PNA is restarted, a VISA error message will appear stating "VI_ERROR_INV_OBJECT." To correct this problem, the 82357 USB/GPIB interface must be reinitialized after hibernation. This is done by clicking on the Accept button in the interface initialization dialog box. The green READY light on the interface will illuminate.

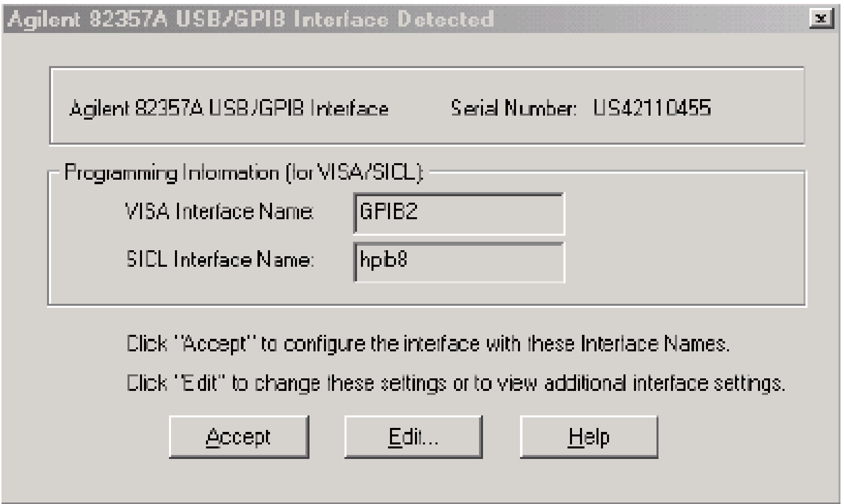
- **PNA Controller -** The 82357A can be used by the PNA to control other GPIB devices. This frees the default GPIB interface to perform other GPIB operations, such as control the PNA from an external PC.
- **Source Power Cal** - The 82357A can be used to run a source power calibration.

Installing the 82357A USB/GPIB Interface

1. Download and install firmware PNA revision 3.0 or greater. To check the revision of your PNA firmware, click **Help** then **About Network Analyzer.**
2. Upgrade to the latest Agilent IO libraries from the CDROM that was shipped with the 82357A. If not available, download them from www.Agilent.com (search for **82357A)**

Configure the 82357A USB/GPIB Interface

When the 82357A is connected to the PNA USB, the following dialog box appears:



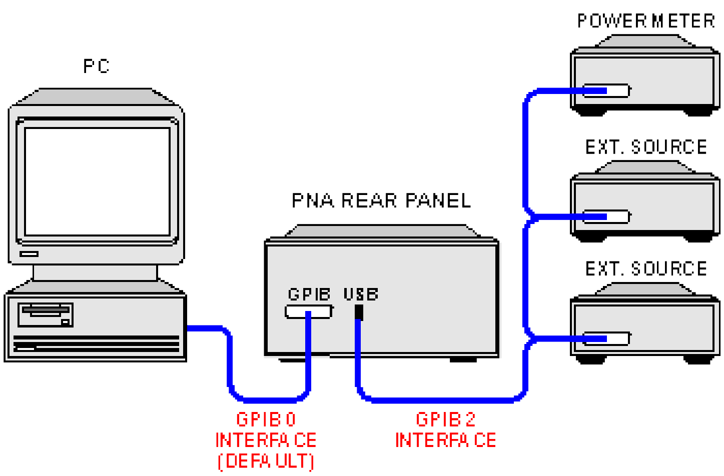
Normally, you do NOT need to edit these settings. The 82357A USB/GPIB Interface is configured automatically as the next unused VISA interface. This is usually **GPIB2** unless you have already configured it for another purpose.

If the VISA Interface Name appears as GPIB0 or GPIB1, these Interfaces must be returned to their default settings for the 82357A to work properly with the PNA. See Configure for VISA / SICL to learn how.

Connecting the 82357A USB/GPIB Interface

The following diagram illustrates how to connect GPIB test equipment using the USB/GPIB Interface.

- Plug the USB/GPIB Interface into any unused PNA USB port.
- The default GPIB Interface and USB/GPIB Interface should never be connected together.



Communicating with Equipment Connected to the USB/GPIB Interface

- The Frequency Converter Application will automatically find and communicate with test equipment that is connected to the USB/GPIB Interface.
- Source power calibration will automatically find and communicate with the power meter that is connected to the USB/GPIB Interface.
- To control other devices through your own program using the 82357A, you must include the new GPIB Interface number when addressing the devices.

Firmware Upgrade

PNA firmware upgrades are available to you at no cost in a self-extracting Install Shield file. The upgrade includes the PNA application, Online help, and Service utilities. Note: The file is at least 45 MB.

The following options are available for you to upgrade your PNA application:

- Auto-Check and AgileUpdate If your PNA is connected to the Internet, these utilities will automatically check for, download, and install, the new firmware and associated files when the PNA application is started. You will be prompted before this occurs.
- Website Access If your PNA is NOT connected to the Internet, but you have a PC that is, you can download the PNA firmware and associated files to a storage medium.

See Also

PNA Cal Kits and Firmware Upgrades

Other Support Topics

Auto-Check

With Internet access to your PNA, Auto-Check automatically and regularly checks the Internet for new PNA firmware revisions. If a new revision is found, a notification message prompts you to run the AgileUpdate utility, which then performs the actual download.

Without Internet access to your PNA, Auto-Check provides a reminder prompt at the selected intervals.

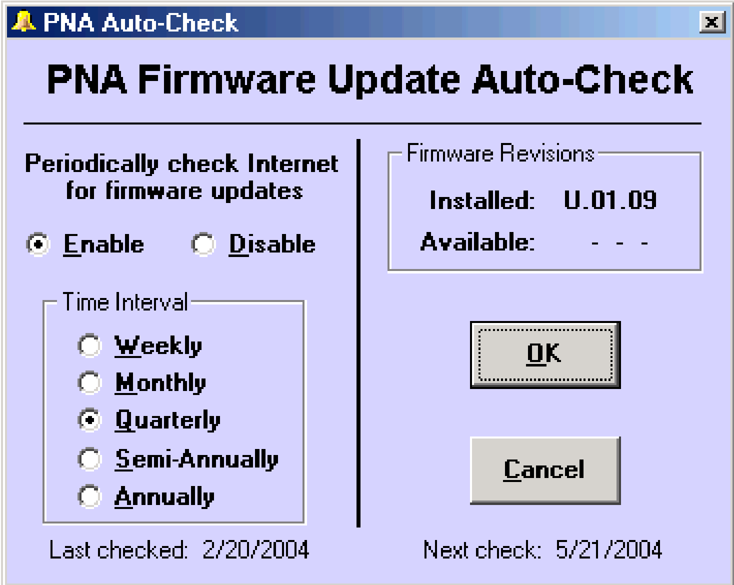
Auto-Check is run only when the PNA application is started. Once the PNA application is running, it will not check for updates again until it is restarted.

When Auto-Check runs, it checks the following conditions:

- Is there an active connection to the Internet?
- Is the Auto-Check utility enabled?
- Is it time to check for new firmware?
- Does new firmware exist?

If all of these conditions are true, Auto-Check shows the following dialog box.

If all of these conditions are NOT true, or to change these settings at any time, click **System, Service,** then **AgileUpdate**. From within AgileUpdate, click **AutoCheck**. These preferences are stored in the PNA registry. Future firmware upgrades will not change these settings.



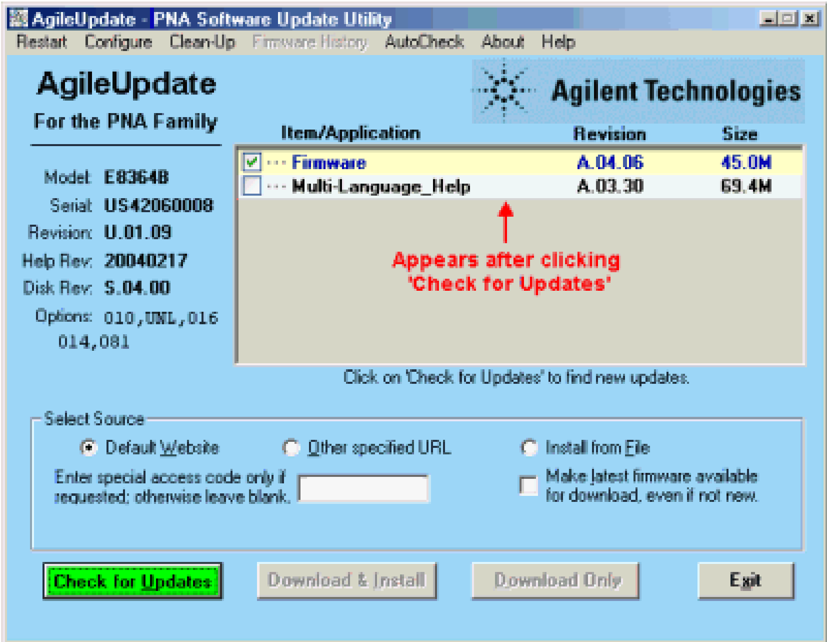
PNA Auto-Check dialog box help
Enable When the PNA application is started, Auto-Check will search the PNA website for firmware updates at the selected time interval. **Disable** When the PNA application is started, Auto-Check will NOT search the PNA website for firmware updates. **Time Interval** Select the time interval Auto-Check is to search for firmware updates. **OK** Starts AgileUpdate. **Cancel** No further action is taken until the selected time interval has elapsed.

AgileUpdate

Note: You must have administrative privileges on the analyzer to run this utility. See Set Up Analyzer Users.

In most cases, the following steps are sufficient to download and install the newest version of PNA firmware:

1. Connect the PNA to the Internet. A LAN connection is recommended because a firmware download can take many hours using a modem.
2. In the **System** menu, click **Service**, then **AgileUpdate**.
3. Click **Check for Updates**.
4. If updates exist, click **Download & Install**.



AgileUpdate dialog box help

Note: Your privacy is important to Agilent. AgileUpdate does NOT send ANY information from the PNA to the server. It only downloads from the server to the PNA.

Restart Click to restart from the beginning.

Configure Click to launch the Configure dialog box.

Clean-up Click to delete all but the two most recent install shield packages from the PNA hard drive.

Firmware History Available after clicking **Check for Updates**.

Auto-Check Launches the Auto-Check dialog box.

Item / Application Lists the items available for download at the firmware website.

- Click on items with **i** to read more information about the download.
- Items in RED should be downloaded and installed individually.
- Multi-language help includes all help files except English.

Note: The firmware includes the help file. Therefore, only the firmware checkbox will be selected if a new version for both the firmware and the help file are available.

Select Source

Default Website The Agilent site that contains upgrade FW.

Other Specified URL Click if you were instructed to get firmware from a different website.

Install from File Click if you have already downloaded the InstallShield package and want AgileUpdate to install it for you.

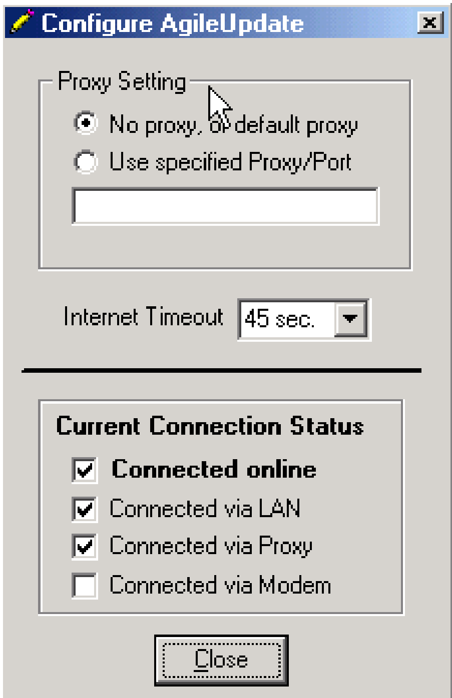
Special Access Code... Type in the code if you were given one from Agilent Technical Support. Otherwise, leave blank.

Make Latest Firmware Available... Select this checkbox if you want to download the latest firmware, even if it is not new.

Check for Updates Click to look for firmware updates at the Agilent website. If there are newer versions, the files will be listed.

Download and Install When updates are found, this selection becomes available. Some files may be pre-checked. Be sure the corresponding boxes are checked for the files you want to download. Then click to download and install the update.

Download Only Click to download the files to the analyzer hard disk and install the files at a later time. At that time, click **Install from File.**



Configuration dialog box help

Note: If AgileUpdate will not connect, try to access ANY Internet website. Contact your local IT department if necessary.

Proxy Setting

No Proxy or Default Proxy Click if you use a LAN connection. AgileUpdate will automatically use the proxy specified in Internet Explorer.

Use specified Proxy / Port Click to enter the proxy name and port. The format is: proxyName:portNumber. (The proxy port number is typically 8088).

Internet timeout If you are using an automatic dial-up Internet connection you may need to increase the timeout.

Current Connection Status Shows the current status of the PNA connection to the Internet.

Note: These settings are NOT saved; they must be re-entered each time AgileUpdate is run.

Agilent Website Access

If you cannot access the Internet directly with your analyzer, you can use an external PC with Internet access to download the file from the Agilent website. You can then transfer the file from your PC to your analyzer over a LAN or other means.

1. Connect to the PNA web page at http://www.agilent.com/find/PNA.
2. Follow the links, or search for the "PNA firmware" download page.
3. Click on the firmware to be downloaded.
4. Save the program to disk (hard drive of your PC).
5. Transfer the file from your PC to your PNA using LAN, CD, or USB Pen drive.
6. Double-click the file on the PNA.

Warning: You can save the upgrade file to your PC, but do not attempt to install the PNA application on your PC. It will alter system settings and can result in system crashes.

PNA Configurations and Options

Included with each PNA is a mouse, keyboard. This topic presents standard PNA models and the available options and upgrades.

- PNA Models
  - PNA L Series
  - Microwave Models
  - RF Models
  - mmWave Model
- Options and Upgrade Kits
- Warranty Period

To view the options that are installed on your analyzer, click **Help** then **About Network Analyzer**

A documentation CD-ROM is no longer included in each PNA shipment (Feb.05).

Other Support Topics

PNA-L Series   Model N5230A

Click the **Option number** to see the block diagram.

Click the **Connector type** to see the connector specifications.

Option	Frequency Range	Test Ports	Connector	Reference Receivers	Test Set[1]
020	300 KHz* to 6 GHz	2	3.5 mm Male	2	Standard
025	300 KHz* to 6 GHz	2	3.5 mm Male	2	Configurable
120	300 KHz* to 13 GHz	2	3.5 mm Male	2	Standard
125	300 KHz* to 13 GHz	2	3.5 mm Male	2	Configurable
220	10 MHz* to 20 GHz	2	3.5 mm Male	2	Standard
225	10 MHz* to 20 GHz	2	3.5 mm Male	2	Configurable
240	300 KHz* to 20 GHz	4	3.5 mm Male	1	Standard
245	300 KHz* to 20 GHz	4	3.5 mm Male	1	Configurable[2]
420	10 MHz* to 40 GHz	2	2.4 mm Male	2	Standard
425	10 MHz* to 40 GHz	2	2.4 mm Male	2	Configurable

520	10 MHz* to 50 GHz	**2**	2.4 mm Male	**2**	**Standard**
525	10 MHz* to 50 GHz	**2**	2.4 mm Male	**2**	**Configurable**

* Typical specs apply

1 **Test Set**

Standard NO Configurable Test Set.

Configurable - Adds six front panel access loops and two 50 dB step attenuators.

2 **Configurable** - Adds nine front panel access loops and one 60 db step attenuator.

Microwave Standard Models (see options)

Click the **PNA model** to see the block diagram.

Click the **Connector type** to see the connector specifications.

PNA Model	Frequency Range	Port	Connector Type	Reference Receivers	Front Panel Jumpers
E8361A	10 MHz* to 67 GHz (tunable to 70 GHz)**	**2**	1.85 mm Male	**2**	**0**
E8362A	45 MHz to 20 GHz	**2**	3.5 mm Male	**2**	**0**
E8363A	45 MHz to 40 GHz	**2**	2.4 mm Male	**2**	**0**
E8364A	45 MHz to 50 GHz	**2**	2.4 mm Male	**2**	**0**
E8362B	10 MHz* to 20 GHz	**2**	3.5 mm Male	**2**	**0**
E8363B	10 MHz* to 40 GHz	**2**	2.4 mm Male	**2**	**0**
E8364B	10 MHz* to 50 GHz	**2**	2.4 mm Male	**2**	**0**

Legend

No longer produced

* Typical specs apply from 10 to 45 MHz

** Typical specs apply from 67 to 70 GHz

RF Standard Models (see options)

Click the **PNA model** to see the block diagram.

Click the **Connector type** to see the connector specifications.

PNA Model	Frequency Range	Ports	Connector Type	Reference Receivers	Front Panel Jumpers
E8356A	300 kHz to 3 GHz	**2**	Type-N Female	**2**	**4**
E8357A	300 kHz to 6 GHz	**2**	Type-N Female	**2**	**4**
E8358A	300 kHz to 9 GHz	**2**	Type-N Female	**2**	**4**
E8801A	300 kHz to 3 GHz	**2**	Type-N Female	**1**	**0**
E8802A	300 kHz to 6 GHz	**2**	Type-N Female	**1**	**0**
E8803A	300 kHz to 9 GHz	**2**	Type-N Female	**1**	**0**
N3381A	300 kHz to 3 GHz	**3**	Type-N Female	**1**	**0**
N3382A	300 kHz to 6 GHz	**3**	Type-N Female	**1**	**0**
N3383A	300 kHz to 9 GHz	**3**	Type-N Female	**1**	**0**

Millimeter Wave PNA

PNA Model	Frequency Range	Ports	Connector Type	Reference Receivers	Front Panel jumpers
N5250A	10 MHz* to 110 GHz	**2**	1.0 mm	**2**	**N5260A**

* Typical specs apply from 10 to 45 MHz

Test heads to 325 GHz are also available

Upgrade your existing E836xB with the following:

- H11 option
- N5260A Test Set
- Millimeter-Wave VNA Frequency Extension Modules from Oleson Microwave Labs Extension Modules

Options and Upgrade Kits

The following options are installed at the time of purchase, and some are also available after the initial purchase of a PNA. To order an upgrade, contact your Agilent representative.

Option	Supported Models	Description

006	E8356A E8801A N3381A	**Upgrade to 6 GHz frequency range.** Adds 6 GHz operation. Includes installation at an Agilent service center. Instrument calibration (required after frequency upgrade) is available for an additional fee.
009	E8356/7A E8801/2A N3381/2A	**Upgrade to 9 GHz frequency range.** Adds 9 GHz operation. Includes installation at an Agilent service center. Instrument calibration (required after frequency upgrade) is available for an additional fee.
010	All	**Time-domain** Adds time-domain capability to analyzer. The serial number of the analyzer must be specified when ordering this kit. Software upgrade. Learn more
014	E8801/ 02/ 03A N3381/ 82/ 83A E8361A E8362/ 63/ 64A E8362/ 63/ 64B	**Configurable test set** Adds front panel access to the source output and coupler input on test ports 1 and 2. Adds front panel access directly to all receivers, including the reference receiver. Upgrade includes installation at an Agilent service center. See the PNA block diagram.
015	E8356/ 57/ 58A	**Configurable test set** Adds front panel access to the source output and coupler input on test ports 1 and 2. Adds 35 dB step attenuators between the couplers and receivers. Upgrade includes installation at an Agilent service center. See the PNA block diagram
016	E8361A E8362/ 63/ 64B	**Receiver Step Attenuators** Adds two step attenuators. (E8361A adds 50 dB step attenuators with 10 dB resolution; all other PNA models add 35 dB step attenuators with 5 dB resolution.) Each attenuator is inserted between a test port and its corresponding receiver. Requires option UNL.
022	E8361A E8362/ 63/ 64B	**Extended Memory** Adds more RAM for a total of 512MB. Learn more about PNA RAM
080	E8361A E8362/ 63/ 64B N5230A	**Frequency Offset** Enables you to set the PNA source independently from where the receivers are tuned. This capability is important for measuring mixers, converters, and amplifiers. To control the additional hardware, a basic user interface, SCPI, and COM commands are provided. Requires option 014 or Active test set (N5230A).

081	E8361A E8362/ 63/ 64B	**Reference Switch** Adds an internal solid state switch in the R1 reference path for controlling an external reference mixer. Requires option 014 and opt 080 See a block diagram of reference switch.
083	E8361A E8362B/ 63B/ 64B	**Frequency Converter Measurement Application** Provides an intuitive user-interface for making extremely accurate conversion loss and absolute group delay measurements on mixers and converters. Exceptional amplitude and phase accuracy is achieved through two new calibration techniques: match-corrected power meter calibration and vector mixer calibration. The application also provides automatic control of all of Agilent's major signal sources. Requires option 014, 080, and 081. Learn more
098	E8356/ 57/ 58A	**CPU board upgrade** Replaces the 266 MHz CPU board with a 500 MHz CPU board. Upgrade includes installation at an Agilent service center. Learn how to check your PNA CPU speed.
1E1	E8801/ 02/ 03A N3381/ 82/ 83A	**70 dB step attenuator** Adds a single 70 dB step attenuator that is switched between the source and each output port to extend the output power to -85 dBm. **Note**: Two 70 dB step attenuators are standard equipment with PNA models E8356/ 57/ 58A. Upgrade includes installation at an Agilent service center.
1E5	E8801/ 02/ 03A N3381/ 82/ 83A	**High stability 10 MHz time base** Replaces 10 ppm time base with a 1 ppm time base. High stability time base is standard with PNA models E8356/ 57/ 58A and E8362/ 63/ 64B.
H08	E8361A E8362B/ 63B/ 64B	**Pulsed Measurement Application** Provides the Pulsed Application for average pulse and point-in-pulse measurements. Requires options H11, UNL, 014, 080, and 081. Learn more
H11	E8361A E8362B/ 63B/ 64B	**IF Access** Provides rear panel access to the PNA IF paths for: • Extended frequency coverage to 325 GHz • Pulsed measurement capability • Advanced antenna measurements Requires options UNL, 014, 080, and 081. Learn more

UNL	E8361A E8362/ 63/ 64A E8362/ 63/ 64B	**Extended power range and bias tees** Adds two step attenuators and two bias tees. (E8361A adds 50 dB step attenuators; E8362/ 63/ 64A/B adds 60 dB step attenuators. All attenuators have a 10 dB resolution.) A step attenuator and bias-tee set is inserted between the source and test port 1 and another set between the source and test port 2. Upgrade includes installation at an Agilent service center.

Certification Options

Option	Supported Models	Description
1A7	E8361A E8362B/ 63B/ 64B N5250A	Complete set of measurement data which was acquired from testing your PNA to published specifications. Includes calibration label, calibration certificate, and data report. Conforms to ISO 9001.
UK6	E8361A E8362B/ 63B/ 64B N5250A	Complete set of measurement data which was acquired from testing your PNA to published specifications. Includes calibration label, ISO 17025 calibration certificate, data report, measurement uncertainties and guard bands on all specifications. Conforms to ISO17025 and ISO 9001.

Documentation and Localization Options

Printed versions of PNA Help in .pdf format are available at Agilent.com. (May.2005
A documentation CD-ROM is no longer included with each PNA shipment (Feb.2005).
To download a service guide for your PNA, or the latest version of PNA Help, visit http://www.agilent.com, search for your PNA model, then click Library.

PNA Warranty Period

The actual warranty on your instrument depends on the date it was ordered as well as whether or not any warranty options were purchased at that time. To determine the exact warranty on your instrument, contact Agilent Technologies with the model and serial number of your instrument.

For online information about Agilent's service and support products visit: www.agilent.com/find/tm_services

Option Enable Utility

The Option Enable utility allows you to perform the following activities.

- Enable or remove software options and some hardware options.
- Recover option data if the hard drive or other data-containing assembly is replaced.
- Input or change a serial number.

- Keywords
- Running the Program
- Removing an Option
- Installing an Option
- Repairing and Recovering Option Data
- Installing or Changing the Serial Number

Keywords

To add certain options, you need a keyword that is provided by Agilent. There are two types of keywords:

- **Option Keywords** to add a software option.
- **Model Keywords** that may be required if you replace multiple assemblies.

Temporary and Permanent Options

Any software option can also be installed on a temporary basis for a specified amount of time. This allows you to evaluate a specific feature or capability at no cost.

If the license key provided by Agilent has an expiration date, you must select the "temporary" option and enter the expiration date exactly as stated in the license statement. If you decide to make this option permanent, Agilent will provide a new keyword that converts the option to permanent status.

For either permanent or temporary software options, a provided keyword must be entered.

Running the Program

1. In the analyzer **System** menu, point to **Service**, and click **Option Enable**.
2. The first screen will display the model number, serial number, and all installed options. To enable or remove an option, select it from the drop-down list of available options.
   - If the desired option is not listed, select the last choice in the list, labeled **Enter Unlisted Option**. Next, enter the 3-character option name and click **Enter**

If a software option was chosen, the following occurs.

- The **Remove** button will be enabled.

- The keyword entry area becomes visible.
- The permanent/temporary selection is enabled.

If a hardware option is selected, the following occurs.

- With the hardware option already installed, the **Remove** button is enabled.
- With the hardware option not installed, the **Enable** button is enabled.

Removing an Option

1. To remove an option, click **Remove**.
2. After the option is removed, restart the network analyzer application for the changes to take effect.

Note: Removal of a licensed option (such as Option 010, Time Domain) will permanently remove the license keyword. If this option **may** be needed in the future, then record the license keyword before removing the option. Do this by copying the file "gen.lic" to another location (such as a floppy disk), or print it using notepad. The file, located at "C:\Program Files\Agilent\Network Analyzer" contains all the information needed to recreate the license.

Installing an Option

1. If the keyword entry area is visible, enter a keyword. (The keyword is not case sensitive.)
2. Click **Enable**.
3. After the option is installed, restart the network analyzer application for the changes to take effect.

Repairing and Recovering Option Data

Use this part of the Option Enable Utility in the following situations:

- If the hard drive is replaced
- If the frequency reference assembly is replaced

This routine rebuilds the option information contained on the hard drive and frequency reference assembly (primary and backup).

1. Select **Repair** from the **Option Enable** menu bar.

Note: If you are unsure if this routine needs to be done, run it; no harm will result.

---

2. The model and serial number are displayed, along with four check boxes.
3. Select the boxes that apply.
4. Click **Begin Repair**. The routine checks all data files and performs any needed repairs. You may be asked to verify certain information and processes.
5. If the routine finds that the model number is incorrect or invalid, you will be asked to select the correct model number.

- Along with this model number, a model keyword will be required. If this is not labeled on the analyzer, or is not otherwise known, contact Agilent
- After you have entered the requested data, click **Change Model**. This process takes about 30 seconds.

6. When done, click **Exit Repair**.

7. If you do not need to install any other options, click **Exit**.

Installing or Changing the Serial Number

It may be necessary to install or change a serial number if certain assemblies are replaced.

1. To change the serial number, select **Change Serial** from the **Option Enable** menu bar. The current serial number will be displayed. If no serial number has previously been entered, the word "NONE" will be displayed.

2. Type the new serial number into the space provided, and click **Change Serial**. (The serial number is not case sensitive.)

Note: Use extreme care when entering the serial number; only one entry chance is allowed!

---

3. To change an incorrect serial number, a clear-code password is required. Contact Agilent to obtain this clear code and have the existing serial number available. Enter the clear code in the space provided, along with the new serial number, then click **Change Serial**.

Other Resources

The following network analysis resources are also available.

Document Resources

Third Party Resources

Other Support Topics

Document Resources

Application Notes

You can access a number of the following application notes from the analyzer.

1. In the **Help** menu, click **Product Overview.**
2. The Product Overview starts with an introduction. To immediately move through this, click **Main Menu.**
3. In the main menu click **Literature**, and then any literature title to view the document.

You can also access all of the following application and product notes at this URL:

http://www.agilent.com/find/PNA

- Application Development with the Agilent PNA Series of Network Analyzers, (literature number 5980-2666ENUS).
- Understanding and Improving Network Analyzer Dynamic Range*,* Agilent Application Note 1363-1 (literature number 5980-2778EN).
- *The "*Need for Speed*"* in Component Manufacturing Test, (literature number 5980-2783EN).
- Generate Component Data Sheets with Agilent's BenchLinkXL, (literature number 5980-2781EN).
- Connectivity Advances in a LAN-enabled Instument, (literature number 5980-2782EN).
- De-embedding and Embedding S-Parameter Networks Using the PNA Series Network Analyzer, Agilent Application Note 1364-1 (literature number 5980-2784EN).
- Advanced Filter Tuning Using TIme Domain, Agilent Application Note 1287-10 (literature number 5980-2785EN).
- Understanding the Fundamental Principles of Vector Network Analysis, Agilent Application Note 1287-1 (literature number 5965-7707E).
- Exploring the Architectures of Network Analyzers, Agilent Application Note 1287-2 (literature number 5965-7708E).
- Applying Error Correction to Network Analyzer Measurements, Agilent Application Note 1287-3 (literature number 5965-7709E).
- Network Analyzer Measurements: Filter and Amplifier Examples, Agilent Application Note 1287-4 (Agilent literature number 5965-7710E).

- Improving Throughput in Network Analyzer Applications, Agilent Application Note 1287-5 (literature number 5966-3317E)
- Using a Network Analyzer to Characterize High-Power Components, Agilent Application Note 1287-6 (literature number 5966-3319E)
- Simplified Filter Tuning Using Time-Domain Analysis, Agilent Application Note 1287-8 (literature number 5968-5328E).
- In-Fixture Measurements Using Vector Network Analysis, Agilent Application Note 1287-9 (literature number 5968-5329E).
- 8 Hints for Making Better Network Analyzer Measurements, Agilent Application Note AN 1291-1 (literature number 5965-8166E)
- Specifying Calibration Standards for the Agilent 8510 Network Analyzer, Agilent Product Note 8510-5A, (literature number 5956-4352, or Agilent part number 08510-90352, February 1988).
- In-Fixture Microstrip Device Measurements Using TRL* Calibration, Agilent Product Note 8720-2, (literature number 5091-1943E, August 1991).

Third-Party Resources

For information about test fixtures and part handlers, contact:

Inter-Continental Microwave
1515 Wyatt Drive
Santa Clara, CA 95054-1524 USA
Telephone: (408) 727-1596
Fax: (408) 727-0105
Web site: www.icmicrowave.com
E-mail: icmfixture@aol.com

For information about probing equipment and accessories, contact:

Cascade Microtech, Inc.
2430 NW 206th Avenue
Beaverton, OR 97006 USA
Telephone: (503) 601-1000
Fax: (503) 601-1002
Web site: www.cascademicrotech.com
E-mail: sales@cmicro.com

SCPI Errors

Standard SCPI Errors

- -100 to -200 Command Errors
- -200 to -299 Execution Errors
- -300 to -399 SCPI Specified Device-Specific Errors
- -400 to -800 Query and System Errors

PNA specific Errors

Note: See also PNA Errors

-100 to -200 Command Errors

A command error indicates that the test set's GPIB parser has detected an IEEE 488.2 syntax error. When one of these errors is generated, the command error bit in the event status register is set.

-100	std_command	Command - This event bit (Bit 5) indicates a syntax error, or a semantic error, or a GET command was entered, see IEEE 488.2, 11.5.1.1.4.
-101	std_invalidChar	Invalid character - Indicates a syntactic elements contains a character which is invalid for that type.
-102	std_syntax	Syntax - Indicates that an unrecognized command or data type was encountered. For example, a string was received when the device does not accept strings.
-103	std_invalidSeparator	Invalid separator - The parser was expecting a separator and encountered an illegal character. For example, the semicolon was omitted after a program message unit.
-104	std_wrongParamType	Data type -The parser recognized a data element different than one allowed. For example, numeric or string data was expected but block data was encountered.
-105	std_GETNotAllowed	GET not allowed - Indicates a Group Execute Trigger was received within a program message. Correct the program so that the GET does not occur within the program code.
-108	std_tooManyParameters	Parameter not allowed - Indicates that more parameters were received than expected for the header. For example, *ESE common command only accepts one parameter, so *ESE 0,1 is not allowed.
-109	std_tooFewParameters	Missing parameter - Indicates that less parameters were received than required for the header. For example, *ESE requires one parameter, *ESE is not allowed.
-110	std_cmdHeader	Command header - Indicates an error was detected in the header. This error is used when the device cannot detect the more specific errors -111 through -119.
-111	std_headerSeparator	Header separator - Indicates that a character that is not a legal header separator was encountered while parsing the header.

-112	std_IDTooLong	Program mnemonic too long - Indicates that the header contains more that twelve characters, see IEEE 488.2, 7.6.1.4.1.
-113	std_undefinedHeader	Undefined header - Indicates the header is syntactically correct, but it is undefined for this specific device. For example, *XYZ is not defined for any device.
-114	std_suffixOutOfRange	Header suffix out of range - Indicates the value of a header suffix attached to a program mnemonic makes the header invalid.
-120	std_numericData	Numeric data - This error, as well as errors
-121	std_invalidCharInNumber	Invalid character in number - Indicates an invalid character for the data type being parsed was encountered. For example, an alpha in a decimal numeric or a "9" in octal data.
-123	std_exponentTooLarge	Exponent too large - Indicates the magnitude of an exponent was greater than 32000, see IEEE 488.2, 7.7.2.4.1.
-124	std_decimalTooLong	Too many digits - Indicates the mantissa of a decimal numeric data element contained more than 255 digits excluding leading zeros, see IEEE 488.2, 7.7.2.4.1.
-128	std_numericNotAllowed	Numeric data not allowed - Indicates that a legal numeric data element was received, but the device does not accept one in this position for the header.
-130	std_suffix	Suffix - This error, as well as errors -131 through -139, are generated when parsing a suffix. This particular error message is used if the device cannot detect a more specific error.
-131	std_badSuffix	Invalid suffix - Indicates the suffix does not follow the syntax described in IEEE 488.2, 7.7.3.2, or the suffix is inappropriate for this device.
-134	std_suffixTooLong	Suffix too long - Indicates the suffix contain more than 12 characters, see IEEE 488.2, 7.7.3.4.
-138	std_suffixNotAllowed	Suffix not allowed - Indicates that a suffix was encountered after a numeric element that does not allow suffixes.
-140	std_charData	Character data - This error, as well as errors
-141	std_invalidCharData	Invalid character data - Indicates that the character data element contains an invalid character or the particular element received is not valid for the header.
-144	std_charDataTooLong	Character data too long - Indicates the character data element contains more than twelve characters, see IEEE 488.2, 7.7.1.4.
-148	std_charNotAllowed	Character data not allowed - Indicates a legal character data element was encountered where prohibited by the device.
-150	std_stringData	String data - This error, as well as errors
-151	std_stringInvalid	Invalid string data - Indicates that a string data element was expected, but was invalid, see IEEE 488.2, 7.7.5.2. For example, an END message was received before the terminal quote character.

-158	std_stringNotAllowed	String data not allowed - Indicates that a string data element was encountered but was not allowed by the device at this point in parsing.
-160	std_blockData	Block data - This error, as well as errors -161 through -169, are generated when parsing a block data element. This particular error message is used if the device cannot detect a more specific error.
-161	std_badBlock	Invalid block data - Indicates a block data element was expected, but was invalid, see IEEE 488.2, 7.7.6.2. For example, and END message was received before the end length was satisfied.
-168	std_blockNotAllowed	Block data not allowed - Indicates a legal block data element was encountered, but not allowed by the device at this point in parsing.
-170	std_expr	Expression - This error, as well as errors -171 through -179, are generated when parsing an expression data element. This particular error message is used if the device cannot detect a more specific error.
-171	std_invalidExpression	Invalid expression - Indicates the expression data element was invalid, see IEEE 488.2, 7.7.7.2. For example, unmatched parentheses or an illegal character.
-178	std_exprNotAllowed	Expression data not allowed - Indicates a legal expression data was encountered, but was not allowed by the device at this point in parsing.
-180	std_macro	Macro - This error, as well as error -181 through -189, are generated when defining a macro or execution a macro. This particular error message is used if the device cannot detect a more specific error.
-181	std_validOnlyInsideMacro	Invalid outside macro definition - Indicates that a macro parameter place holder was encountered outside of a macro definition.
-183	std_invalidWithinMacro	Invalid inside macro definition - Indicates that the program message unit sequence, sent with a *DDT or a *DMC command, is syntactically invalid, see IEEE 488.2, 10.7.6.3.
-184	std_macroParm	Macro parameter - Indicates that a command inside the macro definition had the wrong number or type of parameters.

-200 to -299 Execution Errors

These errors are generated when something occurs that is incorrect in the current state of the instrument. These errors may be generated by a user action from either the remote or the manual user interface

-200	std_execGen	Execution - This event bit (Bit 4) indicates a PROGRAM DATA element following a header was outside the legal input range or otherwise inconsistent with the device's capabilities, see IEEE 488.2, 11.5.1.1.5.
-201	std_invalidWhileInLocal	Invalid while in local
-202	std_settingsLost	Settings lost due to rtl
-203	std_commandProtected	Command protected - Indicates that a legal password-protected program command or query could not be executed because the command was disabled.
-210	std_trigger	Trigger

-211	std_triggerIgnored	Trigger ignored
-212	std_armIgnored	Arm ignored
-213	std_initIgnored	Init ignored
-214	std_triggerDeadlock	Trigger deadlock
-215	std_armDeadlock	Arm deadlock
-220	std_parm	Parameter - Indicates that a program data element related error occurred.
-221	std_settingsConflict	Settings conflict - Indicates that a legal program data element was parsed but could not be executed due to the current device state.
-222	std_dataOutOfRange	Data out of range - Indicates that a legal program data element was parsed but could not be executed because the interpreted value was outside the legal range defined by the devices
-223	std_tooMuchData	Too much data - Indicates that a legal program data element of block, expression, or string type was received that contained more data than the device could handle due to memory or related device-specific requirements.
-224	std_illegalParmValue	Illegal parameter value - Indicates that the value selected was not part of the list of values given.
-225	std_noMemoryForOp	Out of memory - The device has insufficient memory to perform the requested operation.
-226	std_listLength	Lists not same length - Attempted to use LIST structure having individual LIST's of unequal lengths.
-230	std_dataCorruptOrStale	Data corrupt or stale - Indicates invalid data, a new reading started but not completed since the last access.
-231	std_dataQuestionable	Data questionable - Indicates that measurement accuracy is suspect.
-232	std_invalidFormat	Invalid format
-233	std_invalidVersion	Invalid version - Indicates that a legal program data element was parsed but could not be executed because the version of the data is incorrect to the device. For example, a not supported file version, a not supported instrument version.
-240	std_hardware	Hardware - Indicates that a legal program command or query could not be executed because of a hardware problem in the device.
-241	std_hardwareMissing	Hardware missing - Indicates that a legal program command or query could not be executed because of missing device hardware. For example, an option was not installed.
-250	std_massStorage	Mass storage - Indicates that a mass storage error occurred. The device cannot detect the more specific errors described for errors -251 through -259.

-251	std_missingMassStorage	Missing mass storage - Indicates that a legal program command or query could not be executed because of missing mass storage.
-252	std_missingMedia	Missing media - Indicates that a legal program command or query could not be executed because of missing media. For example, no disk.
-253	std_corruptMedia	Corrupt media - Indicates that a legal program command or query could not be executed because of corrupt media. For example, bad disk or wrong format.
-254	std_mediaFull	Media full- Indicates that a legal program command or query could not be executed because the media is full. For example, there is no room left on the disk.
-255	std_directoryFull	Directory full - Indicates that a legal program command or query could not be executed because the media directory was full.
-256	std_fileNotFound	File name not found - Indicates that a legal program command or query could not be executed because the file name was not found on the media.
-257	std_fileName	File name - Indicates that a legal program command or query could not be executed because the file name on the device media was in error. For example, an attempt was made to read or copy a nonexistent file.
-258	std_mediaProtected	Media protected - Indicates that a legal program command or query could not be executed becuse the media was protected. For example, the write-protect switch on a memory card was set.
-260	std_expression	Expression
-261	std_math	Math in expression
-270	std_macroExecution	Macro - Indicates that a macro related execution error occurred.
-271	std_macroSyntax	Macro syntax - Indicates that a syntactically legal macro program data sequence, according to IEEE 488.2, 10.7.2, could not be executed due to a syntax error within the macro definition.
-272	std_macroExec	Macro execution - Indicates that a syntactically legal macro program data sequence could not be executed due to some error in the macro definition, see IEEE 488.2, 10.7.6.3.
-273	std_badMacroName	Illegal macro label - Indicates that the macro label was not accepted, it did not agree with the definition in IEEE 488.2, 10.7.3
-274	std_macroPlaceholder Ma	cro parameter - Indicates that the macro definition improperly used a macro parameter placeholder, see IEEE 4882, 10.7.3.
-275	std_macroTooLong	Macro definition too long - Indicates that a syntactically legal macro program data sequence could not be executed because the string of block contents were too long for the device to handle, IEEE 488.2, 10.7.6.1.
-276	std_macroRecursion	Macro recursion - Indicates that a syntactically legal macro program data sequence count not be executed because it would be recursive, see IEEE 488.2, 10.7.6.6.

-277	std_cantRedefineMacro	Macro redefinition not allowed - Indicates that redefining an existing macro label, see IEEE 488.2, 10.7.6.4.
-278	std_macroNotFound	Macro header not found - Indicates that a legal macro label in the *GMS?, see IEEE 488.2, 10.13, could not be executed because the header was not previously defined.
-280	std_program	Program
-281	std_cantCreateProgram	Cannot create program
-282	std_illegalProgramName	Illegal program name
-283	std_illegalVarName	Illegal variable name
-284	std_programRunning	Program currently running
-285	std_programSyntax	Program syntax
-286	std_programRuntime	Program runtime
-290	std_memoryUse	Memory use
-291	std_execOutOfMemory	Out of memory
-292	std_nameNotFound	Referenced name does not exist
-293	std_nameAlreadyExists	Referenced name already exists
-294	std_incompatibleType	Incompatible type

-300 to -399 SCPI Specified Device-Specific Errors

A device-specific error indicates that the instrument has detected an error that occurred because some operations did not properly complete, possibly due to an abnormal hardware or firmware condition. For example, an attempt by the user to set an out of range value will generate a device specific error. When one of these errors is generated, the device specific error bit in the event status register is set.

-300	std_deviceSpecific	Device specific - This event bit (Bit 3) indicates that a device operation did not properly complete due to some condition, such as overrange see IEEE 488.2, 11.5.1.1.6.
-310	std_system	System
-311	std_memory	Memory - Indicates some physical fault in the devices memory, such as a parity error.
-312	std_PUDmemoryLost	PUD memory lost - Indicates protected user data saved by the *PUD command has been lost, see IEEE 488.2, 10.27.
-313	std_calMemoryLost	Calibration memory lost - Indicates that nonvolatile calibration data used by the *CAL? command has been lost, see IEEE 488.2, 10.2.
-314	std_savRclMemoryLost	Save/recall memory lost - Indicates that the nonvolatile data saved by the *SAV command has been lost, see IEEE 488.2, 10.33.
-315	std_configMemoryLost	Configuration memory lost - Indicates that nonvolatile configuration data saved by the device has been lost.
-320	std_storageFault	Storage fault - Indicates that the firmware detected a fault when using data storage. This is not an indication of physical damage or failure of any mass storage element.
-321	std_outOfMemory	Out of memory - An internal operation needed more memory than was available
-330	std_selfTestFailed	Self-test failed - Indicates a problem with the device that is not covered by a specific error message. The device may require service.
-340	std_calFailed	Calibration failed - Indicates a problem during calibration of the device that is not covered by a specific error.
-350	std_queueOverflow	Queue overflow - Indicates that there is no room in the queue and an error occurred but was not recorded. This code is entered into the queue in lieu of the code that caused the error.
-360	std_comm	Communication - This is the generic communication error for devices that cannot detect the more specific errors described for error -361 through -363.
-361	std_parity	Parity in program message - Parity bit not correct when data received for example, on a serial port.
-362	std_framing	Framing in program message - A stop bit was not detected when data was received for example, on a serial port (for example, a baud rate mismatch).
-363	std_inputBufferOverrun	Input buffer overrun - Software or hardware input buffer on serial port overflows with data caused by improper or nonexistent pacing.

-400 to -800 Query and System Errors

A Query error is generated either when data in the instrument's GPIB output queue has been lost, or when an attempt is being made to read data from the output queue when no output is present or pending.

-400	std_queryGen	Query - This event bit (Bit 2) indicates that an attempt to read data from the Output Queues when no output is present or pending, to data in the Output Queue has been lost see IEEE488.2, 11.5.1.1.7.
-410	std_interrupted	Query INTERRUPTED - Indicates the test set has been interrupted by a new program message before it finishes sending a RESPONSE MESSAGE see IEEE 488.2, 6.3.2.3.
-420	std_unterminated	Query UNTERMINATED - Indicates an incomplete Query in the program see IEEE 488.2, 6.3.2.2.
-430	std_deadlocked	Query DEADLOCKED - Indicates that the Input Buffer and Output Queue are full see IEEE 488.2, 6.3.1.7.
-440	std_responseNotAllowed	Query UNTERMINATED after indefinite response - Indicates that a query was received in the same program message after a query requesting an indefinite response was executed see IEEE 488.2, 6.5.7.5.
-500	std_powerOn	Power on
-600	std_userRequest	User request
-700	std_requestControl	Request control
-800	std_operationComplete	Operation complete

.

PNA Specific SCPI Errors

100	dupWindNum	"Duplicate window number"
101	windNumNotFound	"Window number not found"
102	failedWindCreate	"Window creation failed"
103	noCalcParamSelection	"CALC measurement selection set to none"
104	dupMeasName	"Duplicate measurement name"
105	dataNotFound	"Requested data not available"
106	measNotFound	"Requested measurement not found"
107	traceNotFound	"Requested trace not found"
108	notImplemented	"Mnemonic not yet implemented"
109	noDocument	"No measurement container found"
110	dupTraceNum	"Duplicate trace number"
111	titleStrTooLong	"Title string exceeds 50 characters"
112	memoryNotFound	"Requested memory not found"

113	exceedMaxTraces	"Exceeded the maximum number of traces per window"
114	SerNumNotFound	"The serial number was not found. Please store the serial number."
115	LoadFailed	"The state was not loaded. Please check the file name."
116	StoreFailed	"The state was not stored. Please check the file and path names."
117	File	"An in the File operation occurred. Please check file and path names."
118	measChanConflict	"Measurement does not belong to specified channel."
119	exceedMaxWindows	"Exceeded the maximum number of data windows"
120	markerNotFound	"The specified marker was not found."
121	diagnostic	"Diagnostic ."
122	channelNotFound	"The specified channel was not found."
123	exceedMaxMeasurements	"Exceeded the maximum number of allowed mesurements."
124	parameterOutOfRange	"The specified value was out of range."
125	userRangeNotValid	"The currently selected user range is not valid."
126	referenceMarkerNotFound	"The reference marker is not active."
127	sweepSegmentNotFound	"The sweep segment was not found."
128	markerNotDelta	"The specified marker is not a delta marker."
129	printoutFailed	"Attempt to output to a printer failed."
130	memory_trace_not_compatible	"Memory not compatible. Trace Math not applied."
131	trace_math_reset	"Memory not compatible. Trace Math turned off."
132	hw_read_failed	"Hardware read failed."
133	hw_write_failed	"Hardware write failed."
134	dsp_active	"Failed because DSP was not halted."
135	secure_memory	"Attempt to access secure memory region."
136	snum_protected	"The serial number is protected."
137	snum_format_bad	"The serial number format is bad."
138	snum_already_set	"The serial number is already set."
139	hw_setting_failed	"Hardware setting failed."
140	cal_access_failed	"Calibration data access failed."

141	db_access_failed	"Database access failed."
142	memory_range_exceeded	"Command exceeds usable memory range."
143	lost_phase_lock	"Phase lock has been lost."
144	over_power	"Detected too much power at input."
145	ee_wrt_failed	"EEPROM write failed."
146	yig_cal_failed	"YTO calibration failed."
147	ramp_cal_failed	"Analog ramp calibration failed."
148	dspcom_bad	"DSP communication failed."
149	no_license_found	"Request failed. The required license was not found."
150	argLimited	"The argument was out of range
151	markerBWNotFound	"The Marker Bandwidth was not found."
153	peakNotFound	"The Peak was not found."
154	targetNotFound	"The Target search value was not found."
155	calNotImpl	"The Calibration feature requested is not implemented."
156	calClassNotValidForCalType	"SENS:CORR:CCH measurement selection set to none"
158	calNotValidForConfidenceChe	"Selected measurement does not have a calibration valid for Confidence Check"
159	invalidPort	"Specified port is out of range"
160	invalidPortPath	"ROUT:PATH:DEF:PORT x, y does not match measurement; setting to defaults"
161	ioInvalidWrite	"Attempted I/O write while port set to read only."
162	ioInvalidRead	"Attempted I/O read from write only port."
163	calsetNotFound	"Requested Cal Set was not found in Cal Set Storage."
164	noCalSetSelected	"There is no Cal Set currently selected for the specified channel."
165	cantDeleteCalSetInUse	"Cannot delete a Cal Set while it is being used."
166	calsetStimChange	"Channel stimulus settings changed to match selected Cal Set."
167	exceedMaxCalSets	"Exceeded the maximum number of cal sets."
168	calCouldNotTurnOn	"A valid calibration is required before correction can be turned on."
169	standardMeasurementRequired	"The attempted operation can only be performed on a standard measurement type."

170	noDivisorBuffer	"A valid divisor buffer is required before normalization can be turned on."
171	invalidReceiverPowerCalParagraph	"Receiver power cal requires the measurement to be of unratioed power."
172	ecalCouldNotConfigure	"Could not configure the Electronic Calibration system. Check to see if the module is plugged into the proper connector."
173	measHasNoMemoryAlg	"This measurement does not support memory operations"
174	measHasNoNormalizeAlg	"This measurement does not support normalize operations."
175	userCharacterizationNotFound	"User characterization was not found in the Electronic Calibration module."
176	measInvalidBufferSize	"The data provided has an invalid number of points. It could not be stored."

Technical Support

Click on the region of interest.

- For assistance with your test and measurement needs go to **www.agilent.com/find/assist**
- Or contact the test and measurement experts at Agilent Technologies.

Other Support Topics

United States:

(tel) (+1) 800 452 4844

(alt) (+1) 303 662 3999

(fax) (+1) 888 900 8921

Canada

(tel) 1 877 894 4414

(fax) 1 (905) 206 4120

rope:

Austria

(tel) 0820 87 44 11*

(fax) 0820 87 44 22

Belgium

(tel) (+32) (0)2 404 9340

(alt) (+32) (0)2 404 9000

(fax) (+32) (0)2 404 9395

Denmark

(tel) (+45) 7013 1515

(alt) (+45) 7013 7313

(fax) (+45) 7013 1555

Finland

(tel) 08 0052 4000

(alt) (+358) 10 855 2100

(fax) (+358) 92 536 0176

France

(tel) 0825 010 700*

(alt) (+33) (0)1 6453 5623

(fax) 0825 010 701*

Germany

(tel) 01805 24 6333*

(alt) 01805 24 6330*

(fax) 01805 24 6336*

Ireland

(tel) (+353) (0)1 890 924 204

(alt) (+353) (0)1 890 924 206

(fax) (+353) (0)1 890 924 024

Israel

(tel) (+972) 3 9288 500

(fax) (+972) 3 9288 501

Italy

(tel) (+39) (0)2 9260 8484

(fax) (+39) (0)2 9544 1175

Luxemburg

(tel) (+32) (0)2 404 9340

(alt) (+32) (0)2 404 9000

(fax) (+32) (0)2 404 9395

Netherlands

(tel) (+31) (0)20 547 2111

(alt) (+31) (0)20 547 2000

(fax) (+31) (0)20 547 2190

Russia

(tel) (+7) 095 797 3963

(alt) (+7) 095 797 3900

(fax) (+7) 095 797 3901

Spain

(tel) (+34) 91 631 3300

(alt) (+34) 91 631 3000

(fax) (+34) 91 631 3301

Sweden

(tel) 0200 88 22 55*

(alt) (+46) (0)8 5064 8686

(fax) 020 120 2266*

Switzerland (French)

(tel) 0800 80 5353 opt. 2*

(alt) (+33) (0)1 6453 5623

(fax) (+41) (0)22 567 5313

Switzerland (German)

(tel) 0800 80 5353 opt. 1*

(alt) (+49) (0)7031 464 6333

(fax) (+41) (0)1 272 7373

Switzerland (Italian)

(tel) 0800 80 5353 opt. 3*

(alt) (+39) (0)2 9260 8484

(fax) (+41) (0)22 567 5314

United Kingdom

(tel) (+44) (0)7004 666666

(alt) (+44) (0)7004 123123

(fax) (+44) (0)7004 444555

Japan:

(tel) 0120 421 345

(alt) (+81) 426 56 7832

(fax) 0120 421 678

Latin America:

Mexico

(tel) (+52) 55 5081 9469

(alt) 01800 5064 800

(fax) (+52) 55 5081 9467

Brazil

(tel) (+55) 11 4197 3600

(fax) (+55) 11 4197 3800

Australia:

((tel) 1800 629 485

(alt) 1800 143 243

(fax) 1800 142 134

New Zealand

(tel) 0 800 738 378

(fax) 64 4 495 8950

Asia Pacific:

China

(tel) 800 810 0189

(alt) (+86) 10800 650 0021

(fax) 800 820 2816

Hong Kong

(tel) 800 930 871

(alt) (+852) 3197 7889

(fax) (+852) 2 506 9233

India

(tel) 1600 112 929

(fax) 000800 650 1101

Malaysia

(tel) 1800 888 848

(alt) 1800 828 848

(fax) 1800 801 664

Singapore

(tel) 1800 375 8100

(fax) (+65) 6836 0252

South Korea

(tel) 080 769 0800

(alt) (+82) 2 2004 5004

(fax) (+82) 2 2004 5115

Taiwan

(tel) 0800 047 866

(alt) 00801 651 317

(fax) 0800 286 331

Thailand

(tel) 1800 226 008

(alt) (+66) 2 268 1345

(fax) (+66) 2 661 3714

3.8 GHz Frequency Adjustment

This routing adjusts the internal fixed-frequency YIG Oscillator to 3.8 GHz by changing a DAC value. This DAC value is stored in the analyzer's non-volatile memory.

Typically, the oscillator can be set to within 12 kHz of 3.8GHz; it is not necessary for it to be exactly 3.8GHz.

Spectrum Analyzers Compatibility

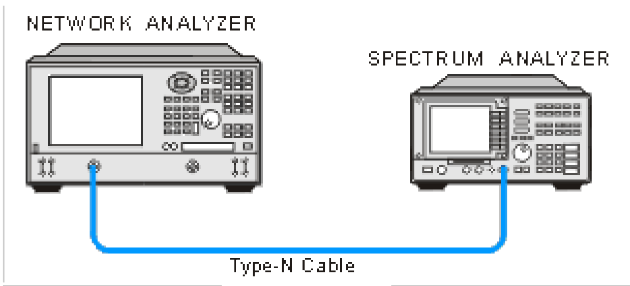
This routine is compatible with Agilent 856x and 859x spectrum analyzers.

If no compatible analyzer is available, select "NONE" for the spectrum analyzer. You can then adjust the DAC manually by viewing the 3.8 GHz signal on another analyzer.

Procedure (For Compatible Spectrum Analyzers Only)

Note: The viewable 3.8 GHz signal level will be low; typically be around -70dBm. Do not use any attenuators in the adjustment, other than the default 10 dB attenuation used in most spectrum analyzers.

1. Connect spectrum analyzer input to the network analyzer's PORT 1 output.
2. Connect GPIB cable from analyzer to spectrum analyzer. Make sure no other controllers are active on the same connection.



3. Set the spectrum analyzer GPIB address to 18.
4. In the analyzer **System** menu, point to **Service**, **Adjustments**, and click **3.8 GHz Freq. Adjust**.
5. Click **Begin Adj.** for the program to adjust the internal oscillator for minimal error and store the results. When the status area indicates the adjustment is complete, select **Exit**.

Procedure (For Non-Compatible Spectrum Analyzers Only)

Note: The viewable 3.8 GHz signal level will be low; typically be around -70dBm. Do not use any attenuators in the adjustment, other than the default 10 dB attenuation used in most spectrum analyzers.

1. Connect the spectrum analyzer input to the network analyzer's PORT 1 output.
2. Set the spectrum analyzer to the following settings:
   - Center frequency=3.8 GHz
   - Span= 100 MHz
   - Bandwidth= 10 kHz

- Scaling where a signal of -70 dBm will be clearly visible

3. In the analyzer **System** menu, point to **Service**, **Adjustments**, and click **3.8 GHz Freq. Adjust**.
4. Under **Spectrum Analyzer**, select **NONE** option for spectrum analyzer.
5. Click **Begin Adj**.
6. The application presets the DAC to an initial value equal to the current value stored. View the spectrum analyzer to see if the signal is above or below 3.800 GHz.
   - If the signal frequency is above 3.8 GHz, move the slider to adjust the DAC to a lower value (left).
   - If the frequency is below 3.8 GHz, move the slider to adjust the DAC to a higher value (right).

Note: The valid DAC values are from 0 to 4095. The oscillator will shift about 23 kHz per DAC value.

---

7. Set the DAC value to reach a frequency very close to 3.8 GHz. If you made large changes in DAC values, allow several seconds for thermal effects to stablize.
8. Change the spectrum analyzer settings to better view the frequency signal:
   - Frequency span = 500 kHz
   - Bandwidth = 3 kHz
9. Change the DAC value to keep the signal centered at 3.8 GHz.
10. Once you have determined the correct DAC value, click **SAVE DAC** to permanently store that value into EEPROM. Click **Exit**.

Note: If large changes are made to the existing DAC value, then this test should be repeated again after 15-30 minutes. This allows the thermal effects to fully stablize.

---

10 MHz Reference Frequency Adjustment

This routine adjusts the analyzer's internal time-base to exactly 10 MHz by changing a DAC value. This DAC value is stored in the analyzer's non-volatile memory. This routine should only be necessary in the following situations:

- The frequency reference assembly is replaced.
- The 10 MHz reference has drifted significantly from the factory adjusted value.

WARNING: The range of this adjustment is only about 20 Hz. It is highly recommended that a very accurate frequency standard be used to measure this 10 MHz signal.

Frequency Counter Compatibility

This procedure uses SCPI commands (over GPIB) to communicate with the frequency counter. It should work with the Agilent 5313xA, 5315xA, 53181A series of counters as well as the older 5350 series.

If no compatible counters are available, select the "Manual" mode of operation. If you do choose the manual mode, you must input the measured frequency manually.

Procedure for GPIB Counters Only

1. Connect the analyzer rear panel 10 MHz Reference output to the frequency counter .
2. Connect a GPIB cable from the analyzer to the counter. Make sure no other controllers are active on the same connection.
3. If applicable, connect the house frequency standard to the counter reference input.
4. Set the counter GPIB address to 03. Ensure that the counter is the only device at this address.
5. In the analyzer **System** menu, point to **Service**, **Adjustments**, and click **10 MHz Freq. Adjust**.
6. Click **Begin Adj**. The application adjusts the internal reference for minimal error and stores the results.
7. Click **Read Freq** to trigger another reading of the 10 MHz signal.
8. Click **Read DAC** to view the current DAC value stored in the analyzer's non-volatile memory (value = 0 - 4095).
9. When the status area indicates the adjustment is complete, click **Exit**.

Procedure for Non-GPIB Counters

1. Connect the counter input to the rear panel 10 MHz Reference Output.
2. Set the counter to at least 1 Hz resolution.
3. If applicable, connect the house-frequency standard to the counter reference input.
4. In the analyzer **System** menu, point to **Service**, **Adjustments** and click **10 MHz Freq. Adjust**.
5. Under **Frequency Counter**, select **Manual.**
6. Click **Begin Adj.**

7. The application presets the DAC to an initial value. Enter the measured frequency offset from 10 MHz. If the measured frequency is less than 10 MHz, use a minus (-) sign to indicate a negative error. For example:

    - If the counter reads 10000003.5 Hz, enter 3.5 (or +3.5) in the indicated window.
    - If the counter reads 9999997.8 Hz, enter -2.2 in the indicated window.

8. The adjustment loops at least 3 times unless the entered value is exactly zero.

    - Click **Read Freq** to trigger another reading of the 10 MHz signal.
    - Click **Read DAC** to view the current DAC value stored in the analyzer's non-volatile memory (value = 0 - 4095).

9. When the status area indicates the adjustment is complete, click **Exit**.

Note: If the counter is misreading the frequency, it may be necessary to attenuate the input, or set the input impedance to 50 ohms, or both.

LO Power Adjustment

This procedure adjusts the receiver's LO input power to a specific level by changing DAC values. These DAC values are then stored in the analyzer's non-volatile memory. The procedure will vary depending upon the model number.

Power Meter Compatibility

This routine is only compatible with the Agilent EPM series of power meters. Different sensors may be used. For 9 GHz analyzers and below, an 8482 or E4412A sensor can be used. For the higher frequency units (20 GHz or above), a sensor must be able to measure a maximum of 20 GHz. At no time during this test will a frequency higher than 20 GHz be measured, even if the PNA has a maximum frequency of 50 GHz.

If the older HP 84xx series of sensors are used, the correct calibration data should be entered into the appropriate cal table of the EPM series power meter, although for this adjustment, high accuracy is not required. Inaccuracies in the order of several tenths of a dB are acceptable.

Procedure

1. Allow the analyzer and power meter to warm up for 30 minutes.
2. Manually zero and calibrate the power sensor. (This allows you to skip this step later)
3. Connect a GPIB cable from the analyzer to the power meter. Make sure no other controllers are active on the same connection.
4. Set the power meter GPIB address to 13. (others can also be used; 13 is the default)
5. Remove the outer cover on the analyzer.
6. In the PNA **System** menu point to **Service**, then **Adjustments**, and click **LO Power Adjust**.
7. Connect the power sensor to the LO output, using adapters if needed.. The LO output location varies with model number. Click on the LO Power Adjust **Setup** menu selection to see a diagram of the exact location.
8. **For 9 GHz units and below:**
   Click **Begin Adj** to start the LO power cal routine. The routine adjusts the power level for each band (1 through 3) to fall within certain bounds. If any changes are made, it automatically stores them.

   For 20GHz units and above:
   If using an 84xx power sensor, click **Configure** and select the proper sensor model number. Click **Close**. Click **Calibrate** to begin the adjustment. The entire calibration process takes about 5 minutes. Once completed, you can verify the current calibration accuracy by clicking **Verify Cal**.
   Note: Correction constants are defaulted at the beginning of calibration. Once the calibration process has started, it must be completed in order to regenerate proper data.
9. Click **Read DAC** to view the current DAC values (0-4095) stored in the PNA non-volatile memory for each band (0-7).
10. When the message/status area indicates the adjustment is complete, click **Exit**.
11. Reconnect the semi-rigid cable and replace the covers.

Offset LO Power Adjustment

Note: This adjustment is only performed on PNAs with Frequency Offset Mode (option 080).

The Offset LO Adjustment sets the LO power for the offset mixer to a consistent value across all bands. It requires access to the internal components of the PNA so that the power sensor can be connected to the LO output. Because the LO frequency does not exceed 3 GHz, almost any power sensor can be used. The adjustment is relatively simple and only takes a couple of minutes.

When to perform

This adjustment should be performed when any of the following occur:

- the A13 Frequency Offset Receiver is significantly modified or replaced
- the A9 Synthesizer is replaced (that is where the correction data resides)

How to perform Offset LO Power adjustment

1. To start the Offset LO adjustment, click **System**, point to **Service**, **Adjustments**, then click **Offset LO Adjust**.
2. You will be prompted to zero and cal the power sensor. (You can do this before beginning.)
3. Connect the sensor to J3 of the A13 LO output by removing the existing cable (or simply disconnecting one end) as shown in the Set-up diagram in the program.
4. Connect the power meter to the PNA using a GPIB cable. Make sure the GPIB address shown in the program matches the actual power meter address (default is 13.)
5. Click **Adjust** to begin the test and follow the instructions.

The program automatically adjusts all bands; no user input is needed. The program repeats several times as this is an iterative process. The progress of the adjustment is shown on the screen.

The Configure menu selection is for factory personnel ONLY.

Once completed, to verify the actual results, click Verify.

Upon exiting, the PNA application will restart; this takes several seconds.

Operator's Check

- [Overview]
- [How to Run the Operator's Check]
- [Operators Check Dialog Box Help]

Overview

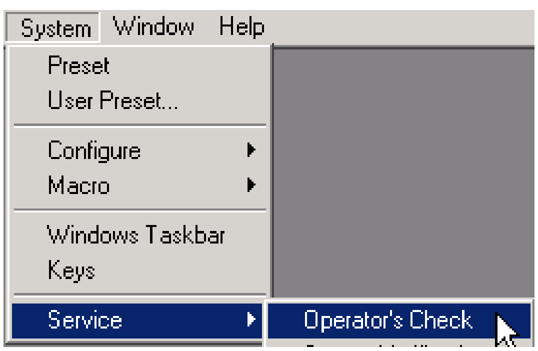
The Operator's Check should be performed when you first receive your PNA, and any time you wish to have confidence that the PNA is working properly.

Note: The Operator's Check does not verify performance to specifications. To verify PNA performance to specifications, run System Verification.

The Pass/Fail criteria used in the Operator's Check identifies **obvious failures** in the following portions of the PNA hardware:

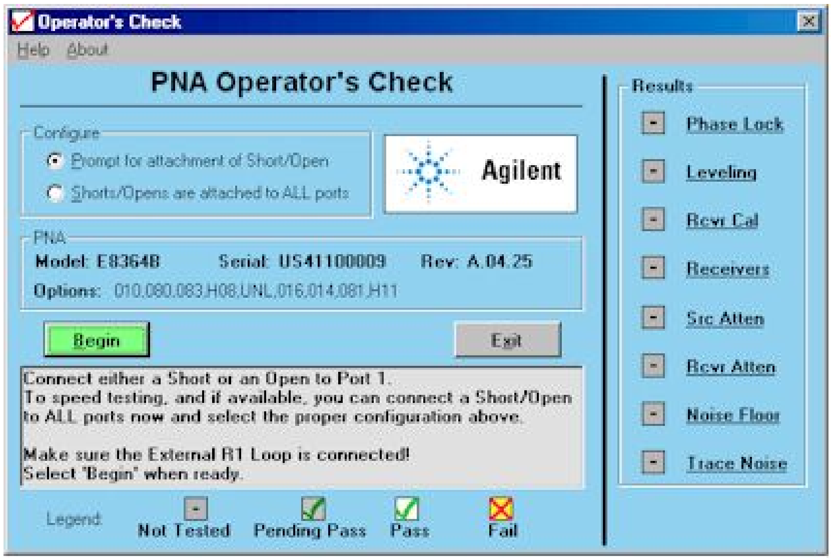
- Repeatability of the RF switch in the test set
- Attenuation ranges of the test port attenuators (if installed)
- Calibration of the receivers
- Frequency response of the receivers
- Phase lock and leveling
- Noise floor and trace noise

How to Run the Operator's Check

1. 

2. Connect one or more standards (see Configure).
3. Click **Begin** and **Continue** (if necessary) until "Operator's Check is complete!" appears.

Learn more about using the front panel interface



This dialog box will look slightly different, depending on PNA model number and installed options. Some of the tests are not run if the appropriate option is not installed.

Operators Check dialog box help
Configure **Prompt for attachment of Short / Open** If you do not have enough shorts or opens for all test ports, you will be prompted to move the standard to the next test port. Connect either a short or open to port 1, then click Begin. **Shorts / Opens are attached to all ports** Connect either a short or open for each test port, then click Begin. All ports are tested without interruption. You can mix shorts and opens on the test ports. **PNA** Shows information about the PNA that is being tested. **Legend** Shows the status icons used in the Operator's Check and their meaning. **Pending Pass** means that a portion of the testing has been completed successfully. **Results** Shows the current status of each test. Click on the test name to learn how that test is performed. This may help in troubleshooting failed tests. If any tests Fail, refer to Chapter 3 of the PNA service guide. **Begin** Starts the Operator's Check. **View Results** Shows all results in text format. Failed items are preceded by **===>>>.** This text file can be printed or saved with a unique file name to compare results with previous or subsequent testing. **Exit** Ends the program and closes the window.

Phase-Lock IF Gain Adjustment

The E836x A/B PNA models have a variable gain control for the phase-lock loop IF signal. By dynamically changing the gain as a function of frequency and power, the phase-lock signal amplitude can be adjusted to a constant level for the entire operating range of the instrument. This constant level is important for phase-lock acquisition and stability.

When to perform

Phase-Lock IF Gain Adjustment should be performed when any of the following occur:

- a source calibration
- an assembly in the reference receiver path (R1,R2) is replaced.
- the Test Set Motherboard is replaced
- the Phase Lock board is replaced
- **Phase Lock Lost** error message appears after replacing a source or receiver assembly

How to perform Phase-Lock IF Gain adjustment

1. From the **System** menu, click **Service**, then **Adjustments**, then **IF Gain Adjustment**.
2. Select any special test set options installed, otherwise leave the selection at **None**.
3. No connections to the test ports are required.
4. Click **Begin Adj.** The adjustment takes about a minute to complete.
5. The advanced screen is for factory personnel only.

System Verification

The system verification utility verifies the PNA system specifications by automatically measuring the magnitude and phase for all four S-parameters for each verification device, and comparing the values against the following:

- Factory measured data from files on the verification disk
- Limit lines based on the measurement uncertainty

System Verification requires the use of a calibration kit and verification kit which have been recently certified using a NIST-traceable process. System Verification can NOT be used to perform this kit certification.

Operator's Check should also be performed to verify the basic operation of the PNA.

- **Equipment Used in the System Verification**
- **Precautions for Handling Airlines**
- **Flow Diagram of Procedure**
- **Procedure for System Verification**
- **If the System Fails the Verification Test**
- **Interpreting the Verification Results**

Note: Although the performance for all S-parameters is measured, the S_{11}, S_{22}, S_{33}, and S_{44} phase uncertainties are less important for verifying system performance. Therefore, the limit lines will not appear on the printouts.

Equipment Used in the System Verification

For PNA models:

E8356A, E8357A, E8358A
N3381A, N3382A, N3383A
E8801A, E8802A, E8803A
(Type-N test ports)

Equipment Type	Type-N	3.5 mm
Calibration kit **or** ECAL Module	85032F 85092B	85033E 85093B
Verification kit	85055A	85053B

RF Cable	N6314A	See Cable substitution

E8362A/B
N5230A (20 GHz)
(3.5 mm test ports)

Equipment Type	3.5 mm	Type-N
Calibration kit **or** ECAL Module	85052B/C/D N4691A	85054B/D N4690A
Verification kit	85053B	85055A
RF Cable(s)	Single: 85131C/E Pair: 85131D/F	Single: 85132C/E Pair: 85132D/F
Adapters	None	Single: 85130C and one 7mm-to-Type-N from 85054B cal kit_ Pair: Two 7mm-to-Type-N from 85054B cal kit

E8363A/B, E8364A/B
N5230A (40 or 50 GHz)
(2.4 mm test ports)

Equipment Type	2.4 mm	3.5 mm	Type-N
Calibration kit **or** ECAL Module	85056A/D N4693A	85052B/C/D N4691A	85054B/D N4690A
Verification kit	85057B	85053B	85055A
RF Cable(s)	Single: 85133C/E Pair: 85133D/F	Single: 85134C/E Pair: 85134D/F	Single: 85135C/E Pair: 85135D/F
Adapters	None	Single: 85130F_ Pair: None	Single: 85130E and two 7mm-to-Type-N from 85054B cal kit_ Pair: Two 7mm-to-Type-N from 85054B cal kit

E8361A
(1.85 mm test ports)

Equipment Type	**1.85 mm**	**2.4 mm** **3.5 mm** **Type-N**
Calibration kit **or** ECAL Module	85058B N4694A	See 2.4 mm test port table above
Verification kit	85058V	See 2.4 mm test port table above
RF Cable(s)	Single: N4697E Pair: N4697F	See 2.4 mm test port table above
Adapters	None	See 2.4 mm test port table above

Cable Substitution

The test port cables specified for the PNA have been characterized for connector repeatability, magnitude and phase stability with flexing, return loss, insertion loss, and aging rate. Since test port cable performance is a significant contributor to the system performance, cables of lower performance will increase the uncertainty of your measurement. It is highly recommended that the test port cables be regularly tested.

If the system verification is performed with a non-Agilent cable, ensure that the cable meets or exceeds the operation of the specified cable. Refer to the cable User's Guide for specifications.

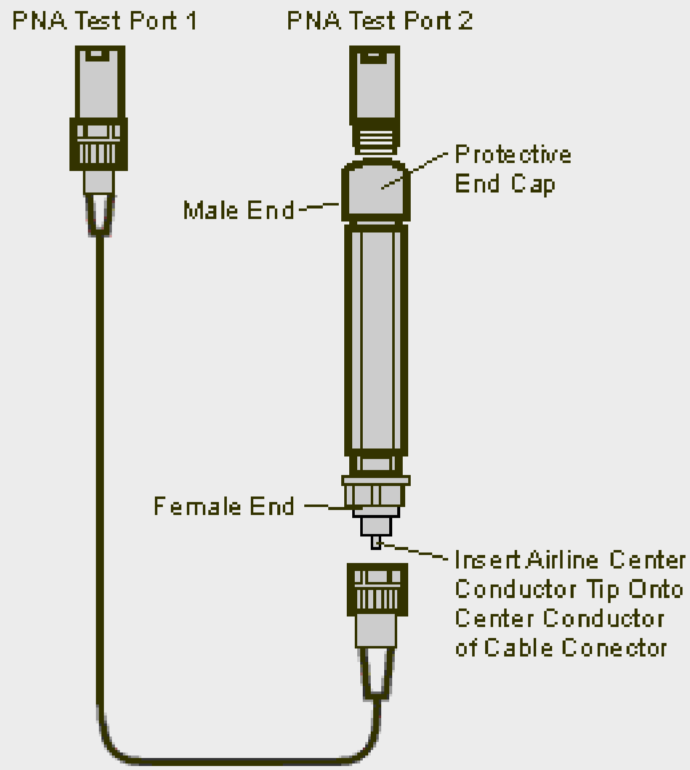
Calibration Kit Substitution

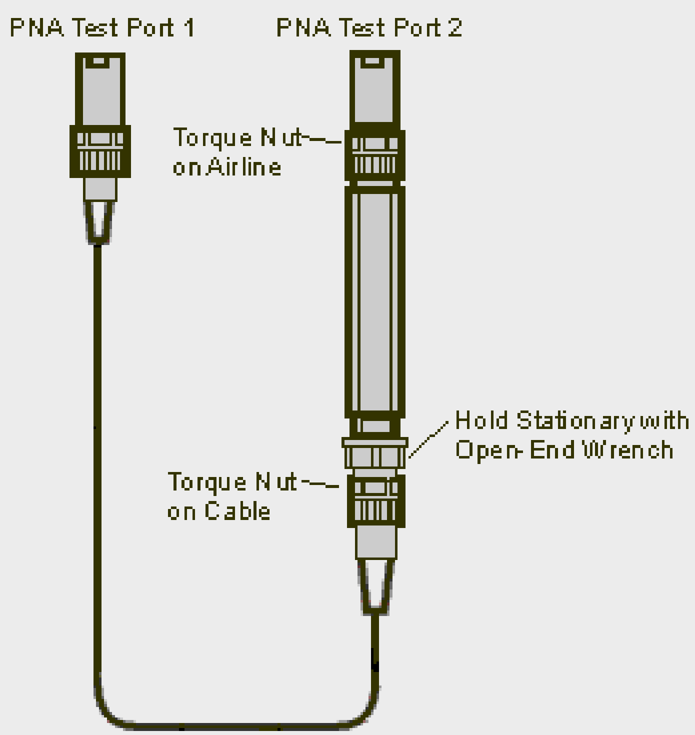
Non-Agilent calibration kits are not recommended nor supported.

Precautions for Handling Airlines

When you are using the airlines in the verification kit, observe the following practices to ensure good measurement techniques.

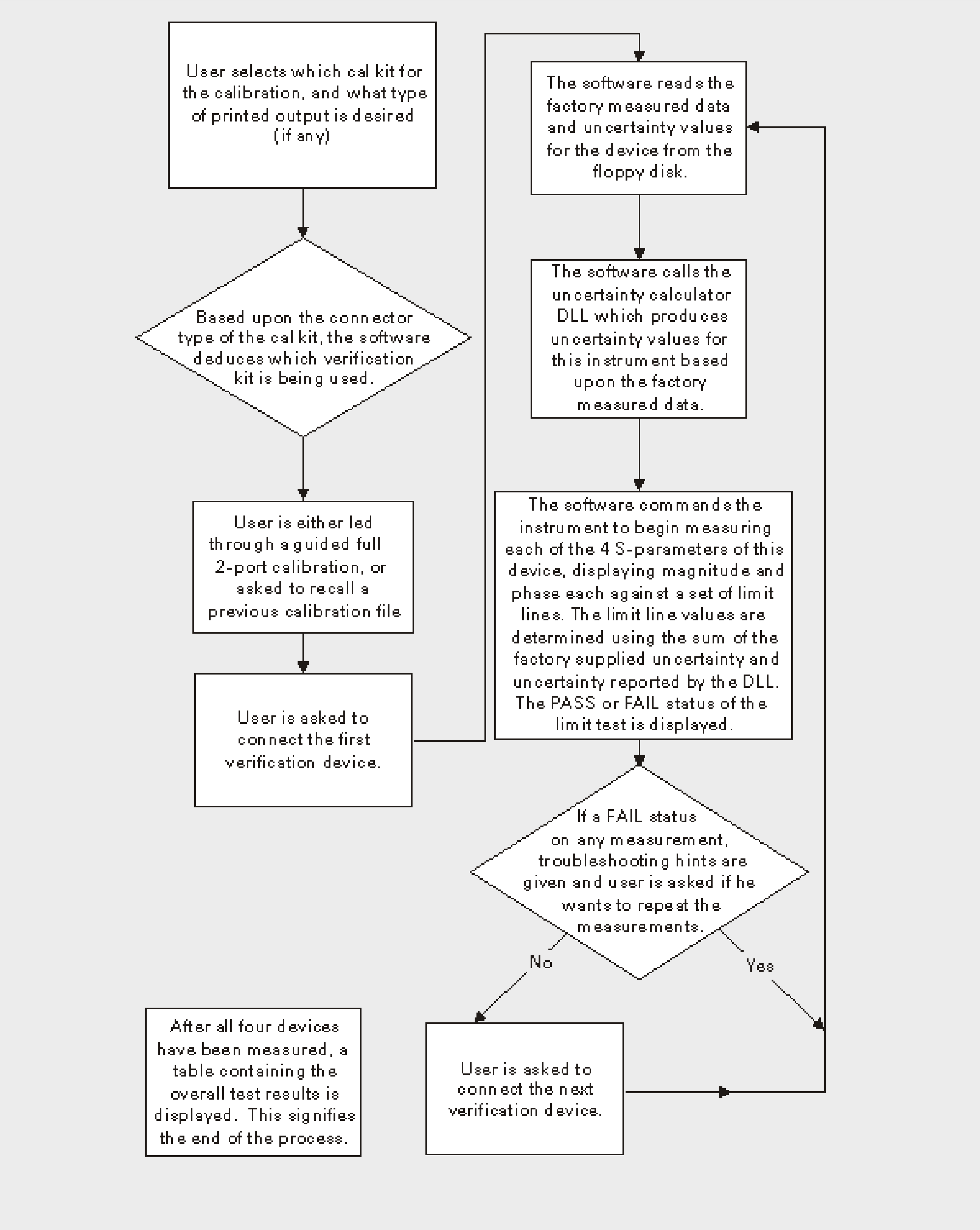
- Be very careful not to drop the airline's center or outer conductor. Damage will result if these devices are dropped.
- Use proper Electro-Static Discharge (ESD) procedures.
- Clean your hands or wear gloves as skin oils will cause a change in electrical performance.





Flow Diagram of Procedure

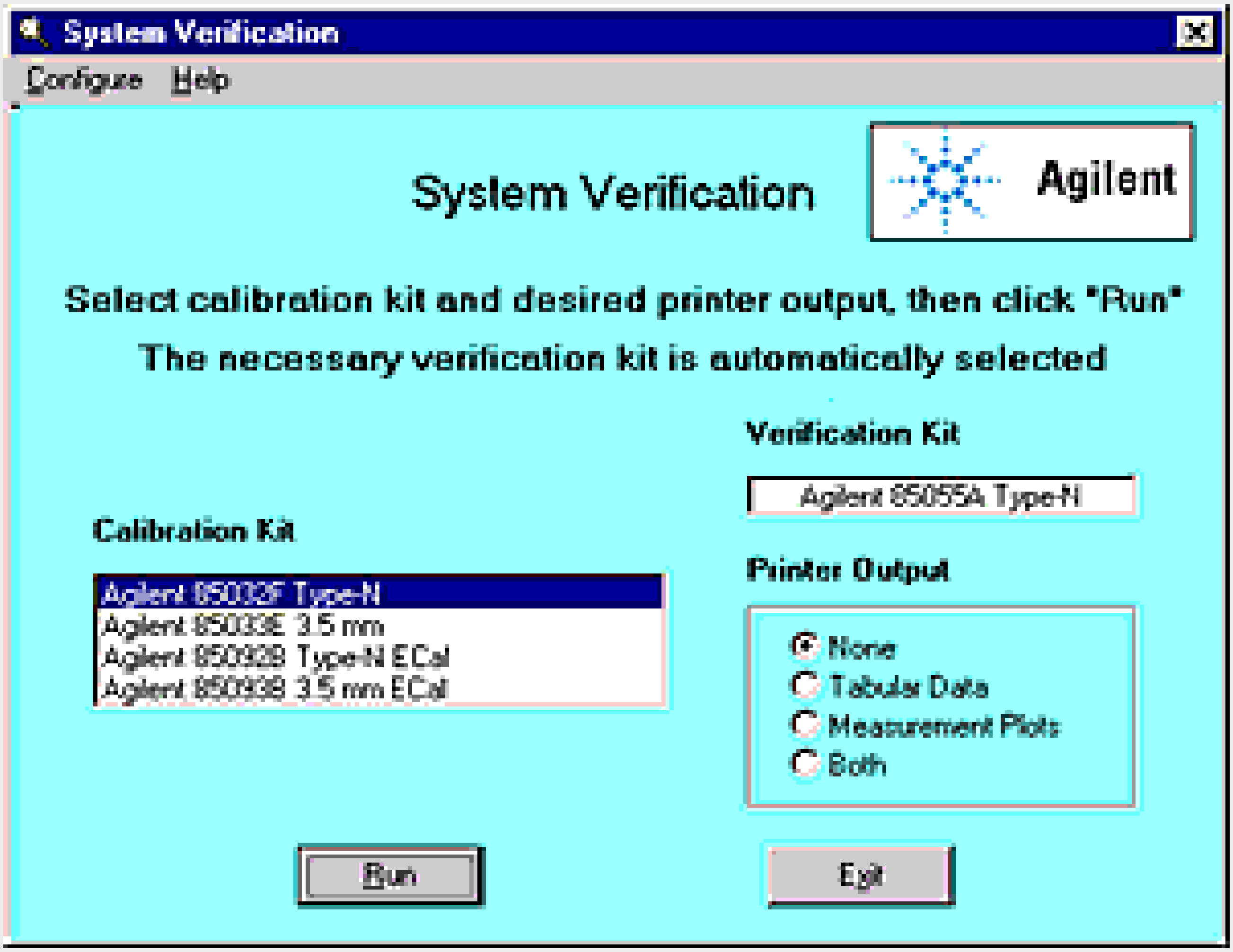
The operational flow of the software is depicted by the flowchart shown below.


User selects which cal kit for the calibration, and what type of printed output is desired (if any)
Based upon the connector type of the cal kit, the software deduces which verification kit is being used.
User is either led through a guided full 2-port calibration, or asked to recall a previous calibration file
User is asked to connect the first verification device.
The software reads the factory measured data and uncertainty values for the device from the floppy disk.
The software calls the uncertainty calculator DLL which produces uncertainty values for this instrument based upon the factory measured data.
The software commands the instrument to begin measuring each of the 4 S-parameters of this device, displaying magnitude and phase each against a set of limit lines. The limit line values are determined using the sum of the factory supplied uncertainty and uncertainty reported by the DLL. The PASS or FAIL status of the limit test is displayed.
If a FAIL status on any measurement, troubleshooting hints are given and user is asked if he wants to repeat the measurements.
No
Yes
User is asked to connect the next verification device.
After all four devices have been measured, a table containing the overall test results is displayed. This signifies the end of the process.

Procedure for System Verification

1. If you want printed test outputs, connect a printer to the analyzer. Let the analyzer warm up for at least 30 minutes.
2. Insert the PNA verification kit floppy disk into the analyzer disk drive.
3. On the **System** menu, point to **Service**, and click **System Verification**. The System Verification window similar to this will be displayed.

System Verification window



4. In the **Calibration Kit** box, select the calibration kit or ECal module that is being used. The corresponding verification kit to use appears in the **Verification Kit** box.
5. Under **Printer Output** click on any of the following options.
    - **Print Tabular Data:** Prints the verification data in tabular form which includes measured data and uncertainty limits. Refer to a tabular data example, later in this topic.
    - **Print Graphs:** Prints the verification data in graphical form. The graphic form includes the measured data trace, factory supplied data trace and uncertainty limits. Refer to a plot data example, later in this topic.
    - **File Tabular Data:** Writes the verification data in tabular form to a text file in the C:\Program

Files\Agilent\Network Analyzer\Documents\ directory.

- **File Graphs:** Saves a screen image in .PNG format in the C:\Program Files\Agilent\Network Analyzer\Documents\ directory.

Note: If you want printed output, it is assumed you have already installed the Windows driver for your particular printer, and have tested that you can print to the printer from the network analyzer. This software is designed to print to whichever printer is currently set as the Default printer (see Printers in the Windows Control Panel).

6. To modify the number of ports to be verified, to change the number of devices to measure, or to use a previously stored verification calibration, click on the **Configure** tab and make the desired selections.
   - For the system verification to be truly adequate, the software must measure all devices in the kit with a recent calibration applied. Removing and reattaching any test port cables or adapters invalidates all previous calibrations.
7. Click **Run**.
8. Follow the instructions on the analyzer for performing the system verification, inserting the verification devices as prompted.

Note for 3 Port PNA:
The System Verification Procedure is **repeated three times**. The first time, **Ports 1 and 2** are measured as a pair; then **Ports 1 and 3** are measured; and lastly, **Ports 2 and 3** are measured.

Note for 4 Port PNA:
The System Verification Procedure is **repeated two times**. The first time, **Ports 1 and 2** are measured as a pair, then **Ports 3 and 4** are measured.

Step-by-Step Process Description

1. Depending upon the selected choice in the Calibration submenu of the Configure menu, the user is either prompted to recall a previous calibrated instrument state, or is guided through a full 2-port calibration using the selected calibration kit. For ECal, the ECal module is connected just once; a standby message is posted while the software is performing the calibration.
2. The user is prompted to connect the first verification device.
3. The software reads the factory measured data for that device and uncertainty values for that data (CITIfiles) from the floppy disk supplied with the verification kit.
4. The software sends the factory measured data, calibration kit and instrument state information to the uncertainty calculator DLL, which generates uncertainty values specific to the PNA.
5. The analyzer first sets up for magnitude measurements of all four S-parameters, each parameter in a separate window (lin mag for S_{11} and S_{22}, log mag for S_{21} and S_{12}). Each of the factory measured S-parameters are fed to the appropriate window as a memory trace. Limit line offsets are calculated as the sum of the factory measured data uncertainties and PNA uncertainties reported by the DLL. Upper and lower limits are displayed (factory measured data + uncertainty

sum, factory measured data - uncertainty sum). The PNA takes a sweep, limit test is turned on and PASS/FAIL status is reported in each of the four windows.

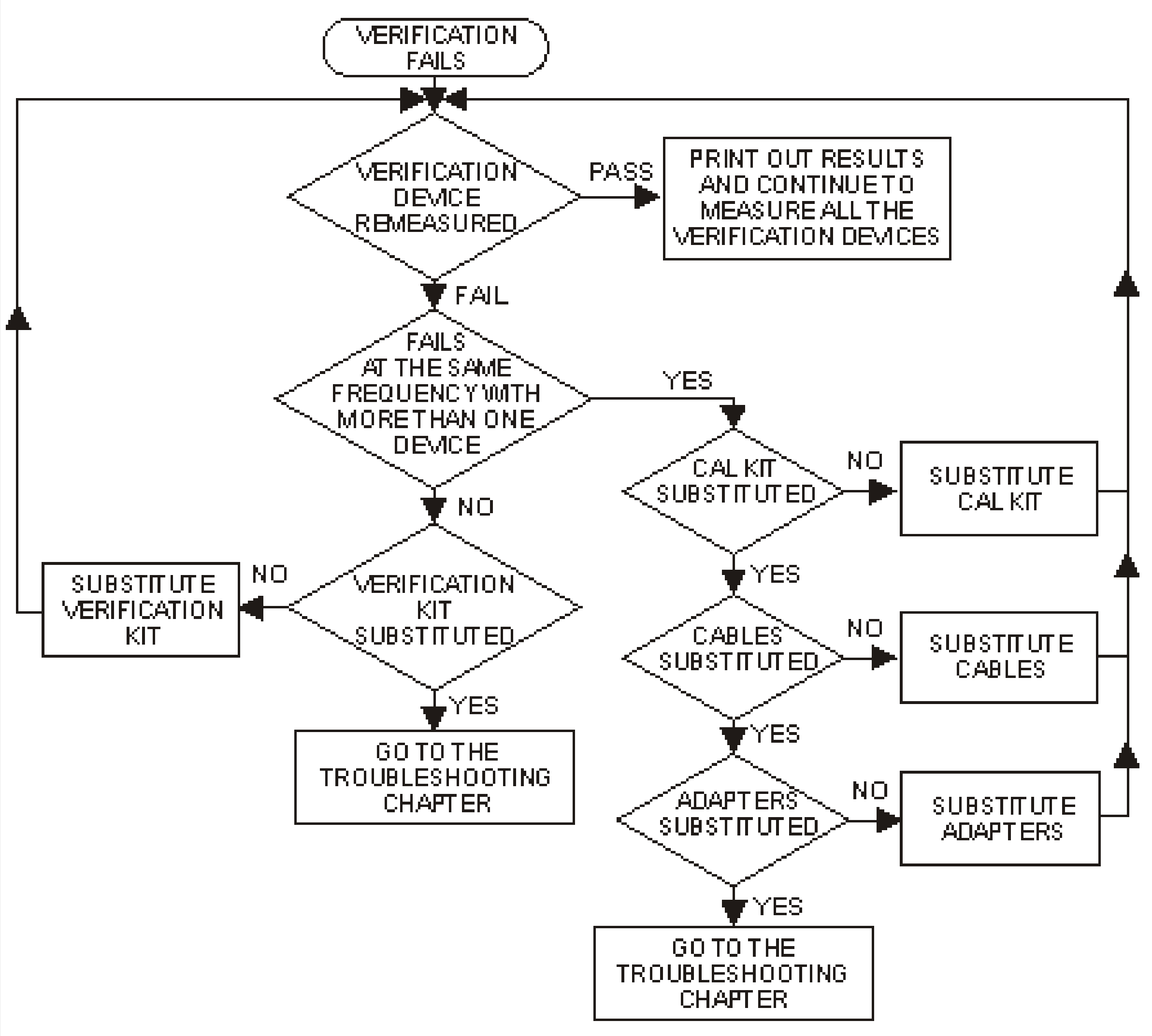
6. The user clicks a button when ready to view phase measurements. The four windows get updated for phase format, phase memory traces, phase limits and PASS/FAIL result.
7. If the limit test of any of the four S-parameters (magnitude or phase) indicates a FAIL status, the software suggests troubleshooting tips and asks if the user would like to repeat measurement of that device or proceed to the next device. If proceeding to the next device, the factory measured data and uncertainties for the next device are read from floppy, the uncertainty DLL gets called with this next set of factory measured data, and the four measurement windows get updated for magnitude measurement of the next device.
8. The software follows this same process until all selected devices have been measured, at which point a summary window is displayed containing the set of PASS/FAIL results for all four parameters of each device.

If the System Fails the Verification Test

IMPORTANT: Inspect all connections. Do not remove the cable from the analyzer test port. This will invalidate the calibration that you have done earlier.

1. Repeat this verification test. Make good connections with correct torque specifications for each verification device.
2. Disconnect, clean and reconnect the device that failed the verification test. Then measure the device again.
3. If the analyzer still fails the test, check the measurement calibration by viewing the error terms as described in "Front Panel Access to Error Terms" on page 4-7 of the Service Guide.
4. Refer to the graphic below, for additional troubleshooting steps.

Verification Fails Flowchart



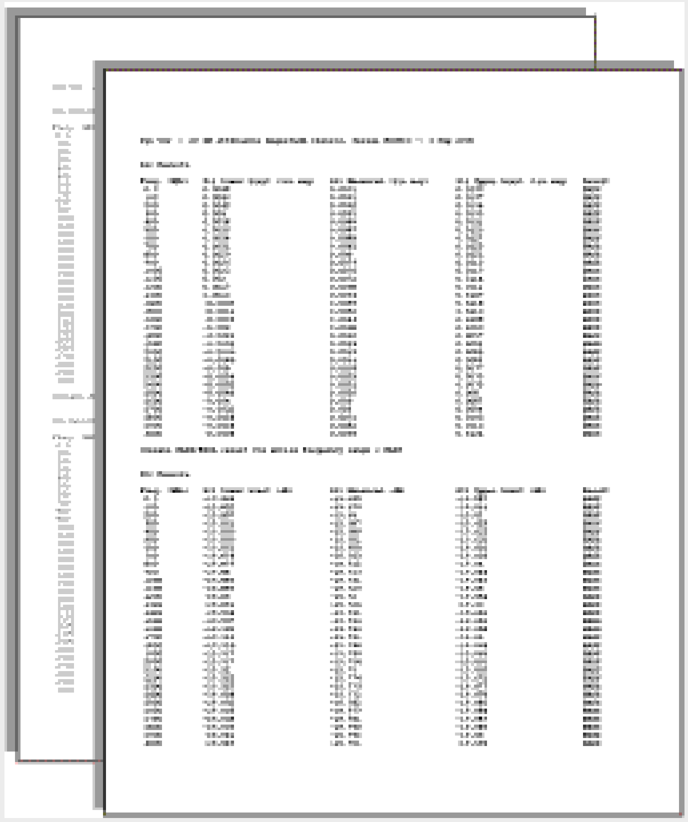
Interpreting the Verification Results

The graphic below shows an example of typical verification results with **Tabular Data** selected in the **Printer Output** area of the **System Verification** window. A graphic later in this topic shows an example of typical verification results with **Measurement Plots** selected in the **Printer Output** area of the **System Verification** windows. These printouts include a comparison of the data from your measurement results with the traceable data and corresponding uncertainty specifications. Use these printouts to determine whether your measured data falls within the total uncertainty limits at all frequencies.

The tabular data consists of:

- Frequency of the data points (in MHz).
- Lower limit line as defined by the total system uncertainty specification.
- Results of the measurement.
- Upper limit line as defined by the total system uncertainty specification.
- Test status (PASS or FAIL) of that measurement point.

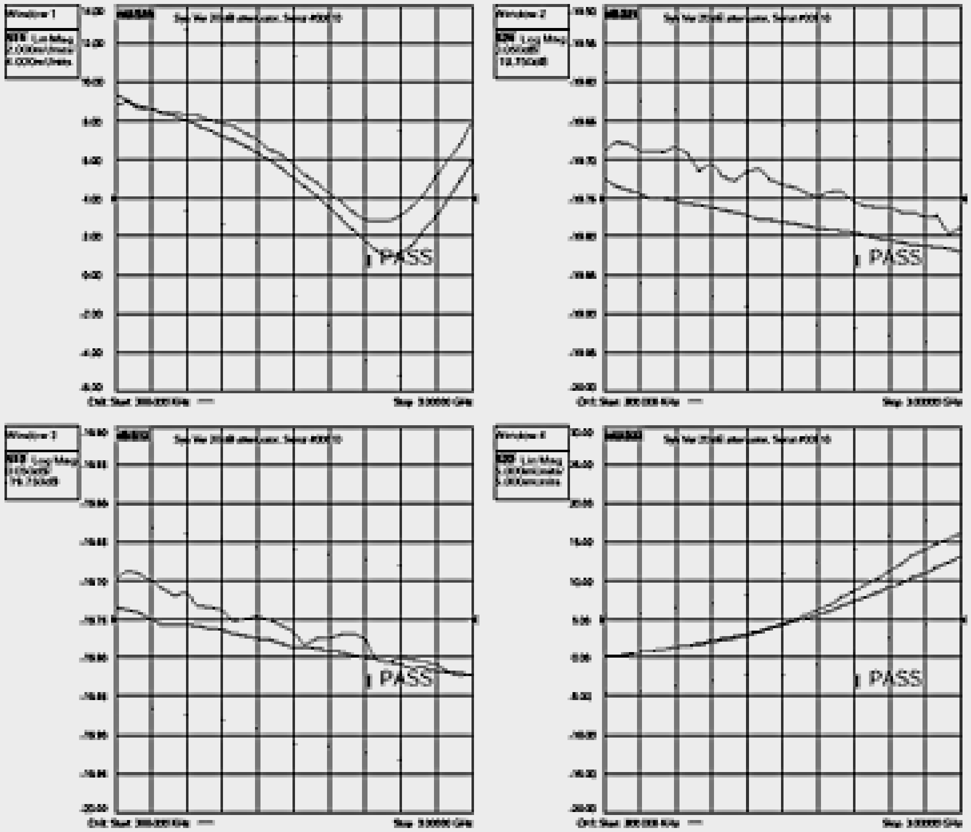
Printout of Tabular Verification Results



The printed graphical results show:

- Upper limit points as defined by the total system uncertainty specifications.
- Lower limit points as defined by the total system uncertainty specifications.
- Data measured at the factory.
- Results of measurements.
- Measurement parameter names and formats (Lin Mag or Log Mag).
- Serial number of device (00810).
- Device being measured (Sys Ver 20 dB attenuator).

Printout of Graphical Verification Results


PASS
PASS
PASS
PASS

Source Calibration

Source calibration adjusts the PNA source power for flatness across its full frequency range. This adjustment is for service only; not for measurement calibration.

Required Equipment

Preferred Power Meter: E4419B
Alternate Power Meters: E4419A or EPM-442A

Note: The power sensor depends on the PNA frequency range. Depending on the PNA model, two power sensors may be required to test the full frequency range.
The PNA front panel connector type will determine the cable used and if an adapter is required with the power sensor(s).

PNA Model	Power Sensor(s)	Cable
E8356A	8482A	N6314A
E8357A E8358A	8482A and E4412A	N6314A
E8801A	8482A	N6314A
E8802A E8803A	8482A and E4412A	N6314A
N3381A	8482A	N6314A
N3382A/N3383A	8482A and E4412A	N6314A
E8361A	8487A (use with adapter 11900B) and V8486A (use with adapter V281B) ** See E8361A procedure below	N4697-60001
E8362A / B	E4413A	85131E
E8363A/B E8364A/B	8487A (use with adapter 11900B)	85133E

See PNA Accessories

Procedure

1. Refer to your power meter documentation to ensure the proper calibration factors for the power sensor have been entered into the table on the power meter.
2. Connect a GPIB cable between the power meter and network analyzer.
3. Ensure the power sensor(s) are connected to the power meter.
4. In the analyzer **System** menu, point to **Service**, **Adjustments**, and click **Source Calibration**.

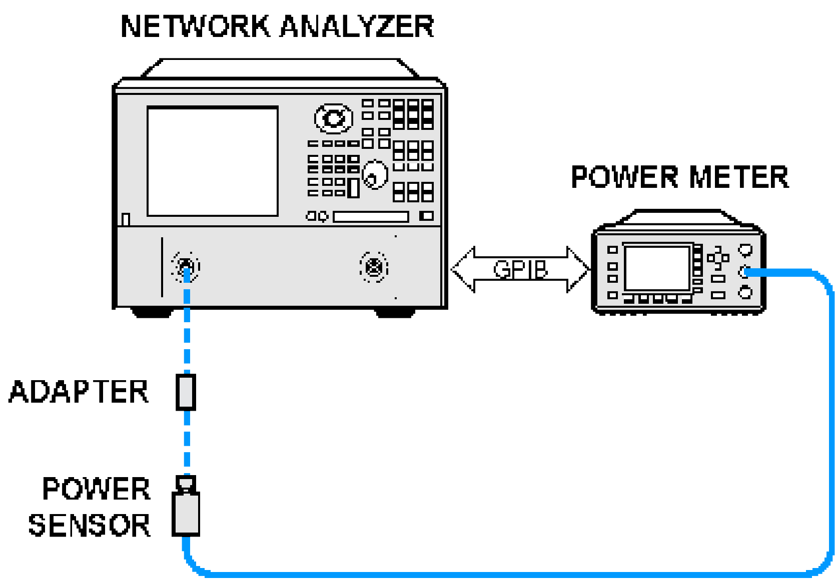
5. The software presents you with three choices:

   a. Click **Inspect Linearity** to observe how accurately the power steps from one power level to the next. When finished, Test Output Power Linearity should meet specification for your PNA Model.

   b. Click **Inspect Amplitude** to observe flatness of the source power versus frequency for three power levels: -5 dBm, 0 dBm and +5 dBm. When finished, Test Output Power Accuracy should meet specification for your PNA Model.

   c. Click **Calibrate** to begin the source calibration process. The software begins by identifying the power meter and sensor. Then you are prompted to connect the sensor(s) and cable as needed.

Connecting sensors to the PNA



Additional Information

All ports are tested on all PNAs. Source calibration takes approximately 20 to 90 minutes to complete depending on the frequency range of the PNA.

Troubleshooting

In the event there is a problem with Source Calibration, please refer to the "Troubleshooting" chapter in the PNA Service Guide.

E8361A Procedure

Source and Receiver calibration requires the power meter to measure the source power over the full range of each of the PNA internal bands. Because the 8487A can not measure accurately above 50 GHz, it can only be used up to the next highest band switch frequency at 46.2 GHz. The V8486A sensor and V281B adapter are used from 46.2 GHz to 67 GHz.

For highest accuracy, the V8486A and V281B should be sent to Agilent for a custom calibration from 45 GHz to 70 GHz.

For the next highest accuracy level, the following procedure shows how to measure correction factors yourself from 46 to 50 GHz. This procedure assumes you have already loaded correction factors for both sensors into the power meter.

1. On your power meter, add 46 and 48 GHz to the Cal Factor Table.
2. Preset the PNA
3. Tune the PNA to 46 GHz (CW frequency)
4. Using the 8487A, measure power at port 1. Record this value.
5. Tune the PNA to 48 GHz (CW frequency)
6. Using the 8487A, measure power at port 1. Record this value.
7. Connect the V8486A, V281A, and 1.85 f-f adapter to the power meter.
8. Tune the PNA to 46 GHz (CW frequency)
9. Adjust the cal factor table 46 GHz setting until the power meter reading matches the power readings from step 4.
10. Tune the PNA to 48 GHz (CW frequency)
11. Adjust the cal factor table 48 GHz setting until the power meter reading matches the power readings from step 8.

Receiver Calibration

Receiver calibration adjusts the network analyzer receivers for a flat response across its full frequency range. This adjustment is for service only; not for measurement calibration.

Required Equipment

Preferred Power Meter: E4419B
Alternate Power Meters: E4419A or EPM-442A

Note: The power sensor depends on the PNA frequency range. Depending on the PNA model, two power sensors may be required to test the full frequency range.
The PNA front panel connector type will determine the cable used and if an adapter is required with the power sensor(s).

PNA Model	Power Sensor(s)	Cable
E8356A	8482A	N6314A
E8357A/E8358A	8482A and E4412A	N6314A
E8801A	8482A	N6314A
E8802A/E8803A	8482A and E4412A	N6314A
N3381A	8482A	N6314A
N3382A/N3383A	8482A and E4412A	N6314A
E8361A	8487A (use with adapter 11900B) and V8486A (use with adapter V281B) ** See E8361A procedure	N4697-60001
E8362A / B	E4413A	85131E
E8363A/B E8364A/B	8487A (use with adapter 11900B)	85133E

See PNA Accessories

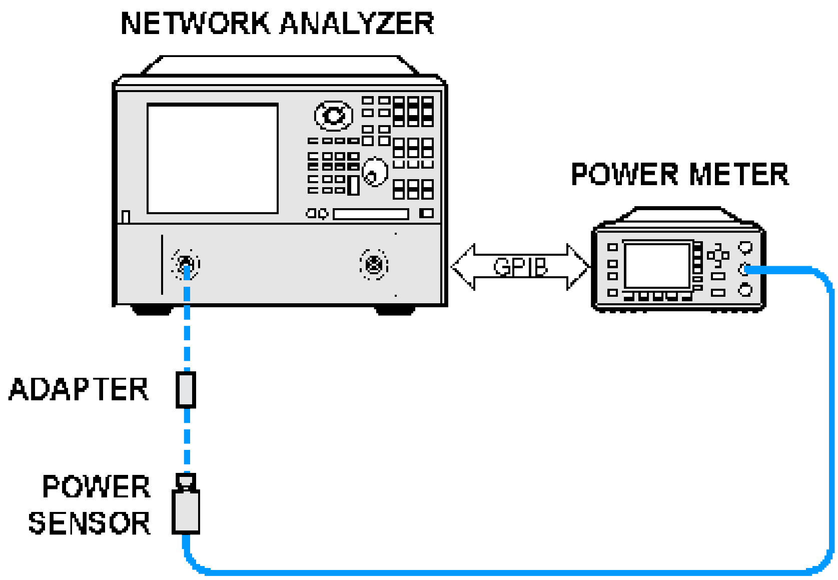
Procedure

1. Refer to your power meter documentation to ensure the proper calibration factors for the power sensor have been entered into a table on the power meter.
2. Connect a GPIB cable between the power meter and network analyzer.
3. Ensure the power sensor(s) are connected to the power meter.
4. In the analyzer **System** menu, point to **Service**, **Adjustments**,and click **Receiver Calibration**.
5. The software presents you with two choices:

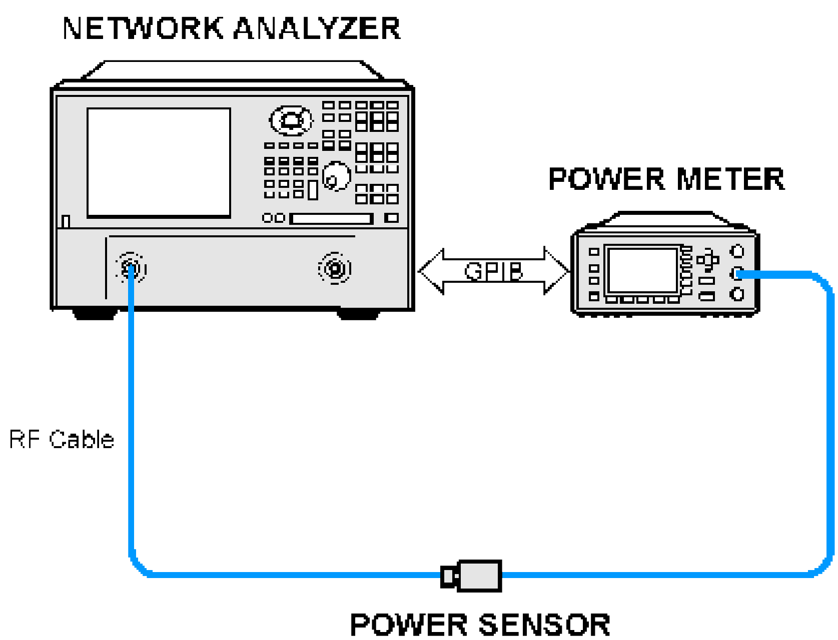
a. Click **Inspect Flatness** to observe flatness of receiver response versus frequency. Although there is no explicit specification for receiver flatness, Receiver Calibration should improve Transmission and Reflection Tracking error terms which are specified.

b. Click **Calibrate** to begin the receiver calibration process. The software prompts you to connect the sensor(s), cable and adapter as needed (see the following graphics).

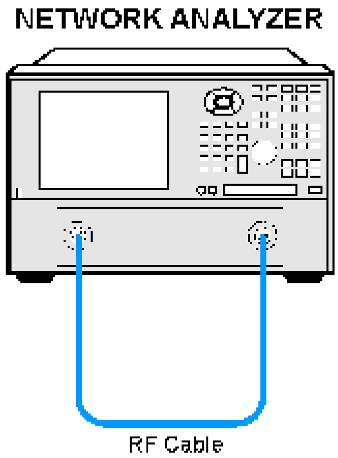
Connecting sensor(s) to the PNA



Connecting adapter and cable between sensor and PNA



Through connection using the specified cable



Additional Information

Receiver Calibration tests all PNA receivers, taking approximately 15 and 45 minutes. Length is dependent on frequency range and number of ports.

Troubleshooting

In the event there is a problem with Receiver Calibration, please refer to the "Troubleshooting" chapter in the Network Analyzer Service Guide.

Receiver Display

[The Receiver Display as a Troubleshooting Tool](#)

[How to start the Receiver Display](#)

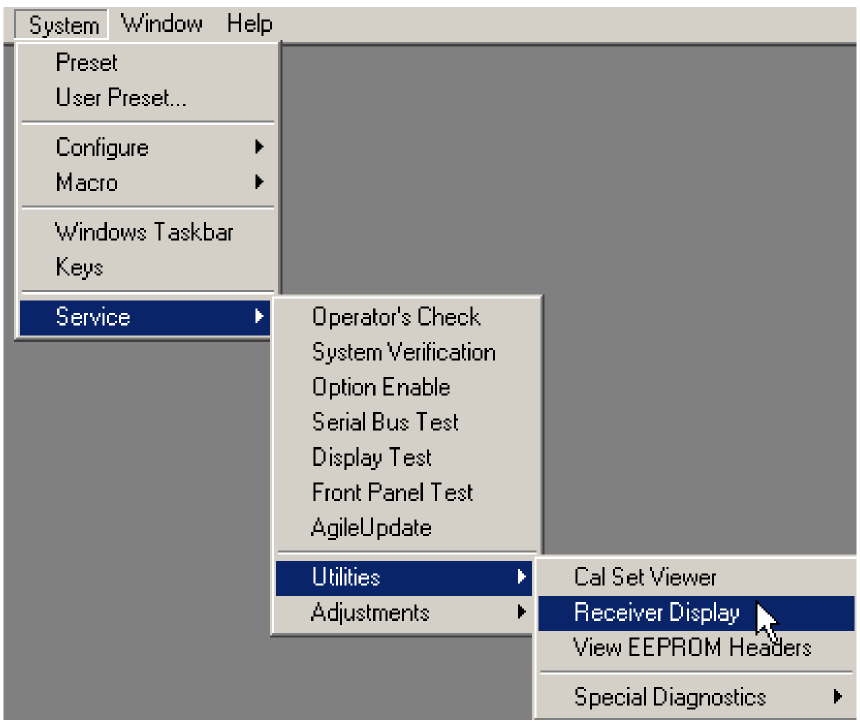
Other Support Topics

The Receiver Display as a Troubleshooting Tool

The Receiver Display is a Troubleshooting Tool. It enables the analyzer to isolate faulty functional groups within its own Measurement System. Traces for each Receiver are Displayed in individual windows. Identifying discrepancies of the traces in these windows can help isolate the faulty assembly.

For a thorough description of Receiver Display and the troubleshooting steps see Chapter 3 of the PNA Service Guide. You can download a copy of the Service Guide from our Web site at http://www.agilent.com/find/pna

How to Start the Receiver Display



Learn more about using the front panel interface

Serial Bus Test

This test is designed to work with the PNA Series Network Analyzer as it was originally manufactured. If any of the following assemblies have been replaced in the analyzer, this test may not show accurate pass/fail results.

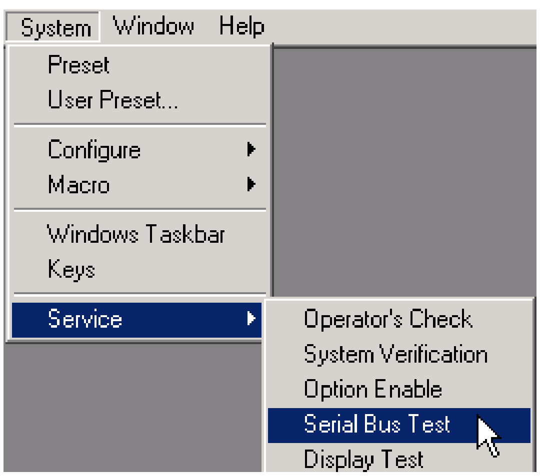
- A8
- A10
- A11
- A12

This test sequentially tests all 32 nodes at 300 kHz, and then all nodes whose values change with frequency, at 11 other frequencies. All measured values are written to an ASCII text file in the service directory any measured values which exceeed their tolerances are highlighted in the application and the data file.

The assemblies tested and their associated node numbers are as follows:

Board Assembly	**Node Numbers**
A12 Source Assembly	111 to 118
A10 Frequency Reference	211 to 218
A11 Phase Lock	311 to 318
A8 Fractional-N Synthesizer	411 to 418

How to Start Serial Bus Test



Learn more about using the front panel interface

Note: If any errors are reported, refer to the service guide.

Instrument Calibration

An instrument calibration is a process where the analyzer performance is measured to ensure that the analyzer operates within specifications. If any performance parameter does not conform to the published specifications, adjustments are made to bring the performance into conformance.

Why Should I Get an Instrument Calibrated?

Over time, the active components in the analyzer age and the performance may degrade or drift. To ensure that the analyzer is performing to the published specifications, you must have an instrument calibration performed periodically.

How Often Should I Get an Instrument Calibrated?

The instrument specifications are set to consider the performance drift that may occur over a 12 month period. Therefore, getting the instrument calibrated at 12 month intervals ensures that the analyzer maintains performance within the operating specifications. If you need the analyzer to maintain more consistent operation parameters, you may want to have the instrument calibrated more often than the suggested 12-month interval.

How Do I Get an Instrument Calibrated?

To get the instrument calibrated, send it to one of the Agilent Technologies service centers.

The PNA must be fully functional when it is sent to the service center, or they will charge for their repair services. If the PNA is being used in a secure environment where the hard drive can not be sent with the PNA, a second hard drive must be purchased and configured for use with the PNA in an "unclassified" environment before the PNA is sent to the service center. See Technical Support.

If you want to perform the instrument calibration, you must have the following required items:

- Instrument Calibration Test Equipment
- Performance Test Software

What Are My Choices of Instrument Calibration?

The following types of instrument calibration are available from Agilent Technologies:

Standard	Includes a certificate of calibration, stating the instrument has been calibrated and is operating within the published specifications.
Option UK6	Includes the test data from the calibration and the standard certificate, stating the instrument has been calibrated and is operating within the published specifications.
Option M40 (Special)	Includes the test data from the calibration and a certificate, stating the instrument has been calibrated using a process in compliance with ANSI Z540 and is operating within the published specifications.

Other Support Topics

About IF Access (H11 Option)

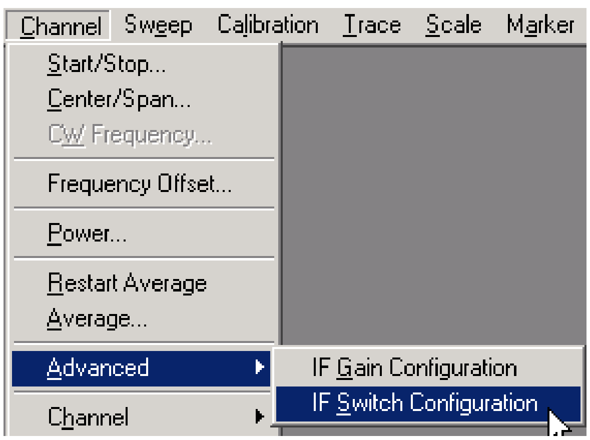
The PNA H11 option provides rear panel access to the PNA IF paths. You can also set the gain of the IF amplifiers.

- **How to Make IF Access Settings**
- **IF Gain Configuration**
- **IF Switch Configuration**

Other IF Access Topics

N5250A Typical System Performance

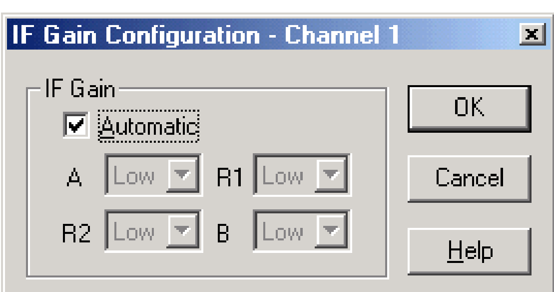
How to Make IF Access Settings



Learn more about using the front panel interface

IF Gain Configuration

The IF Gain Configuration settings allow you to manually set the gain of the PNA IF amplifiers.



IF Gain Configuration dialog box help

Automatic - check to allow the PNA to set the IF gain automatically. Clear to enable manual IF gain settings.

Note: Clearing this checkbox can result in inaccurate measurements.

A, R1, R2, B Receivers

Auto Gain - The PNA selects the best gain level to make pulsed measurements.

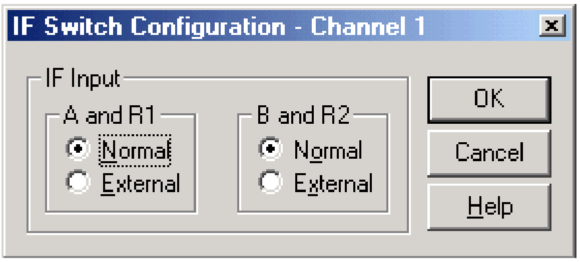
Low - about 0 dB of gain

Med - about 17 dB of gain

Hi - about 34 dB of gain

IF Switch Configuration

The IF Switch Configuration settings allow you to select the input path to the PNA IF amplifiers.



IF Switch Configuration dialog box help

Available only with the H11 option

Normal - allow the PNA to decide the input path to the PNA IF amplifiers.

External - always use the rear panel inputs to the PNA IF amplifiers.

External Millimeter Module Configuration

You can use external Millimeter Modules to extend the frequency coverage of your PNA. To use this feature your PNA must have the H11 Option.

- **PNA Limitations when using External Millimeter Modules**
- **How to Configure Millimeter Modules**

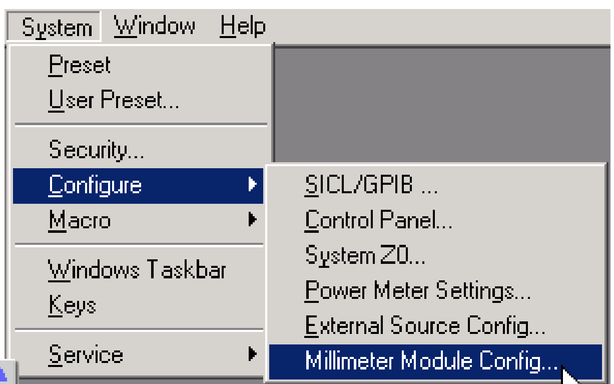
Other IF Access Topics

PNA Limitations when using External Millimeter Modules

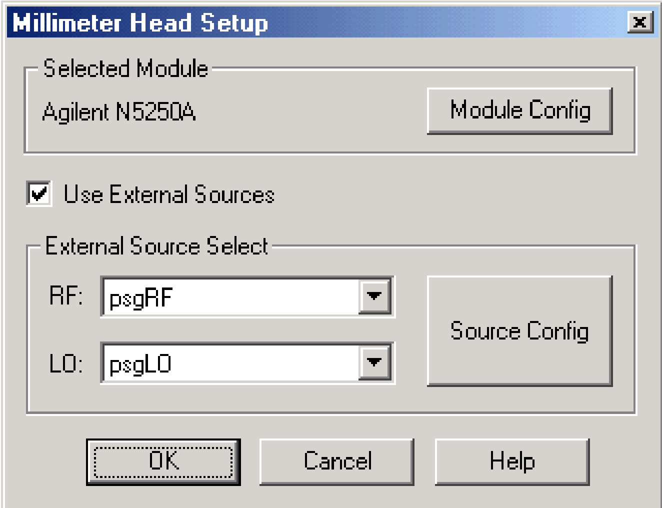
Power Settings When using external Millimeter Modules, the PNA cannot control the power level into your DUT above 67 GHz. Because of this limitation, PNA power settings will not function correctly. Some of these settings are: Power level in standard or segment sweep, source and receiver power calibrations, and calibration interpolation. Your modules may have a manual power control.

Frequency Offset Measurements Because of the many configurations used in frequency offset, we do not support Frequency Offset measurements when using external Millimeter Modules.

How to Configure Millimeter Modules



Learn more about using the front panel interface



Millimeter Head Setup dialog box help

Important Notes

- The External Sources will become UNSYNCHRONIZED after performing an Instrument Preset, recalling an instrument state, or adding a Channel. When this occurs, launch this dialog box again and click **OK.** We strongly recommend using only one channel to make measurements using external sources.
- For measurements up to 110 GHz, the amount of phase noise is comparable when using either the internal PNA sources or external sources. For the best measurement accuracy at frequencies above 110 GHz, it is strongly recommended that external sources be used.
- Only Agilent PSG sources are supported.

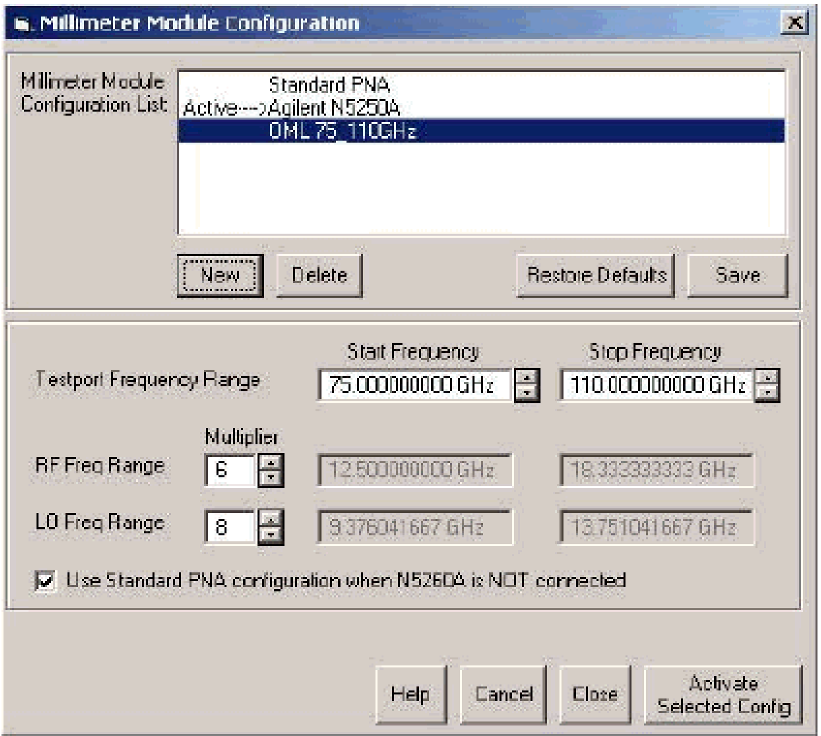
Selected Module Displays the currently selected module. To select and configure a different module, click Module Config.

Use External Sources Check to use external sources to provide an LO for the Millimeter Modules.

Source Config Click to invoke the External Source Configuration dialog box. This dialog box is also used to configure external sources for the FCA application. Ignore references to FCA in the help topic.

In addition, the following settings on that dialog box are NOT relevant for Millimeter Head Setup:

- **Trigger Mode** Hardware List (BNC) triggering is always used to control the RF and LO sources for Millimeter Wave Heads. Read the notes and follow the setup diagram for two external sources.
- **LO Calibration** The power out of the sources is set and controlled by the PNA H11 option.



Millimeter Module Configuration dialog box help

Millimeter Module Configuration List Lists the standard PNA configuration and others that you have created.

New Click to create a new Millimeter Module Configuration. Invokes the **Name** dialog box. Type a unique name for the new configuration using only alphanumeric characters and underscore, then click OK.

Delete Remove a Millimeter Module Configuration.

Restore Defaults Prompts, then REMOVES all configurations except for the standard PNA (and N5250A if installed).

Save Saves the current settings to the PNA hard drive.

Testport Frequency Range Set the Start and Stop frequencies of the selected configuration at the test ports. When Activated (click **Activate Selected Config**), this becomes the displayed Start and Stop frequency of the PNA.

RF Freq Range Multiplier RF Frequency Range (displayed in grey fields) multiplied by this value = Testport Frequency Range.

Use the Multiplier values that are specified in your test head documentation.

LO Freq Range Multiplier LO Frequency Range (displayed in grey field) multiplied by this value + 8.33 MHz = Testport Frequency Range.

Note: If the LO and RF frequency ranges are not within the operating range of the PNA, a warning message appears along with a red box around the invalid field. Click the appropriate Multiplier value up or down to correct the problem.

Use Standard PNA operation when N5260A is NOT connected When Activate Selected Config is clicked, the PNA detects if a N5260A is connected. If one is NOT connected and:

- This box is checked, then the selected configuration is NOT activated, but uses the Standard PNA configuration.

- This box is cleared, then the selected configuration IS activated anyhow.

Cancel Closes dialog box without saving changes.

Close Prompts to save changes, then closes the dialog box.

Activate Selected Config Saves the configuration, then closes and restarts the PNA application with the new configuration. To change to another configuration, including the standard PNA configuration, you must make this selection again.

Pulsed Application

The Pulsed Application is a Visual Basic program that provides a user interface for making pulsed measurements.

- Required Options
- Using the Pulsed Application
- How to Configure Pulse Generators and Receivers
- Pulse Profiling
- Signal Reduction versus Gate Width
- Pulsed Frequency Converter Measurements
- Writing your own Pulsed Application

For more conceptual information on pulsed measurements, visit www.Agilent.com and search for **AN 1408-11**.

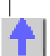
Other IF Access Topics

Required Options

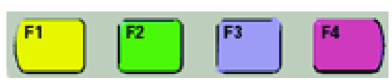
The PNA H08 option provides the Pulsed Application. The H08 option requires that your PNA also have Opt 014 (front panel access) and Opt 080 (frequency offset). To use the internal receiver gating feature of the Pulsed Application, your PNA must have the H11 hardware option. If your PNA does not have the required options, a message is displayed on the screen.

Using the Pulsed Application

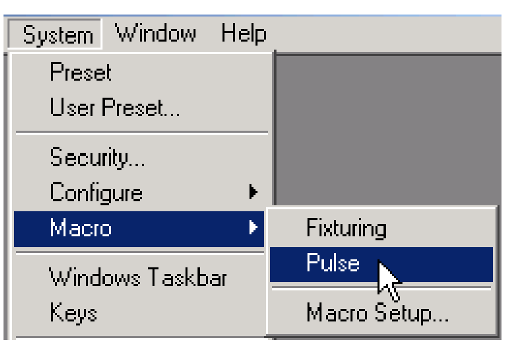
How to start the Pulsed Application



1. Press 

until **Pulse** is visible, then 

or

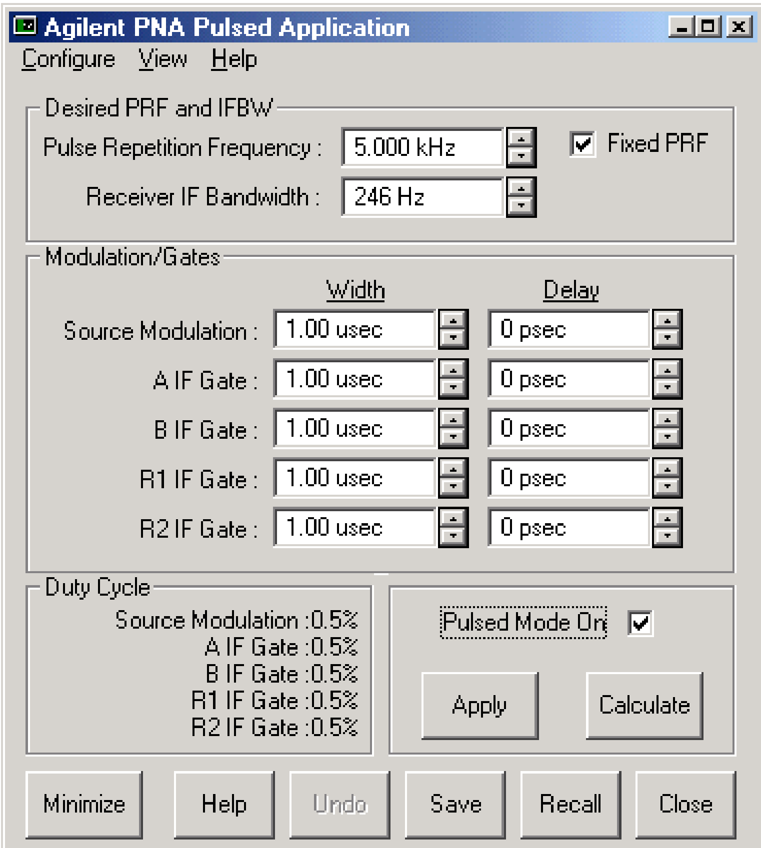
2. 

Learn more about using the front panel interface

Keypad Data Entry

The PNA Numeric Entry and Navigation keys can be used for dialog box input. Also, a keyboard can be used to enter values, including alpha characters for prefixes (for example, **u** for usec.) . After typing values, first press **Enter**, then press **Tab** to go to the next field.

The following is an image of the main dialog box:



Pulsed Application Main dialog box help

Note: An **error message** may appear on the PNA stating that the response frequency has exceeded the maximum allowed frequency.

The pulsed application may set the offset frequency (option 080) of the PNA to some value other than zero (the default value). If the stop frequency is set to the maximum of the PNA model, then the error message will appear.

To fix this, set the stop frequency to a value that is at least 2 KHz less than the maximum allowed. For example, if you have a 20 GHz PNA, and the stop frequency is set to 20 GHz, and the error message appears, then set the stop frequency to 19.999998 GHz

Configure

You can configure more than one channel to make pulsed measurements, but the channels must use the same pulse generator settings.

Only the Agilent 81110A Pulse Generator is supported with the Pulsed Application. Refer to the 81110A documentation for pulse repetition frequency and duty cycle capabilities.

See also:

Configure Receivers

Converter Measurements

Edit / Undo Pulse Application settings revert to those when Apply was last pressed.

Desired PRF and IFBW Enter the DESIRED values. When **Calculate** is pressed, one or both of these values may change.

Pulse Repetition Frequency: Frequency of the pulses from the Pulse Generator.

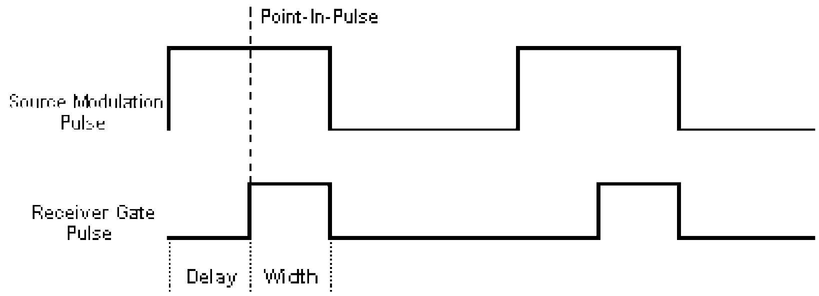
Receiver IF Bandwidth: IF Bandwidth of the PNA. Choose a setting from 1 Hz to 10 KHz.

Fixed PRF When checked, (default setting) the **Calculate** algorithm will NOT adjust the PRF, but only change the IF Bandwidth.

Modulation/Gates The Source Modulation and four PNA receiver gates can each have their own, or share, Pulse Generator outputs. Shared outputs have identical Width and Delay values. To configure and enable outputs, click **Configure**, then **Pulse Generators** to launch the Pulsed Generator Configuration dialog box.

Width Pulse Width.

Delay The delay that occurs before the pulse.



Duty Cycle Calculated Duty Cycle of the source and each of the selected receivers.

Pulse Mode On When this box is checked, the PNA is enabled for Pulsed measurements. The PNA Status Bar annotation indicates the following:

- **G** Internal IF gates enabled.
- **F** Filtering for Pulsed Measurements enabled.

Apply All selections are sent to the pulse generator and the active channel of the PNA.

Calculate All selections are calculated and valid PRF and IFBW values are entered in their fields. If these settings are not acceptable, try changing the values you previously entered and click Calculate again. When acceptable values are attained, click Apply to send these values to the pulse generator and PNA.

Pulse Profile Launches the Pulse Profile dialog box. Same as clicking **View / Pulse Profile.**

Minimize Click to minimize the dialog box to make changes in the PNA application.

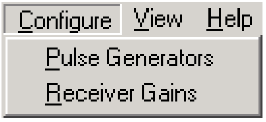
Save All settings from the Pulsed Application are saved in a *.ppf file. These settings are NOT saved with PNA instrument state.

Recall Restore settings from the specified *.ppf file that were previously saved.

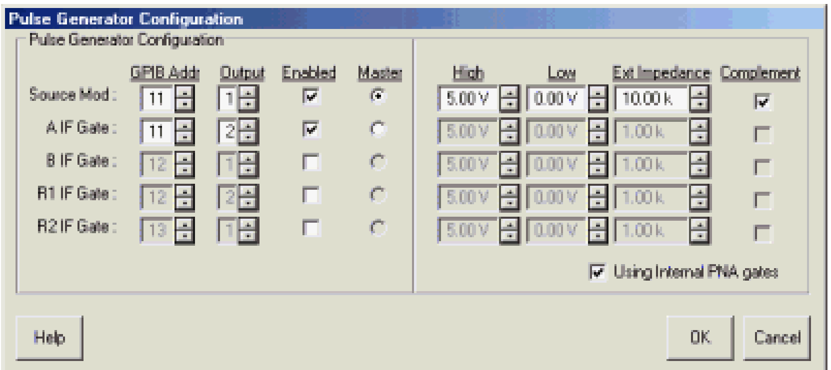
Close Closes the dialog box without saving changes.

How to configure Pulse Generators and Receivers

From the Pulse App main dialog box



If this setting is unavailable, click Calculate.



Pulsed Generator Configuration dialog box help

Configures the Agilent 81110A Pulse Generator outputs. You can configure each 81110A Pulse Generator with either one or two 81111A output modules.

The Source Mod and four PNA receiver gates can each have their own, or share, pulsed outputs. To share a generator output between one or more PNA inputs, use the same GPIB address and output module for each PNA input. This causes the shared PNA inputs to have identical **Width** and **Delay** values selected on the Main dialog.

Source Mod: The output that pulses the PNA source.

A, B, R1, R2 IF Gate: The PNA rear panel connector to receive the gating pulse.

GPIB Addr: The GPIB address of the 81110A.

Output: The output module of the 81110A.

Master: The 81110A that uses the 10 MHz reference signal from the PNA.

Enabled: Turns the pulse output ON.

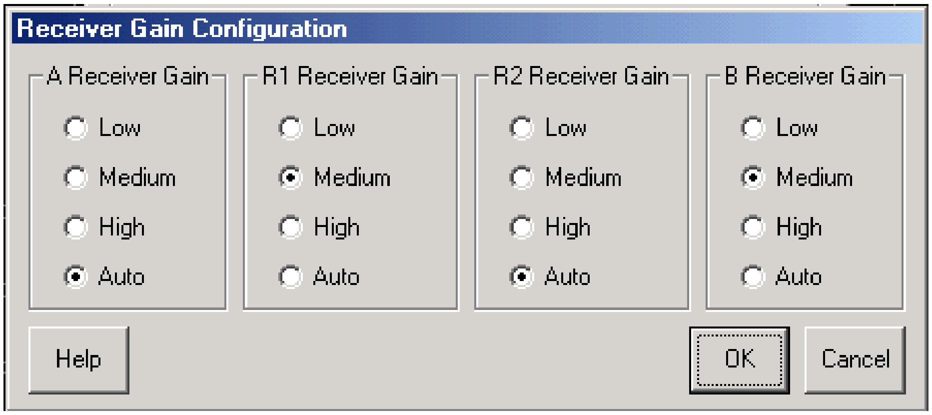
High: Specify a 'TTL-High' voltage level

Low: Specify a 'TTL-Low' voltage level

Ext Impedance: Impedance of the DUT

Complement: When this box is cleared, TTL HIGH is the pulse. When checked, TTL LOW is the pulse.

Using Internal PNA gates When this box is checked, the voltage, impedance, and complement values are forced to settings that prevent damage to the internal gates.



Receiver Gain Configuration dialog box help
Sets the gain of each receiver manually or automatically. **Auto** - The PNA selects the best gain level to make pulsed measurements. Use these to manually set the gain for each receiver. **Low -** about 0 dB of gain **Medium -** about 17 dB of gain **High** - about 24 dB of gain Some PNA models have only two gain settings instead of three as in the previous graphic.

Pulse Profiling

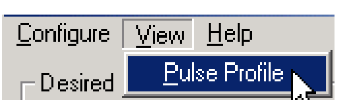
Pulse profiling provides a time domain view of the pulse envelope. Profiling is performed using a measurement technique that "walks" a narrow receiver "snapshot" across the width of the pulse. This is analogous to using a camera to take many small snapshots of a wide image, then piecing them together to form a single, panoramic view.

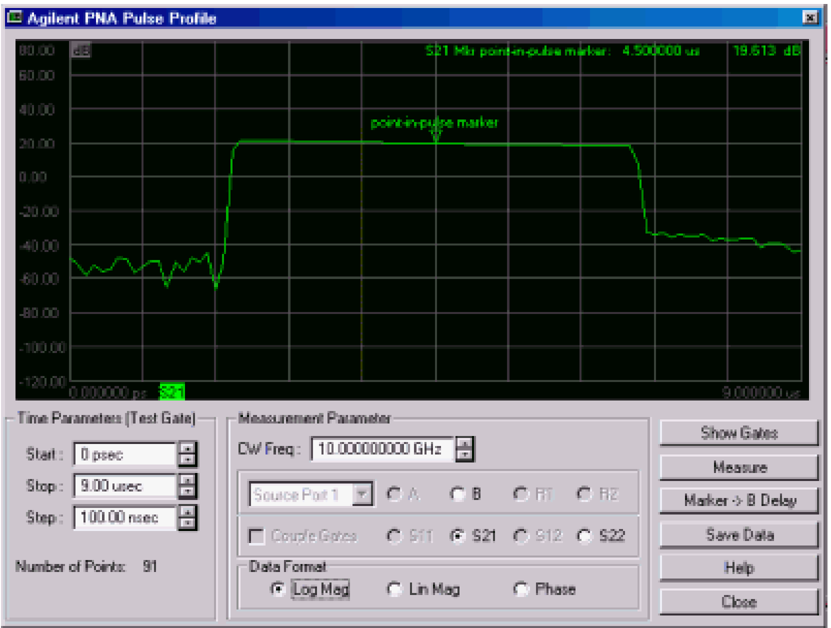
- Pulse Profiling can be performed using ratioed or unratioed measurements.
- Pulse Profiling is performed at a single CW frequency.

How to perform Pulse Profiling

From the Pulse App main dialog box

If this setting is unavailable, click Calculate.





Pulse Profile dialog box help

Learn about Pulse Profiling (scroll up)

Time Parameters

Start, Stop These two combine to make the window of the assembled pulse profile. To view the entire pulse, the start and stop values must be at least as wide as the Source Modulation Width plus Delay value (from the main Pulse App dialog box).

Step Each consecutive snapshot is incremented by this value until the stop value is reached. Therefore, the number of points for the pulse profile measurement can be calculated as: (Stop - Start) / Step. The higher the number of points, the longer it takes to make the measurement.

Measurement Parameter

CW Freq. Frequency of the PNA source.

Source Port The PNA port supplying the source power for single receiver measurements (Receivers A, B, R1, R2)

Select a Measurement Only those receiver gates that are configured in Pulsed Generator Configuration are available. Select from the following:

Unratioed (single receiver) - A, B, R1, R2

OR

S-parameters (ratioed)

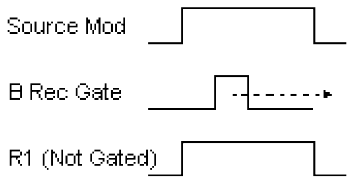
Measurements that are available depend on which receiver gates are configured:

- To measure S11, configure A and (optionally) R1.
- To measure S21, configure B and (optionally) R1.
- To measure S12, configure A and (optionally) R2.
- To measure S22, configure B and (optionally) R2.

If the reference receiver gate (R1 or R2) is NOT configured, the average of the Source Modulation pulse is used as the reference.

For example:

S21 Selected, ONLY **B** receiver gate is configured



B Gate is walked across the Source Modulation pulse.

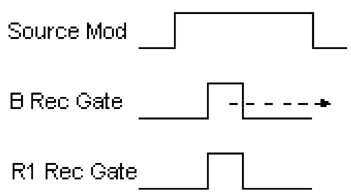
Source Modulation pulse average is used as reference (not gated).

Coupled Gates Available when the appropriate receiver gates are configured for your ratioed measurement.

- **Uncoupled** (box cleared) The reference gate is FIXED at the delay setting as the test gate is walked across the Source Modulation pulse.

For example:

S21 Selected, **B** and **R1** receiver gates configured, Gates **Uncoupled**



B Gate is walked across the Source Modulation pulse.

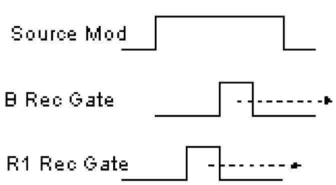
R1 gate is fixed at pulse width and delay setting.

- **Coupled** (box checked) The reference gate is walked simultaneously with the test gate at the specified delay values.

For example:

S21 Selected, **B** and **R1** receiver gates configured, Gates **Coupled**

B gate delay = 3 microseconds, **R1** gate delay = 2 microseconds,



B Gate is walked across the Source Modulation pulse.

R1 gate is fixed at pulse width and delay setting.

B gate leads R1 gate by one microsecond.

Data Format Log Magnitude, Linear Magnitude, or Phase (only available if S-parameter selected).

Buttons

Show Gates Allows you do change the receiver gating width and delay while looking at the results.

Measure Click to start the pulse profile measurement. A bar above the button indicates progress.

Marker to Delay After making a measurement, you can drag the display maker to any point along the trace. Click this button and the marker time is entered into the Receiver **Delay** field on the main dialog box.

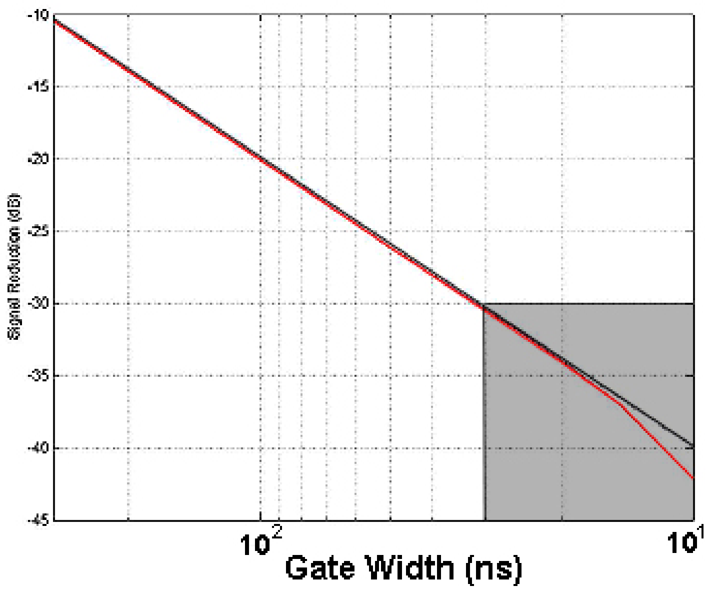
Save Data Saves time domain data to the PNA hard drive in any of the following formats:

- Touchstone (*.s1p)
- Comma delimited (*.prn)
- Citifile (*.cti)

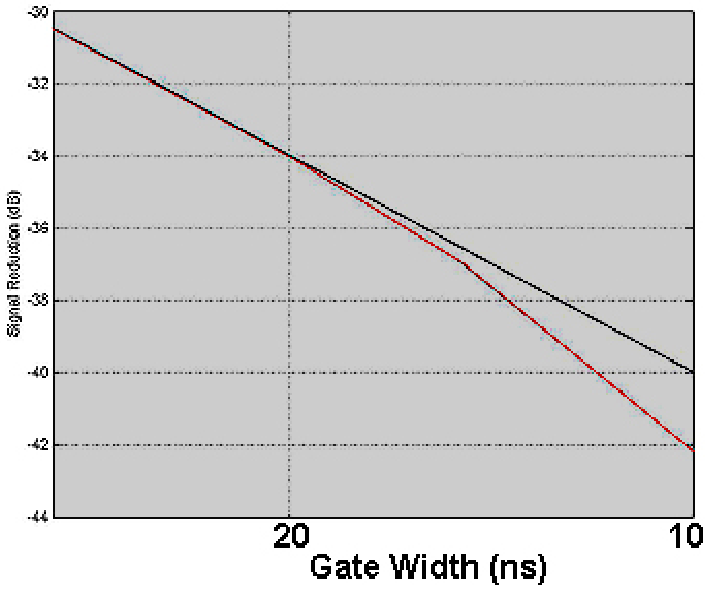
Learn more about these data formats.

Signal Reduction versus Gate Width

Signal Reduction versus Gate Width **PRF = 1 MHz**
The following two figures show the performance of the internal IF gates as the width is narrowed.



The following is a zoomed image of the shaded area (above).



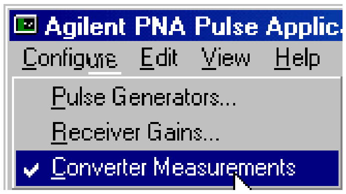
- The straight line shows the theoretical loss in dynamic range due to duty cycle effects when using narrowband detection.
- The curved (red) line shows the actual measured performance of the gates.
- The minimum gate width for <1dB deviation from theoretical is approximately 20ns.

See the specifications for the option H11 and option H08.

Pulsed Frequency Converter Measurements

The Pulsed Application works with both FCA (option 083) and standard Frequency Offset (opt 080) measurements. On the **Configure** menu, check **Converter Measurements**. When checked, this setting prevents the Pulsed Application from overwriting frequency offset values. This may limit the number of **PRF** and **IFBW** solutions that are returned when **Calculate** is pressed on the main Pulsed Application dialog box.

Note: Pulse Profiling can NOT be performed with frequency converter measurements.



Writing your own Pulsed Application

You can use the Pulsed Application or use the example program as a template for making your own Pulsed Application.

The Pulsed Application uses a custom .dll to perform the calculations that are necessary to make pulsed measurements. Use the ConfigNarrowBand3 Method to send and return values to **agilentpnapulsed.dll.** Then use SCPI or COM commands to control the PNA.

Install and Register the Pulsed .dll on your PC

To create your own Pulsed Application, or run the Pulsed Application from a remote PC, you must do the following:

1. Copy the following files from the PNA C:\program files\agilent\network analyzer\ to a directory on your PC.
   - **agilentpnapulsed.dll**
   - **OffsetList.txt**
   - **prfbw.txt**
   - **prfbwmixer.txt**
2. To register the ActiveX DLL in Microsoft Windows Operating System:
   - From a command prompt on your PC, navigate to the directory where you copied the DLL.
   - Type: **regsvr32 agilentpnapulsed.dll** and press **Enter**

For Operating Systems other than Windows, see their associated help files to learn how to register DLL files.

Technical Specifications for the E8356A, E8357A, E8358A

(Rev. 2005-07-12)

This is a complete list of the E8356A, E8357A, and E8358A network analyzer technical specifications.

- To optimize viewing of uncertainty curves, click the Maximize button.
- To view or print the .pdf version of the specifications, visit our web site at http://www.agilent.com/find/pna, and search for "E835xA Specifications"
- The uncertainty curves contained in this document apply only to the setup conditions listed. Please download our free Uncertainty Calculator from http://www.agilent.com/find/na_calculator to generate the curves for your PNA setup. View the equations used to generate the uncertainty curves.

Definitions

Corrected System Performance

- **System Dynamic Range**
- **Type-N Connectors**
- **3.5 mm Connectors**
- **7-16 Connectors**

Uncorrected Instrument Performance

Test Port Output Characteristics (Source)

Test Port and Receiver Input Characteristics

General Information

Front-panel Jumper Specs

Measurement Throughput Summary

Test Set Block Diagram

Test Set with Option 015 Block Diagram

See Specs for other PNA models

Definitions

All specifications and characteristics apply over a 25 °C ±5 °C range (unless otherwise stated) and 90 minutes after the instrument has been turned on.

Specification (spec.): Warranted performance. Specifications include guardbands to account for the expected statistical performance distribution, measurement uncertainties, and changes in performance due to environmental conditions.

Characteristic (char.): A performance parameter that the product is expected to meet before it leaves the factory, but that is not verified in the field and is not covered by the product warranty. A characteristic includes the same guardbands as a specification.

Typical (typ.): Expected performance of an average unit which does not include guardbands. It is not covered by the product warranty.

Nominal (nom.): A general, descriptive term that does not imply a level of performance. It is not covered by the product warranty.

Calibration: The process of measuring known standards to characterize a network analyzer's systematic (repeatable) errors.

Corrected (residual): Indicates performance after error correction (calibration). It is determined by the quality of calibration standards and how well "known" they are, plus system repeatability, stability, and noise.

Uncorrected (raw): Indicates instrument performance without error correction. The uncorrected performance affects the stability of a calibration.

Standard: When referring to the analyzer, this includes no options unless noted otherwise.

Corrected System Performance

The specifications in this section apply for measurements made with the E8356A, E8357A, and E8358A analyzer with the following conditions:

- 10 Hz IF bandwidth
- No averaging applied to data
- Environmental temperature of 25 °C ±5 °C, with < 1 °C deviation from calibration temperature
- Isolation calibration not omitted

Note: The uncertainty curves contained in these specifications apply only to the setup conditions listed. Please download our free Uncertainty Calculator from http://www.agilent.com/find/na_calculator to generate the curves for your PNA setup. View the equations used to generate the uncertainty curves.

- **System Dynamic Range**
- **Type-N Connectors**
- **3.5 mm Connectors**
- **7-16 Connectors**

Table 1. System Dynamic Range

Description	Specification (dB)	Characteristic (dB)
Dynamic range[a] (at test port)		
300 kHz to 25 MHz[b]	125	
25 MHz to 3 GHz[b]	128	
3 GHz to 6 GHz	118	
6 GHz to 9 GHz	113	
Dynamic range[c] (at receiver input)		
300 kHz to 25 MHz[d]		140
25 MHz to 3 GHz[d]		143
3 GHz to 6 GHz		133
6 GHz to 9 GHz		128

a The test port dynamic range is calculated as the difference between the test port rms noise floor and the source maximum output power. The effective dynamic range must take measurement uncertainties and interfering signals into account.

b May be limited to 100 dB at particular frequencies below 750 MHz due to spurious receiver residuals.

c The receiver input dynamic range is calculated as the difference between the receiver rms noise floor and the source maximum output power. The effective dynamic range must take measurement uncertainties and interfering signals into account. This set-up should only be used when the receiver input will never exceed its damage level. When the analyzer is in segment sweep mode, frequency segments can be defined with a higher power level when the extended dynamic range is required (i.e. the portion of the device's response with high insertion loss), and reduced power when receiver damage may occur (i.e. the portion of the devices's response with low insertion loss).

d May be limited to 115 dB at particular frequencies below 750 MHz due to spurious receiver residuals.

Note: Receiver Dynamic Range specifications are not included in this E8356/7/8A document.

Corrected System Performance with Type-N Connectors

- **85032F Calibration Kit**
- **85092C Electronic Calibration Module**

Note: This document provides technical specifications for the following calibration kits only: 85032F, 85092C, 85033E, 85093C and 85038A.

Table 2. Corrected System Performance With Type-N Device Connectors, 85032F Calibration Kit

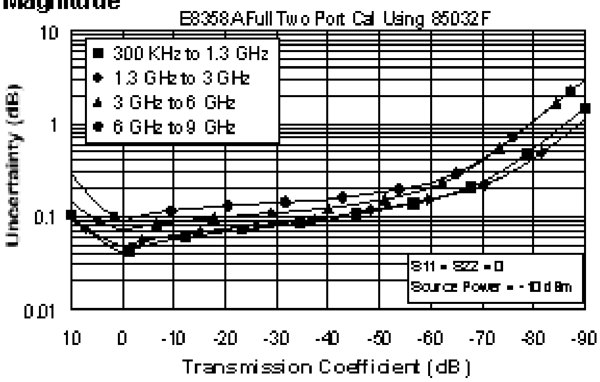
Applies to the E8356A, E8357A, and E8358A analyzer, 85032F (Type-N, 50Ω) calibration kit, N6314A test port cable, and a full 2-port calibration. Also applies to the following conditions:

- IF bandwidth = 10 Hz
- No averaging applied to data
- Environmental temperature 25° ±5 °C, with < 1 °C deviation from calibration temperature
- Isolation calibration not omitted

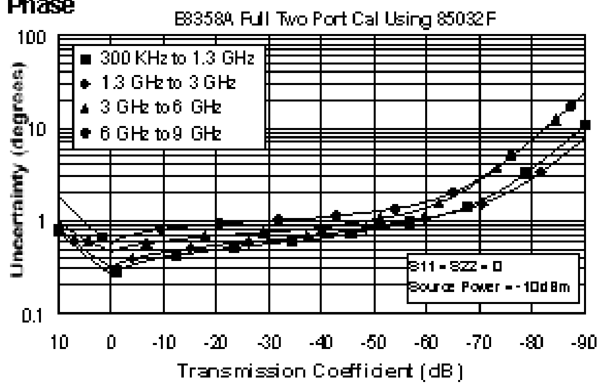
Description	Specification (dB)			
	300 kHz to 1.3 GHz	1.3 GHz to 3 GHz	3 to 6 GHz	6 to 9 GHz
Directivity	49	46	40	38
Source Match	41	40	36	35
Load Match	49	45	39	37
Reflection Tracking	±0.011	±0.021	±0.032	±0.054
Transmission Tracking	±0.011	±0.019	±0.041	±0.051

Transmission Uncertainty (Specifications)

Magnitude

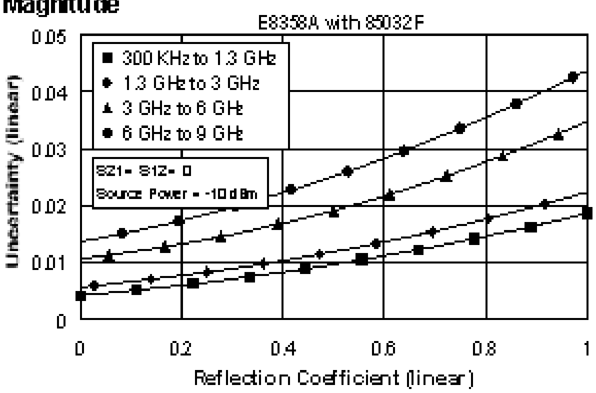


Phase



Reflection Uncertainty (Specifications)

Magnitude



Phase

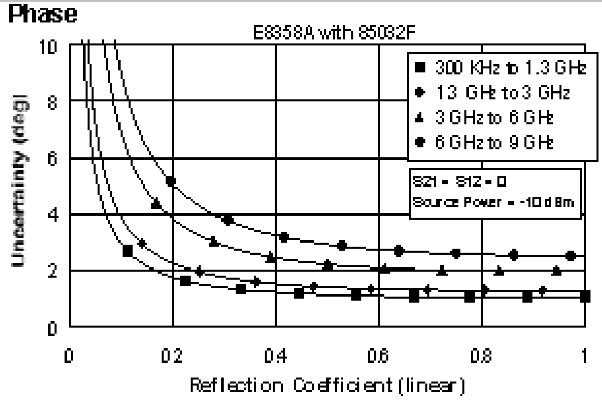


Table 3. Corrected System Performance With Type-N Device Connectors, Option 015 With 85032F Calibration Kit

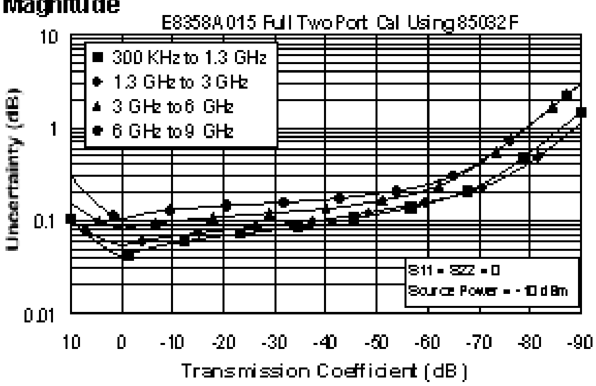
Applies to the E8356A, E8357A, and E8358A analyzer with Option 015, 85032F (Type-N, 50Ω) calibration kit, N6314A test port cable, and a full 2-port calibration. Also applies to the following conditions:

- IF bandwidth = 10 Hz
- No averaging applied to data
- Environmental temperature 25° ±5 °C, with < 1 °C deviation from calibration temperature
- Isolation calibration not omitted

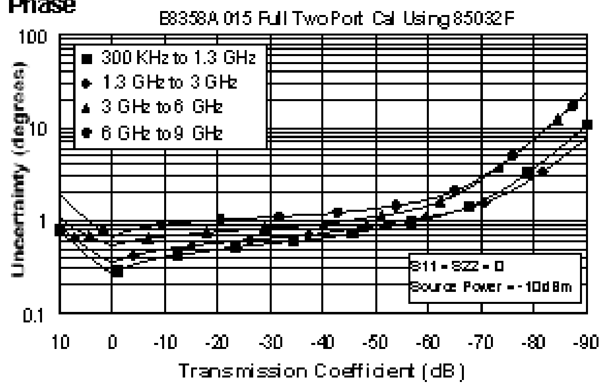
Description	Specification (dB)			
	300 kHz to 1.3 GHz	1.3 GHz to 3 GHz	3 to 6 GHz	6 to 9 GHz
Directivity	49	46	40	38
Source Match	41	40	36	35
Load Match	49	45	39	37
Reflection Tracking	±0.011	±0.021	±0.032	±0.054
Transmission Tracking	±0.011	±0.024	±0.052	±0.065

Transmission Uncertainty (Specifications)

Magnitude

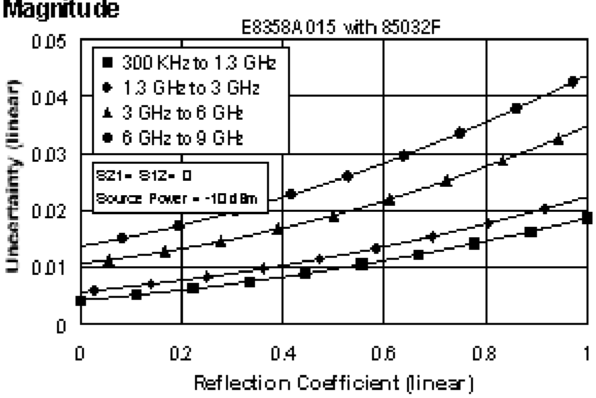


Phase



Reflection Uncertainty (Specifications)

Magnitude



Phase

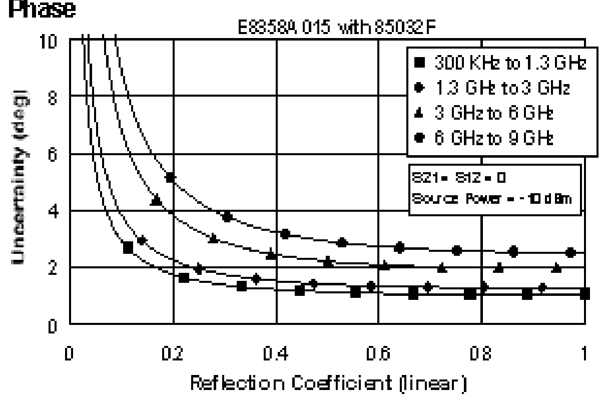


Table 4. Corrected System Performance With Type-N Device Connectors, 85092C Electronic Calibration Module

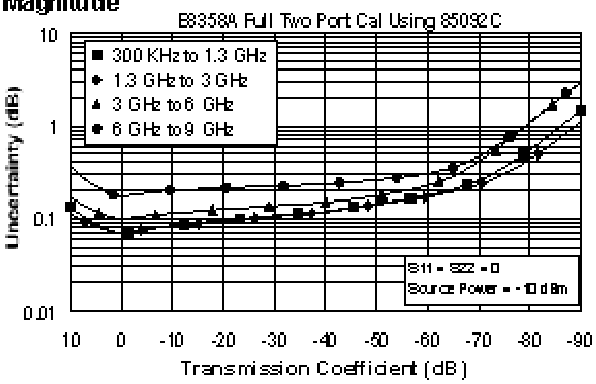
Applies to the E8356A, E8357A, and E8358A analyzer, 85092C (Type-N, 50Ω) electronic calibration (ECal) module, N6314A test port cable, and a full 2-port calibration. Also applies to the following conditions:

- IF bandwidth = 10 Hz
- No averaging applied to data
- Environmental temperature 25° ±5 °C, with < 1 °C deviation from calibration temperature
- Isolation calibration not omitted

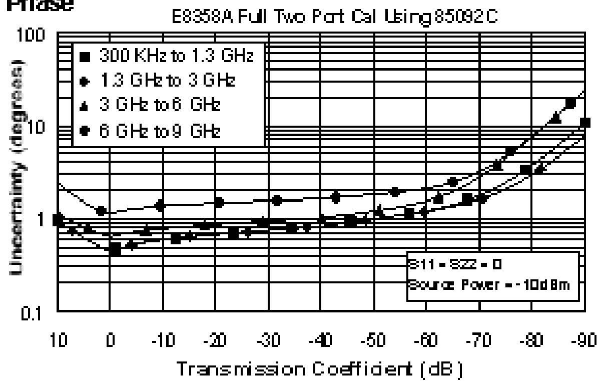
Description	Specification (dB)			
	300 kHz to 1.3 GHz	1.3 GHz to 3 GHz	3 to 6 Hz	6 to 9 GHz[a]
Directivity	52	54	52	47
Source Match	45	44	41	36
Load Match	47	47	44	39
Reflection Tracking	±0.040	±0.040	±0.060	±0.070
Transmission Tracking	±0.039	±0.039	±0.068	±0.135

Transmission Uncertainty (Specifications)

Magnitude

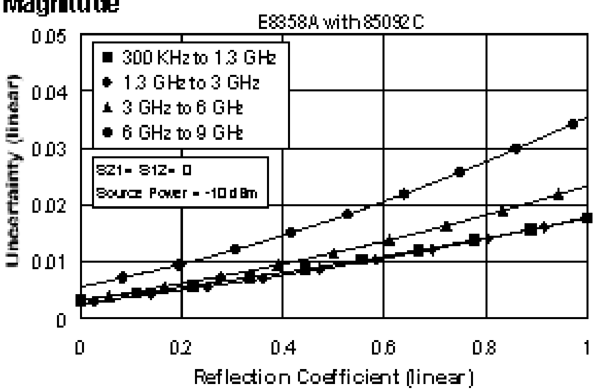


Phase



Reflection Uncertainty (Specifications)

Magnitude



Phase

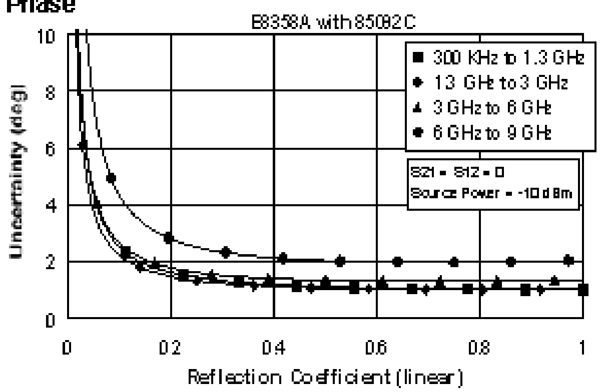


Table 5. Corrected System Performance With Type-N Device Connectors, Option 015 With 85092C Electronic Calibration Module

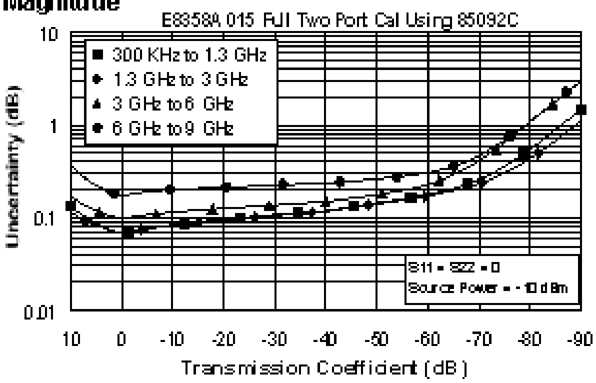
Applies to the E8356A, E8357A, and E8358A analyzer with Option 015, 85092C (Type-N, 50Ω) electronic calibration (ECal) module, N6314A test port cable, and a full 2-port calibration. Also applies to the following conditions:

- IF bandwidth = 10 Hz
- No averaging applied to data
- Environmental temperature 25° ±5 °C, with < 1 °C deviation from calibration temperature
- Isolation calibration not omitted

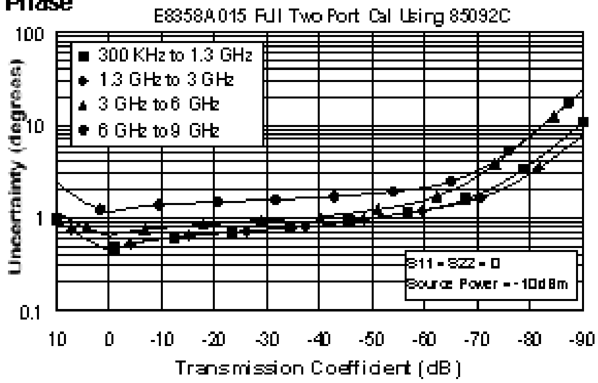
Description	Specification (dB)			
	300 kHz to 1.3 GHz	1.3 GHz to 3 GHz	3 to 6 GHz	6 to 9 GHz[a]
Directivity	52	54	52	47
Source Match	45	44	41	36
Load Match	47	47	44	39
Reflection Tracking	±0.040	±0.040	±0.060	±0.070
Transmission Tracking	±0.039	±0.039	±0.068	±0.135

Transmission Uncertainty (Specifications)

Magnitude

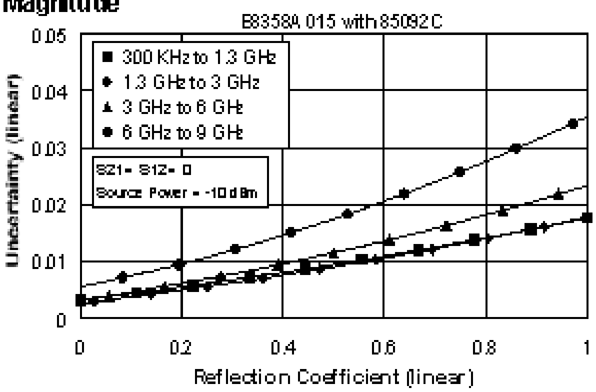


Phase

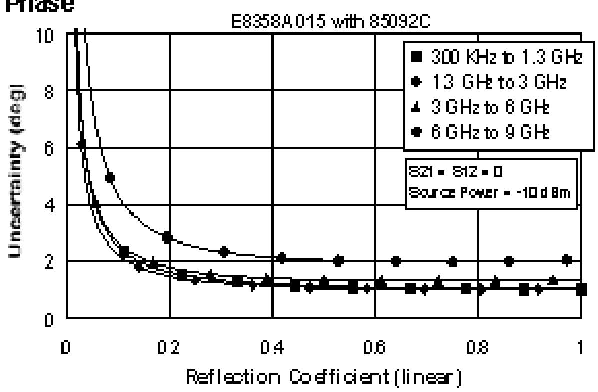


Reflection Uncertainty (Specifications)

Magnitude



Phase



Corrected System Performance with 3.5 mm Connectors

- 85033E Calibration Kit
- Option 015 with 85033E Calibration Kit
- 85093C Electronic Calibration Module

Option 015 with 85093C Electronic Calibration Module

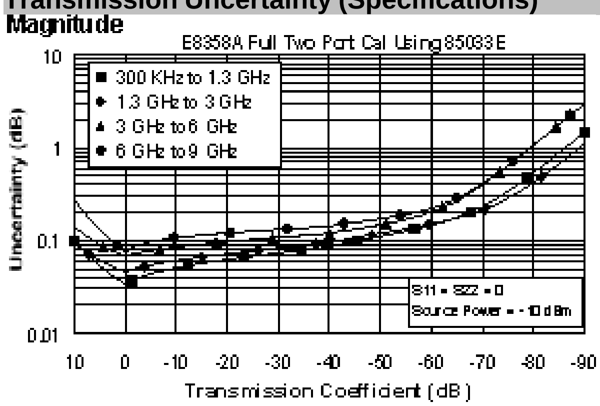
Table 6. Corrected System Performance With 3.5 mm Device Connector Type, 85033E Calibration Kit

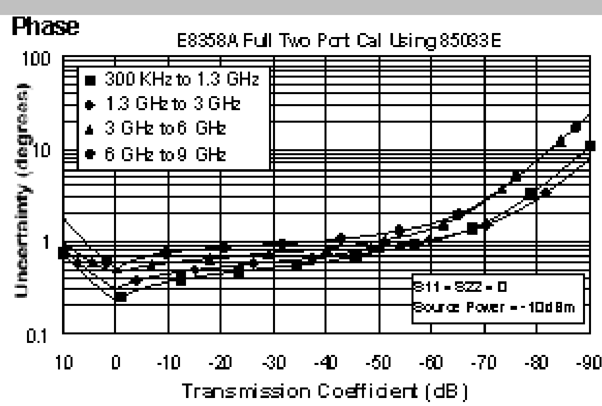
Applies to the E8356A, E8357A, and E8358A analyzer, 85033E (3.5 mm, 50Ω) calibration kit, N6314A test port cable, and a full 2-port calibration. Also applies to the following conditions:

- IF bandwidth = 10 Hz
- No averaging applied to data
- Environmental temperature 25° ±5 °C, with < 1 °C deviation from calibration temperature
- Isolation calibration not omitted

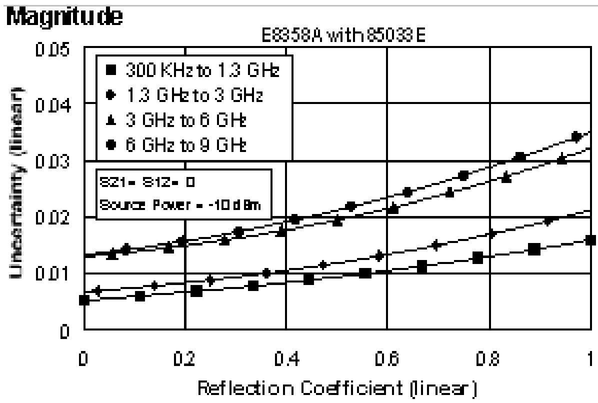
Description	Specification (dB)			
	300 kHz to 1.3 GHz	1.3 GHz to 3 GHz	3 to 6 GHz	6 to 9 GHz
Directivity	46	44	38	38
Source Match	43	40	37	36
Load Match	46	44	38	38
Reflection Tracking	±0.006	±0.007	±0.009	±0.010
Transmission Tracking	±0.011	±0.020	±0.041	±0.047

Transmission Uncertainty (Specifications)





Reflection Uncertainty (Specifications)



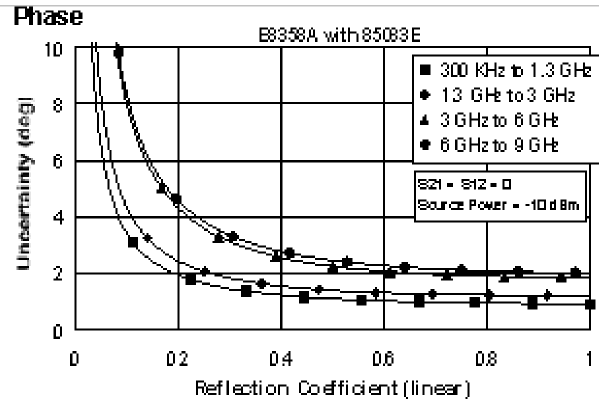


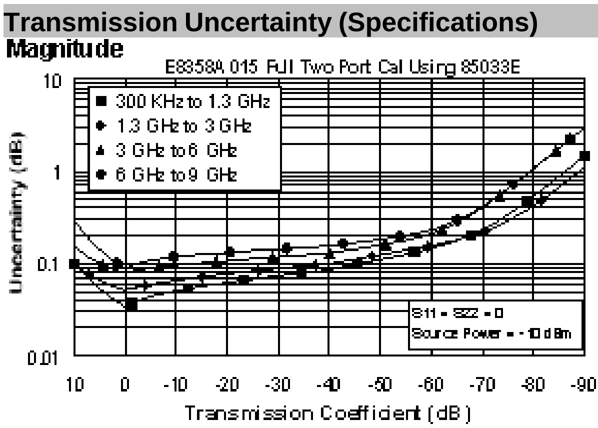
Table 7. Corrected System Performance With 3.5 mm Device Connector Type, Option 015 With 85033E Calibration Kit

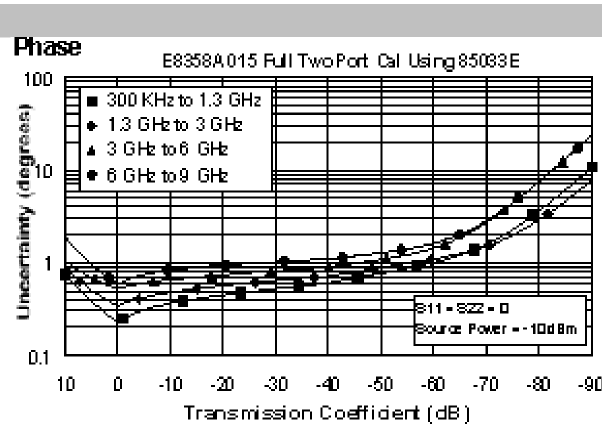
Applies to the E8356A, E8357A, and E8358A analyzer with Option 015, 85033E (3.5 mm, 50Ω) calibration kit, N6314A test port cable, and a full 2-port calibration. Also applies to the following conditions:

- IF bandwidth = 10 Hz
- No averaging applied to data
- Environmental temperature 25° ±5 °C, with < 1 °C deviation from calibration temperature
- Isolation calibration not omitted

Description	Specification (dB)			
	300 kHz to 1.3 GHz	1.3 GHz to 3 GHz	3 to 6 GHz	6 to 9 GHz
Directivity	46	44	38	38
Source Match	43	40	37	36
Load Match	46	44	38	38
Reflection Tracking	±0.006	±0.007	±0.009	±0.010
Transmission Tracking	±0.011	±0.025	±0.052	±0.059

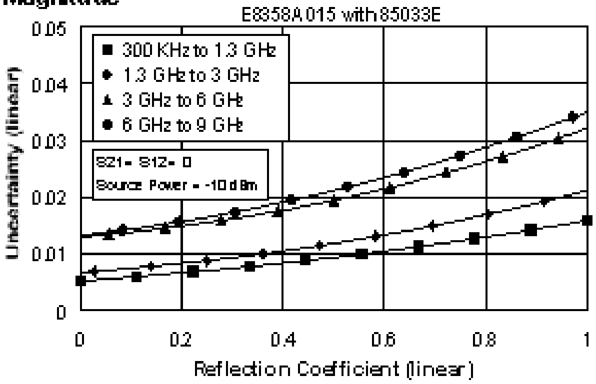
Transmission Uncertainty (Specifications)





Reflection Uncertainty (Specifications)

Magnitude



Phase

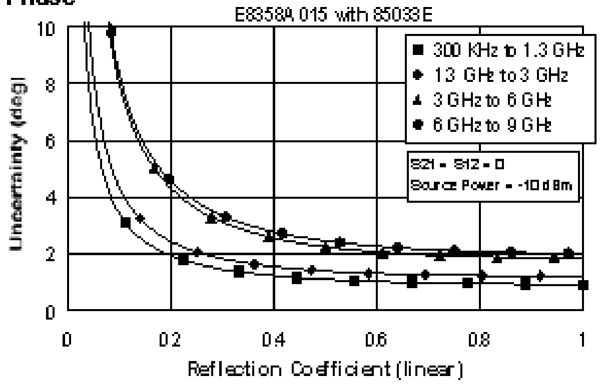


Table 8. Corrected System Performance With 3.5 mm Device Connector Type, 85093C Electronic Calibration Module

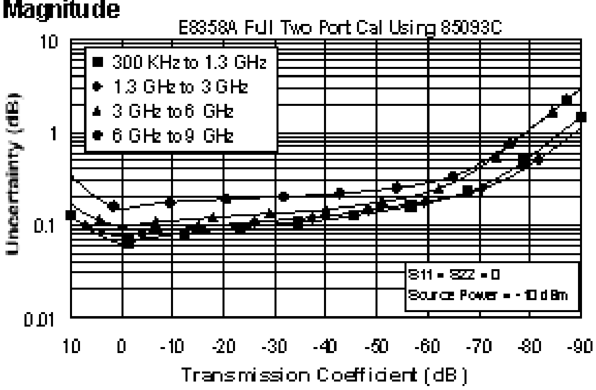
Applies to the E8356A, E8357A, and E8358A analyzer, 85093C (3.5 mm, 50Ω) electronic calibration (ECal) module, N6314A test port cable, and a full 2-port calibration. Also applies to the following conditions:

- IF bandwidth = 10 Hz
- No averaging applied to data
- Environmental temperature 25° ±5 °C, with < 1 °C deviation from calibration temperature
- Isolation calibration not omitted

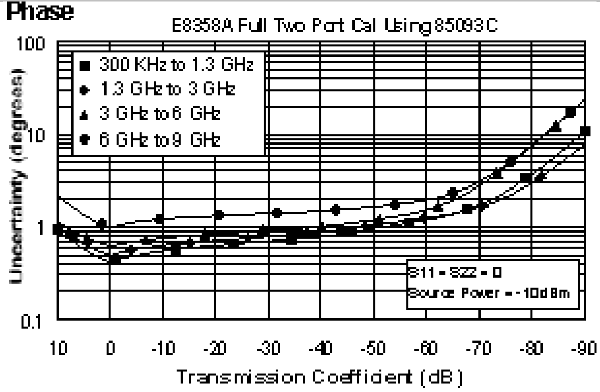
Description	**Specification (dB)**			
	300 kHz to 1.3 GHz	**1.3 GHz to 3 GHz**	**3 to 6 GHz**	**6 to 9 GHz**[a]
Directivity	52	52	51	47
Source Match	44	44	39	34
Load Match	47	47	44	40
Reflection Tracking	±0.030	±0.040	±0.050	±0.070
Transmission Tracking	±0.039	±0.049	±0.068	±0.116

Transmission Uncertainty (Specifications)

Magnitude

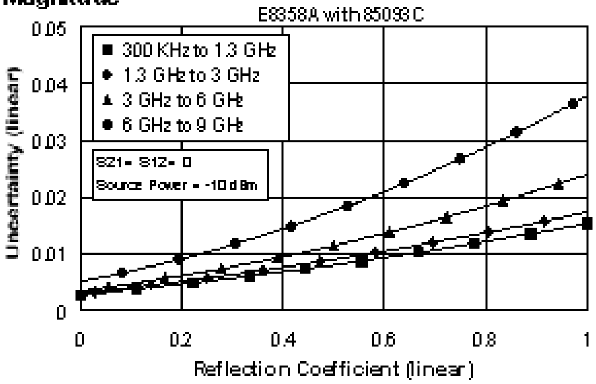


Phase



Reflection Uncertainty (Specifications)

Magnitude



Phase

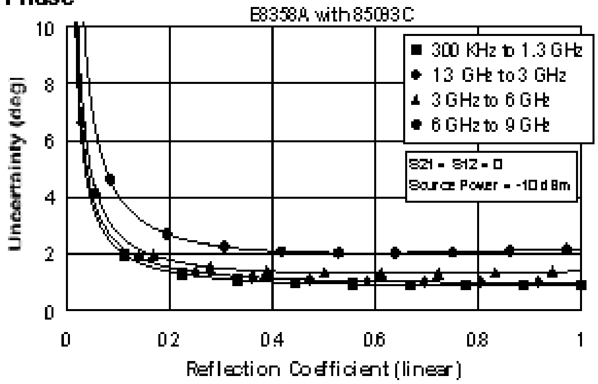


Table 9. Corrected System Performance With 3.5 mm Device Connector Type, Option 015 With 85093C Electronic Calibration Module

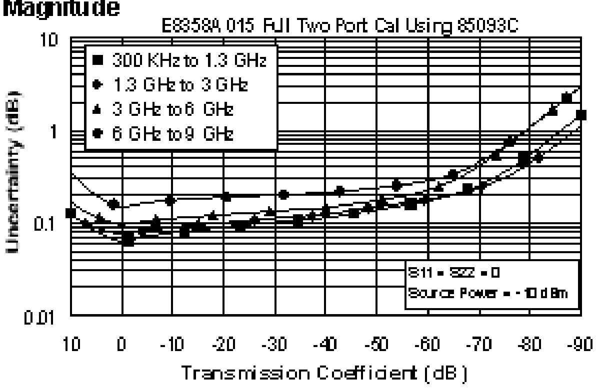
Applies to the E8356A, E8357A, and E8358A analyzer with Option 015, 85093C (3.5 mm, 50Ω) electronic calibration (ECal) module, N6314A test port cable, and a full 2-port calibration. Also applies to the following conditions:

- IF bandwidth = 10 Hz
- No averaging applied to data
- Environmental temperature 25° ±5 °C, with < 1 °C deviation from calibration temperature
- Isolation calibration not omitted

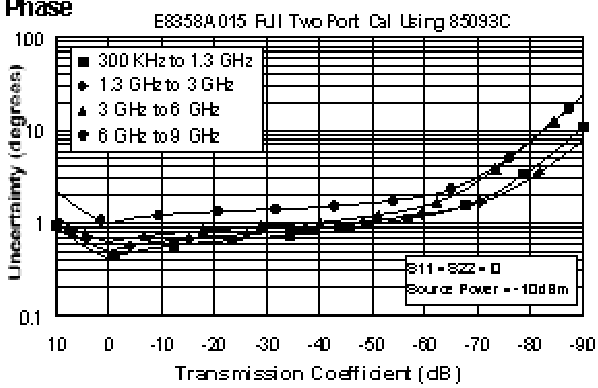
Description	**Specification (dB)**			
	300 kHz to 1.3 GHz	**1.3 GHz to 3 GHz**	**3 to 6 GHz**	**6 to 9 GHz[a]**
Directivity	52	52	51	47
Source Match	44	44	39	34
Load Match	47	47	44	40
Reflection Tracking	±0.030	±0.040	±0.050	±0.070
Transmission Tracking	±0.039	±0.049	±0.068	±0.116

Transmission Uncertainty (Specifications)

Magnitude

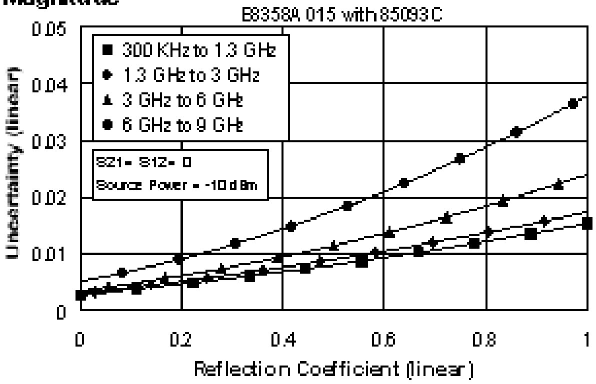


Phase

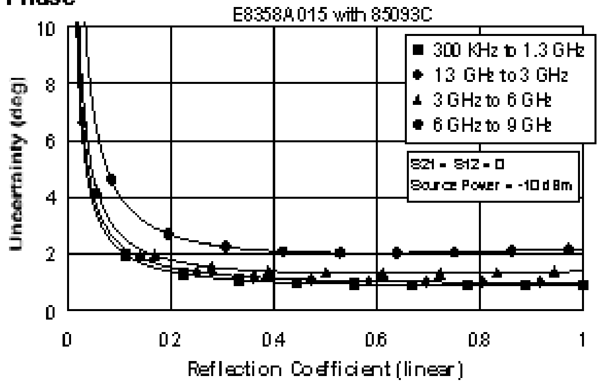


Reflection Uncertainty (Specifications)

Magnitude



Phase



Corrected System Performance with 7-16 Connectors

- **85038A Calibration Kit**
- **Option 015 with 85038A Calibration Kit**

Table 10. Corrected System Performance With 7-16 Device Connector Type, 85038A Calibration Kit

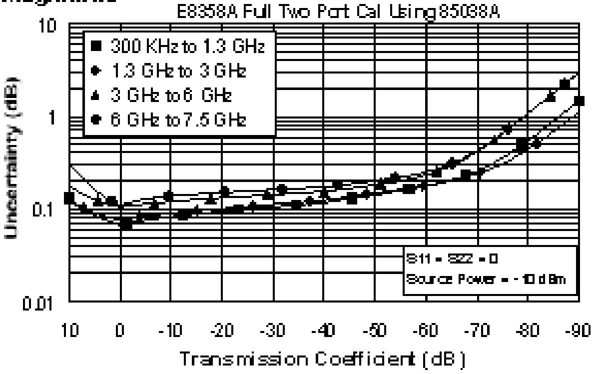
Applies to the E8356A, E8357A, and E8358A analyzer, 85038A (7-16, 50Ω) calibration kit, N6314A test port cable, and a full 2-port calibration. Also applies to the following conditions:

- IF bandwidth = 10 Hz
- No averaging applied to data
- Environmental temperature 25° ±5 °C, with < 1 °C deviation from calibration temperature
- Isolation calibration not omitted

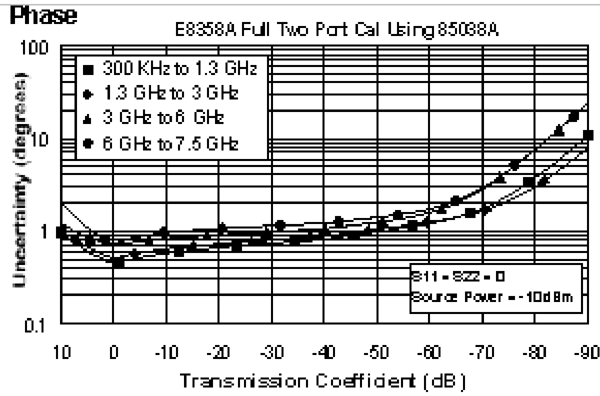
Description	**Specification (dB)**			
	300 kHz to 1.3 GHz	**1.3 GHz to 3 GHz**	**3 to 6 GHz**	**6 to 9 GHza**
Directivity	40	40	36	36
Source Match	37	37	34	34
Load Match	39	39	35	35
Reflection Tracking	±0.089	±0.089	±0.115	±0.115
Transmission Tracking	±0.022	±0.031	±0.059	±0.062

Transmission Uncertainty (Specifications)

Magnitude

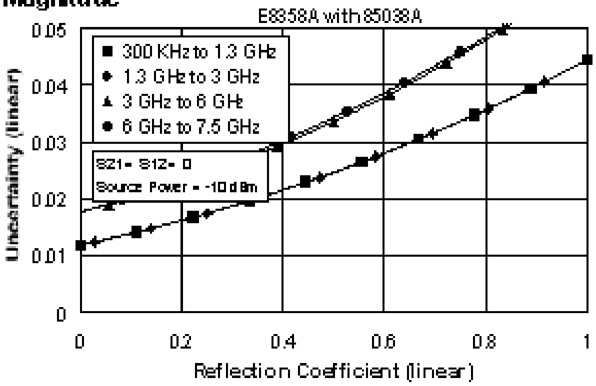


Phase



Reflection Uncertainty (Specifications)

Magnitude



Phase

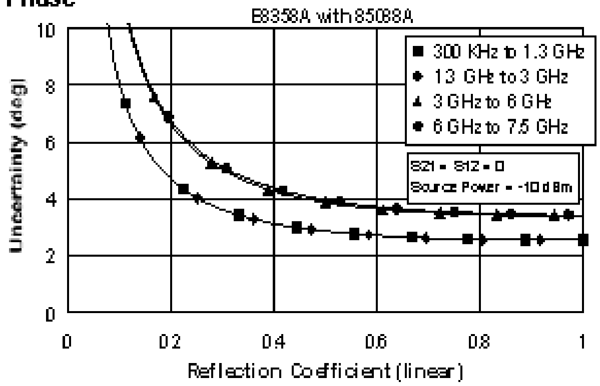


Table 11. Corrected System Performance With 7-16 Device Connector Type, Option 015 With 85038A Calibration Kit

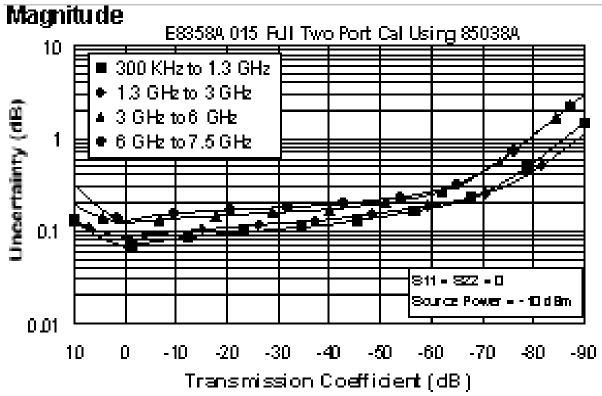
Applies to the E8356A, E8357A, and E8358A analyzer with Option 015, 85038A (7-16, 50Ω) calibration kit, N6314A test port cable, and a full 2-port calibration. Also applies to the following conditions:

- IF bandwidth = 10 Hz
- No averaging applied to data
- Environmental temperature 25° ±5 °C, with < 1 °C deviation from calibration temperature
- Isolation calibration not omitted

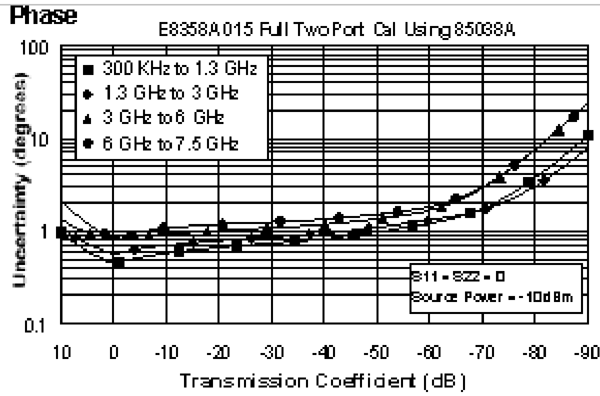
Description	Specification (dB)			
	300 kHz to 1.3 GHz	1.3 GHz to 3 GHz	3 to 6 GHz	6 to 9 GHz[a]
Directivity	40	40	36	36
Source Match	37	37	34	34
Load Match	39	39	35	35
Reflection Tracking	±0.089	±0.089	±0.115	±0.115
Transmission Tracking	±0.022	±0.040	±0.075	±0.080

Transmission Uncertainty (Specifications)

Magnitude

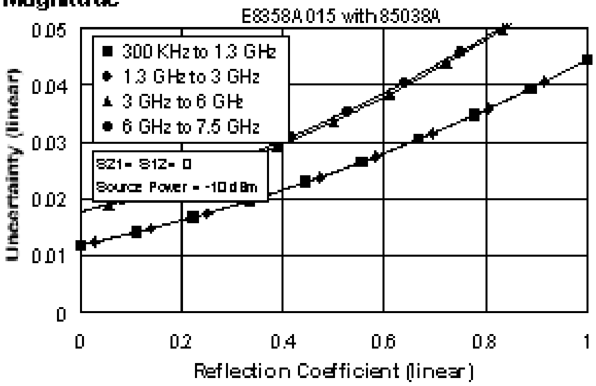


Phase

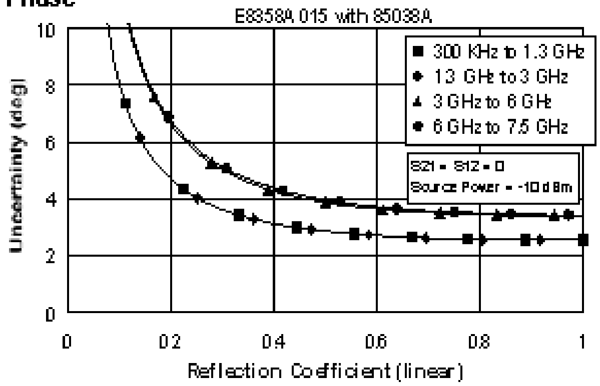


Reflection Uncertainty (Specifications)

Magnitude



Phase



Uncorrected Instrument Performance

Table 12. Uncorrected Instrument Performance

Description	Specification (dB)				
	300 kHz to 1 MHz	1MHz to 1.3 GHz	1.3 GHz to 3 GHz	3 GHz to 6 GHz	6 GHz to 9 GHz
Directivity	30	33	27	20	13
Source Match	20	20	17	15	14
Source Match (Opt. 015)	20	20	15	13	12
Load Match	20	20	17	15	15
Load Match (Opt. 015)	20	20	15	13	13
Reflection Tracking	±1.5	±1.5	±1.5	±2.5	±3.0
Transmission Tracking	±1.5	±1.5	±1.5	±2.5	±3.0

Test Port Output Characteristics (Source)

Output Frequency

- Output Power
- Output Signal Purity

Table 13. Test Port Output Frequency

Description	Specification	Supplemental Information
Range		
E8356A	300 kHz to 3.0 GHz	
E8357A	300 kHz to 6.0 GHz	
E8358A	300 kHz to 9.0 GHz	
Resolution	1 Hz	
Source Stability		±0.05 ppm, -0° to 40 °C, typical ±0.1 ppm/year, typical
CW Accuracy	±1 ppm	

Table 14. Test Port Output Power[a]

Description	Specification	Supplemental Information
Level Accuracy		
		Variation from 0 dBm in power range 0 (step attenuator at 0 dB) ±1.5dB below 10 MHz
300 kHz to 10 MHz	±1.5 dB	
10 MHz to 6 GHz	±1.0 dB	
6 GHz to 9 GHz	±2.0 dB	
Level Linearity		
		Variation from 0 dBm in power range 0
300 kHz to 9 GHz	±0.3 dB	-15 to +5 dBm
300 kHz to 1 MHz	±1.0 dB	+5 to +10 dBm
1 MHz to 6 GHz	±0.5 dB	+5 to +10 dBm
Range[b]		
300 kHz to 6 GHz	-85 to +10 dBm	
6 GHz to 9 GHz	-85 to +5 dBm	

Sweep Range		
300 kHz to 6 GHz	25 dB	
6 GHz to 9 GHz	20 dB	
Level Resolution	0.01 dB	

a Source output performance on port 1 only. Port 2 output performance is typical.

b Power to which the source can be set and phase lock is assured.

Table 15. Test Port Output Signal Purity

Description	**Specification**	**Supplemental Information**
Harmonics (2nd or 3rd)		
at max output power (< 25 MHz)		< -25 dBc, typical
at max output power (25 MHz to 9 GHz)		< -25 dBc, characteristic[a]
at 0 dBm output		< -35 dBc, typical
at -10 dBm output		< -38 dBc, typical, in power range 0 (step attenuator at 0 dB)
Non-harmonic Spurious		
at max output		-30 dBc, typical for offset freq >1kHz
at -10 dBm output		-50 dBc, typical for offset freq >1kHz

a Typical below 25 MHz.

Test Port and Receiver Input Characteristics

- **Input Levels**
- **Trace Noise**
- **Reference Level and Stability**
- **Dynamic Accuracy**
- **Group Delay**

Table 16. Test Port and Receiver Input Levels

Description	**Specification**	**Supplemental Information**
Maximum Test Port Input Level		
Test Ports 1 and 2:		
300 kHz to 25 MHz	+10 dBm	< 0.6 dB compression

25 MHz to 3 GHz	+10 dBm	< 0.4 dB compression
3 GHz to 6 GHz	+10 dBm	< 0.7 dB compression
6 GHz to 9 GHz	+5 dBm	< 0.7 dB compression
Damage Level		
Test Port 1, 2		+30 dBm or ±30 VDC, typ.
R1, R2 IN		+15 dBm or ±5 VDC, typ.
A, B IN (standard)		+15 dBm or ±5 VDC, typ.
A, B IN (Opt. 015)		+15 dBm or ±0 VDC, typ.
Coupler IN (Opt. 015)		+33 dBm or ±0 VDC, typ.
Test Port Noise Floor[a]		
300 kHz to 25 MHz[b]		
10 Hz IF Bandwidth	= -115 dBm	
1 kHz IF Bandwidth	= -95 dBm	
25 MHz to 3 GHz[b]		
10 Hz IF Bandwidth	= -118 dBm	
1 kHz IF Bandwidth	= -98 dBm	
3 GHz to 9 GHz		
10 Hz IF Bandwidth	≤ -108 dBm	
1 kHz IF Bandwidth	≤ -88 dBm	
Receiver Noise Floor[a]		
300 kHz to 25 MHz[c]		
10 Hz IF Bandwidth	≤ -130 dBm	
1 kHz IF Bandwidth	≤ -110 dBm	
25 MHz to 3 GHz[c]		
10 Hz IF Bandwidth	≤ -133 dBm	
1 kHz IF Bandwidth	≤ -113 dBm	
6 GHz to 9 GHz		
10 Hz IF Bandwidth	≤ -123 dBm	

1 kHz IF Bandwidth	≤ -103 dBm	
Crosstalk		
		Between test ports 1 and 2, with short circuits at both ports
300 kHz to 1 MHz	<-120 dB	
1 MHz to 25 MHz	<-125 dB	
25 MHz to 3 GHz	<-128 dB	
3 GHz to 6 GHz	<-118 dB	
6 GHz to 9 GHz	<-113 dB	
Maximum Receiver Input Level (A, B, R1, R2)		
300 kHz to 6 GHz		-6 dBm, typical
6 GHz to 9 GHz		-11 dBm, typical
Reference Input Level (R1, R2)[d]		
300 kHz to 9 GHz		-10 to -35 dBm, typical
Maximum Coupler Input Level (Opt 015)		
300 kHz to 9 GHz		+33 dBm, typical

a Total average (RMS) noise power calculated as the mean value of a linear magnitude trace expressed in dBm.
b May be limited to -90 dBm at particular frequencies below 750 MHz due to spurious receiver residuals.
c May be limited to -105 dBm at particular frequencies below 750 MHz due to spurious receiver residuals.

d Input level to maintain phase lock.

Table 17. Test Port Input (Trace Noise)

Description	Specification	Supplemental Information
Trace Noise[a] Magnitude		
1 kHz IF Bandwidth	< 0.002 dB rms	
10 kHz IF Bandwidth	< 0.005 dB rms	
Trace Noise[a] Phase		
1 kHz IF Bandwidth	< 0.010° rms	
10 kHz IF Bandwidth	< 0.035° rms	

a Trace noise is defined as a ratio measurement of a through or a full reflection, with the source set to 0 dBm.

Table 18. Test Port Input (Reference Level and Stability)

Description	Specification	Supplemental Information
Reference Level Magnitude		
Range	±200 dB	
Resolution	0.001 dB	
Reference Level Phase		
Range	±500°	
Resolution	0.01°	
Stability Magnitude[a]		
300 kHz to 3 GHz		0.02 dB/°C, typical
3 GHz to 6 GHz		0.04 dB/°C, typical
6 GHz to 9 GHz		0.06 dB/°C, typical
Stability Phase[a]		
300 kHz to 3 GHz		0.2°/°C, typical
3 GHz to 6 GHz		0.3°/°C, typical
6 GHz to 9 GHz		0.6°/°C, typical

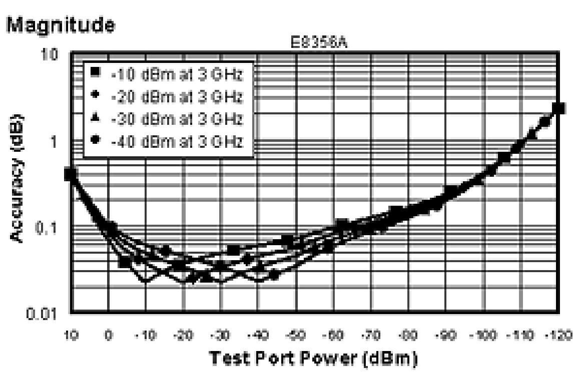
a Stability is defined as a ratio measurement at the test port.

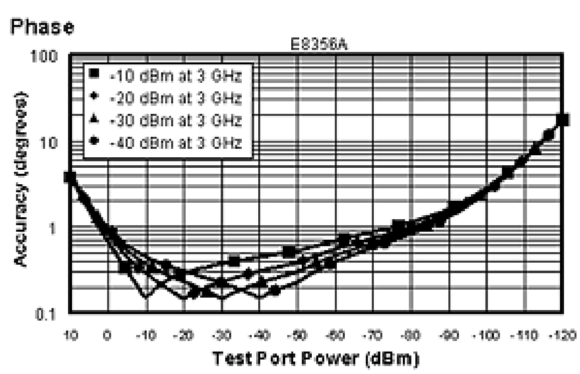
Table 19. Test Port Input (Dynamic Accuracy specification[a])

Accuracy of the test port input power reading is relative to the reference input power level. Applies to input ports 1 and 2 with the following conditions:

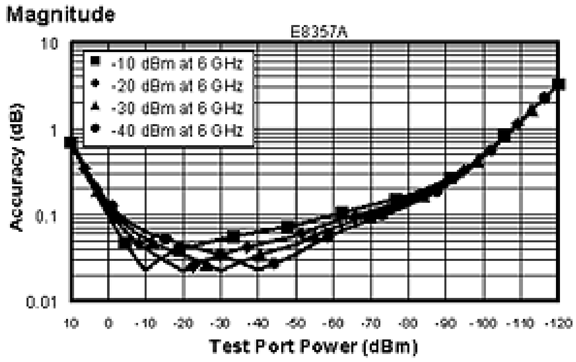
- IF bandwidth = 10 Hz
- Environmental temperature 25° ±5 °C, with < 1 °C deviation from calibration temperature

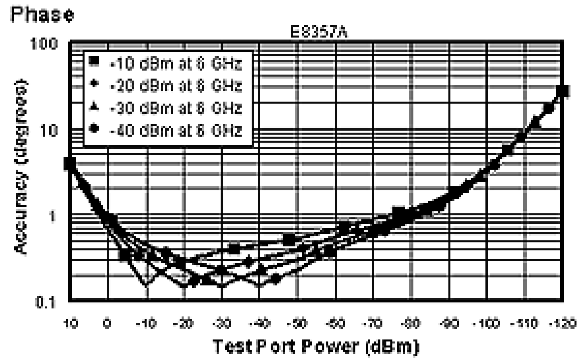
300 kHz to 3 GHz	300 kHz to 3 GHz



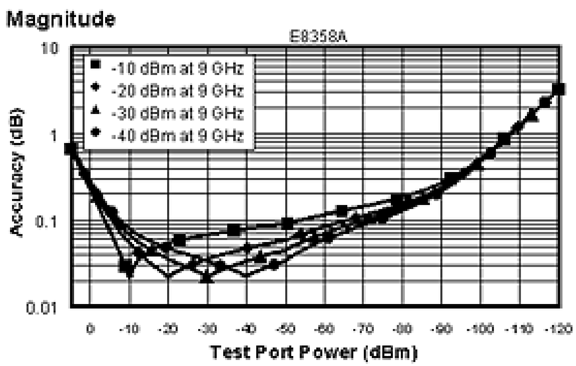


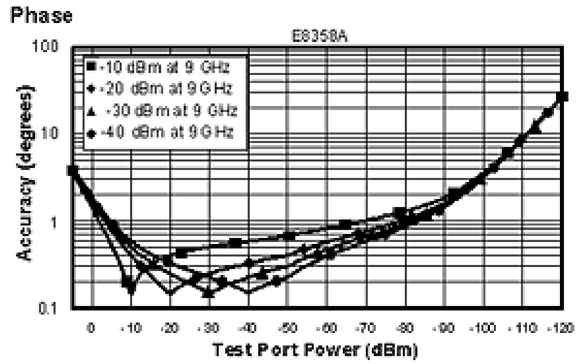
300 kHz to 6 GHz	300 kHz to 6 GHz





300 kHz to 9 GHz	300 kHz to 9 GHz



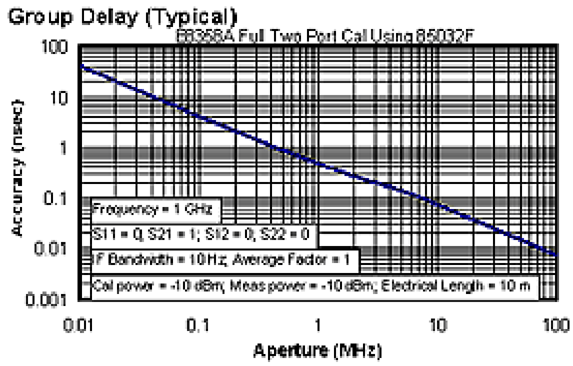


a Dynamic accuracy is verified with the following measurements:

- compression over frequency
- IF linearity at a single frequency of 1.195 GHz and a reference level of -20 dBm

Table 20. Test Port Input (Group Delay)[a]

Description	Specification	Supplemental Information
Aperture (selectable)	(frequency span)/(number of points - 1)	
Maximum Aperture	20% of frequency span	
Range	0.5 x (1/minimum aperture)	
Maximum Delay		Limited to measuring no more than 180° of phase change within the minimum aperture.)
Accuracy		See graph below. Char.

The following graph shows group delay accuracy with type-N full 2-port calibration and a 10 Hz IF bandwidth. Insertion loss is assumed to be < 2 dB and electrical length to be ten meters. 
In general, the following formula can be used to determine the accuracy, in seconds, of specific group delay measurement: ±Phase Accuracy (deg)/[360 × Aperture (Hz)] Depending on the aperture and device length, the phase accuracy used is either incremental phase accuracy or worst case phase accuracy.

a Group delay is computed by measuring the phase change within a specified frequency step (determined by the frequency span and the number of points per sweep).

General Information

- **System Bandwidths**
- **Front Panel**
- **Rear Panel**
- **Environment and Dimensions**

Table 21. System Bandwidths

Description	**Specification**	**Supplemental Information**
IF Bandwidth Settings		
Range		1 Hz to 40 kHz in a 1, 2, 3, 5, 7,10 sequence up to 30 kHz, 35 kHz, 40kHz, nominal

Table 22. Front Panel Information

Description	Supplemental Information
RF Connectors	
Type	Type-N, female; 50 Ω, nominal
Center Pin Protrusion	0.204 to 0.207 in., characteristic
Probe Power	
Connector	3-pin connector, male
Positive Supply	+15 VDC ±2%, 400 mA, max, characteristic
Negative Supply	-12.6 VDC ±5%, 300 mA, max, characteristic
Display	
Size	21.3 cm (8.4 in) diagonal color active matrix LCD; 640 (horizontal) X 480 (vertical) resolution
Refresh Rate	Vertical 59.83 Hz; Horizontal 31.41 Hz
Display Range	
Magnitude	±200 dB (at 20 dB/div), max
Phase	±180°, max
Polar	10 pUnits, min 1000 Units, max
Display Resolution	
Magnitude	0.001 dB/div, min
Phase	0.01°/div, min
Marker Resolution	
Magnitude	0.001 dB, min
Phase	0.01°, min
Polar	0.01 mUnit, min; 0.01°, min

Table 23. Rear Panel Information

Description	Supplemental Information
Test Port Bias Input	
Connector	BNC, female
Maximum Voltage	±30 VDC, typical
Maximum Current (no degradation in RF specifications)	±200 mA, typical

Maximum Current	±1 A, typical
Bias Input Connectors	
Bias current	500 mA, maximum
Bias voltage	40 Volts, maximum
10 MHz Reference In	
Connector	BNC, female
Input Frequency	10 MHz ± 1 ppm, typical
Input Level	-15 dBm to +20 dBm, typical
Input Impedance	200 Ω, nom.
10 MHz Reference Out	
Connector	BNC, female
Output Frequency	10 MHz ± 1 ppm, typical
Signal Type	Sine Wave, typical
Output Level	+10 dBm ± 4 dB into 50 Ω, typical
Output Impedance	50 Ω, nominal
Harmonics	<-40 dBc, typical
VGA Video Output	
Connector	15-pin mini D-Sub; Drives VGA compatible monitors
Devices Supported:	Resolutions:
Flat Panel (TFT)	1024 X 768, 800 X 600, 640 X 480
Flat Panel (DSTN)	800 X 600, 640 X 480
CRT Monitor	1280 X 1024, 1024 X 768, 800 X 600, 640 X 480
	Simultaneous operation of the internal and external displays is allowed, but with 640 X 480 resolution only. If you change resolution, you can only view the external display (internal display will "white out").
Test Set IO	25-pin D-Sub connector, available for external test set control
Aux IO	25-pin D-Sub connector, male, analog and digital IO
Handler IO	36-pin IDC D-ribbon socket connector; all input/output signals are default set to negative logic; can be reset to positive logic via GPIB command
GPIB	24-pin D-sub (Type D-24), female; compatible with IEEE-488.
Parallel Port (LPT1)	25-pin D-Sub connector, female; provides connection to printers or any other parallel port peripheral

Serial Port (COM 1)	9-pin D-Sub, male; compatible with RS-232
USB Port	
	Universal Serial Bus jack, Type A configuration (4 contacts inline, contact 1 on left); female
Contact 1	Vcc: 4.75 to 5.25 VDC, 500 mA, maximum
Contact 2	-Data
Contact 3	+Data
Contact 4	Ground
LAN	10/100BaseT Ethernet; 8-pin configuration; auto selects between the two data rates
Line Power (A third-wire ground is required.)	
Frequency at 110/115 V	50/60/400 Hz
Voltage	120/240 VAC (Power supply is auto switching.)
Maximum Watts	500 Watts

Note: Option H08 and Option H11 specifications are not provided in this E8356/7/8A specifications document.

Table 24. Rear Panel Information (continued)

Description	Supplemental Information
External AM Input	
Description	Input provides low-frequency AM modulation to test port output signal, or shifts the test port output. Zero volts input gives the power level set by the instrument, a positive voltage gives a higher level, and a negative voltage gives a lower level.
Connector	BNC, female
Input Sensitivity	8 dB/V, typical
Bandwidth	1 kHz, typical
Input Impedance	1 kΩ, typical
External Detector Input	
Description	Input from an external, negative polarity diode detector provides ALC for a test port remote from instrument's front panel
Connector	BNC, female
Input Sensitivity	-500 mV yields approximately -3 dBm at detector's input, typical
Bandwidth	50 kHz, typical
Input Impedance	1 kΩ, nominal

Table 25. Analyzer Environment and Dimensions

Description	Supplemental Information		
General Environmental			
RFI/EMI Susceptibility	Defined by CISPR Pub. 11, Group 1, Class A, and IEC 50082-1		
ESD	Minimize using static-safe work procedures and an antistatic bench mat		
Dust	Minimize for optimum reliability		
Operating Environment			
Temperature	0 °C to +40 °C Instrument powers up, phase locks, and displays no error messages within this temperature range.		
Error-Corrected Temperature Range	25°C ± 5°C with less than 1°C deviation from calibration temp.		
Humidity	5% to 95% at +40 °C		
Altitude	0 to 4500 m (14,760 ft.)		
Non-Operating Storage Environment			
Temperature	-40 °C to +70 °C		
Humidity	0% to 90% at +65 °C (non-condensing)		
Altitude	0 to 15,240 m (50,000 ft.)		
Cabinet Dimensions			
	Height	**Width**	**Depth**
Excluding front and rear panel hardware and feet	223 mm 8.75 in	426 mm 16.75 in	427 mm 16.8 in
As shipped - includes front panel connectors, rear panel bumpers, and feet.	235 mm 9.25 in	435 mm 17.10 in	470 mm 18.5 in
As shipped plus handles	235 mm 9.5 in	458 mm 18 in	501 mm 19.70 in
As shipped plus rack-mount flanges	235 mm 9.25 in	483 mm 19 in	470 mm 18.5 in
As shipped plus handles and flanges	235 mm 9.25 in	483 mm 19 in	501 mm 19.70 in
Weight			
Net	24 kg (54 lb), nominal		
Shipping	32 kg (70 lb), nominal		

Note: "Misc. Information" specifications are not included in this E8356/7/8A document.

Measurement Throughput Summary

- **Typical Cycle Time for Measurement Completion**
- **Cycle Time with Changes in IF Bandwidth**
- **Cycle Time with Changes in Number of Points**
- **Data Transfer Time**
- **Recall and Sweep Speed**

Table 26. Typical Cycle Time[a,b] (ms)

	Number of Points			
	101	**201**	**401**	**1601**
Start 1.8 GHz, Stop 2 GHz, 35 kHz IF bandwidth				
Uncorrected, 1-port cal	9	12	18	54
2-Port cal	22	29	42	117
Start 300 kHz, Stop 3 GHz, 35 kHz IF bandwidth				
Uncorrected, 1-port cal	39	47	56	96
2-Port cal	80	101	121	204
Start 300 kHz, Stop 9 GHz, 35 kHz IF bandwidth				
Uncorrected, 1-port cal	51	57	64	103
2-Port cal	112	124	138	220

a Typical performance.
b Includes sweep time, retrace time and band-crossing time. Analyzer display turned off with DISPLAY:ENABLE OFF. Add 21 ms for display on. Data for one trace (S11) measurement.

Table 27. Cycle Time vs. IF Bandwidth[a]

Applies to the Preset condition (201 points, correction off) except for the following changes:

- CF = 1 GHz
- Span = 100 MHz
- Display off (add 21 ms for display on)

IF Bandwidth (Hz)	Cycle Time (ms)[b]
40,000	8
35,000	9
30,000	11
20,000	13
10,000	28
7000	36
5000	48
3000	72
1000	196
300	620
100	1875
30	8062
10	17877

a Typical performance.
b Cycle time includes sweep and retrace time.

Table 28. Cycle Time vs. Number of Points[a]

Applies to the Preset condition (35 kHz IF bandwidth, correction off) except for the following changes:

- CF = 1 GHz
- Span = 100 MHz
- Display off (add 21 ms for display on)

Number of Points	Cycle Time (ms)[b]
3	4
11	4
51	5
101	6
201	9
401	16
801	29
1601	52

a Typical performance.
b Cycle time includes sweep and retrace time.

Table 29. Data Transfer Time[a] (ms)

	Number of Points			
	51	**201**	**401**	**1601**
SCPI over GPIB (program executed on external PC)[b]				
32-bit floating point	3	7	12	43
64-bit floating point	4	12	22	84
ASCII	7	64	124	489
SCPI over 100 Mbit/s LAN (program executed on external PC)[b]				
32-bit floating point	1	1	1	1
64-bit floating point	1	1	1	2
ASCII	5	15	26	96
SCPI (program executed in the analyzer)[d]				
32-bit floating point	1	1	2	3
64-bit floating point	1	2	2	4
ASCII	8	29	56	222
COM (program executed in the analyzer)[e]				
32-bit floating point	1	1	1	1
Variant type	1	1	2	6
DCOM over 100 Mbit/s LAN (program executed on external PC)[f]				
32-bit floating point[g]	1	1	1	2
Variant type[h]	1	3	6	19

a Typical performance of unit with new 500 MHz Pentium III Processor.

b Measured using a VEE 5.0 program running on a 600 MHz HP Kayak, National InstrumentsTM GPIB card. Transferred complex S11 data , using "CALC:DATA?SDATA".

c Measured using a VEE 5.0 program running on a 600 MHz HP Kayak. Transferred complex S11 data, using "CALC:DATA?SDATA". Speed dependent on LAN traffic, if connected to network.

d Measured using a VEE 5.0 program running inside PNA Series Analyzer. Transferred complex S11 data, using "CALC:DATA?SDATA".

e Measured using a Visual Basic 6.0 program running inside PNA Series Analyzer. Transferred complex S11 data.

f Measured using a Visual Basic 6.0 program running on a 600 MHz HP Kayak. Transferred complex S11 data. Speed dependent on LAN traffic, if connected to network.

g Used IArray Transfer.getComplex method for 32-bit floating point.

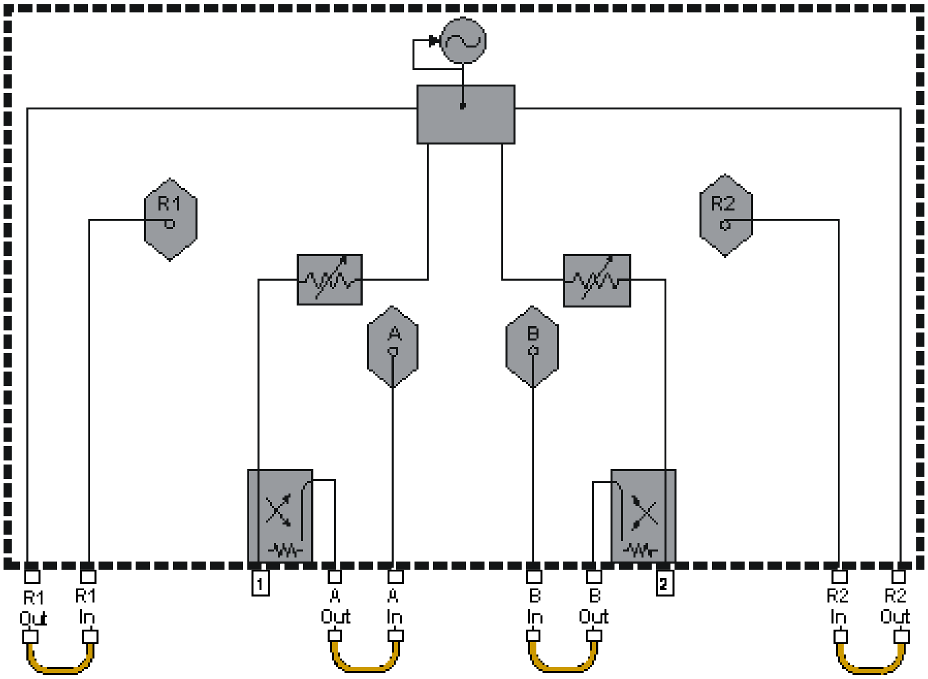
h Used meas.getData method for Variant data type.

Table 30. Recall and Sweep Speed[a]

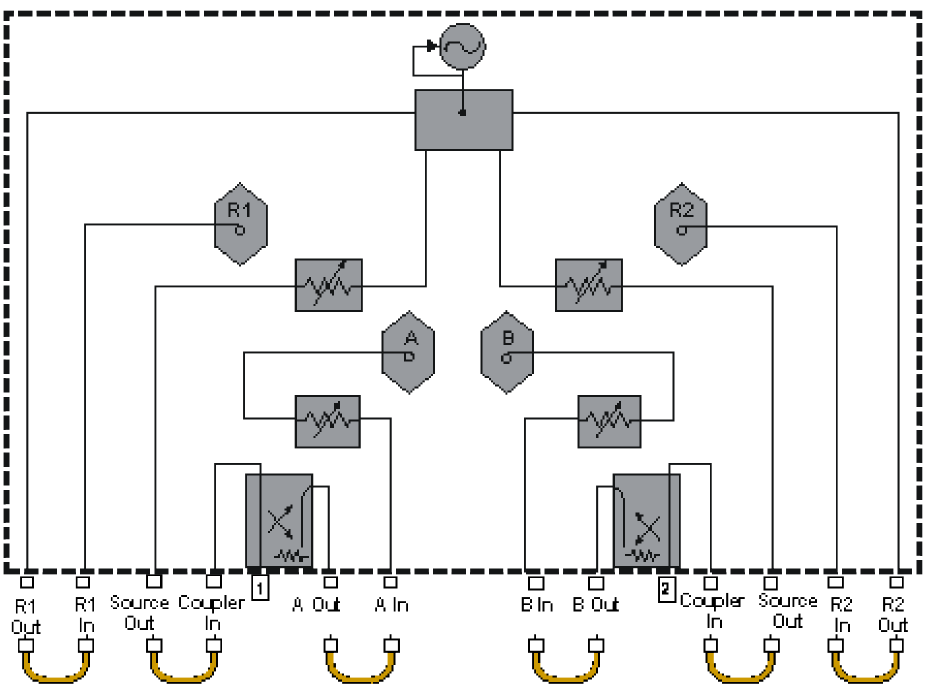
Operations	Number of Window(s)	Number of Trace(s)	Recall Time (ms)
Recall	1	1	49
Recall and Sweep	1	1	59
Recall	1	2	82
Recall and Sweep	1	2	96
Recall	1	4	159
Recall and Sweep	1	4	203
Recall	2	2	93
Recall and Sweep	2	2	115
Recall	3	4	158
Recall and Sweep	3	4	218
Recall	4	4	187
Recall and Sweep	4	4	247
Recall	4	8	340
Recall and Sweep	4	8	507

a CF= 177 MHz, Span = 200 MHz, 201 points, 35 kHz IF BW

E8356A, E8357A, and E8358A – Simplified Test Set Block Diagram



E8356A, E8357A, and E8358A with Option 015 – Simplified Test Set Block Diagram



Technical Specifications for the E8801A, E8802A, E8803A

(Rev. 2005-07-12)

This is a complete list of the E8801A, E8802A, and E8803A network analyzer technical specifications.

- To optimize viewing of uncertainty curves, click the Maximize button.
- To view or print the .pdf version of the specifications, visit our web site at http://www.agilent.com/find/pna, and search for "E880xA Specifications"
- The uncertainty curves contained in this document apply only to the setup conditions listed. Please download our free Uncertainty Calculator from http://www.agilent.com/find/na_calculator to generate the curves for your PNA setup. View the equations used to generate the uncertainty curves.

Definitions

Corrected System Performance

- **System Dynamic Range**
- **Type-N Connectors**
- **3.5 mm Connectors**
- **7-16 Connectors**

Uncorrected Instrument Performance

Test Port Output Characteristics (Source)

Test Port and Receiver Input Characteristics

General Information

Front-panel Jumper Specs (Option 014 only)

Measurement Throughput Summary

Test Set Block Diagram

Test Set with Option 015 Block Diagram

See Specs for other PNA models

Definitions

All specifications and characteristics apply over a 25 °C ±5 °C range (unless otherwise stated) and 90 minutes after the instrument has been turned on.

Specification (spec.): Warranted performance. Specifications include guardbands to account for the expected statistical performance distribution, measurement uncertainties, and changes in performance due to environmental conditions.

Characteristic (char.): A performance parameter that the product is expected to meet before it leaves the factory, but that is not verified in the field and is not covered by the product warranty. A characteristic includes the same guardbands as a specification.

Typical (typ.): Expected performance of an average unit which does not include guardbands. It is not covered by the product warranty.

Nominal (nom.): A general, descriptive term that does not imply a level of performance. It is not covered by the product warranty.

Calibration: The process of measuring known standards to characterize a network analyzer's systematic (repeatable) errors.

Corrected (residual): Indicates performance after error correction (calibration). It is determined by the quality of calibration standards and how well "known" they are, plus system repeatability, stability, and noise.

Uncorrected (raw): Indicates instrument performance without error correction. The uncorrected performance affects the stability of a calibration.

Standard: When referring to the analyzer, this includes no options unless noted otherwise.

Corrected System Performance

The specifications in this section apply for measurements made with the E8801A, E8802A, and E8803A analyzer with the following conditions:

- 10 Hz IF bandwidth
- No averaging applied to data
- Environmental temperature of 25 °C ±5 °C, with < 1 °C deviation from calibration temperature
- Isolation calibration not omitted

Note: The uncertainty curves contained in these specifications apply only to the setup conditions listed. Please download our free Uncertainty Calculator from http://www.agilent.com/find/na_calculator to generate the curves for your PNA setup. View the equations used to generate the uncertainty curves.

- **System Dynamic Range**
- **Type-N Connectors**
- **3.5 mm Connectors**
- **7-16 Connectors**

Table 1. System Dynamic Range

Description	Specification (dB)	Characteristic (dB)
Dynamic range[a] (at test port)		
300 kHz to 25 MHz[b]	125	
25 MHz to 3 GHz[b]	128	
3 GHz to 6 GHz	118	
6 GHz to 9 GHz	115	
Dynamic range[c] (at receiver input)		
300 kHz to 25 MHz[d]		140
25 MHz to 3 GHz[d]		143
3 GHz to 6 GHz		133
6 GHz to 9 GHz		130

a The test port dynamic range is calculated as the difference between the test port rms noise floor and the source maximum output power. The effective dynamic range must take measurement uncertainties and interfering signals into account.
b May be limited to 100 dB at particular frequencies below 750 MHz due to spurious receiver residuals.
c The receiver input dynamic range is calculated as the difference between the receiver rms noise floor and the source maximum output power. The effective dynamic range must take measurement uncertainties and interfering signals into account. This set-up should only be used when the receiver input will never exceed its damage level. When the analyzer is in segment sweep mode, frequency segments can be defined with a higher power level when the extended dynamic range is required (i.e. the portion of the device's response with high insertion loss), and reduced power when receiver damage may occur (i.e. the portion of the devices's response with low insertion loss).
d May be limited to 115 dB at particular frequencies below 750 MHz due to spurious receiver residuals.

Note: Receiver Dynamic Range specifications are not included in this E8801/2/3A document.

Corrected System Performance with Type-N Connectors

- **85032F Calibration Kit**
- **85092C Electronic Calibration Module**

Note: This document provides technical specifications for the following calibration kits only: 85032F, 85092C, 85033E, 85093C and 85038A.

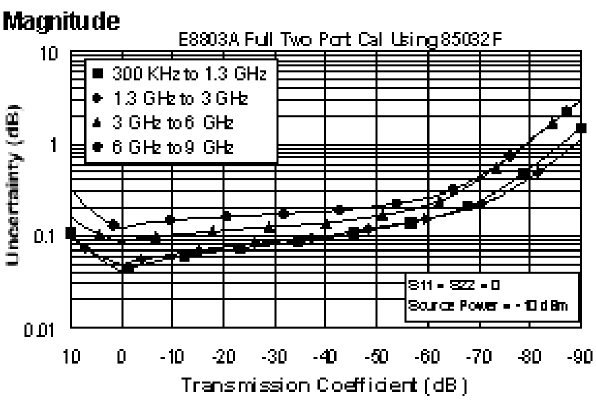
Table 2. Corrected System Performance With Type-N Device Connectors, 85032F Calibration Kit

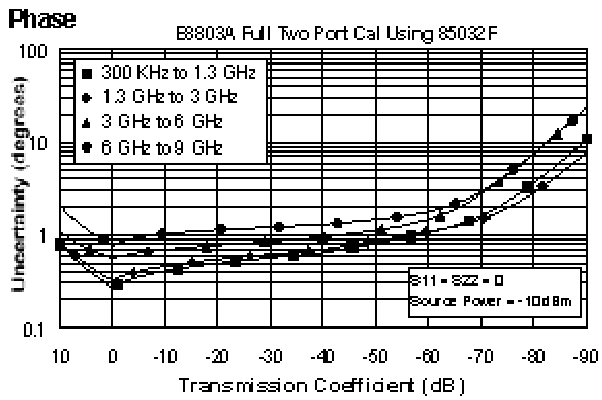
Applies to the E8801A, E8802A, and E8803A analyzer, 85032F (Type-N, 50Ω) calibration kit, N6314A test port cable, and a full 2-port calibration. Also applies to the following conditions:

- IF bandwidth = 10 Hz
- No averaging applied to data
- Environmental temperature 25° ±5 °C, with < 1 °C deviation from calibration temperature
- Isolation calibration not omitted

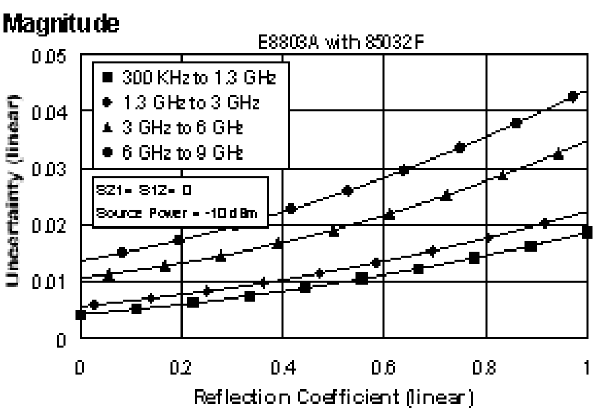
Description	Specification (dB)			
	300 kHz to 1.3 GHz	1.3 GHz to 3 GHz	3 GHz to 6 GHz	6GHz to 9 GHz
Directivity	49	46	40	38
Source Match	41	40	36	35
Load Match	49	45	39	37
Reflection Tracking	±0.011	±0.021	±0.032	±0.054
Transmission Tracking	±0.012	±0.020	±0.055	±0.083

Transmission Uncertainty (Specifications)





Reflection Uncertainty (Specifications)



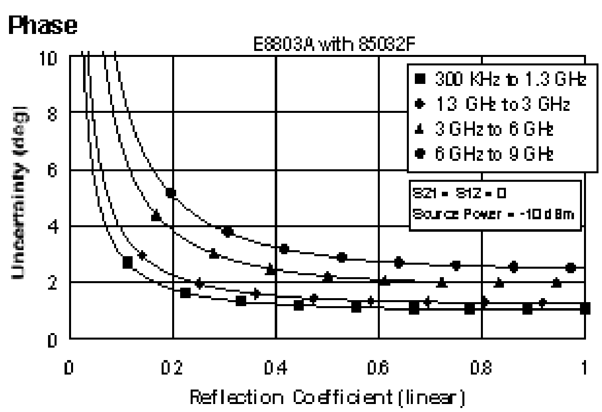


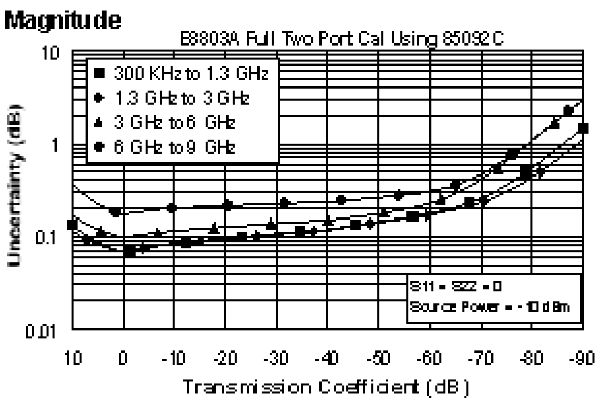
Table 3. Corrected System Performance With Type-N Device Connectors, 85092C Electronic Calibration Module

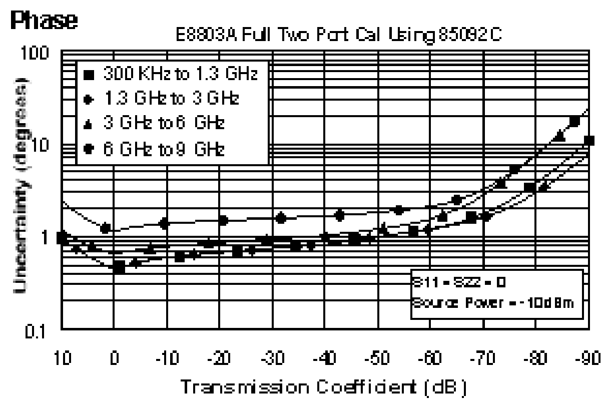
Applies to the E8801A, E8802A, and E8803A analyzer, 85092C (Type-N, 50Ω) electronic calibration (ECal) module, N6314A test port cable, and a full 2-port calibration. Also applies to the following conditions:

- IF bandwidth = 10 Hz
- No averaging applied to data
- Environmental temperature 25° ±5 °C, with < 1 °C deviation from calibration temperature
- Isolation calibration not omitted

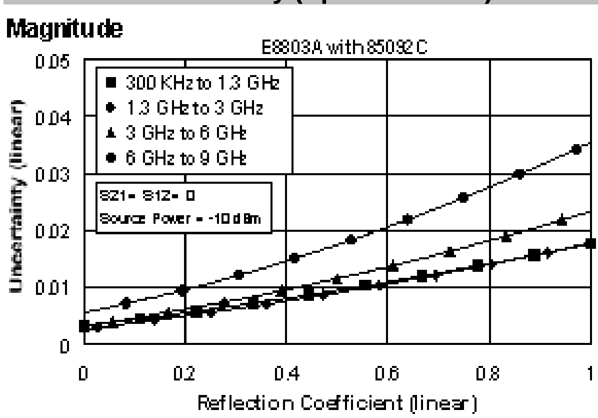
Description	Specification (dB)			
	300 kHz to 1.3 GHz	1.3 GHz to 3 GHz	3 GHz to 6 GHz	6 to 9 GHz[a]
Directivity	52	54	52	47
Source Match	45	44	41	36
Load Match	47	47	44	39
Reflection Tracking	±0.040	±0.040	±0.060	±0.070
Transmission Tracking	±0.039	±0.039	±0.068	±0.136

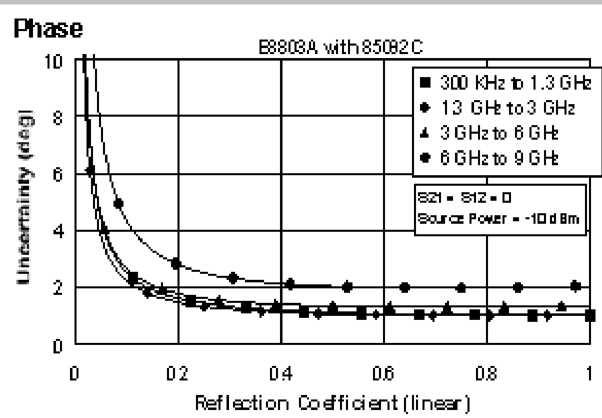
Transmission Uncertainty (Specifications)





Reflection Uncertainty (Specifications)





Corrected System Performance with 3.5 mm Connectors

- 85033E Calibration Kit
- 85093C Electronic Calibration Module

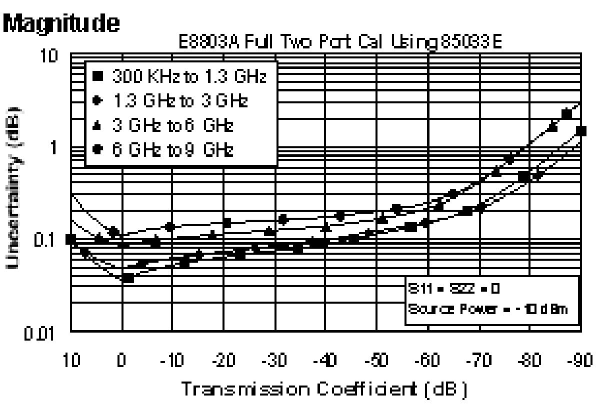
Table 4. Corrected System Performance With 3.5 mm Device Connector Type, 85033E Calibration Kit

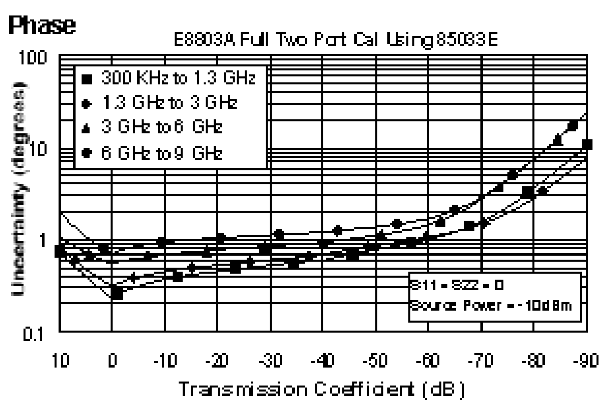
Applies to the E8801A, E8802A, and E8803A analyzer, 85033E (3.5 mm, 50Ω) calibration kit, N6314A test port cable, and a full 2-port calibration. Also applies to the following conditions:

- IF bandwidth = 10 Hz
- No averaging applied to data
- Environmental temperature 25° ±5 °C, with < 1 °C deviation from calibration temperature
- Isolation calibration not omitted

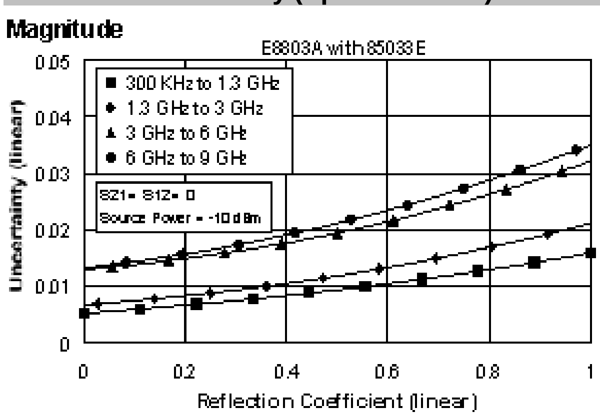
Description	Specification (dB)			
	300 kHz to 1.3 GHz	1.3 GHz to 3 GHz	3 GHz to 6 GHz	6 GHz to 9 GHz
Directivity	46	44	38	38
Source Match	43	40	37	36
Load Match	46	44	38	38
Reflection Tracking	±0.006	±0.007	±0.009	±0.010
Transmission Tracking	±0.012	±0.021	±0.057	±0.075

Transmission Uncertainty (Specifications)





Reflection Uncertainty (Specifications)



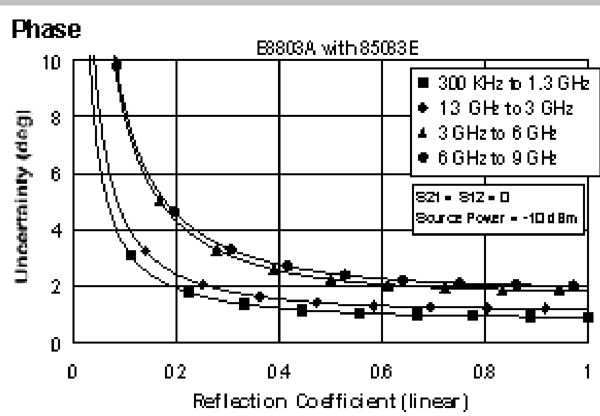


Table 5. Corrected System Performance With 3.5 mm Device Connector Type, 85093C Electronic Calibration Module

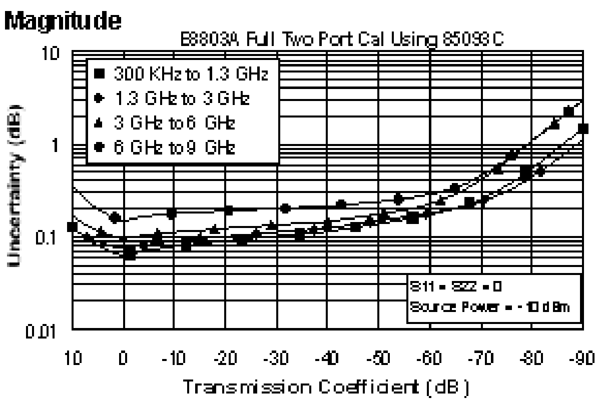
Applies to the E8801A, E8802A, and E8803A analyzer, 85093C (3.5 mm, 50Ω) electronic calibration (ECal) module, N6314A test port cable, and a full 2-port calibration. Also applies to the following conditions:

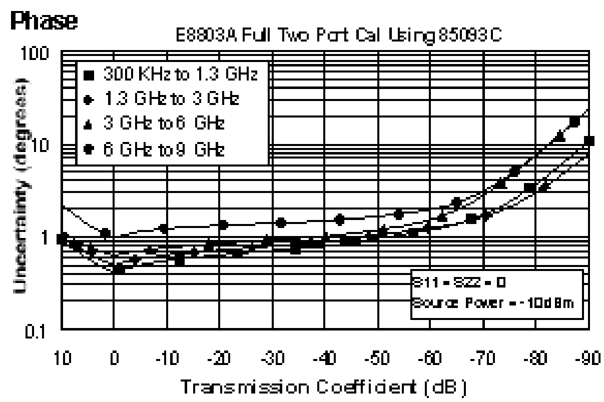
- IF bandwidth = 10 Hz

- No averaging applied to data
- Environmental temperature 25° ±5 °C, with < 1 °C deviation from calibration temperature
- Isolation calibration not omitted

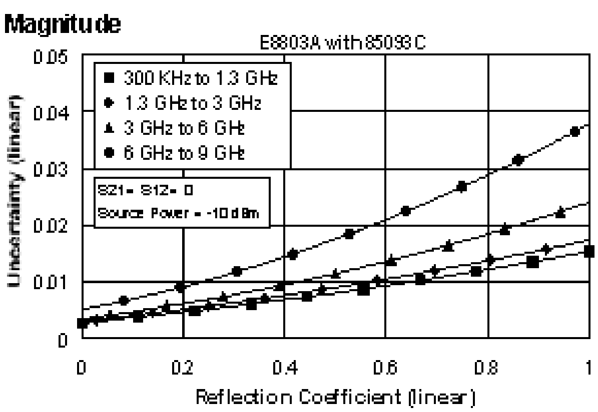
Description	Specification (dB)			
	300 kHz to 1.3 GHz	1.3 GHz to 3 GHz	3 GHz to 6 GHz	6 GHz to 9 GHz[a]
Directivity	52	52	51	47
Source Match	44	44	39	34
Load Match	47	47	44	40
Reflection Tracking	±0.030	±0.040	±0.050	±0.070
Transmission Tracking	±0.039	±0.049	±0.068	±0.117

Transmission Uncertainty (Specifications)





Reflection Uncertainty (Specifications)



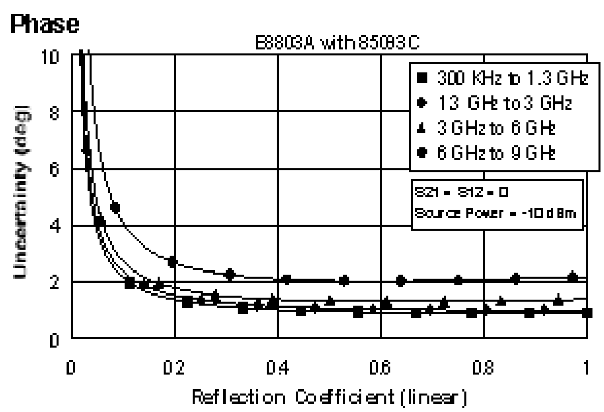


Table 6. Corrected System Performance With 7-16 Device Connector Type, 85038A Calibration Kit

Applies to the E8801A, E8802A, and E8803A analyzer, 85038A (7-16, 50Ω) calibration kit, N6314A test port cable, and a full 2-port calibration. Also applies to the following conditions:

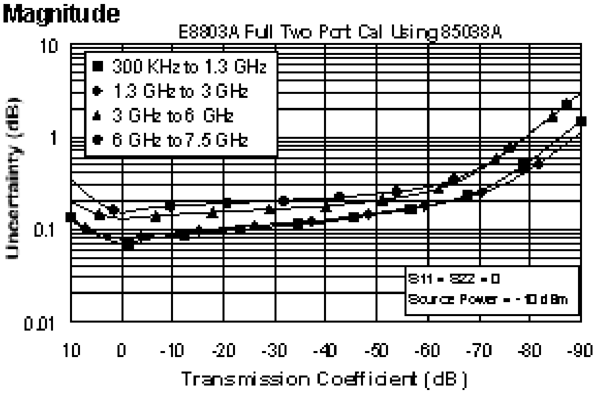
- IF bandwidth = 10 Hz
- No averaging applied to data

- Environmental temperature 25° ±5 °C, with < 1 °C deviation from calibration temperature
- Isolation calibration not omitted

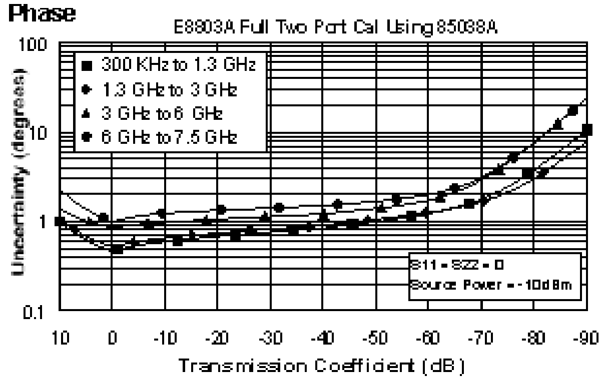
Description	Specification (dB)			
	300 kHz to 1.3 GHz	1.3 GHz to 3 GHz	3 to 6 GHz	6 to 9 GHz[a]
Directivity	40	40	36	36
Source Match	37	37	34	34
Load Match	39	39	35	35
Reflection Tracking	±0.089	±0.089	±0.115	±0.115
Transmission Tracking	±0.024	±0.033	±0.082	±0.103

Transmission Uncertainty (Specifications)

Magnitude

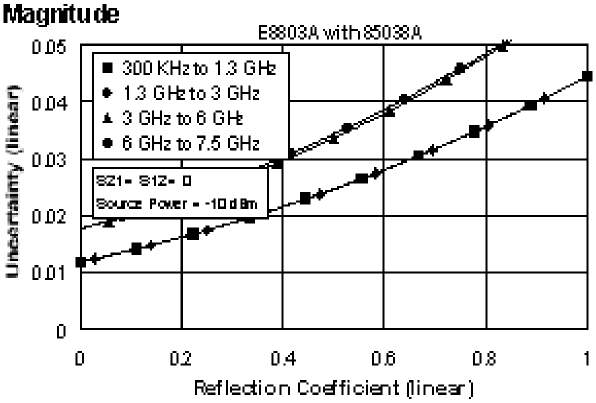


Phase



Reflection Uncertainty (Specifications)

Magnitude



Phase

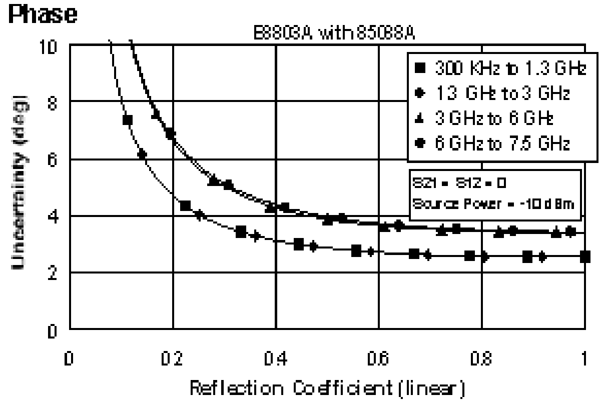


Table 7. Uncorrected Instrument Performance

Description	Specification (dB)				
	300 kHz to 1 MHz	1MHz to 1.3 GHz	1.3 GHz to 3 GHz	3 GHz to 6 GHz	6 GHz to 9 GHz
Directivity	30	33	27	20	13
Source Match	18	18	16	11	8
Load Match	20	20	17	13.5	13
Reflection Tracking	±1.5	±1.5	±1.5	±2.5	±3.0
Transmission Tracking	±1.5	±1.5	±1.5	±2.5	±3.0

Test Port Output Characteristics (Source)

- Output Frequency
- Output Power
- Output Signal Purity

Table 8. Test Port Output Frequency

Description	Specification	Supplemental Information
Range:		
E8801A	300 kHz to 3.0 GHz	
E8802A	300 kHz to 6.0 GHz	
E8803A	300 kHz to 9.0 GHz	
Resolution:	1 Hz	
Source Stability		±1 ppm, 0°C to 40 °C, typical ±1ppm/year maximum
Source Stability (Option 1E5)		±0.05 ppm, 0° to 70 °C, typical ±0.1 ppm/year maximum
CW Accuracy	±3 ppm	
CW Accuracy (Option 1E5)	±1 ppm	

Table 9. Test Port Output Power[a]

Description	Specification	Supplemental Information
Level Accuracy:		
300 kHz to 6 GHz	±1.0 dB	Variation from 0 dBm in power range 0
6 GHz to 9 GHz	±2.0 dB	±1.5dB below 10 MHz
Level Linearity:		
		Variation from 0 dBm in power range 0
300 kHz to 9 GHz	±0.3 dB	-15 to +5 dBm
300 kHz to 1 MHz	±1.0 dB	+5 to +10 dBm
1 MHz to 6 GHz	±0.5 dB	+5 to +10 dBm
6 GHz to 9 GHz	±0.5 dB	+5 to +7 dBm
Range[b]:		
300 kHz to 6 GHz	-15 to +10 dBm	
6 GHz to 9 GHz	-15 to +7 dBm	
Range[b]:		
(Option 1E1):		
300 kHz to 6 GHz	-85 to +10 dBm	
6 GHz to 9 GHz	-85 to +7 dBm	
Sweep Range		
300 kHz to 6 GHz	25 dB	
6 GHz to 9 GHz	22 dB	
Level Resolution	0.01 dB	

a Source output performance on port 1 only. Port 2 output performance is typical.

b Power to which the source can be set and phase lock is assured.

Table 10. Test Port Output Signal Purity

Description	Specification	Supplemental Information
Harmonics (2nd or 3rd)		
at max output power (< 25 MHz)		< -25 dBc, typical
at max output power (25 MHz to 9 GHz)		< -25 dBc, characteristic[a]
at 0 dBm output		< -35 dBc, typical
at -10 dBm output		< -38 dBc, typical, in power range 0
Non-harmonic Spurious		
at max output		-30 dBc, typical for offset freq>1kHz
at -10 dBm output		-50 dBc, typical for offset freq >1kHz

a Typical below 25 MHz.

Test Port and Receiver Input Characteristics

- **Input Levels**
- **Trace Noise**
- **Reference Level and Stability**
- **Dynamic Accuracy**
- **Group Delay**

Table 11. Test Port and Receiver Input Levels

Description	Specification	Supplemental Information
Maximum Test Port Input Level		
Test Ports 1 and 2:		
300 kHz to 25 MHz	+10 dBm	< 0.6 dB compression
25 MHz to 3 GHz	+10 dBm	< 0.4 dB compression
3 GHz to 6 GHz	+10 dBm	< 0.7 dB compression
6 GHz to 9 GHz	+5 dBm	< 0.7 dB compression
Damage Level		
Test Port 1, 2		+30 dBm or ±30 VDC, typ.
R, A, B (Opt. 014)		+15 dBm or ±5 VDC, typ.
Coupler Thru (Opt. 014)		+33 dBm or ±0 VDC, typ.

Test Port Noise Floor[a]		
300 kHz to 25 MHz[b]		
10 Hz IF Bandwidth	-115 dBm	
1 kHz IF Bandwidth	-95 dBm	
25 MHz to 3 GHz[b]		
10 Hz IF Bandwidth	-118 dBm	
1 kHz IF Bandwidth	-98 dBm	
3 GHz to 9 GHz		
10 Hz IF Bandwidth	≤ -108 dBm	
1 kHz IF Bandwidth	≤ -88 dBm	
Receiver Noise Floor[a]		
300 kHz to 25 MHz[c]		
10 Hz IF Bandwidth	≤ -130 dBm	
1 kHz IF Bandwidth	≤ -110 dBm	
25 MHz to 3 GHz[c]		
10 Hz IF Bandwidth	≤ -133 dBm	
1 kHz IF Bandwidth	≤ -113 dBm	
6 GHz to 9 GHz		
10 Hz IF Bandwidth	≤ -123 dBm	
1 kHz IF Bandwidth	≤ -103 dBm	
Crosstalk		
		Between test ports 1 and 2, with short circuits at both ports
300 kHz to 1 MHz	<-120 dB	
1 MHz to 25 MHz	<-125 dB	
25 MHz to 3 GHz	<-126 dB	
3 GHz to 6 GHz	<-117 dB	
6 GHz to 9 GHz	<-106 dB	
Maximum Receiver Input Level (A, B, R)		

300 kHz to 6 GHz		-6 dBm, typical
6 GHz to 9 GHz		-9 dBm, typical
Reference Input Level (R)[d]		
300 kHz to 9 GHz		-10 to -35 dBm, typical
Maximum Coupler Input Level (Opt 014)		
300 kHz to 9 GHz		+33 dBm, typical

a Total average (RMS) noise power calculated as the mean value of a linear magnitude trace expressed in dBm.

b May be limited to -90 dBm at particular frequencies below 750 MHz due to spurious receiver residuals.

c May be limited to -105 dBm at particular frequencies below 750 MHz due to spurious receiver residuals.

d Input level to maintain phase lock.

Table 12. Test Port Input (Trace Noise)

Description	Specification	Supplemental Information
Trace Noise[a] Magnitude		
1 kHz IF Bandwidth	< 0.002 dB rms	
10 kHz IF Bandwidth	< 0.005 dB rms	
Trace Noise[a] Phase		
1 kHz IF Bandwidth	< 0.010° rms	
10 kHz IF Bandwidth	< 0.035° rms	

a Trace noise is defined as a ratio measurement of a through or a full reflection, with the source set to 0 dBm.

Table 13. Test Port Input (Reference Level and Stability)

Description	Specification	Supplemental Information
Reference Level Magnitude		
Range	±200 dB	
Resolution	0.001 dB	
Reference Level Phase		
Range	±500°	
Resolution	0.01°	
Stability Magnitude[a]		
300 kHz to 3 GHz		0.02 dB/°C, typical
3 GHz to 6 GHz		0.04 dB/°C, typical
6 GHz to 9 GHz		0.06 dB/°C, typical
Stability Phase[a]		
300 kHz to 3 GHz 3 GHz to 6 GHz 6 GHz to 9 GHz		0.2°/°C, typical 0.3°/°C, typical 0.6°/°C, typical

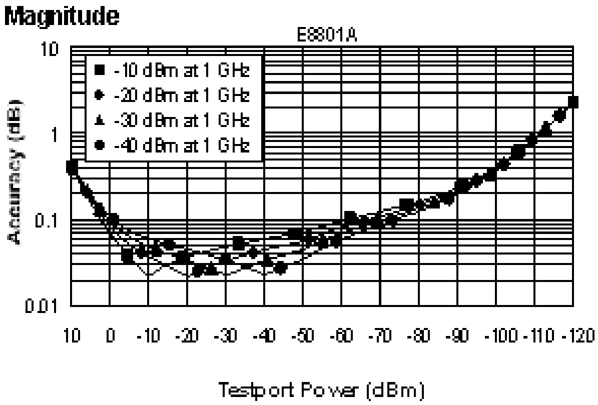
a Stability is defined as a ratio measurement at the test port.

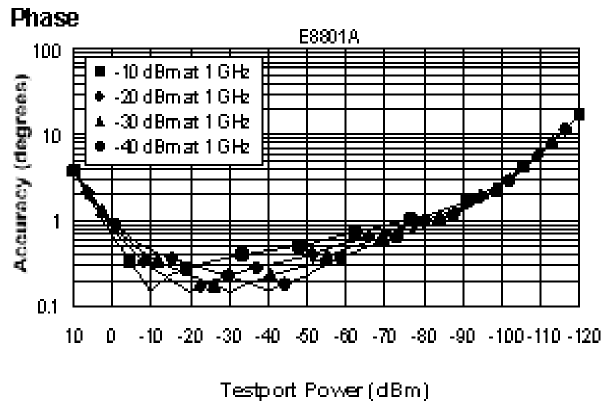
Table 14. Test Port Input (Dynamic Accuracy specification[a])

Accuracy of the test port input power reading is relative to the reference input power level. Applies to input ports 1 and 2 with the following conditions:

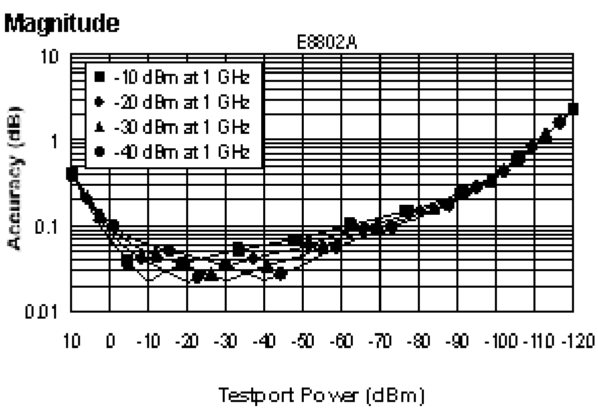
- IF bandwidth = 10 Hz
- Environmental temperature 25° ±5 °C, with < 1 °C deviation from calibration temperature

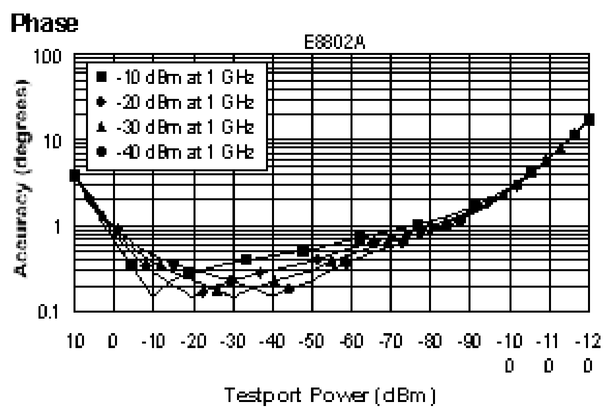
300 kHz to 3 GHz	300 kHz to 3 GHz



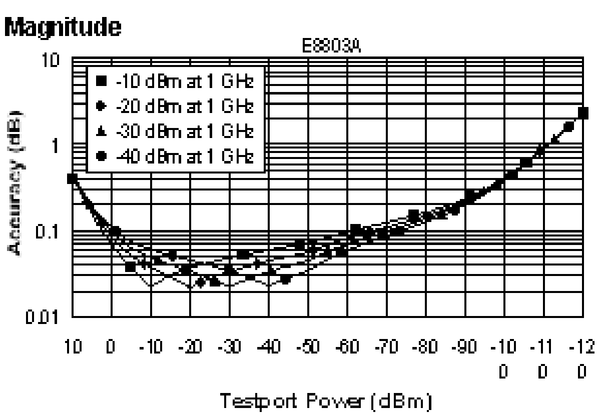


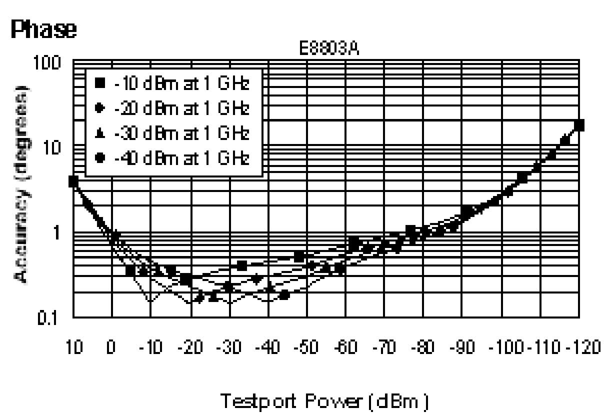
300 kHz to 6 GHz	300 kHz to 6 GHz





300 kHz to 9 GHz	300 kHz to 9 GHz





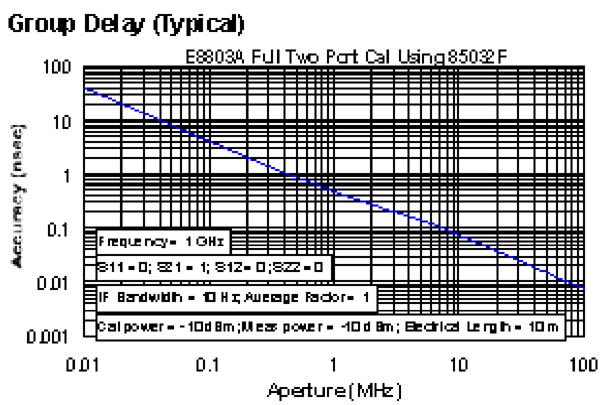
a Dynamic accuracy is verified with the following measurements:

- compression over frequency
- IF linearity at a single frequency of 1.195 GHz and a reference level of -20 dBm

Table 15. Test Port Input (Group Delay)[a]

Description	Specification	Supplemental Information
Aperture (selectable)	(frequency span)/(number of points - 1)	
Maximum Aperture	20% of frequency span	
Range	0.5 x (1/minimum aperture)	
Maximum Delay		Limited to measuring no more than 180° of phase change within the minimum aperture.)
Accuracy		See graph below. Char.

The following graph shows group delay accuracy with type-N full 2-port calibration and a 10 Hz IF bandwidth. Insertion loss is assumed to be < 2 dB and electrical length to be ten meters.



In general, the following formula can be used to determine the accuracy, in seconds, of specific group delay measurement:

±Phase Accuracy (deg)/[360 × Aperture (Hz)]

Depending on the aperture and device length, the phase accuracy used is either incremental phase accuracy or worst case phase accuracy.

a Group delay is computed by measuring the phase change within a specified frequency step (determined by the frequency span and the number of points per sweep).

General Information

- **System Bandwidths**
- **Front Panel**
- **Rear Panel**
- **Environment and Dimensions**

Table 16. System Bandwidths

Description	Specification	Supplemental Information
IF Bandwidth Settings		
Range		1 Hz to 40 kHz in a 1, 2, 3, 5, 7,10 sequence up to 30 kHz, 35 kHz, 40kHz, nominal

Table 17. Front Panel Information

Description	Supplemental Information
RF Connectors	
Type	Type-N, female; 50 Ω, nominal
Center Pin Protrusion	0.204 to 0.207 in., characteristic
Probe Power	
Connector	3-pin connector, male
Positive Supply	+15 VDC ±2%, 400 mA, max, characteristic
Negative Supply	-12.6 VDC ±5%, 300 mA, max, characteristic
Display	
Size	21.3 cm (8.4 in) diagonal color active matrix LCD; 640 (horizontal) X 480 (vertical) resolution
Refresh Rate	Vertical 59.83 Hz; Horizontal 31.41 Hz
Display Range	
Magnitude	±200 dB (at 20 dB/div), max
Phase	±180°, max
Polar	10 pUnits, min 1000 Units, max
Display Resolution	
Magnitude	0.001 dB/div, min
Phase	0.01°/div, min
Marker Resolution	
Magnitude	0.001 dB, min
Phase	0.01°, min
Polar	0.01 mUnit, min; 0.01°,min

Table 18. Rear Panel Information

Description	Supplemental Information
10 MHz Reference In	

Connector	BNC, female
Input Frequency	10 MHz ± 1 ppm, typical
Input Level	-15 dBm to +20 dBm, typical
Input Impedance	200 Ω, nom.
10 MHz Reference Out	
Connector	BNC, female
Output Frequency	10 MHz ± 10 ppm, typical
Signal Type	Sine Wave, typical
Output Level	+10 dBm ± 4 dB into 50 Ω, typical
Output Impedance	50 Ω, nominal
Harmonics	<-40 dBc, typical
VGA Video Output	
Connector	15-pin mini D-Sub; Drives VGA compatible monitors
Devices Supported	Resolutions
Flat Panel (TFT	1024 X 768, 800 X 600, 640 X 480
Flat Panel (DSTN)	800 X 600, 640 X 480
CRT Monitor	1280 X 1024, 1024 X 768, 800 X 600, 640 X 480
	Simultaneous operation of the internal and external displays is allowed, but with 640 X 480 resolution only. If you change resolution, you can only view the external display (internal display will "white out").
Test Set IO	25-pin D-Sub connector, available for external test set control
Aux IO	25-pin D-Sub connector, male, analog and digital IO
Handler IO	36-pin IDC D-ribbon socket connector; all input/output signals are default set to negative logic; can be reset to positive logic via GPIB command
GPIB	24-pin D-sub (Type D-24), female; compatible with IEEE-488.
Parallel Port (LPT1)	25-pin D-Sub connector, female; provides connection to printers or any other parallel port peripherals
Serial Port (COM 1)	9-pin D-Sub, male; compatible with RS-232
USB Port	
	Universal Serial Bus jack, Type A configuration (4 contacts inline, contact 1 on left); female
Contact 1	Vcc: 4.75 to 5.25 VDC, 500 mA, maximum

Contact 2	-Data
Contact 3	+Data
Contact 4	Ground
LAN	10/100BaseT Ethernet, 8-pin configuration; auto selects between the two data rates
Line Power (A third-wire ground is required.)	
Frequency	50/60/400 Hz
Voltage	120/240 VAC (Power supply is auto switching.)
Max	500 Watts

Note: Option H08 and Option H11 specifications are not provided in this E8801/2/3A specifications document.

Table 19. Rear Panel Information (continued)

Description	Supplemental Information
External AM Input	
Description	Input provides low-frequency AM modulation to test port output signal, or shifts the test port output. Zero volts input gives the power level set by the instrument, a positive voltage gives a higher level, and a negative voltage gives a lower level.
Connector	BNC, female
Input Sensitivity	8 dB/V, typical
Bandwidth	1 kHz, typical
Input Impedance	1 kΩ, typical
External Detector Input	
Description	Input from an external, negative polarity diode detector provides ALC for a test port remote from instrument's front panel
Connector	BNC, female
Input Sensitivity	-500 mV yields approximately -3 dBm at detector's input, typical
Bandwidth	50 kHz, typical
Input Impedance	1 kΩ, nominal

Table 20. Analyzer Environment and Dimensions

Description	Supplemental Information
General Environmental	
RFI/EMI Susceptibility	Defined by CISPR Pub. 11, Group 1, Class A, and IEC 50082-1
ESD	Minimize using static-safe work procedures and an antistatic bench mat
Dust	Minimize for optimum reliability
Operating Environment	
Temperature	0 °C to +40 °C Instrument powers up, phase locks, and displays no error messages within this temperature range.
Error-Corrected Temperature Range	25°C ± 5°C with less than 1°C deviation from calibration temp.
Humidity	5% to 95% at +40 °C
Altitude	0 to 4500 m (14,760 ft.)
Non-Operating Storage Environment	
Temperature	-40 °C to +70 °C
Humidity	0% to 90% at +65 °C (non-condensing)
Altitude	0 to 15,240 m (50,000 ft.)

Cabinet Dimensions			
	Height	**Width**	**Depth**
Excluding front and rear panel hardware and feet	223 mm 8.75 in	426 mm 16.75 in	427 mm 16.8 in
As shipped - includes front panel connectors, rear panel bumpers, and feet.	235 mm 9.25 in	435 mm 17.10 in	470 mm 18.5 in
As shipped plus handles	235 mm 9.25 in	458 mm 18 in	501 mm 19.70 in
As shipped plus rack-mount flanges	235 mm 9.25 in	483 mm 19 in	470 mm 18.5 in
As shipped plus handles and flanges	235 mm 9.25 in	483 mm 19 in	501 mm 19.70 in

Weight	
Net	24 kg (54 lb), nominal
Shipping	32 kg (70 lb), nominal

Note: "Misc. Information" specifications are not included in this E8801/2/3A document.

Measurement Throughput Summary

- Typical Cycle Time for Measurement Completion
- Cycle Time with Changes in IF Bandwidth
- Cycle Time with Changes in Number of Points
- Data Transfer Time
- Recall and Sweep Speed

Table 21. Typical Cycle Time[a,b] (ms)

	Number of Points			
	101	**201**	**401**	**1601**
Start 1.8 GHz, Stop 2 GHz, 35 kHz IF bandwidth				
Uncorrected, 1-port cal	7	10	16	52
2-Port cal	27	36	55	164
Start 300 kHz, Stop 3 GHz, 35 kHz IF bandwidth				
Uncorrected, 1-port cal	48	54	64	104
2-Port cal	103	119	145	254
Start 300 kHz, Stop 9 GHz, 35 kHz IF bandwidth				
Uncorrected, 1-port cal	51	57	64	103
2-Port cal	112	124	138	220

a Typical performance.

b Includes sweep time, retrace time and band-crossing time. Analyzer display turned off with DISPLAY:ENABLE OFF. Add 21 ms for display on. Data for one trace (S11) measurement..

Table 22. Cycle Time vs. IF Bandwidth[a]

Applies to the Preset condition (201 points, correction off) except for the following changes:

- CF = 1 GHz
- Span = 100 MHz
- Display off (add 21 ms for display on)

IF Bandwidth (Hz)	Cycle Time (ms)[b]
40,000	8
35,000	9
30,000	11
20,000	13
10,000	28
7000	36
5000	48
3000	72
1000	196
300	620
100	1875
30	8062
10	17877

a Typical performance.
b Cycle time includes sweep and retrace time.

Table 23. Cycle Time vs. Number of Points[a]

Applies to the Preset condition (35 kHz IF bandwidth, correction off) except for the following changes:

- CF = 1 GHz
- Span = 100 MHz
- Display off (add 21 ms for display on)

Number of Points	Cycle Time (ms)[b]
3	4
11	4
51	5
101	6
201	9
401	16
801	29
1601	52

a Typical performance.
b Cycle time includes sweep and retrace time.

Table 24. Data Transfer Time[a] (ms)

	Number of Points			
	51	**201**	**401**	**1601**
SCPI over GPIB (program executed on external PC)[b]				
32-bit floating point	3	7	12	43
64-bit floating point	4	12	22	84
ASCII	7	64	24	489
SCPI over 100 Mbit/s LAN (program executed on external PC)[b]				
32-bit floating point	1	1	1	1
64-bit floating point	1	1	1	2
ASCII	5	15	26	96
SCPI (program executed in the analyzer)[d]				
32-bit floating point	1	1	2	3
64-bit floating point	1	2	2	4
ASCII	8	29	56	222
COM (program executed in the analyzer)[e]				
32-bit floating point	1	1	1	1
Variant type	1	1	2	6
DCOM over 100 Mbit/s LAN (program executed on external PC)[f]				
32-bit floating point[g]	1	1	1	2
Variant type[h]	1	3	6	19

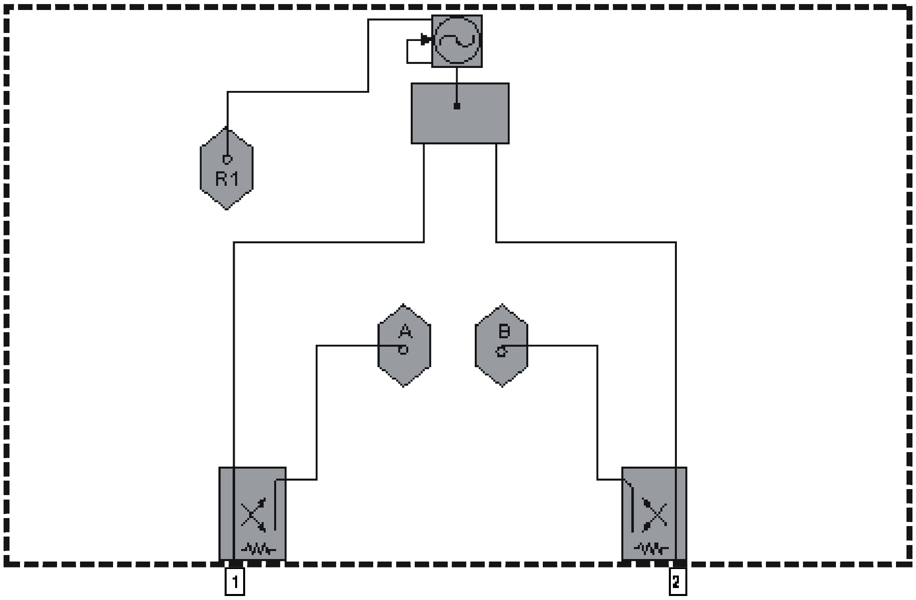
a Typical performance of unit with 500 MHz Pentium III processor.
b Measured using a VEE 5.0 program running on a 600 MHz HP Kayak, National InstrumentsTM GPIB card. Transferred complex S11 data , using "CALC:DATA?SDATA".
c Measured using a VEE 5.0 program running on a 600 MHz HP Kayak. Transferred complex S11 data, using "CALC:DATA?SDATA". Speed dependent on LAN traffic, if connected to network.
d Measured using a VEE 5.0 program running inside PNA Series Analyzer. Transferred complex S11 data, using "CALC:DATA?SDATA".
e Measured using a Visual Basic 6.0 program running inside PNA Series Analyzer. Transferred complex S11 data.
f Measured using a Visual Basic 6.0 program running on a 600 MHz HP Kayak. Transferred complex S11 data. Speed dependent on LAN traffic, if connected to network.
g Used IArray Transfer.getComplex method for 32-bit floating point.
h Used meas.getData method for Variant data type.

Table 25. Recall and Sweep Speed[a]

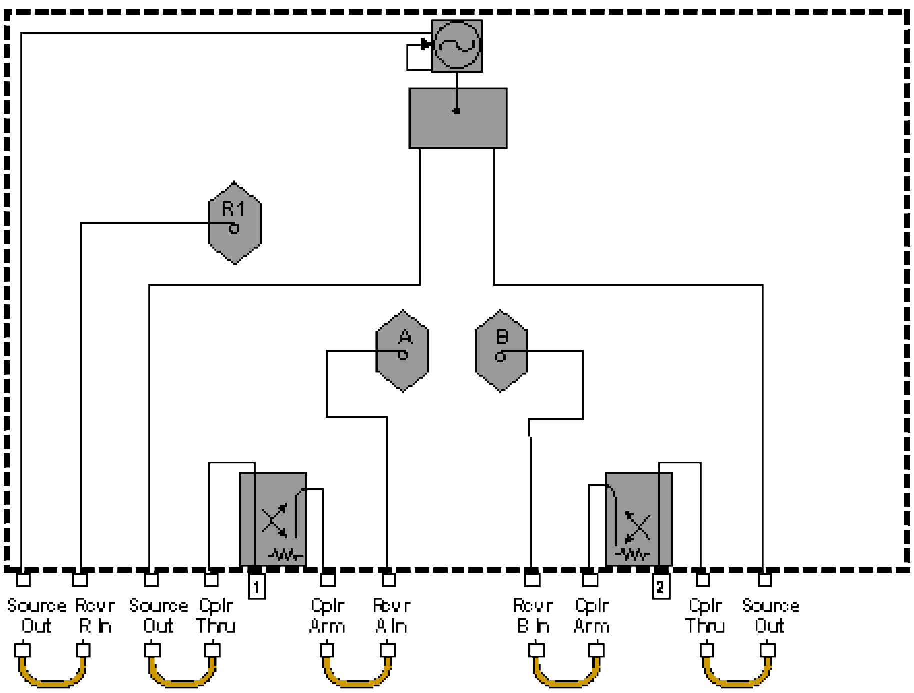
Operations	Number of Window(s)	Number of Trace(s)	Recall Time (ms)
Recall	1	1	49
Recall and Sweep	1	1	59
Recall	1	2	82
Recall and Sweep	1	2	96
Recall	1	4	159
Recall and Sweep	1	4	203
Recall	2	2	93
Recall and Sweep	2	2	115
Recall	3	4	158
Recall and Sweep	3	4	218
Recall	4	4	187
Recall and Sweep	4	4	247
Recall	4	8	340
Recall and Sweep	4	8	507

a CF=177 MHz, Span=200 MHz, 201 points, 35 kHz IF BW

E8801A, E8802A, and E8803A – Simplified Test Set Block Diagram



E8801A, E8802A, and E8803A with Option 014 – Simplified Test Set Block Diagram



Technical Specifications for the N3381A, N3382A, N3383A

(Rev. 2005-07-12)

This is a complete list of the N3381A, N3382A, and N3383A network analyzer technical specifications.

- To optimize viewing of uncertainty curves, click the Maximize button.
- To view or print the .pdf version of the specifications, visit our web site at http://www.agilent.com/find/pna, and search for "N338xA Specifications"
- The uncertainty curves contained in this document apply only to the setup conditions listed. Please download our free Uncertainty Calculator from http://www.agilent.com/find/na_calculator to generate the curves for your PNA setup. View the equations used to generate the uncertainty curves.

Definitions

Corrected System Performance

- **System Dynamic Range**
- **Type-N Connectors**
- **3.5 mm Connectors**
- **7-16 Connectors**

Uncorrected Instrument Performance

Test Port Output Characteristics (Source)

Test Port and Receiver Input Characteristics

General Information

Front-panel Jumper Specs (Option 014 only)

Measurement Throughput Summary

Test Set Block Diagram

Test Set with Option 015 Block Diagram

See Specs for other PNA models

Definitions

All specifications and characteristics apply over a 25 °C ±5 °C range (unless otherwise stated) and 90 minutes after the instrument has been turned on.

Specification (spec.): Warranted performance. Specifications include guardbands to account for the expected statistical performance distribution, measurement uncertainties, and changes in performance due to environmental conditions.

Characteristic (char.): A performance parameter that the product is expected to meet before it leaves the factory, but that is not verified in the field and is not covered by the product warranty. A characteristic includes the same guardbands as a specification.

Typical (typ.): Expected performance of an average unit which does not include guardbands. It is not covered by the product warranty.

Nominal (nom.): A general, descriptive term that does not imply a level of performance. It is not covered by the product warranty.

Calibration: The process of measuring known standards to characterize a network analyzer's systematic (repeatable) errors.

Corrected (residual): Indicates performance after error correction (calibration). It is determined by the quality of calibration standards and how well "known" they are, plus system repeatability, stability, and noise.

Uncorrected (raw): Indicates instrument performance without error correction. The uncorrected performance affects the stability of a calibration.

Standard: When referring to the analyzer, this includes no options unless noted otherwise.

Corrected System Performance

The specifications in this section apply for measurements made with the N3381A, N3382A, and N3383A analyzer with the following conditions:

- 10 Hz IF bandwidth
- No averaging applied to data
- Environmental temperature of 25 °C ±5 °C, with < 1 °C deviation from calibration temperature
- Isolation calibration not omitted

Note: The uncertainty curves contained in these specifications apply only to the setup conditions listed. Please download our free Uncertainty Calculator from http://www.agilent.com/find/na_calculator to generate the curves for your PNA setup. View the equations used to generate the uncertainty curves.

- **System Dynamic Range**
- **Type-N Connectors**
- **3.5 mm Connectors**
- **7-16 Connectors**

Table 1. System Dynamic Range

Description	Specification (dB)	Characteristic (dB)
Dynamic range[a] (at test port)		
300 kHz to 25 MHz[b]	125	
25 MHz to 3 GHz[b]	128	
3 GHz to 6 GHz	118	
6 GHz to 9 GHz	115	
Dynamic range[c] (at receiver input)		
300 kHz to 25 MHz[d]		140
25 MHz to 3 GHz[d]		143
3 GHz to 6 GHz		133
6 GHz to 9 GHz		130

a The test port dynamic range is calculated as the difference between the test port rms noise floor and the source maximum output power. The effective dynamic range must take measurement uncertainties and interfering signals into account.

b May be limited to 100 dB at particular frequencies below 750 MHz due to spurious receiver residuals.

c The receiver input dynamic range is calculated as the difference between the receiver rms noise floor and the source maximum output power. The effective dynamic range must take measurement uncertainties and interfering signals into account. This set-up should only be used when the receiver input will never exceed its damage level. When the analyzer is in segment sweep mode, frequency segments can be defined with a higher power level when the extended dynamic range is required (i.e. the portion of the device's response with high insertion loss), and reduced power when receiver damage may occur (i.e. the portion of the devices's response with low insertion loss). Specification applies only when power is sourced from Port 1. If power is sourced from either Port 2 or Port 3, dynamic range decreases by 3 dB.

d May be limited to 115 dB at particular frequencies below 750 MHz due to spurious receiver residuals.

Note: Receiver Dynamic Range specifications are not included in this N3381/2/3A document.

Note: This document provides technical specifications for the following calibration kits only: 85032F, 85092C, 85033E, 85093C and 85038A.

Corrected System Performance with Type-N Connectors

85032F Calibration Kit

85092C Electronic Calibration Module

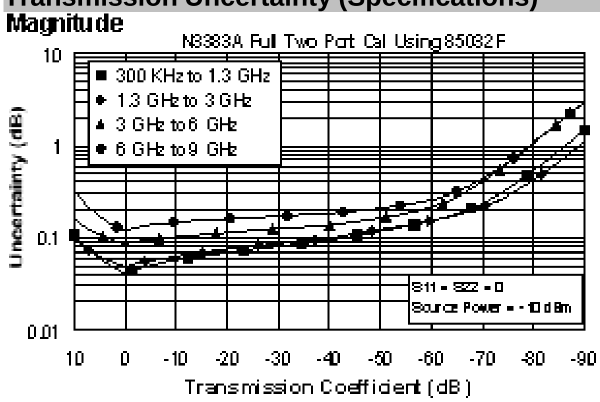
Table 2. Corrected System Performance With Type-N Device Connectors, 85032F Calibration Kit

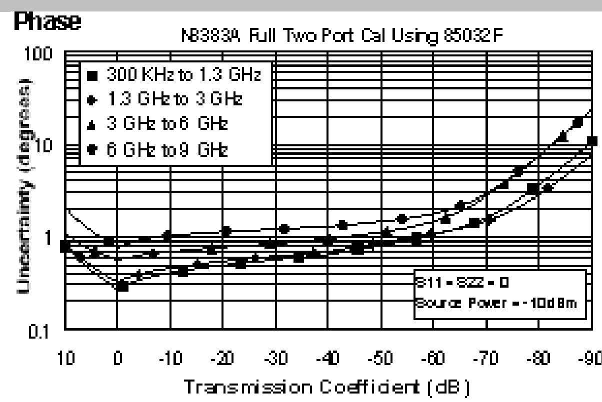
Applies to the N3381A, N3382A, and N3383A analyzer, 85032F (Type-N, 50Ω) calibration kit, N6314A test port cable, and a full 2-port calibration. Also applies to the following conditions:

- IF bandwidth = 10 Hz
- No averaging applied to data
- Environmental temperature 25° ±5 °C, with < 1 °C deviation from calibration temperature
- Isolation calibration not omitted

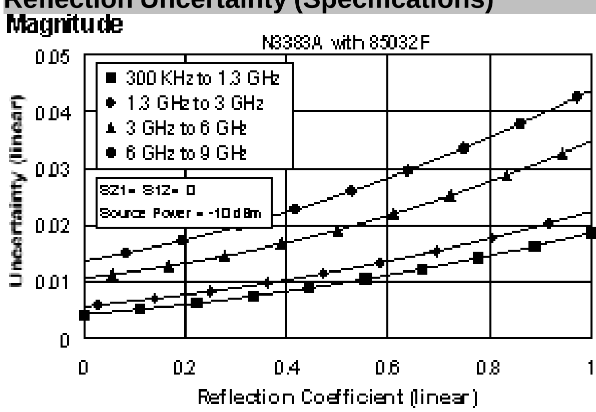
Description	Specification (dB)			
	300 kHz to 1.3 GHz	1.3 GHz to 3 GHz	3 to 6 GHz	6 to 9 GHz
Directivity	49	46	40	38
Source Match	41	40	36	35
Load Match	49	45	39	37
Reflection Tracking	±0.011	±0.021	±0.032	±0.054
Transmission Tracking	±0.012	±0.020	±0.055	±0.083

Transmission Uncertainty (Specifications)





Reflection Uncertainty (Specifications)



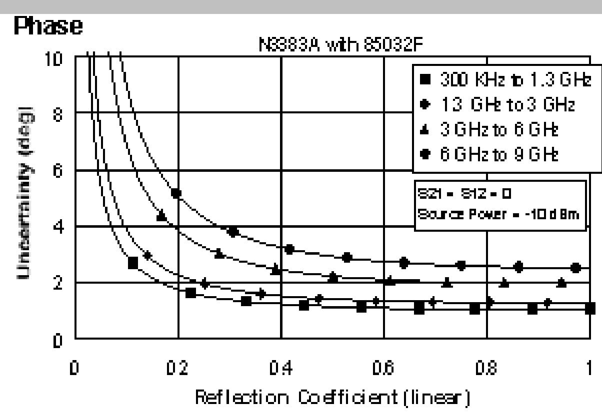


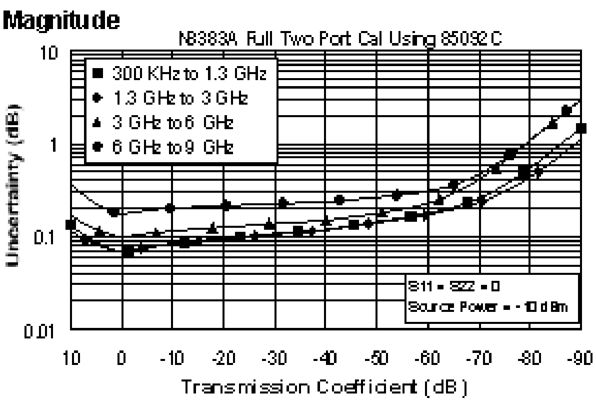
Table 3. Corrected System Performance With Type-N Device Connectors, 85092C Electronic Calibration Module

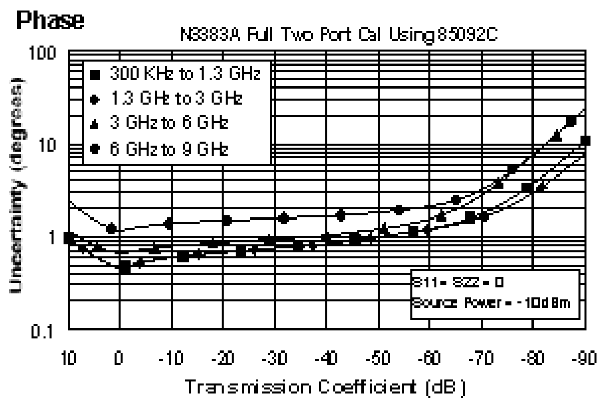
Applies to the N3381A, N3382A, and N3383A analyzer, 85092C (Type-N, 50Ω) electronic calibration (ECal) module, N6314A test port cable, and a full 2-port calibration. Also applies to the following conditions:

- IF bandwidth = 10 Hz
- No averaging applied to data
- Environmental temperature 25° ±5 °C, with < 1 °C deviation from calibration temperature
- Isolation calibration not omitted

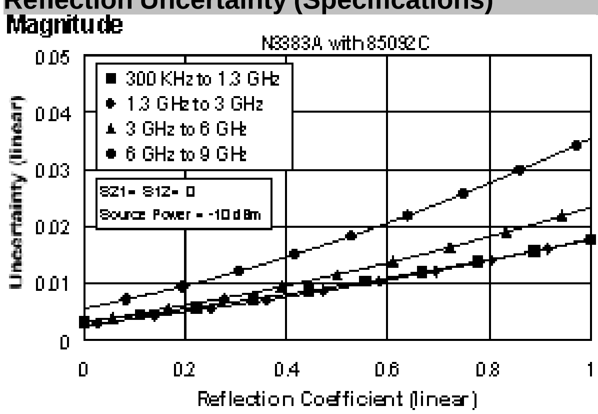
Description	Specification (dB)			
	300 kHz to 1.3 GHz	1.3 GHz to 3 GHz	3 to 6 GHz	6 to 9 GHz[a]
Directivity	52	54	52	47
Source Match	45	44	41	36
Load Match	47	47	44	39
Reflection Tracking	±0.040	±0.040	±0.060	±0.070
Transmission Tracking	±0.039	±0.039	±0.068	±0.136

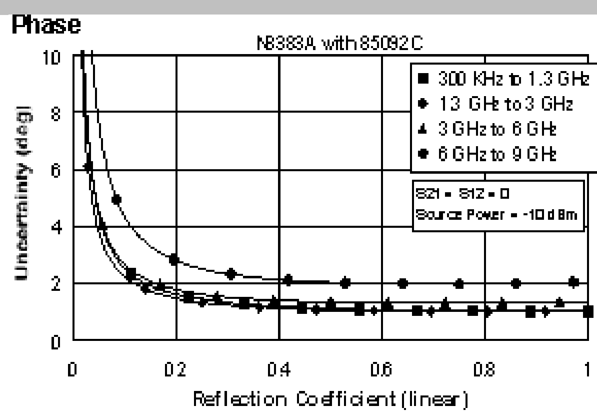
Transmission Uncertainty (Specifications)





Reflection Uncertainty (Specifications)





Corrected System Performance with 3.5 mm Connectors

- 85033E Calibration Kit
- 85093C Electronic Calibration Module

Table 4. Corrected System Performance With 3.5 mm Device Connector Type, 85033E Calibration Kit

Applies to the N3381A, N3382A, and N3383A analyzer, 85033E (3.5 mm, 50Ω) calibration kit, N6314A test port cable, and a full 2-port calibration. Also applies to the following conditions:

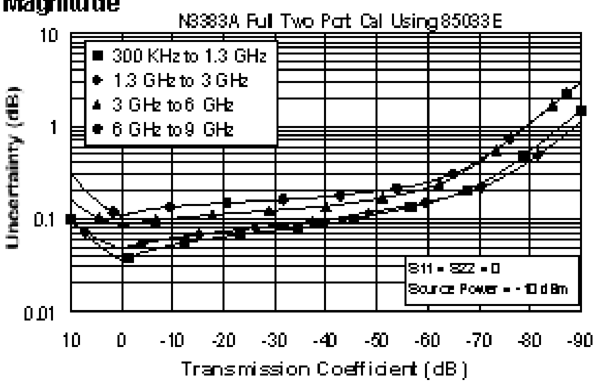
- IF bandwidth = 10 Hz

- No averaging applied to data
- Environmental temperature 25° ±5 °C, with < 1 °C deviation from calibration temperature
- Isolation calibration not omitted

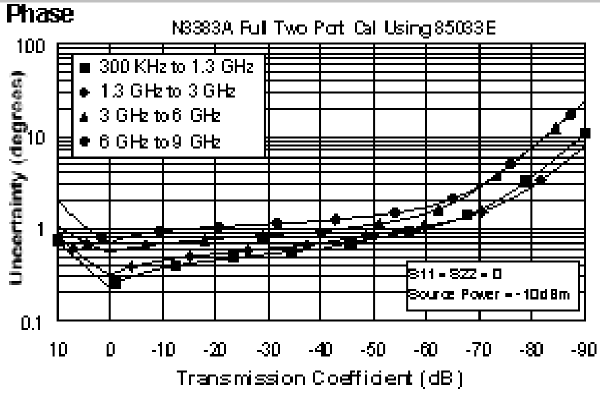
Description	Specification (dB)			
	300 kHz to 1.3 GHz	1.3 GHz to 3 GHz	3 to 6 GHz	6 to 9 GHz
Directivity	46	44	38	38
Source Match	43	40	37	36
Load Match	46	44	38	38
Reflection Tracking	±0.006	±0.007	±0.009	±0.010
Transmission Tracking	±0.012	±0.021	±0.057	±0.075

Transmission Uncertainty (Specifications)

Magnitude

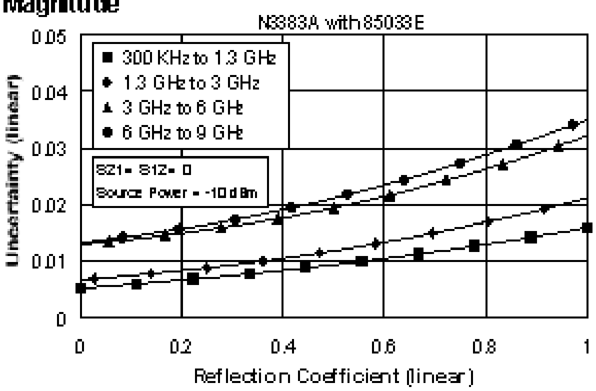


Phase



Reflection Uncertainty (Specifications)

Magnitude



Magnitude

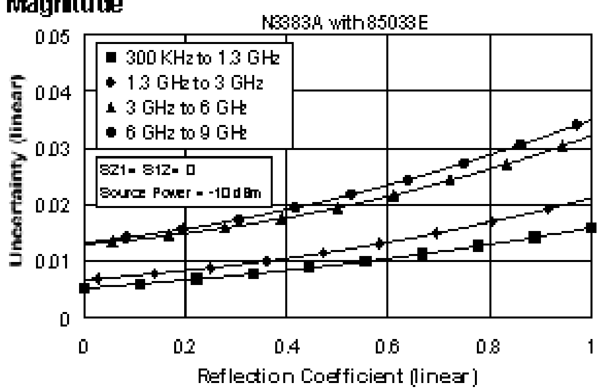


Table 5. Corrected System Performance With 3.5 mm Device Connector Type, 85093C Electronic Calibration Module

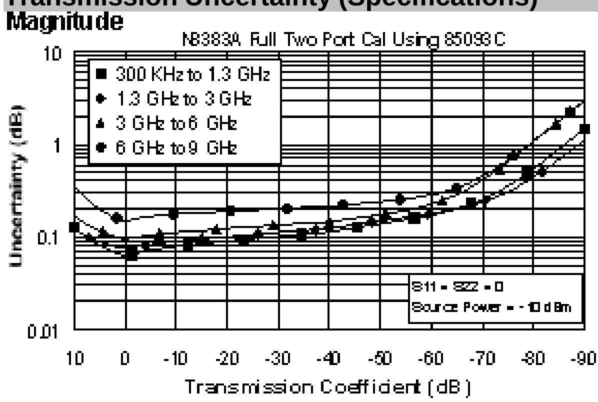
Applies to the N3381A, N3382A, and N3383A analyzer, 85093C (3.5 mm, 50Ω) electronic calibration (ECal) module, N6314A test port cable, and a full 2-port calibration. Also applies to the following conditions:

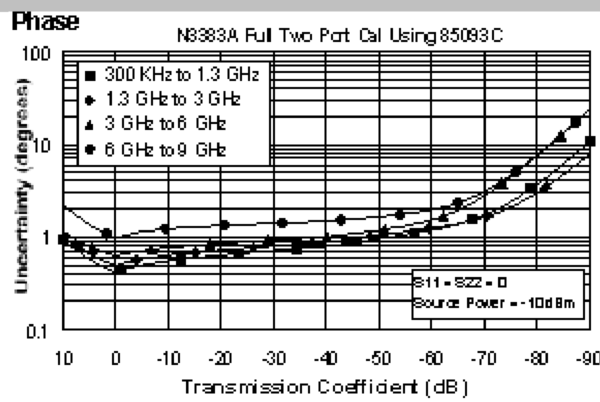
- IF bandwidth = 10 Hz
- No averaging applied to data

- Environmental temperature 25° ±5 °C, with < 1 °C deviation from calibration temperature
- Isolation calibration not omitted

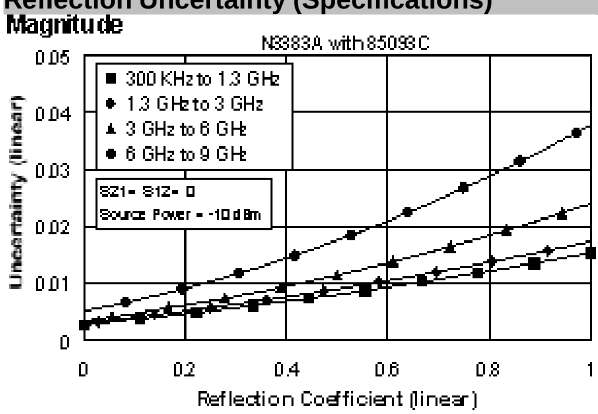
Description	Specification (dB)			
	300 kHz to 1.3 GHz	1.3 GHz to 3 GHz	3 to 6 GHz	6 to 9 GHz[a]
Directivity	52	52	51	47
Source Match	44	44	39	34
Load Match	47	47	44	40
Reflection Tracking	±0.030	±0.040	±0.050	±0.070
Transmission Tracking	±0.039	±0.049	±0.068	±0.117

Transmission Uncertainty (Specifications)





Reflection Uncertainty (Specifications)



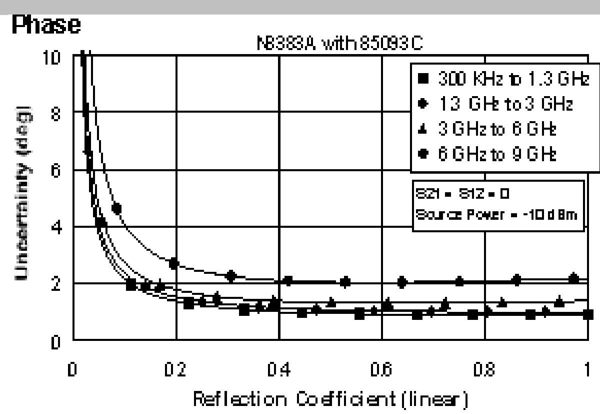


Table 6. Corrected System Performance With 7-16 Device Connector Type, 85038A Calibration Kit

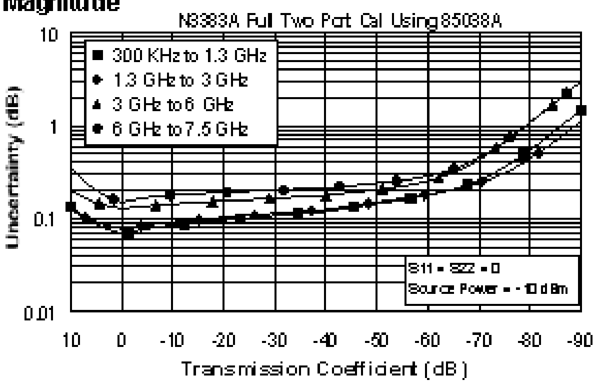
Applies to the N3381A, N3382A, and N3383A analyzer, 85038A (7-16, 50Ω) calibration kit, N6314A test port cable, and a full 2-port calibration. Also applies to the following conditions:

- IF bandwidth = 10 Hz
- No averaging applied to data
- Environmental temperature 25° ±5 °C, with < 1 °C deviation from calibration temperature
- Isolation calibration not omitted

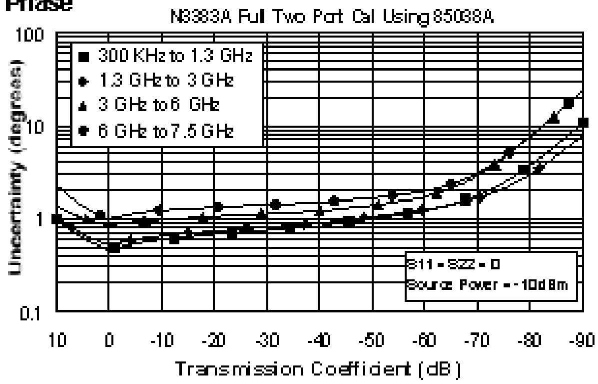
Description	Specification (dB)			
	300 kHz to 1.3 GHz	1.3 GHz to 3 GHz	3 to 6 GHz	6 to 9 GHz[a]
Directivity	40	40	36	36
Source Match	37	37	34	34
Load Match	39	39	35	35
Reflection Tracking	±0.089	±0.089	±0.115	±0.115
Transmission Tracking	±0.024	±0.033	±0.082	±0.103

Transmission Uncertainty (Specifications)

Magnitude

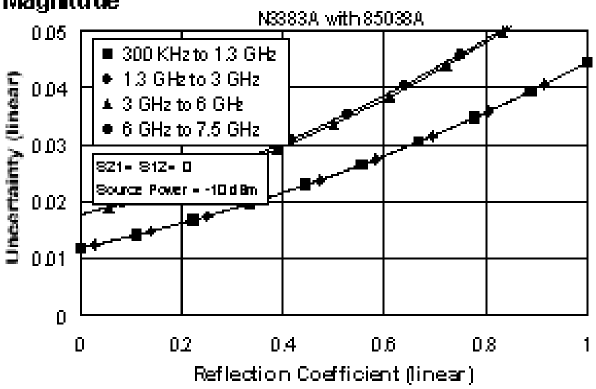


Phase



Reflection Uncertainty (Specifications)

Magnitude



Phase

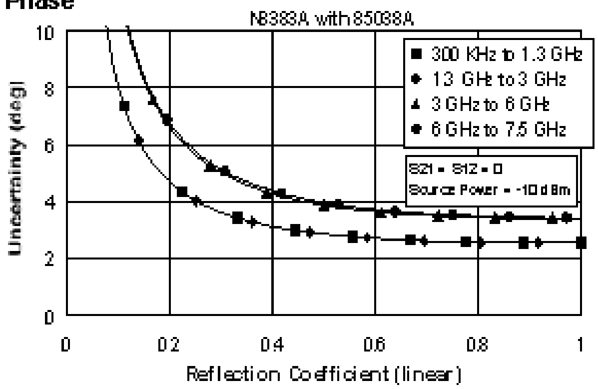


Table 7. Uncorrected Instrument Performance

Description	Specification (dB)				
	300 kHz to 1 MHz	1MHz to 1.3 GHz	1.3 GHz to 3 GHz	3 GHz to 6 GHz	6 GHz to 9 GHz
Directivity	30	33	27	20	13
Source Match Ports 1 and 2	18	18	17	14	12
Source Match Port 3 only	18	18	17	14	12
Load Match Ports 1 and 2	20	20	17	13.5	11.5
Load Match Port 3 only	20	20	17	13.5	11.5
Reflection Tracking	±1.5	±1.5	±1.5	±2.5	±3.0
Transmission Tracking	±1.5	±1.5	±1.5	±2.5	±3.0

Test Port Output Characteristics (Source)

- Output Frequency
- Output Power
- Output Signal Purity

Table 8. Test Port Output Frequency

Description	Specification	Supplemental Information
Range:		
N3381A	300 kHz to 3.0 GHz	
N3382A	300 kHz to 6.0 GHz	
N3383A	300 kHz to 9.0 GHz	
Resolution	1 Hz	
Source Stability		±1 ppm, 0° to 40 °C, typical ±1ppm/year maximum
Source Stability (Option 1E5)		±0.05 ppm, 0° to 40 °C, typical ±0.1 ppm/year maximum
CW Accuracy	±3 ppm	

CW Accuracy (Option 1E5)	±1 ppm	

Table 9. Test Port Output Power[a]

Description	Specification	Supplemental Information
Level Accuracy		
		Variation from 0 dBm in power range 0 (step attenuator at 0 dB) ±1.5dB below 10 MHz
300 kHz to 6 GHz	±1.0 dB	
6 GHz to 9 GHz	±2.0 dB	
Level Linearity		
		Variation from 0 dBm in power range 0
300 kHz to 9 GHz	±0.3 dB	-15 to +5 dBm
300 kHz to 1 MHz	±1.0 dB	+5 to +10 dBm
1 MHz to 6 GHz	±0.5 dB	+5 to +10 dBm
6 GHz to 9 GHz	±0.5 dB	+5 to +7 dBm
Range[b]		
300 kHz to 6 GHz	-15 to +10 dBm	
6 GHz to 9 GHz	-15 to +7 dBm	
Range[b] (Option 1E1)		
300 kHz to 6 GHz	-85 to +10 dBm	
6 GHz to 9 GHz	-85 to +7 dBm	
Sweep Range		
Port 1:		
300 kHz to 6 GHz	25 dB	
6 GHz to 9 GHz	22 dB	
Level Resolution	0.01 dB	

a Source output performance on port 1 only. Port 2 and port 3 output performance is typically 3 dB less.

b Power to which the source can be set and phase lock is assured.

Table 10. Test Port Output Signal Purity

Description	Specification	Supplemental Information
Harmonics (2nd or 3rd)		
at max output power (< 25 MHz)		< -25 dBc, typical
at max output power (25 MHz to 9 GHz)		< -25 dBc, characteristic[a]
at 0 dBm output		< -35 dBc, typical
at -10 dBm output		< -38 dBc, typical, in power range 0
Non-harmonic Spurious		
at max output		-30 dBc, typical for offset freq>1kHz
at -10 dBm output		-50 dBc, typical for offset freq >1kHz

a Typical below 25 MHz.

Test Port and Receiver Input Characteristics

- **Input Levels**
- **Trace Noise**
- **Reference Level and Stability**
- **Dynamic Accuracy**
- **Group Delay**

Table 11. Test Port and Receiver Input Levels

Description	Specification	Supplemental Information
Maximum Test Port Input Level		
300 kHz to 25 MHz	+10 dBm	<0.6 dB compression
25 MHz to 3 GHz	+10 dBm	<0.4 dB compression
3 GHz to 6 GHz	+10 dBm	<0.7 dB compression
6 GHz to 9 GHz	+7 dBm	<0.7 dB compression
Damage Level		
Test Port 1, 2. 3		+30 dBm or ±30 VDC, typ.
R, A, B, C (Opt. 014)		+15 dBm or ±5 VDC, typ.
Coupler Thru (Opt. 014)		+33 dBm or ±0 VDC, typ.

Test Port Noise Floor[a]		
300 kHz to 25 MHz[b]		
10 Hz IF Bandwidth	= -115 dBm	
1 kHz IF Bandwidth	= -95 dBm	
25 MHz to 3 GHz[b]		
10 Hz IF Bandwidth	= -118 dBm	
1 kHz IF Bandwidth	= -98 dBm	
3 GHz to 9 GHz		
10 Hz IF Bandwidth	≤ -108 dBm	
1 kHz IF Bandwidth	≤ -88 dBm	
Receiver Noise Floor[a]		
300 kHz to 25 MHz[c]		
10 Hz IF Bandwidth	≤ -130 dBm, characteristic	
1 kHz IF Bandwidth	≤ -110 dBm, characteristic	
25 MHz to 3 GHz[c]		
10 Hz IF Bandwidth	= -133 dBm, characteristic	
1 kHz IF Bandwidth	= -113 dBm, characteristic	
6 GHz to 9 GHz		
10 Hz IF Bandwidth	≤ -123 dBm, characteristic	
1 kHz IF Bandwidth	≤ -103 dBm, characteristic	
Crosstalk		
(S_{21}, S_{31}):		
300 kHz to 1 MHz	<-120 dB	
1 MHz to 25 MHz	<-125 dB	
25 MHz to 3 GHz	<-126 dB	
3 GHz to 6 GHz	<-117 dB	

6 GHz to 9 GHz	<-106 dB	
(S_{12}, S_{13}):		
300 kHz to 1 MHz	<-120 dB	
1 MHz to 25 MHz	<-125 dB	
25 MHz to 3 GHz	<-126 dB	
3 GHz to 6 GHz	<-113 dB	
6 GHz to 9 GHz	<-106 dB	
(S_{23}, S_{32}):		
300 kHz to 1 MHz	<-120 dB	
1 MHz to 3 GHz	<-125 dB	
3 GHz to 6 GHz	<-115 dB	
6 GHz to 9 GHz	<-107 dB	
Maximum Receiver Input Level (A, B, R, C)		
300 kHz to 6 GHz		dBm, typical
6 GHz to 9 GHz		dBm, typical
Reference Input Level (R)[d]		
300 kHz to 9 GHz		-10 to -35 dBm, typical
Maximum Coupler Input Level (Opt 014)		
300 kHz to 9 GHz		+33 dBm, typical

a Total average (RMS) noise power calculated as the mean value of a linear magnitude trace expressed in dBm.
b May be limited to -90 dBm at particular frequencies below 750 MHz due to spurious receiver residuals.
c May be limited to -105 dBm at particular frequencies below 750 MHz due to spurious receiver residuals.

d Input level to maintain phase lock.

Table 12. Test Port Input (Trace Noise)

Description	Specification	Supplemental Information
Trace Noise[a] Magnitude		
1 kHz IF Bandwidth	< 0.002 dB rms	
10 kHz IF Bandwidth	< 0.005 dB rms	
Trace Noise[a] Phase		
1 kHz IF Bandwidth	< 0.010° rms	
10 kHz IF Bandwidth	< 0.035° rms	

a Trace noise is defined as a ratio measurement of a through or a full reflection, with the source set to 0 dBm.

Table 13. Test Port Input (Reference Level and Stability)

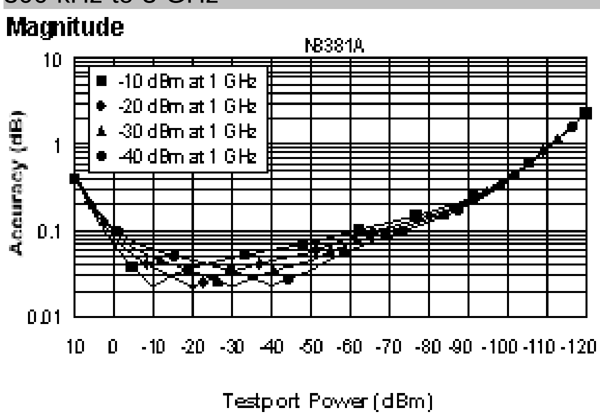
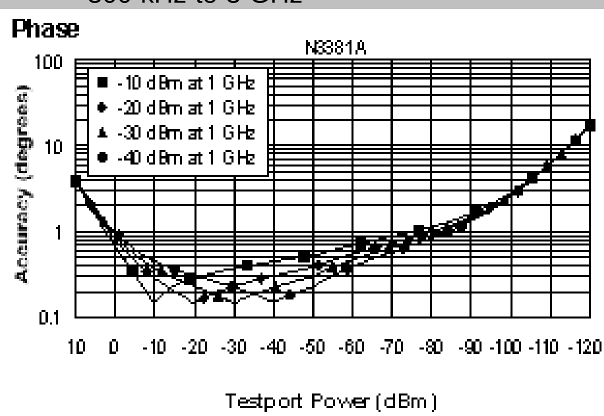
Description	Specification	Supplemental Information
Reference Level Magnitude		
Range	±200 dB	
Resolution	0.001 dB	
Reference Level Phase		
Range	±500°	
Resolution	0.01°	
Stability Magnitude[a]		
300 kHz to 3 GHz		0.02 dB/°C, typical
3 GHz to 6 GHz		0.04 dB/°C, typical
6 GHz to 9 GHz		0.06 dB/°C, typical
Stability Phase[a]		
300 kHz to 3 GHz		0.2°/°C, typical
3 GHz to 6 GHz		0.3°/°C, typical
6 GHz to 9 GHz		0.6°/°C, typical

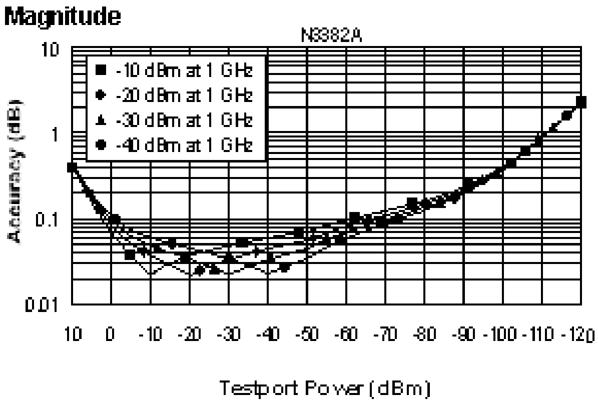
a Stability is defined as a ratio measurement at the test port.

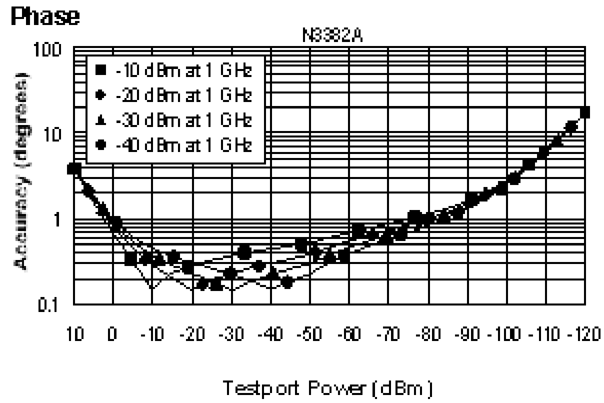
Table 14. Test Port Input (Dynamic Accuracy specification[a])

Accuracy of the test port input power reading is relative to the reference input power level. Applies to input ports 1 and 2 with the following conditions:

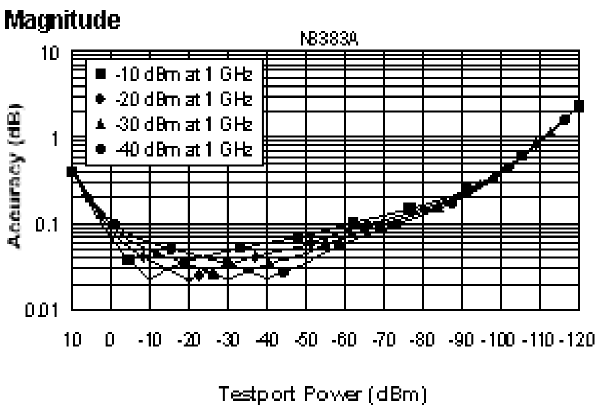
- IF bandwidth = 10 Hz
- Environmental temperature 25° ±5 °C, with < 1 °C deviation from calibration temperature

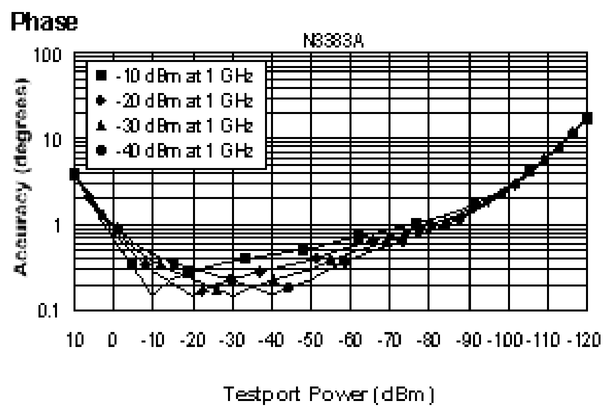
300 kHz to 3 GHz	300 kHz to 3 GHz
	
300 kHz to 6 GHz	300 kHz to 6 GHz





300 kHz to 9 GHz 300 kHz to 9 GHz



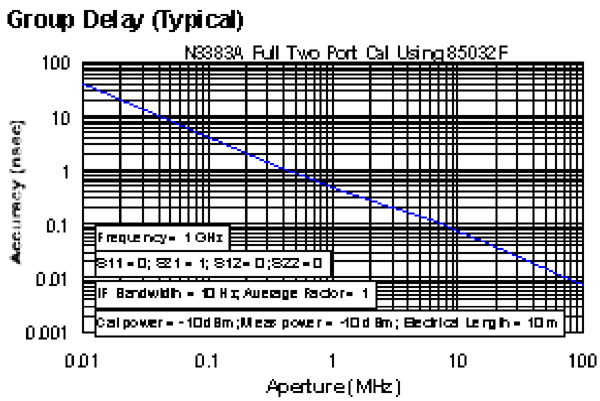


aDynamic accuracy is verified with the following measurements:

- compression over frequency
- IF linearity at a single frequency of 1.195 GHz and a reference level of -20 dBm

Table 15. Test Port Input (Group Delay)[a]

Description	Specification	Supplemental Information
Aperture (selectable)	(frequency span)/(number of points -1)	
Maximum Aperture	20% of frequency span	
Range	0.5 x (1/minimum aperture)	
Maximum Delay		Limited to measuring no more than 180° of phase change within the minimum aperture.)
Accuracy		See graph below. Char.

The following graph shows group delay accuracy with type-N full 2-port calibration and a 10 Hz IF bandwidth. Insertion loss is assumed to be < 2 dB and electrical length to be ten meters. 
In general, the following formula can be used to determine the accuracy, in seconds, of specific group delay measurement: ±Phase Accuracy (deg)/[360 × Aperture (Hz)] Depending on the aperture and device length, the phase accuracy used is either incremental phase accuracy or worst case phase accuracy.

a Group delay is computed by measuring the phase change within a specified frequency step (determined by the frequency span and the number of points per sweep).

General Information

- **System Bandwidths**
- **Front Panel**
- **Rear Panel**
- **Environment and Dimensions**

Table 16. System Bandwidths

Description	Specification	Supplemental Information
IF Bandwidth Settings		
Range		1 Hz to 40 kHz in a 1, 2, 3, 5, 7,10 sequence up to 30 kHz, 35 kHz, 40kHz, nominal

Table 17. Front Panel Information

Description	Supplemental Information
RF Connectors	
Type	Type-N, female; 50 Ω, nominal
Center Pin Protrusion	0.204 to 0.207 in., characteristic
Probe Power	
Connector	3-pin connector, male
Positive Supply	+15 VDC ±2%, 400 mA, max, characteristic
Negative Supply	-12.6 VDC ±5%, 300 mA, max, characteristic
Display	
Size	21.3 cm (8.4 in) diagonal color active matrix LCD; 640 (horizontal) X 480 (vertical) resolution
Refresh Rate	Vertical 59.83 Hz; Horizontal 31.41 Hz
Display Range	
Magnitude	±200 dB (at 20 dB/div), max
Phase	±180°, max
Polar	10 pUnits, min 1000 Units, max
Display Resolution	
Magnitude	0.001 dB/div, min
Phase	0.01°/div, min
Marker Resolution	
Magnitude	0.001 dB, min
Phase	0.01°, min
Polar	0.01 mUnit, min; 0.01°,min

Table 18. Rear Panel Information

Description	Supplemental Information
10 MHz Reference In	
Connector	BNC, female
Input Frequency	10 MHz ± 1 ppm, typical
Input Level	-15 dBm to +20 dBm, typical
Input Impedance	200 Ω, nom.

10 MHz Reference Out	
Connector	BNC, female
Output Frequency	10 MHz ± 10 ppm, typical
Signal Type	Sine Wave, typical
Output Level	+10 dBm ± 4 dB into 50 Ω, typical
Output Impedance	50 Ω, nominal
Harmonics	<-40 dBc, typical
VGA Video Output	
Connector	15-pin mini D-Sub; Drives VGA compatible monitors
Devices Supported:	Resolutions:
Flat Panel (TFT)	1024 X 768, 800 X 600, 640 X 480
Flat Panel (DSTN)	800 X 600, 640 X 480
CRT Monitor	1280 X 1024, 1024 X 768, 800 X 600, 640 X 480
	Simultaneous operation of the internal and external displays is allowed, but with 640 X 480 resolution only. If you change resolution, you can only view the external display (internal display will "white out").
Test Set IO	25-pin D-Sub connector, available for external test set control
Aux IO	25-pin D-Sub connector, male, analog and digital IO
Handler IO	36-pin IDC D-ribbon socket connector; all input/output signals are default set to negative logic; can be reset to positive logic via GPIB command
GPIB	24-pin D-sub (Type D-24), female; compatible with IEEE-488.
Parallel Port (LPT1)	25-pin D-Sub connector, female; provides connection to printers or any other parallel port peripherals
Serial Port (COM 1)	9-pin D-Sub, male; compatible with RS-232
USB Port	
	Universal Serial Bus jack, Type A configuration (4 contacts inline, contact 1 on left); female
Contact 1	Vcc: 4.75 to 5.25 VDC, 500 mA, maximum
Contact 2	-Data
Contact 3	+Data
Contact 4	Ground

LAN	10/100BaseT Ethernet, 8-pin configuration; auto selects between the two data rates
Line Power (A third-wire ground is required.)	
Frequency	50/60/400 Hz
Voltage	120/240 VAC (Power supply is auto switching.)
Max	500 Watts

Note: Option H08 and Option H11 specifications are not provided in this N3381/2/3A specifications document.

Table 19. Rear Panel Information (continued)

Description	**Supplemental Information**
External AM Input	
Description	Input provides low-frequency AM modulation to test port output signal, or shifts the test port output. Zero volts input gives the power level set by the instrument, a positive voltage gives a higher level, and a negative voltage gives a lower level.
Connector	BNC, female
Input Sensitivity	8 dB/V, typical
Bandwidth	1 kHz, typical
Input Impedance	1 kΩ, nominal
External Detector Input	
Description	Input from an external, negative polarity diode detector provides ALC for a test port remote from instrument's front panel
Connector	BNC, female
Input Sensitivity	-500 mV yields approximately -3 dBm at detector's input, typical
Bandwidth	50 kHz, typical
Input Impedance	1 kΩ, typical

Table 20. Analyzer Environment and Dimensions

Description	Supplemental Information
General Environmental	
RFI/EMI Susceptibility	Defined by CISPR Pub. 11, Group 1, Class A, and IEC 50082-1
ESD	Minimize using static-safe work procedures and an antistatic bench mat
Dust	Minimize for optimum reliability
Operating Environment	
Temperature	0 °C to +40 °C Instrument powers up, phase locks, and displays no error messages within this temperature range.
Error-Corrected Temperature Range	25°C ± 5°C with less than 1°C deviation from calibration temp.
Humidity	5% to 95% at +40 °C
Altitude	0 to 4500 m (14,760 ft.)
Non-Operating Storage Environment	
Temperature	-40 °C to +70 °C
Humidity	0% to 90% at +65 °C (non-condensing)
Altitude	0 to 15,240 m (50,000 ft.)

Cabinet Dimensions			
	Height	**Width**	**Depth**
Excluding front and rear panel hardware and feet	223 mm 8.75 in	426 mm 16.75 in	427 mm 16.8 in
As shipped - includes front panel connectors, rear panel bumpers, and feet.	235 mm 9.25 in	435 mm 17.10 in	470 mm 18.5 in
As shipped plus handles	235 mm 9.25 in	458 mm 18 in	501 mm 19.70 in
As shipped plus rack-mount flanges	235 mm 9.25 in	483 mm 19 in	470 mm 18.5 in
As shipped plus handles and flanges	235 mm 9.25 in	483 mm 19 in	501 mm 19.70 in
Weight			
Net	24 kg (54 lb), nominal		
Shipping	32 kg (70 lb), nominal		

Note: "Misc. Information" specifications are not included in this N3381/2/3A document.

Measurement Throughput Summary

- **Typical Cycle Time for Measurement Completion**
- **Cycle Time with Changes in IF Bandwidth**
- **Cycle Time with Changes in Number of Points**
- **Data Transfer Time**
- **Recall and Sweep Speed**

Table 21. Typical Cycle Time[a,b] (ms)

	Number of Points			
	101	**201**	**401**	**1601**
Start 1.8 GHz, Stop 2 GHz, 35 kHz IF bandwidth				
Uncorrected, 1-port cal	8	11	17	53
2-Port cal	27	36	55	164
Start 300 kHz, Stop 3 GHz, 35 kHz IF bandwidth				
Uncorrected, 1-port cal	48	54	64	104
2-Port cal	103	119	145	254
Start 300 kHz, Stop 9 GHz, 35 kHz IF bandwidth				
Uncorrected, 1-port cal	45	55	61	99
2-Port cal	99	119	133	212

a Typical performance.
b Includes sweep time, retrace time and band-crossing time. Analyzer display turned off with DISPLAY:ENABLE OFF. Add 21 ms for display on. Data for one trace (S11) measurement.
c Option 010 only. Analyzer display turned off with DISPLAY:ENABLE OFF. Add 21 ms for display on. Data for one trace (S11) measurement.

Table 22. Cycle Time vs. IF Bandwidth[a]

Applies to the Preset condition (201 points, correction off) except for the following changes:

- CF = 1 GHz
- Span = 100 MHz
- Display off (add 21 ms for display on)

IF Bandwidth (Hz)	Cycle Time (ms)[b]
40,000	8
35,000	9
30,000	11
20,000	13
10,000	28
7000	36
5000	48
3000	72
1000	196
300	620
100	1875
30	8062
10	17877

a Typical performance.
b Cycle time includes sweep and retrace time.

Table 23. Cycle Time vs. Number of Points[a]

Applies to the Preset condition (35 kHz IF bandwidth, correction off) except for the following changes:

- CF = 1 GHz
- Span = 100 MHz
- Display off (add 21 ms for display on)

Number of Points	Cycle Time (ms)[b]
3	4
11	4
51	5
101	6
201	9
401	16
801	29
1601	52

a Typical performance.
b Cycle time includes sweep and retrace time.

Table 24. Data Transfer Timea (ms)

	Number of Points			
	51	**201**	**401**	**1601**
SCPI over GPIB (program executed on external PC)[b]				
32-bit floating point	3	7	12	43
64-bit floating point	4	12	22	84
ASCII	7	64	124	489
SCPI over 100 Mbit/s LAN (program executed on external PC)[b]				
32-bit floating point	1	1	1	1
64-bit floating point	1	1	1	2
ASCII	5	15	26	96
SCPI (program executed in the analyzer)[d]				
32-bit floating point	1	1	2	3
64-bit floating point	1	2	2	4
ASCII	8	29	56	222
COM (program executed in the analyzer)[e]				
32-bit floating point	1	1	1	1
Variant type	1	1	2	6
DCOM over 100 Mbit/s LAN (program executed on external PC)[f]				
32-bit floating point[g]	1	1	1	2
Variant type[h]	1	3	6	19

a Typical performance of unit with 500 MHz Pentium III processor.

b Measured using a VEE 5.0 program running on a 600 MHz HP Kayak, National InstrumentsTM GPIB card. Transferred complex S11 data , using "CALC:DATA?SDATA".

c Measured using a VEE 5.0 program running on a 600 MHz HP Kayak. Transferred complex S11 data, using "CALC:DATA?SDATA". Speed dependent on LAN traffic, if connected to network.

d Measured using a VEE 5.0 program running inside PNA Series Analyzer. Transferred complex S11 data, using "CALC:DATA?SDATA".

e Measured using a Visual Basic 6.0 program running inside PNA Series Analyzer. Transferred complex S11 data.

f Measured using a Visual Basic 6.0 program running on a 600 MHz HP Kayak. Transferred complex S11 data. Speed dependent on LAN traffic, if connected to network.

g Used IArray Transfer.getComplex method for 32-bit floating point.

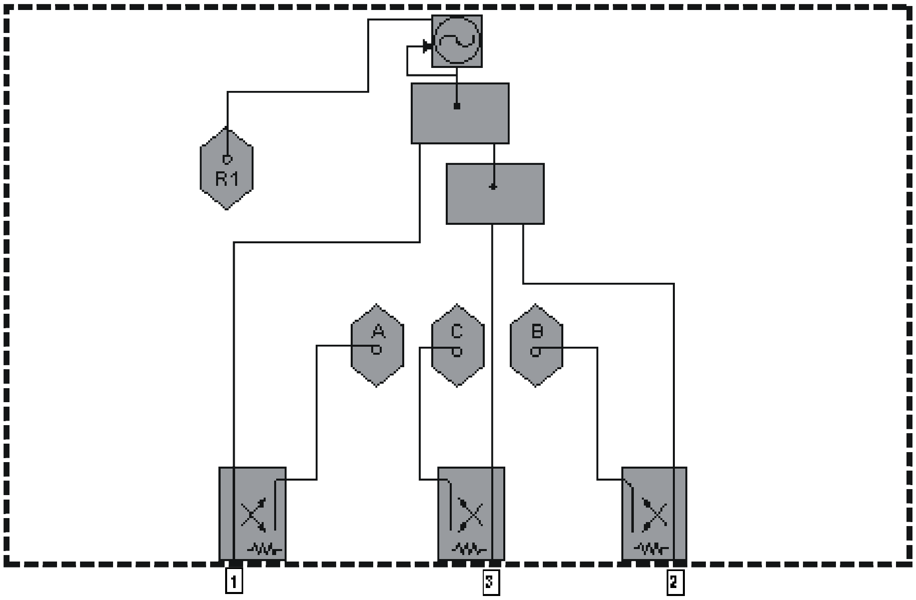
h Used meas.getData method for Variant data type.

Table 25. Recall and Sweep Speed[a]

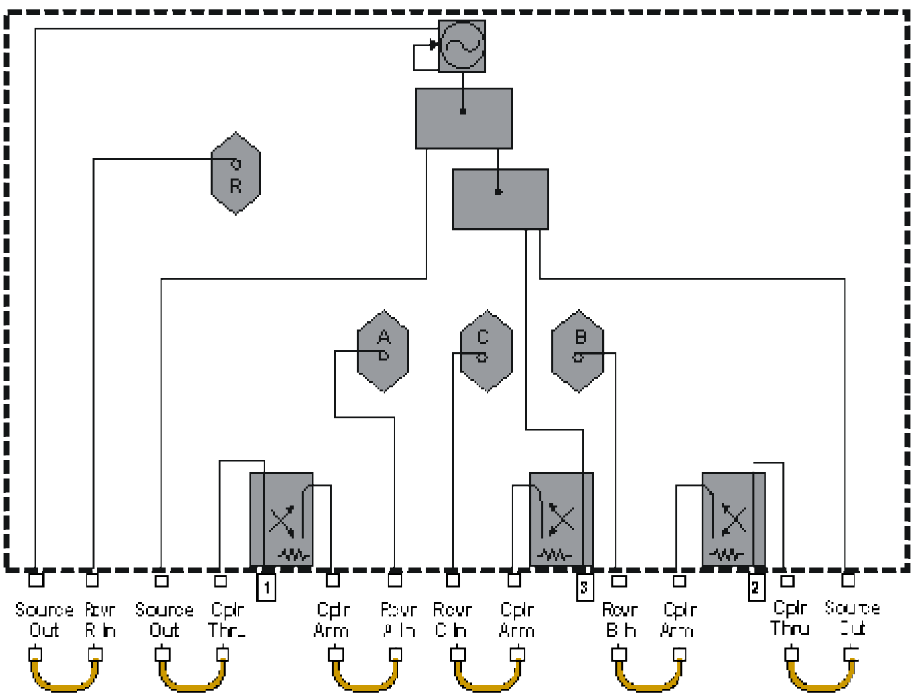
Operations	**Number of Window(s)**	**Number of Trace(s)**	**Recall Time (ms)**
Recall	1	1	49
Recall and Sweep	1	1	59
Recall	1	2	82
Recall and Sweep	1	2	96
Recall	1	4	159
Recall and Sweep	1	4	203
Recall	2	2	93
Recall and Sweep	2	2	115
Recall	3	4	158
Recall and Sweep	3	4	218
Recall	4	4	187
Recall and Sweep	4	4	247
Recall	4	8	340
Recall and Sweep	4	8	507

aCF=177MHz, Span=200 MHz, 201 points, 35 kHz IF BW

N3381A, N3382A, and N3383A – Simplified Test Set Block Diagram



N3381A, N3382A, and N3383A with Option 014 – Simplified Test Set Block Diagram



Technical Specifications for the E8361A

(Rev. 2005-07-12)

This is a complete list of the E8361A network analyzer technical specifications.

- To optimize viewing of uncertainty curves, click the Maximize button.

- To view or print the .pdf version of the specifications, visit our web site at http://www.agilent.com/find/pna, and search for "E8361A Specifications"

- The uncertainty curves contained in this document apply only to the setup conditions listed. Please download our free Uncertainty Calculator from http://www.agilent.com/find/na_calculator to generate the curves for your PNA setup. View the equations used to generate the uncertainty curves.

Definitions

Corrected System Performance

- **System Dynamic Range**
- **Receiver Dynamic Range**
- **1.85mm Connectors**
- **2.4mm Connectors**

Uncorrected System Performance

Test Port Output

Test Port Input

Dynamic Accuracy

Group Delay

General Information

Measurement Throughput Summary

Front-panel Jumper Specs (Option 014 only)

Test Set Block Diagrams

Test Set with Option 014 Block Diagrams

See Specs for other PNA models

Definitions

All specifications and characteristics apply over a 25 °C ±5 °C range (unless otherwise stated) and 90 minutes after the instrument has been turned on.

Specification (spec.): Warranted performance. Specifications include guardbands to account for the expected statistical performance distribution, measurement uncertainties, and changes in performance due to environmental conditions.

Characteristic (char.): A performance parameter that the product is expected to meet before it leaves the factory, but that is not verified in the field and is not covered by the product warranty. A characteristic includes the same guardbands as a specification.

Typical (typ.): Expected performance of an average unit which does not include guardbands. It is not covered by the product warranty.

Nominal (nom.): A general, descriptive term that does not imply a level of performance. It is not covered by the product warranty.

Calibration: The process of measuring known standards to characterize a network analyzer's systematic (repeatable) errors.

Corrected (residual): Indicates performance after error correction (calibration). It is determined by the quality of calibration standards and how well "known" they are, plus system repeatability, stability, and noise.

Uncorrected (raw): Indicates instrument performance without error correction. The uncorrected performance affects the stability of a calibration.

Standard: When referring to the analyzer, this includes no options unless noted otherwise.

Corrected System Performance

The specifications in this section apply for measurements made with the E8361A analyzer with the following conditions:

- 10 Hz IF bandwidth
- No averaging applied to data

- **System Dynamic Range**
- **Receiver Dynamic Range**
- **1.85mm Connectors**
- **2.4mm Connectors**

Table 1. System Dynamic Range[a]

Description	Specification (dB) at Test Port[b]	Typical (dB) at Direct Receiver Access Input[c]	Supplemental Information
Dynamic Range (in a 10 Hz BW)			
Standard Configuration (E8361A - Standard)			
10 MHz to 45 MHz[d]	61	NA	
45 MHz to 500 MHz[e]	87	NA	
500 MHz to 750 MHz	112	NA	

750 MHz to 2 GHz	111	NA	
2 GHz to 10 GHz	111	NA	
10 GHz to 24 GHz	114	NA	
24 GHz to 30 GHz	103	NA	
30 GHz to 40 GHz	104	NA	
40 GHz to 45 GHz	96	NA	
45 GHz to 50 GHz	100	NA	
50 GHz to 60 GHz	97	NA	
60 GHz to 67 GHz	94	NA	
67 GHz to 70 GHz[d]	94	NA	
Configurable Test Set **(E8361A - Option 014 or Option 014 and 080)**			
10 MHz to 45 MHz[d]	61	99	
45 MHz to 500 MHz[e]	87	102	
500 MHz to 750 MHz	112	125.5	
750 MHz to 2 GHz	111	125.5	
2 GHz to 10 GHz	111	125	
10 GHz to 24 GHz	112	128	
24 GHz to 30 GHz	101	117.5	Option 016 degrades performance by 2 dB.
30 GHz to 40 GHz	102	115	
40 GHz to 45 GHz	94	109	
45 GHz to 50 GHz	98	110.5	
50 GHz to 60 GHz	95	107	
60 GHz to 67 GHz	90	101	Option 016 degrades performance by 3 dB.
67 GHz to 70 GHz[d]	90	100	
Configurable Test Set & Extended Power Range **(E8361A - Option 014 & UNL or Option 014 & UNL & 080)**			
10 MHz to 45 MHz[d]	61	99	
45 MHz to 500 MHz[e]	87	102	
500 MHz to 750 MHz	112	125.5	
750 MHz to 2 GHz	111	124	

2 GHz to 10 GHz	111	124	
10 GHz to 24 GHz	112	125	
24 GHz to 30 GHz	101	114.5	Option 016 degrades performance by 2 dB.
30 GHz to 40 GHz	99	112	
40 GHz to 45 GHz	92	105	
45 GHz to 50 GHz	94	106.5	
50 GHz to 60 GHz	91	103	
60 GHz to 67 GHz	84	95	Option 016 degrades performance by 3 dB.
67 GHz to 70 GHz[d]	84	94	

a The system dynamic range is calculated as the difference between the noise floor and the source maximum output power. System Dynamic Range is a specification when the source is set to Port 1, and a characteristic when the source is set to Port 2. The effective dynamic range must take measurement uncertainties and interfering signals into account as well as the insertion loss resulting from a thru cable connected between Port 1 and Port 2..

b The test port system dynamic range is calculated as the difference between the test port noise floor and the source maximum output power. The effective dynamic range must take measurement uncertainties and interfering signals into account as well as the insertion loss resulting from a thru cable connected between Port 1 and Port 2..

c The direct receiver access input system dynamic range is calculated as the difference between the receiver access input noise floor and the source maximum output power. The effective dynamic range must take measurement uncertainties and interfering signals into account. This set-up should only be used when the receiver input will never exceed its damage level. When the analyzer is in segment sweep mode, the analyzer can have predefined frequency segments which will output a higher power level when the extended dynamic range is required (i.e. devices with high insertion loss), and reduced power when receiver damage may occur (i.e. devices with low insertion loss). The extended range is only available in one-path transmission measurements.

d Typical performance.

e May be limited to 100 dB at particular frequencies below 500 MHz due to spurious receiver residuals. Methods are available to regain the full dynamic range.

Note: This E8361A document does NOT provide technical specifications for Receiver Dynamic Range.

Note: This E8361A document provides technical specifications for the following calibration kits and Ecal modules only: 85056A, 85058B, N4693A, N4694A. Please download our free Uncertainty Calculator from http://www.agilent.com/find/na_calculator to generate the curves for your PNA setup.

Table 10. Uncorrected System Performance[a]

Specifications apply over environmental temperature of 23° ±3 °C, with < 1 °C deviation from the calibration temperature

Description	Specification	Supplemental Information
Directivity		
		typical:
10 MHz to 45 MHz[b]	22 dB	22 dB
45 MHz to 2 GHz	24 dB	27 dB
2 GHz to 10 GHz	20 dB	24 dB
10 GHz to 20 GHz	16 dB	20 dB
20 GHz to 30 GHz	14 dB	17 dB

30 GHz to 50 GHz	13 dB	17 dB
50 GHz to 60 GHz	13 dB	17 dB
60 GHz to 67 GHz	10 dB	18 dB
67 GHz to 70 GHz[b]	14 dB	14 dB
Source Match - Standard		
		typical:
10 MHz to 45 MHz[b]	7 dB	7 dB
45 MHz to 2 GHz	18 dB	23 dB
2 GHz to 10 GHz	14 dB	18 dB
10 GHz to 20 GHz	12 dB	15 dB
20 GHz to 30 GHz	8 dB	11.5 dB
30 GHz to 40 GHz	7.5 dB	10 dB
40 GHz to 45 GHz	8 dB	11 dB
45 GHz to 50 GHz	7 dB	10 dB
50 GHz to 60 GHz	6 dB	8.5 dB
60 GHz to 67 GHz	5.5 dB	7.5 dB
67 GHz to 70 GHz[b]	7.5 dB	7.5 dB
Source Match - Option 014		
		typical:
10 MHz to 45 MHz[b]	7 dB	7 dB
45 MHz to 2 GHz	17 dB	21 dB
2 GHz to 10 GHz	12 dB	17 dB
10 GHz to 20 GHz	11 dB	14 dB
20 GHz to 30 GHz	10 dB	13 dB
30 GHz to 40 GHz	8.5 dB	11 dB
40 GHz to 45 GHz	8.5 dB	11 dB
45 GHz to 50 GHz	8.5 dB	11.5 dB

50 GHz to 60 GHz	6.5 dB	9 dB
60 GHz to 67 GHz	6 dB	8.5 dB
67 GHz to 70 GHz[b]	8.5 dB	8.5 dB
Source Match - UNL & 014		
		typical:
10 MHz to 45 MHz[b]	5 dB	5 dB
45 MHz to 2 GHz	15 dB	20 dB
2 GHz to 10 GHz	9 dB	13 dB
10 GHz to 20 GHz	7.5 dB	10.5 dB
20 GHz to 30 GHz	8.5 dB	11 dB
30 GHz to 40 GHz	8 dB	11 dB
40 GHz to 45 GHz	8.5 dB	12 dB
45 GHz to 50 GHz	8 dB	12 dB
50 GHz to 60 GHz	7 dB	11 dB
60 GHz to 67 GHz	6 dB	10 dB
67 GHz to 70 GHz[b]	10 dB	10 dB
Load Match - Standard		
		typical:
10 MHz to 45 MHz[b]	5.5 dB	5.5 dB
45 MHz to 2 GHz	9 dB	10 dB
2 GHz to 10 GHz	9 dB	11 dB
10 GHz to 20 GHz	8.5 dB	10 dB
20 GHz to 30 GHz	7 dB	9 dB
30 GHz to 40 GHz	6 dB	8 dB
40 GHz to 45 GHz	6.5 dB	9 dB
45 GHz to 50 GHz	6.5 dB	8.5 dB
50 GHz to 60 GHz	5.5 dB	7.5 dB

60 GHz to 67 GHz	5.5 dB	7.5 dB
67 GHz to 70 GHz[b]	5 dB	5 dB
Load Match - Option 014		
		typical:
10 MHz to 45 MHz[b]	5.5 dB	5.5 dB
45 MHz to 2 GHz	8.5 dB	10 dB
2 GHz to 10 GHz	8 dB	10 dB
10 GHz to 20 GHz	8 dB	10 dB
20 GHz to 30 GHz	7.5 dB	10 dB
30 GHz to 40 GHz	7 dB	9.5 dB
40 GHz to 45 GHz	7.5 dB	9.5 dB
45 GHz to 50 GHz	7.5 dB	10 dB
50 GHz to 60 GHz	6 dB	8.5 dB
60 GHz to 67 GHz	6 dB	8.5 dB
67 GHz to 70 GHz[b]	5 dB	5 dB
Load Match - Option UNL & 014		
		typical:
10 MHz to 45 MHz[b]	6 dB	6 dB
45 MHz to 2 GHz	8.5 dB	10 dB
2 GHz to 10 GHz	7 dB	9 dB
10 GHz to 20 GHz	6 dB	9 dB
20 GHz to 30 GHz	7.5 dB	11 dB
30 GHz to 40 GHz	8 dB	11.5 dB
40 GHz to 45 GHz	8 dB	12 dB
45 GHz to 50 GHz	8 dB	12 dB
50 GHz to 60 GHz	7.5 dB	11.5 dB
60 GHz to 67 GHz	6 dB	10 dB

67 GHz to 70 GHz[b]	13 dB	13 dB
Reflection Tracking		
		typical:
10 MHz to 45 MHz		±1.5 dB
45 MHz to 20 GHz		±1.5 dB
20 GHz to 40 GHz		±2.0 dB
40 GHz to 50 GHz		±2.0 dB
50 GHz to 67 GHz		±3.0 dB
67 GHz to 70 GHz		±4.5 dB
Transmission Tracking[c]		
		typical:
10 MHz to 45 MHz		±1.5 dB
45 MHz to 20 GHz		±1.5 dB
20 GHz to 40 GHz		±2.0 dB
40 GHz to 50 GHz		±2.0 dB
50 GHz to 67 GHz		±3.0 dB
67 GHz to 70 GHz		±4.5 dB
Crosstalk[d] - Standard		
10 MHz to 45 MHz[b]	-63 dB	
45 MHz to 500 MHz	-87 dB	
500 MHz to 2 GHz	-110 dB	
2 GHz to 10 GHz	-105 dB	
10 GHz to 24 GHz	-111 dB	
24 GHz to 30 GHz	-106 dB	
30 GHz to 40 GHz	-104 dB	
40 GHz to 45 GHz	-98 dB	
45 GHz to 50 GHz	-100 dB	

50 GHz to 60 GHz	-97 dB	
60 GHz to 67 GHz	-94 dB	
67 GHz to 70 GHz[b]	-94 dB	
Crosstalk[d] - Option 014		
		typical (for Option 080 enabled[e])
10 MHz to 45 MHz[b]	-63 dB	-63 dB
45 MHz to 500 MHz	-87 dB	-87 dB
500 MHz to 2 GHz	-110 dB	-110 dB
2 GHz to 10 GHz	-105 dB	-105 dB
10 GHz to 24 GHz	-111 dB	-111 dB
24 GHz to 30 GHz	-104 dB	-104 dB
30 GHz to 40 GHz	-102 dB	-102 dB
40 GHz to 45 GHz	-96 dB	-96 dB
45 GHz to 50 GHz	-98 dB	-98 dB
50 GHz to 60 GHz	-95 dB	-95 dB
60 GHz to 67 GHz	-90 dB	-90 dB
67 GHz to 70 GHz[b]	-90 dB	-90 dB
Crosstalk[d] - Option 014 & UNL		
		typical (for Option 080 enabled[e])
10 MHz to 45 MHz[b]	-63 dB	-63 dB
45 MHz to 500 MHz	-87 dB	-87 dB
500 MHz to 2 GHz	-110 dB	-110 dB
2 GHz to 10 GHz	-104 dB	-104 dB
10 GHz to 24 GHz	-108 dB	-108 dB
24 GHz to 30 GHz	-101 dB	-101 dB
30 GHz to 40 GHz	-99 dB	-99 dB
40 GHz to 45 GHz	-92 dB	-92 dB

45 GHz to 50 GHz	-94 dB	-94 dB
50 GHz to 60 GHz	-91 dB	-91 dB
60 GHz to 67 GHz	-84 dB	-84 dB
67 GHz to 70 GHz[b]	-84 dB	-84 dB

a Specifications apply over environment temperature of 23°C +/- 3°C, with less than 1°C deviation from the calibration temperature.

b Typical performance.

cTransmission tracking performance noted here is normalized to the insertion loss characteristics of the cable used so that the indicated performance is independent of the cable used.

d Measurement conditions: normalized to a thru, measured with two shorts, 10 Hz IF bandwidth, averaging factor of 16, alternate mode, source power set to the lesser of the maximum power out or the maximum receiver power.

e 0 Hz offset.

Table 11. Test Port Output

Description	Specification			Supplemental
Frequency Range				
	Standard	**Options**		
E8361A	45 MHz to 67 GHz (Operates 10 MHz to 70 GHz)			
Nominal Power[c]				
	Std and Options without UNL	**Options with UNL**		
E8361A	-15 dBm	-17 dBm		
Frequency Resolution				
	1 Hz			
CW Accuracy				
	+/-1 ppm			
Frequency Stability				
				+/-0.05 ppm. -10° to 70° C, typical; +/-0.1 ppm/yr maximum, typical
Power Level Accuracy[a]				
	Standard	**Opt 014**	**Opt 014 & UNL**	

10 MHz to 45 MHz[b]	+/-1.5 dB	+/-1.5 dB	+/-1.5 dB	
45 MHz to 10 GHz	+/-1.5 dB	+/-1.5 dB	+/-1.5 dB	Variation from nominal power in range 0 (step attenuator at 0 dB)
10 GHz to 20 GHz	+/-1.5 dB	+/-1.5 dB	+/-2.0 dB	
20 GHz to 30 GHz	+/-2.0 dB	+/-2.0 dB	+/-2.5 dB	
30 GHz to 40 GHz	+/-3.0 dB	+/-3.0 dB	+/-3 dB	
40 GHz to 45 GHz	+/-3.0 dB	+/-3.0 dB	+/-3 dB	
45 GHz to 50 GHz	+/-3.5 dB	+/-3.5 dB	+/-3.5 dB	
50 GHz to 60 GHz	+/-4.0 dB	+/-4.0 dB	+/-4.0 dB	
60 GHz to 67 GHz	+/-4.0 dB	+/-4.0 dB	+/-4.5 dB	
67 GHz to 70 GHz[b]	+/-4.0 dB	+/-4.0 dB	+/-4.5 dB	
Power Level Linearity[d]				
				Any Option
Test reference is at the nominal power level (step attenuator at 0 dB)				
10 MHz to 45 MHz[b]	+/-1.0 dB for power<=-5 dBm[g]			
45 MHz to 67 GHz	+/-1.0 dB for power<=-5 dBm[g]			
67 GHz to 70 GHz[b]	+/-1.0 dB for power<=-5 dBm[g]			
Power Range[a,e,f]				
Note: If power is set above the maximum specified leveled power, the test port output signal may show non-linear effects that are dependent on the DUT.				
	Standard	**Opt 014**	**Opt 014 & UNL**	
10 MHz to 45 MHz[b]	-25 to -9 dBm	-25 to -9 dBm	-75 to -9 dBm	

45 MHz to 500 MHz	-25 to -3 dBm	-25 to -3 dBm	-75 to -3 dBm	
500 MHz to 750 MHz	-25 to 0 dBm	-25 to 0 dBm	-75 to 0 dBm	
750 MHz to 10 GHz	-27 to -1 dBm	-27 to -1 dBm	-77 to -1 dBm	
10 GHz to 30 GHz	-27 to -2 dBm	-27 to -3 dBm	-77 to -3 dBm	
30 GHz to 40 GHz	-27 to -1 dBm	-27 to -2 dBm	-77 to -5 dBm	
40 GHz to 45 GHz	-27 to -7 dBm	-27 to -8 dBm	-77 to -10 dBm	
45 GHz to 50 GHz	-27 to -1 dBm	-27 to -2 dBm	-77 to -6 dBm	
50 GHz to 60 GHz	-27 to -3 dBm	-27 to -4 dBm	-77 to -8 dBm	
60 GHz to 67 GHz	-27 to -5 dBm	-27 to -7 dBm	-77 to -13 dBm	
67 GHz to 70 GHz[b]	-27 to -5 dBm	-27 to -7 dBm	-77 to -13 dBm	
Power Sweep Range (ALC)				
	Standard	**Opt 014**	**Opt 014 & UNL**	ALC range starts at maximum leveled output power and decreases by power level indicated in the table.
10 MHz to 45 MHz[b]	16 dB	16 dB	16 dB	
45 MHz to 500 MHz	22 dB	22 dB	22 dB	
500 MHz to 750 MHz	25 dB	25 dB	25 dB	
750 GHz to 10 GHz	26 dB	26 dB	26 dB	
10 GHz to 30 GHz	25 dB	24 dB	24 dB	

30 GHz to 40 GHz	26 dB	25 dB	22 dB	
40 GHz to 45 GHz	20 dB	19 dB	17 dB	
45 GHz to 50 GHz	26 dB	25 dB	21 dB	
50 GHz to 60 GHz	24 dB	23 dB	19 dB	
60 GHz to 67 GHz	22 dB	20 dB	14 dB	
67 GHz to 70 GHz[b]	22 dB	20 dB	14 dB	
Power Resolution				
	0.01 dB			
Phase Noise				
				Any Option
10 kHz offset from center frequency, nominal power at test port				
				typical:
10 MHz to 45 MHz				-80 dBc
45 MHz to 10 GHz				-70 dBc
10 GHz to 24 GHz				-60 dBc
24 GHz to 70 GHz				-55 dBc
10 kHz offset from center frequency, nominal power at test port - Option 080 enabled				
				typical:
10 MHz to 45 MHz				-80 dBc
45 MHz to 10 GHz				-70 dBc
10 GHz to 24 GHz				-60 dBc

24 GHz to 70 GHz				-55 dBc
100 kHz offset from center frequency, nominal power at test port				
				typical:
10 MHz to 45 MHz				-90 dBc
45 MHz to 10 GHz				-90 dBc
10 GHz to 24 GHz				-85 dBc
24 GHz to 70 GHz				-75 dBc
100 kHz offset from center frequency, nominal power at test port - Option 080 enabled				
				typical:
10 MHz to 45 MHz				-85 dBc
45 MHz to 10 GHz				-80 dBc
10 GHz to 24 GHz				-70 dBc
24 GHz to 70 GHz				-60 dBc
1 MHz offset from center frequency, nominal power at test port				
				typical:
10 MHz to 45 MHz				-115 dBc
45 MHz to 10 GHz				-110 dBc
10 GHz to 24 GHz				-105 dBc
24 GHz to 70 GHz				-95 dBc
1 MHz offset from center frequency, nominal power at test port - Option 080 enabled				
				typical:

10 MHz to 45 MHz				-110 dBc
45 MHz to 10 GHz				-105 dBc
10 GHz to 24 GHz				-95 dBc
24 GHz to 70 GHz				-85 dBc
Harmonics (2nd or 3rd)				
				Any Option
				typical:
10 MHz to 500 MHz				-10 dBc
500 MHz to 10 GHz				-15 dBc
10 GHz to 24 GHz				-23 dBc
24 GHz to 50 GHz				-16 dBc
50 GHz to 60 GHz				-13 dBc
60 GHz to 70 GHz				-19 dBc
Non-Harmonic Spurious (at Nominal Output Power)				
10 MHz to 20 GHz				-50 dBc typical, for offset frequency > 1 kHz
20 GHz to 70 GHz				-30 dBc typical, for offset frequency > 1 kHz

a Test port output is a specification when the source is set to Port 1, and a characteristic when the source is set to Port 2.

b Typical performance.

c Preset power.

d Power Level Linearity is a specification when the source is set to Port 1, and a typical when the source is set to Port 2.

e Test port power is specified into nominal 50 ohms.

f Power to which the source can be set and phase lock is assured.

g +/-1.6 dB for power>-5 dBm.

Table 12: Test Port Input

Description	Specification		Supplemental
	Standard or UNL	Opt 014 or 014 & UNL	
Test Port Noise Floor[1]			
10 Hz IF Bandwidth			
10 MHz to 45 MHz[2]	<-70 dBm	<-70 dBm	
45 MHz to 500 MHz[3, 4]	<-90 dBm	<-90 dBm	
500 MHz to 2 GHz	<-112 dBm	<-112 dBm	
2 GHz to 10 GHz	<-112 dBm	<-112 dBm	
10 GHz to 24 GHz	<-116 dBm	<-115 dBm	
24 GHz to 30 GHz	<-105 dBm	<-104 dBm	Option 016 degrades performance by 2 dB.
30 GHz to 40 GHz	<-105 dBm	<-104 dBm	
40 GHz to 45 GHz	<-103 dBm	<-102 dBm	
45 GHz to 50 GHz	<-101 dBm	<-100 dBm	
50 GHz to 60 GHz	<-100 dBm	<-99 dBm	
60 GHz to 67 GHz	<-99 dBm	<-97 dBm	Option 016 degrades performance by 3 dB.
67 GHz to 70 GHz[2]	<-99 dBm	<-97 dBm	
1 KHz IF Bandwidth			
10 MHz to 45 MHz[2]	<-50 dBm	<-50 dBm	
45 MHz to 500 MHz[3, 4]	<-70 dBm	<-70 dBm	
500 MHz to 2 GHz	<-92 dBm	<-92 dBm	
2 GHz to 10 GHz	<-92 dBm	<-92 dBm	
10 GHz to 24 GHz	<-96 dBm	<-95 dBm	

24 GHz to 30 GHz	<-85 dBm	<-84 dBm	Option 016 degrades performance by 2 dB.
30 GHz to 40 GHz	<-85 dBm	<-84 dBm	
40 GHz to 45 GHz	<-83 dBm	<-82 dBm	
45 GHz to 50 GHz	<-81 dBm	<-80 dBm	
50 GHz to 60 GHz	<-80 dBm	<-79 dBm	
60 GHz to 67 GHz	<-79 dBm	<-77 dBm	Option 016 degrades performance by 3 dB.
67 GHz to 70 GHz[2]	<-79 dBm	<-77 dBm	
Test Port Noise Floor[1] Option 080 enabled[5]			
			Option 014 or 014 & UNL (typical)
10 Hz IF Bandwidth			
10 MHz to 45 MHz			<-70 dBm
45 MHz to 500 MHz[3, 4]			<-90 dBm
500 MHz to 2 GHz			<-112 dBm
2 GHz to 10 GHz			<-112 dBm
10 GHz to 24 GHz			<-115 dBm
24 GHz to 30 GHz		Option 016 degrades performance by 2 dB.	<-104 dBm
30 GHz to 40 GHz			<-104 dBm
40 GHz to 45 GHz			<-102 dBm
45 GHz to 50 GHz			<-100 dBm
50 GHz to 60 GHz			<-99 dBm
60 GHz to 67 GHz		Option 016 degrades performance by 3 dB.	<-97 dBm
67 GHz to 70 GHz			<-97 dBm
1 KHz IF Bandwidth			
			typical:
10 MHz to 45 MHz			<-50 dBm
45 MHz to 500 MHz[3, 4]			<-70 dBm

500 MHz to 2 GHz			<-92 dBm
2 GHz to 10 GHz			<-92 dBm
10 GHz to 24 GHz			<-95 dBm
24 GHz to 30 GHz		Option 016 degrades performance by 2 dB.	<-84 dBm
30 GHz to 40 GHz			<-84 dBm
40 GHz to 45 GHz			<-82 dBm
45 GHz to 50 GHz			<-80 dBm
50 GHz to 60 GHz			<-79 dBm
60 GHz to 67 GHz		Option 016 degrades performance by 3 dB.	<-77 dBm
67 GHz to 70 GHz			<-77 dBm
Direct Receiver Access Input Noise Floor[1]			
	Standard		**Option 014 or 014 & UNL (typical)**
10 Hz IF Bandwidth			
10 MHz to 45 MHz[2]			<-106 dBm
45 MHz to 500 MHz[3, 4]			<-105 dBm
500 MHz to 2 GHz			<-125.5 dBm
2 GHz to 10 GHz			<-125 dBm
10 GHz to 24 GHz			<-128 dBm
24 GHz to 30 GHz		Option 016 degrades performance by 2 dB.	<-117.5 dBm
30 GHz to 40 GHz			<-117 dBm
40 GHz to 45 GHz			<-115 dBm
45 GHz to 50 GHz			<-112.5 dBm
50 GHz to 60 GHz			<-111 dBm
60 GHz to 67 GHz		Option 016 degrades performance by 3 dB	. <-108 dBm
67 GHz to 70 GHz[2]			<-107 dBm
1 KHz IF Bandwidth			

10 MHz to 45 MHz[2]			<-86 dBm
45 MHz to 500 MHz[3, 4]			<-85 dBm
500 MHz to 2 GHz			<-105.5 dBm
2 GHz to 10 GHz			<-105 dBm
10 GHz to 24 GHz			<-108 dBm
24 GHz to 30 GHz		Option 016 degrades performance by 2 dB.	<-97.5 dBm
30 GHz to 40 GHz			<-97 dBm
40 GHz to 45 GHz			<-95 dBm
45 GHz to 50 GHz			<-92.5 dBm
50 GHz to 60 GHz			<-91 dBm
60 GHz to 67 GHz		Option 016 degrades performance by 3 dB.	<-88 dBm
67 GHz to 70 GHz[2]			<-87 dBm
Direct Receiver Access Input Noise Floor[1] - Option 080 enabled[5]			
	Standard		**Option 014**
10 Hz IF Bandwidth			
			typical:
10 MHz to 45 MHz[2]			<-106 dBm
45 MHz to 500 MHz[3, 4]			<-105 dBm
500 MHz to 2 GHz			<-125.5 dBm
2 GHz to 10 GHz			<-125 dBm
10 GHz to 24 GHz			<-128 dBm
24 GHz to 30 GHz		Option 016 degrades performance by 2 dB.	<-117.5 dBm
30 GHz to 40 GHz			<-117 dBm
40 GHz to 45 GHz			<-115 dBm
45 GHz to 50 GHz			<-112.5 dBm
50 GHz to 60 GHz			<-111 dBm

60 GHz to 67 GHz		Option 016 degrades performance by 3 dB.	<-108 dBm
67 GHz to 70 GHz[2]			<-107 dBm
1 KHz IF Bandwidth			
			typical:
10 MHz to 45 MHz[2]			<-86 dBm
45 MHz to 500 MHz[3, 4]			<-85 dBm
500 MHz to 2 GHz			<-105.5 dBm
2 GHz to 10 GHz			<-105 dBm
10 GHz to 24 GHz			<-108 dBm
24 GHz to 30 GHz		Option 016 degrades performance by 2 dB.	<-97.5 dBm
30 GHz to 40 GHz			<-97 dBm
40 GHz to 45 GHz			<-95 dBm
45 GHz to 50 GHz			<-92.5 dBm
50 GHz to 60 GHz			<-91 dBm
60 GHz to 67 GHz		Option 016 degrades performance by 3 dB.	<-88 dBm
67 GHz to 70 GHz[2]			<-87 dBm

Receiver Compression Level (measured at test ports)				
	Specifications			**Supplemental Information**
	Standard	**Option 014**	**014 & UNL**	**Typical**
10 MHz to 45 MHz[2, 6]	negligible			
45 MHz to 500 MHz[6, 7]	<0.1 dB at -9.5 dBm[8] and <0.25 dB at -3 dBm	<0.1 dB at -4.0 dBm[8] and <0.25 dB at +3 dBm		<0.1 dB at +0.5 dBm[8] and <0.25 dB at +8 dBm
500 MHz to 5 GHz	<0.1 dB at -8 dBm[8] and <0.25 dB at -1 dBm	<0.1 dB at -8 dBm[8] and <0.25 dB at -1 dBm	<0.1 dB at -7 dBm[8] and <0.25 dB at 0 dBm	<0.1 dB at -4.0 dBm[8] and <0.25 dB at +3 dBm

5 GHz to 30 GHz	<0.1 dB at -8.5 dBm[8] and <0.25 dB at -2 dBm	<0.1 dB at -9 dBm[8] and <0.25 dB at -2 dBm	<0.1 dB at -6 dBm[8] and <0.25 dB at +1 dBm	<0.1 dB at -1.0 dBm[8] and <0.25 dB at +6 dBm
30 GHz to 67 GHz	<0.1 dB at -10.5 dBm[8] and <0.15 dB at -7 dBm	<0.1 dB at -6.0 dBm[8] and <0.15 dB at -4 dBm	<0.1 dB at -9.5 dBm[8] and <0.15 dB at -6 dBm	<0.1 dB at -2.0 dBm[8, 9] and <0.15 dB at +2 dBm[9]
67 GHz to 70 GHz[2]	NA			

System Compression Level			
	Standard	**Option 014 or 014 & UNL**	
	maximum leveled output power		See dynamic accuracy table
Third Order Intercept[10] - Tone spacing from 100 kHz - 5 MHz			
			Any Option
			typical
10 MHz to 500 MHz			+30 dBm
500 MHz to 24 GHz			+24 dBm
24 GHz to 40 GHz			+23 dBm
40 to 50 GHz			+24 dBm
50 to 67 GHz			+26 dBm
Third Order Intercept[10] - Tone spacing from 5 MHz - 20 MHz			
			Any Option
			typical
10 MHz to 500 MHz			NA
500 MHz to 24 GHz			+20 dBm
24 GHz to 40 GHz			+20 dBm
40 to 50 GHz			+22 dBm
50 to 67 GHz			+24 dBm

Third Order Intercept[10] - Tone spacing from 20 MHz - 50 MHz			
			Any Option
			typical
10 MHz to 500 MHz			NA
500 MHz to 24 GHz			+26 dBm
24 to 40 GHz			+24 dBm
40 to 50 GHz			+25 dBm
50 to 67 GHz			+27 dBm
Trace Noise Magnitude 1 kHz IF bandwidth. Ratio measurement, nominal power at test port.			
	Standard or 014	**014 & UNL**	
10 MHz to 45 MHz[2]	<0.150 dB rms	<0.150 dB rms	
45 MHz to 500 MHz[11, 4]	<0.010 dB rms	<0.010 dB rms	
500 MHz to 24 GHz	<0.006 dB rms	<0.006 dB rms	
24 GHz to 67 GHz	<0.006 dB rms	<0.009 dB rms	
67 GHz to 70 GHz[2]	<0.006 dB rms	<0.009 dB rms	
Trace Noise Magnitude - Option 080 enabled[2,5] 1 kHz IF bandwidth. Ratio measurement, nominal power at test port.			
	Standard or 014	**014 & UNL**	
10 MHz to 45 MHz	<0.150 dB rms	<0.150 dB rms	
45 MHz to 500 MHz[11, 4]	<0.010 dB rms	<0.010 dB rms	

500 MHz to 24 GHz	<0.006 dB rms	<0.006 dB rms	
24 GHz to 67 GHz	<0.009 dB rms	<0.012 dB rms	
67 GHz to 70 GHz	<0.009 dB rms	<0.012 dB rms	
Trace Noise Phase 1 kHz IF bandwidth. Ratio measurement, nominal power at test port.			
	Any Option		
10 MHz to 45 MHz[2]	<0.800° rms		
45 MHz to 500 MHz[4]	<0.100° rms		
500 MHz to 24 GHz	<0.060° rms		
24 GHz to 67 GHz	<0.100° rms		
67 GHz to 70 GHz[2]	<0.100° rms		
Trace Noise Phase - Option 080 enabled[2, 5] 1 kHz IF bandwidth. Ratio measurement, nominal power at test port.			
			typical:
10 MHz to 45 MHz			<0.800° rms
45 MHz to 500 MHz[4]			<0.100° rms
500 MHz to 24 GHz			<0.060° rms
24 GHz to 67 GHz			<0.100° rms
67 GHz to 70 GHz			<0.100° rms

Reference Level Magnitude			
Range	+/-500 dB		
Resolution	0.001 dB		
Reference Level Phase			
Range	+/-500°		
Resolution	0.01°		
Stability Magnitude[12]			
			Any Option
Typical ratio measurement, made at the test port.			
10 MHz to 45 MHz			+/-0.05 dB/°C
45 MHz to 50 GHz			+/-0.02 dB/°C
50 GHz to 70 GHz			+/-0.04 dB/°C
Stability Phase[12]			
			Any Option
Typical ratio measurement, measured at the test port.			
10 MHz to 45 MHz			+/-0.5°/°C
45 MHz to 20 GHz			+/-0.2°/°C
20 GHz to 40 GHz			+/-0.5°/°C
40 GHz to 70 GHz			+/-0.8°/°C
Damage Input Level			
			typical:

Test Port 1 and 2			+27 dBm or +/-40 VDC
R1, R2 in			+15 dBm or +/-15 VDC
A, B in			+15 dBm or +/-7 VDC
Coupler Thru (Option 014 or 014 & UNL)			+27 dBm or +/-40 VDC
Coupler Arm (Option 014 or 014 & UNL)			+30 dBm or +/-7 VDC

1Total average (rms) noise power calculated as the mean value of a linear magnitude trace expressed in dBm.

2Typical performance.

3Noise floor may be degraded by 10 dB at particular frequencies (multiples of 5 MHz) due to spurious receiver residuals.

4 Specified value is for worst-case noise floor at 45 MHz

50 Hz offset

6 Coupler roll-off will reduce compression below 500 MHz. Ultimately, at 45 MHz, compression is negligible.

7 Specified value is for worst-case compression at 500 MHz.

8 This compression level comes from the dynamic accuracy curve with -30 dBm reference test port power.

9 Option 016 degrades performance by 3 dB.

10 TOI is a typical specification that applies while the network analyzer receiver is in its linear range.

11Trace noise magnitude may be degraded to 20 mdB rms at harmonic frequencies of the first IF (8.33 MHz) below 80 MHz.

12Stability is defined as a ratio measurement made at the test port.

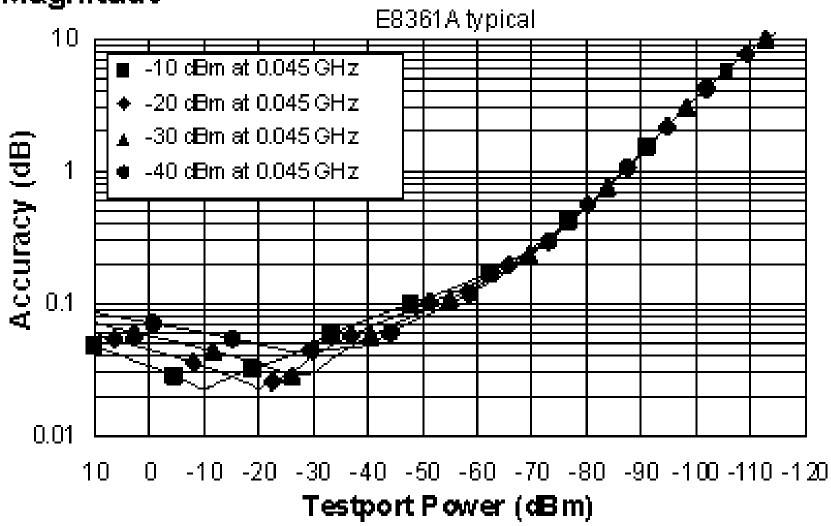
Table 13. Dynamic Accuracy (Specification[a])

Accuracy of the test port input power reading relative to the reference input power level.

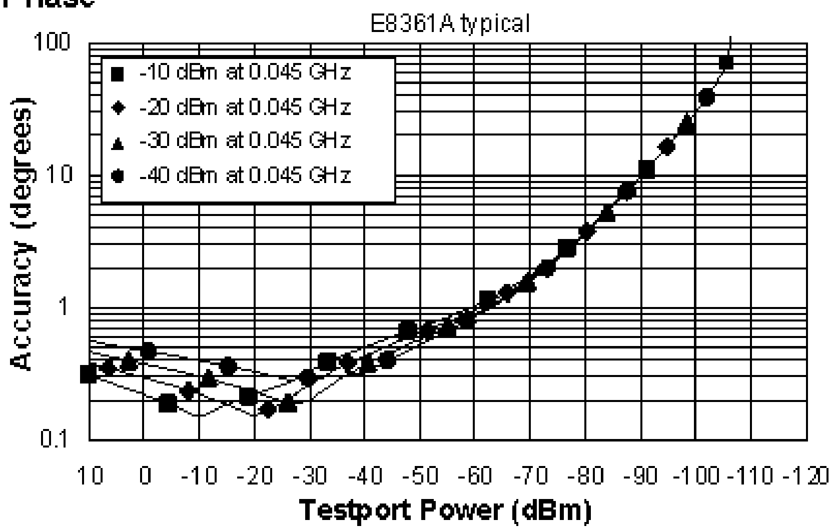
Note: If power is set above the maximum specified leveled power, the test port output signal may show non-linear effects that are dependent on the DUT

Dynamic Accuracy, 0.045 GHz

Magnitude

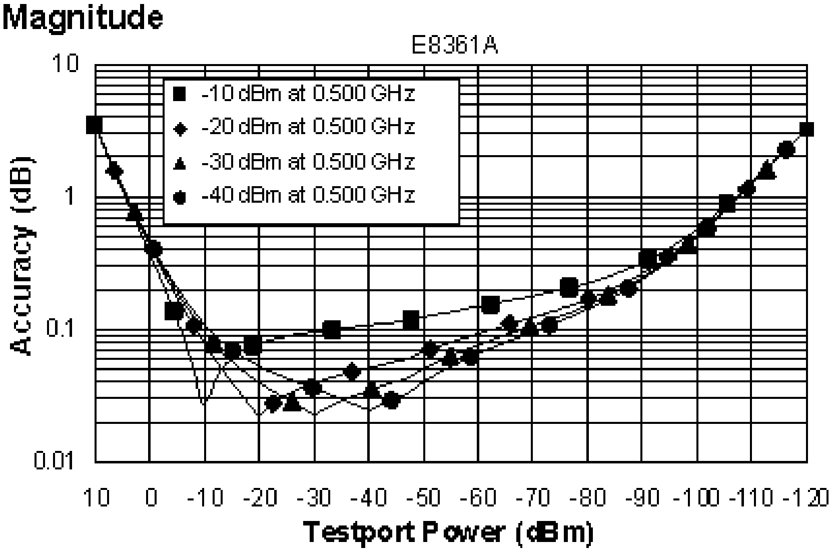


Phase

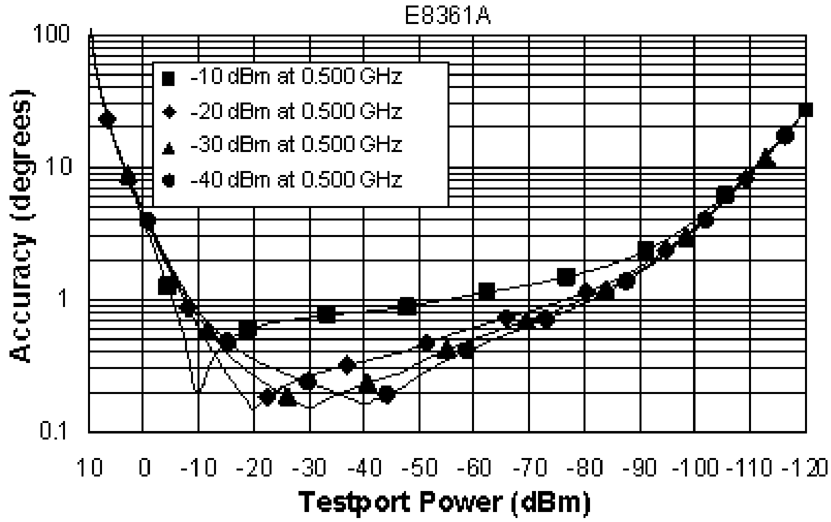


Dynamic Accuracy, 0.500 GHz

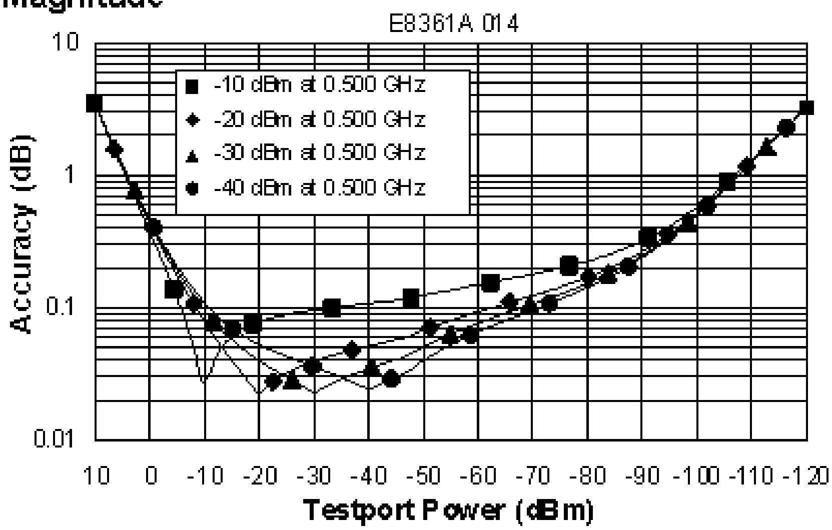
Magnitude



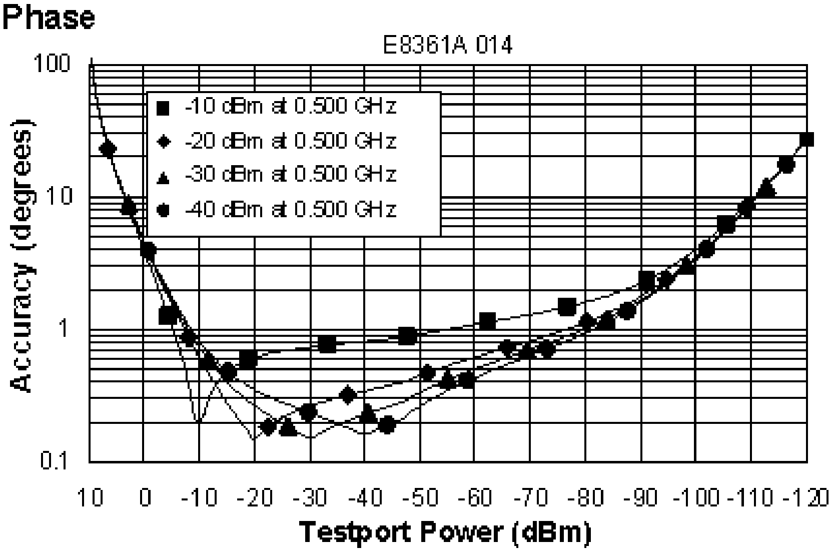
Phase



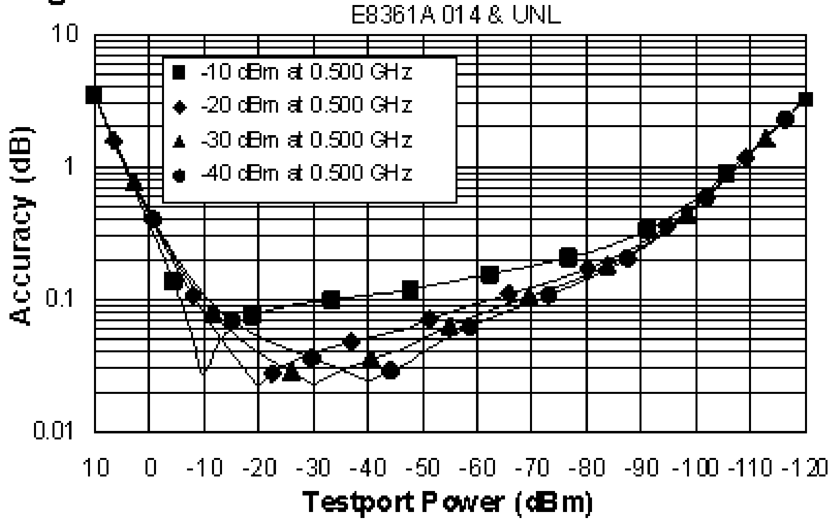
Magnitude



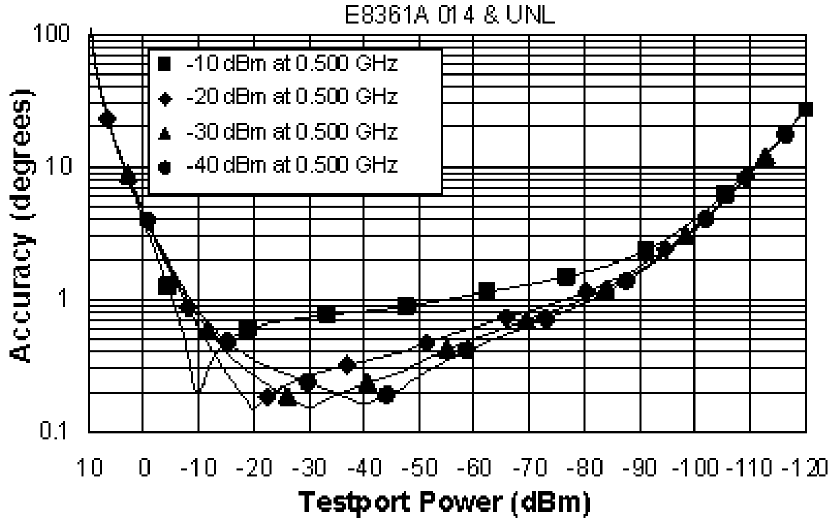
Phase



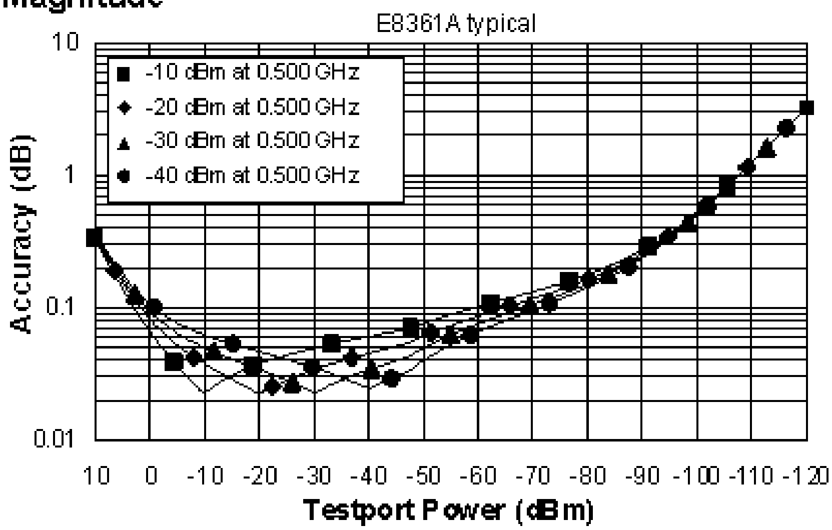
Magnitude



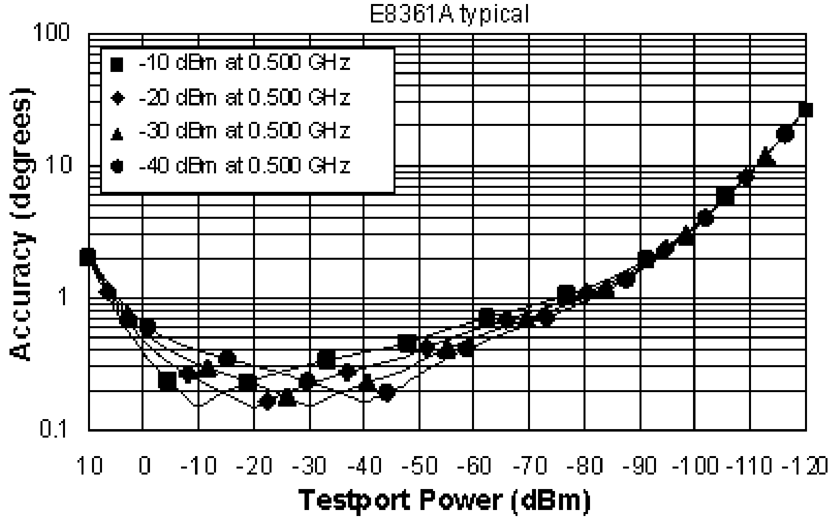
Phase



Magnitude

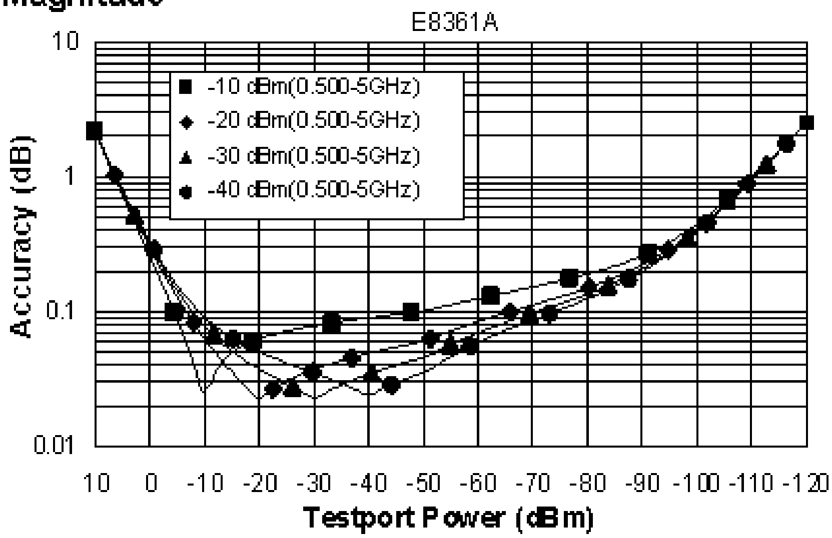


Phase

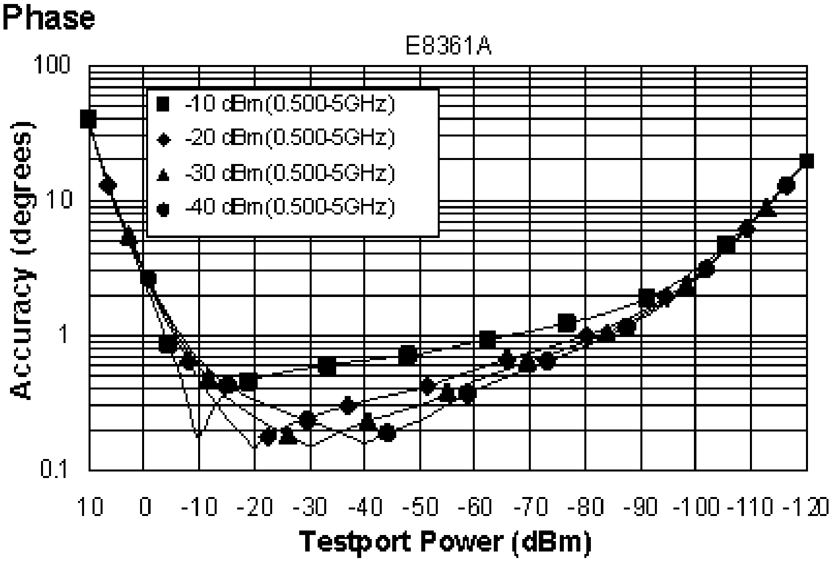


Dynamic Accuracy, 0.500 - 5 GHz

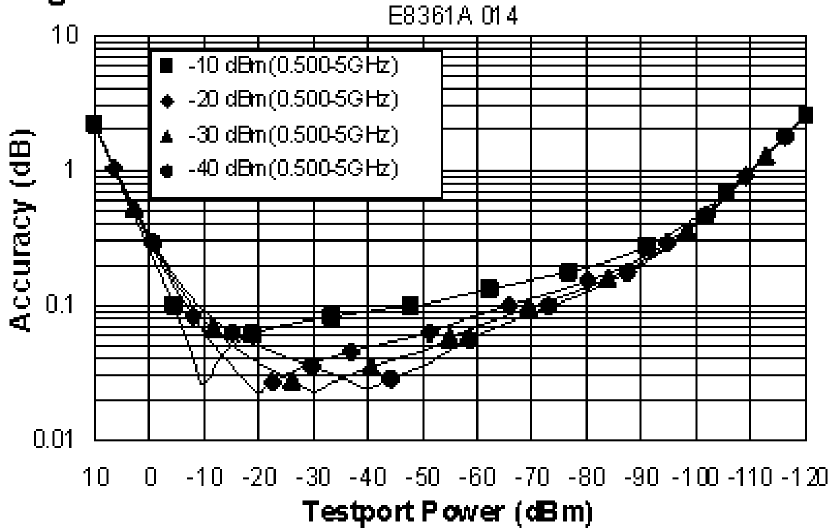
Magnitude



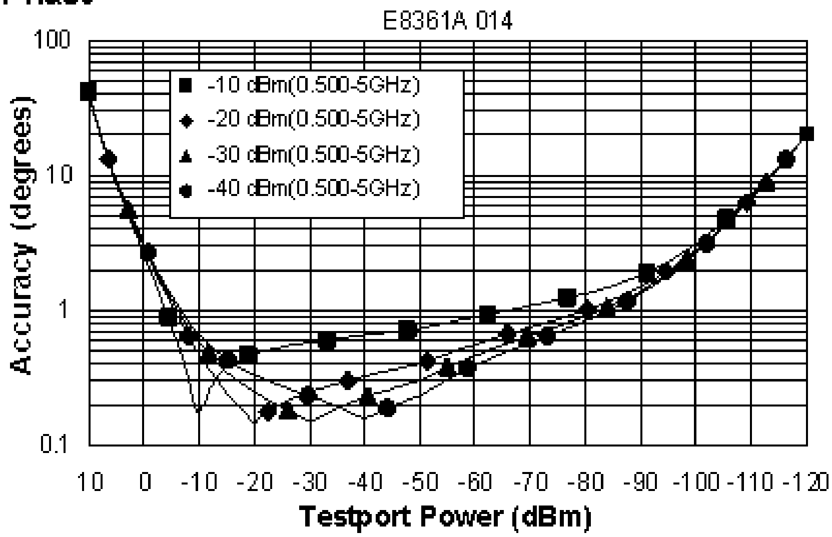
Phase



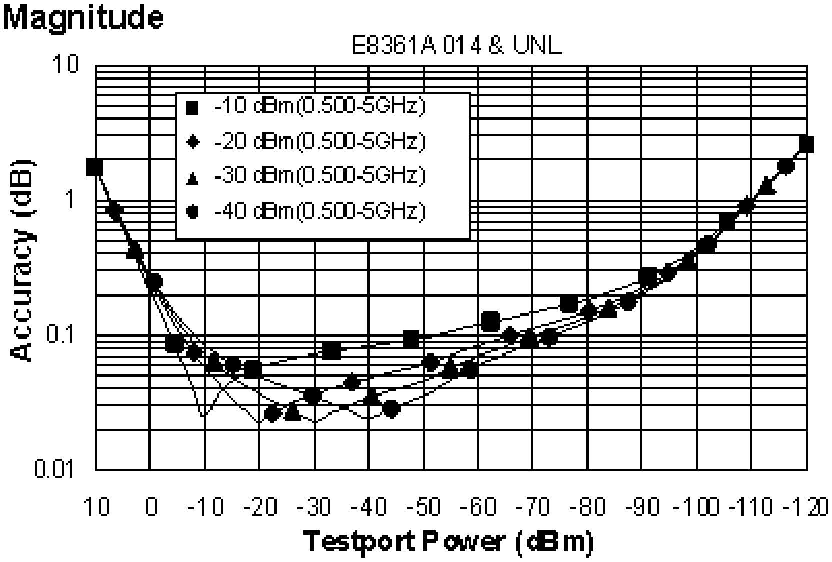
Magnitude



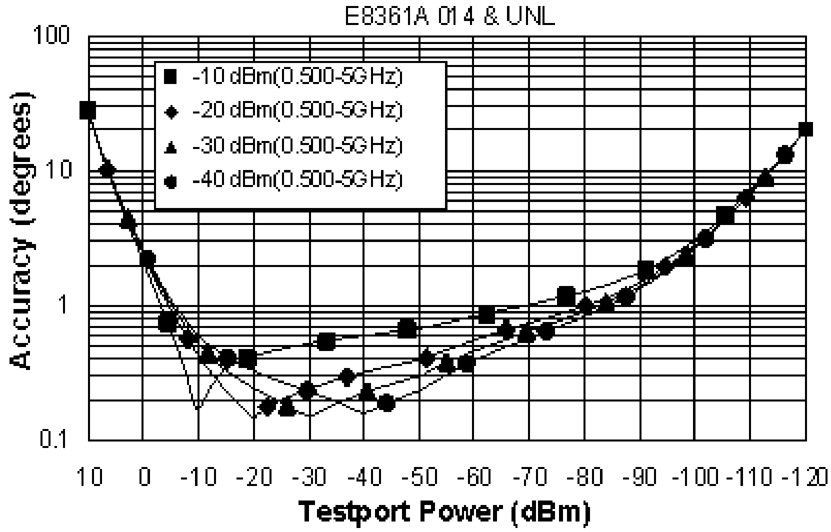
Phase



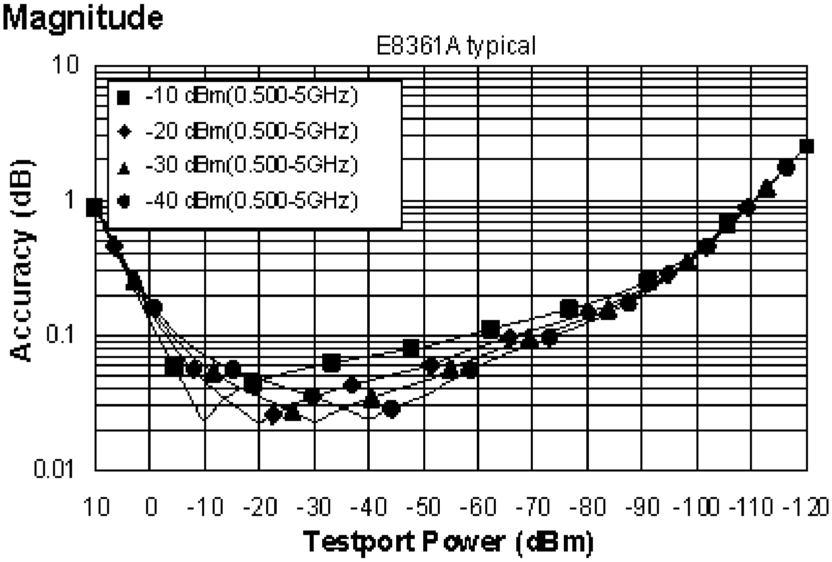
Magnitude



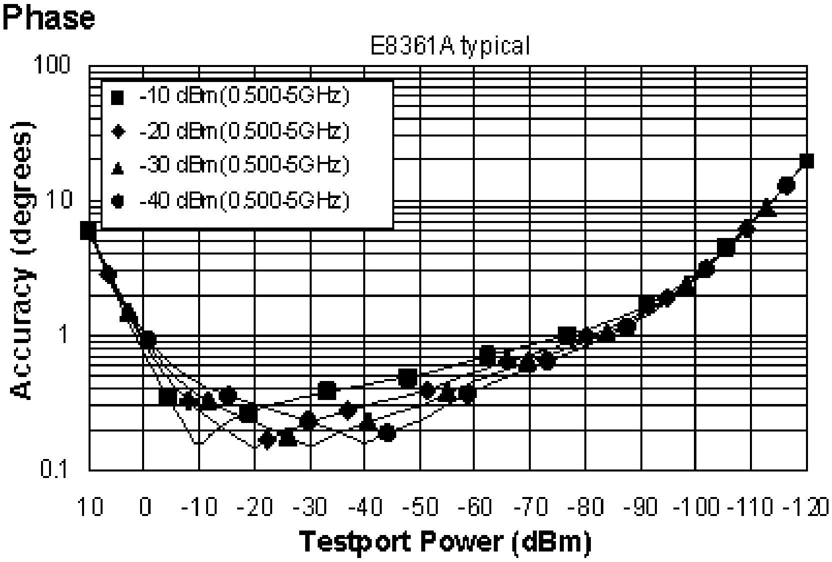
Phase



Magnitude

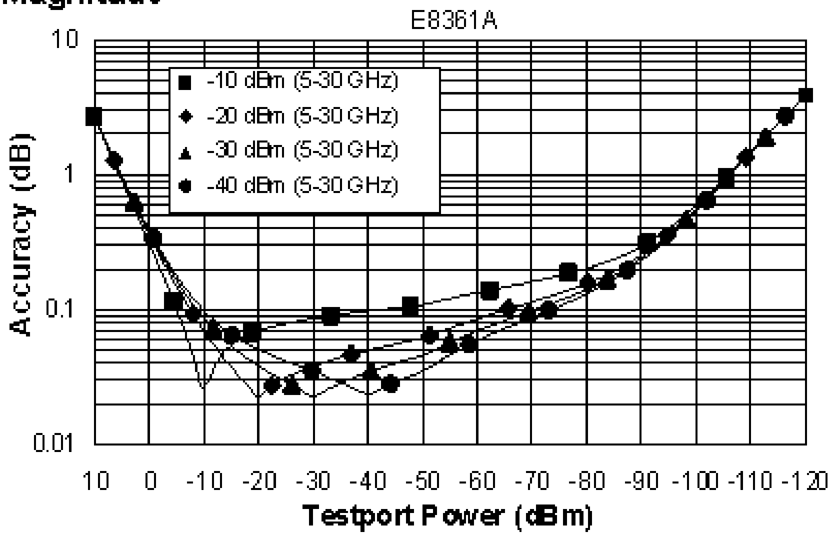


Phase

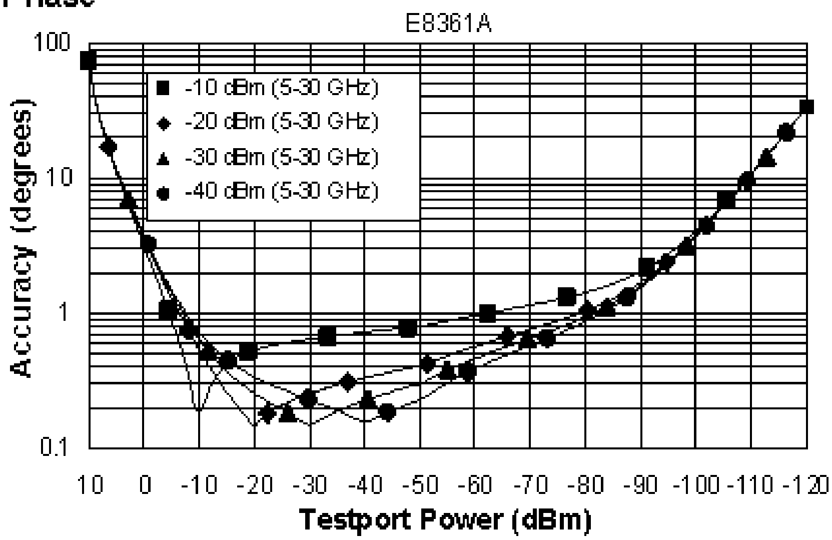


Dynamic Accuracy, 5 - 30 GHz

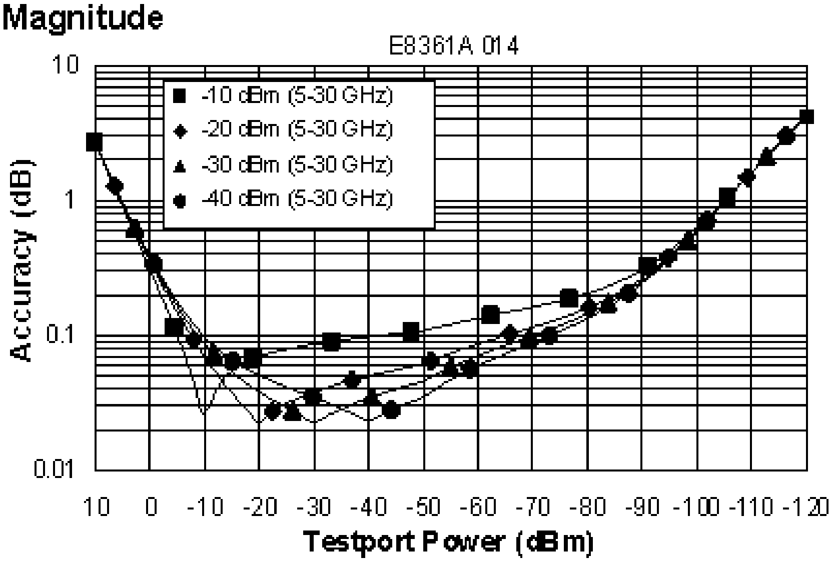
Magnitude



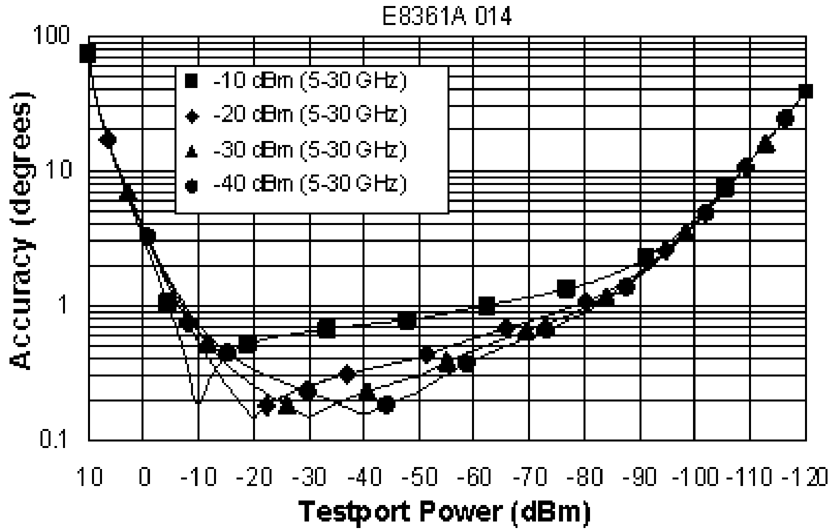
Phase



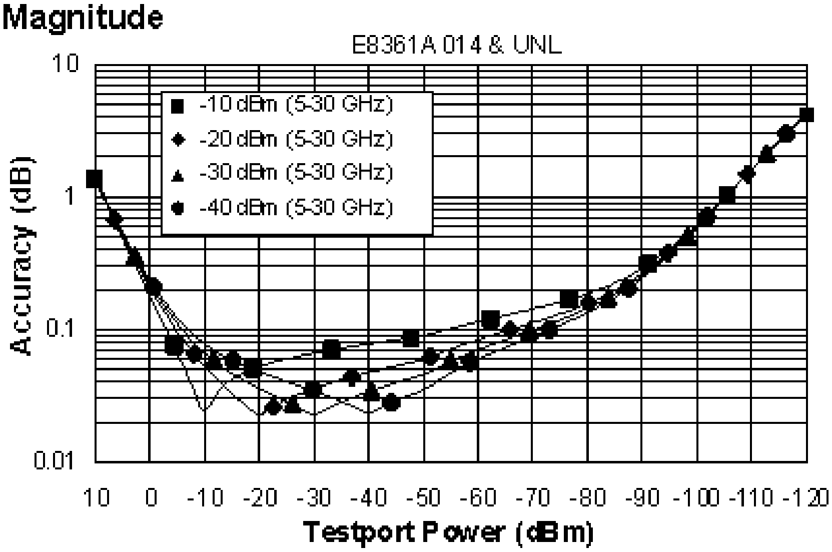
Magnitude



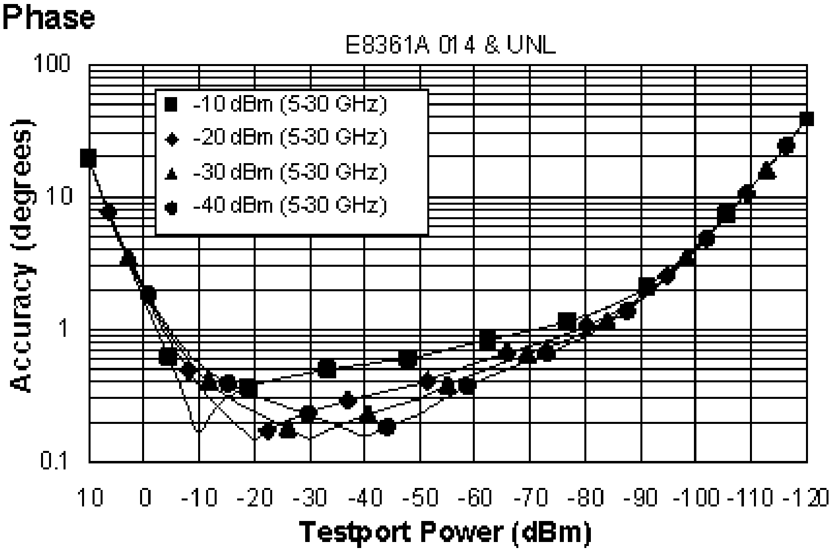
Phase



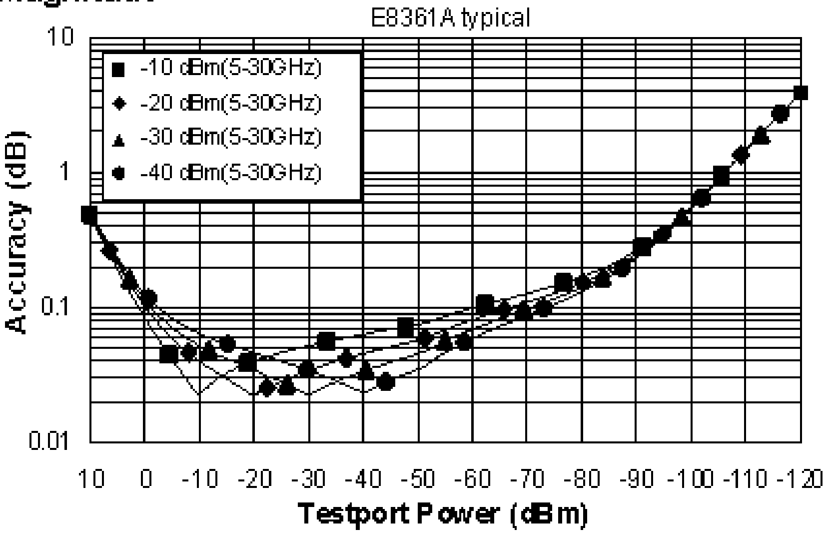
Magnitude



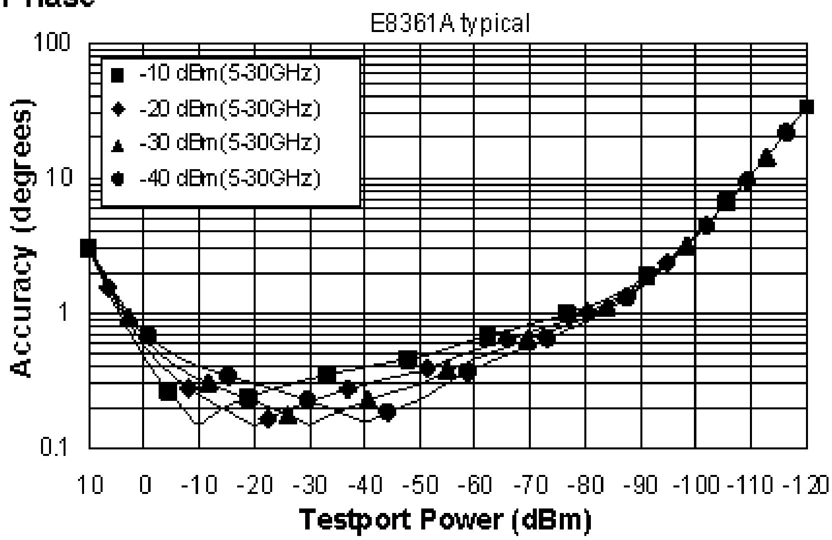
Phase



Magnitude

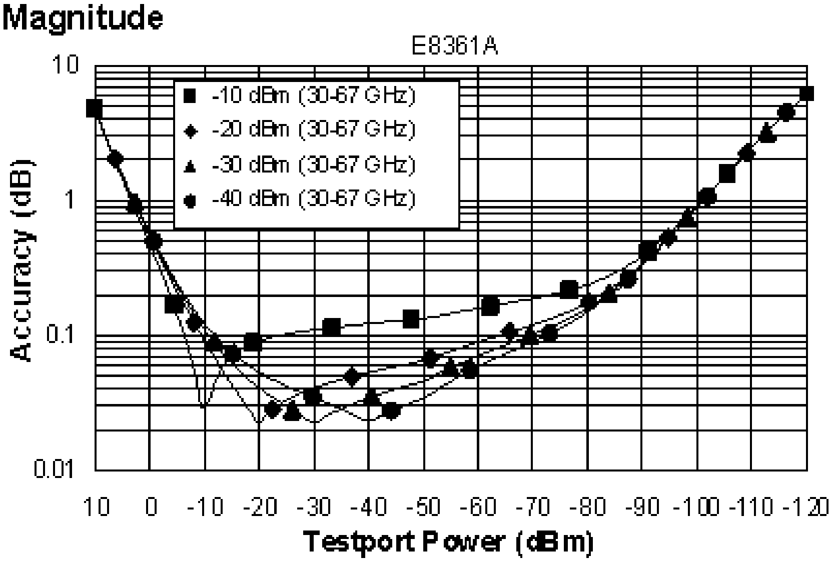


Phase

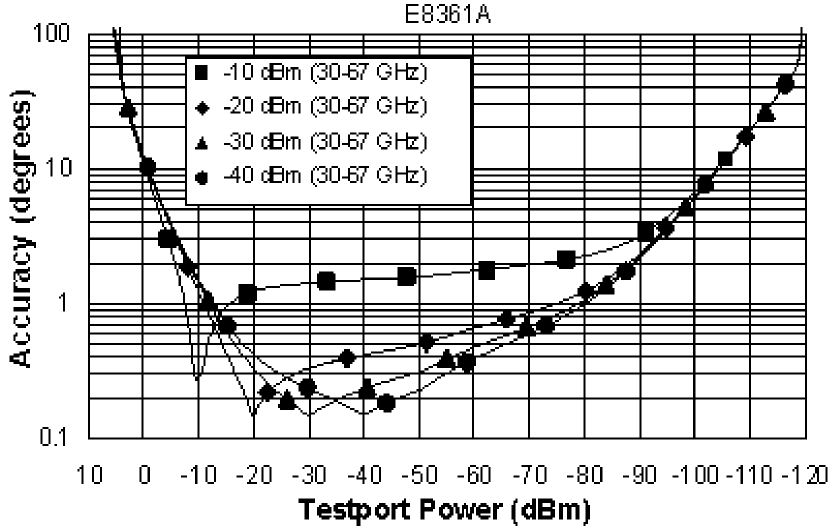


Dynamic Accuracy, 30 - 67 GHz

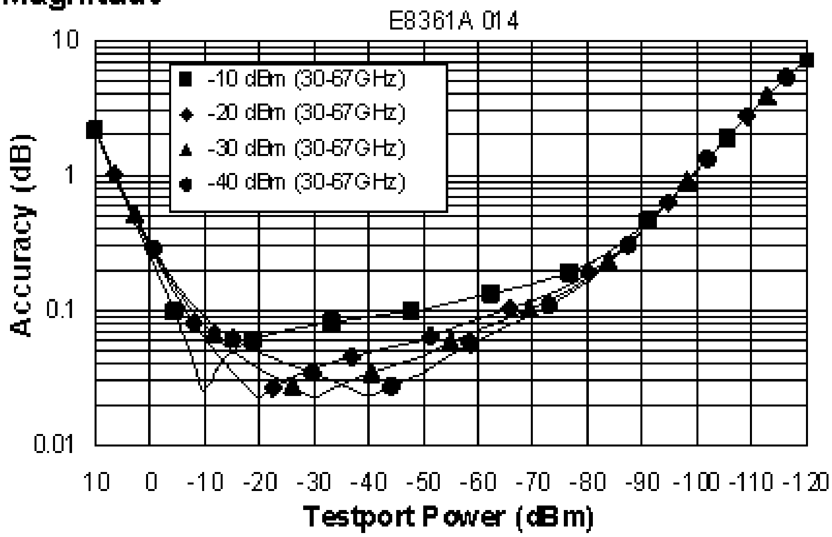
Magnitude



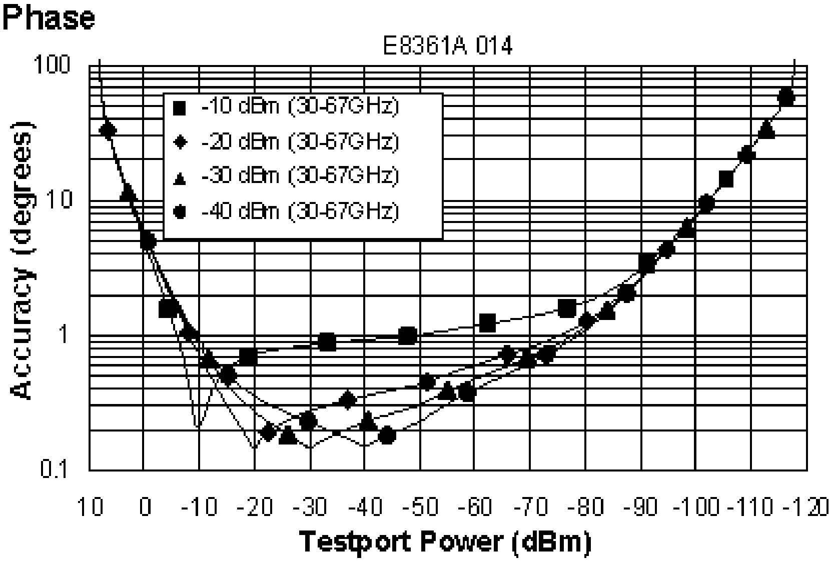
Phase



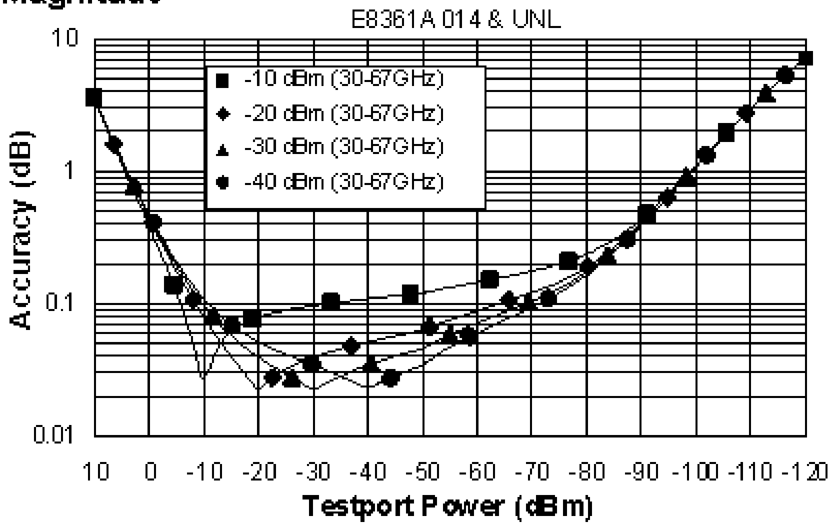
Magnitude



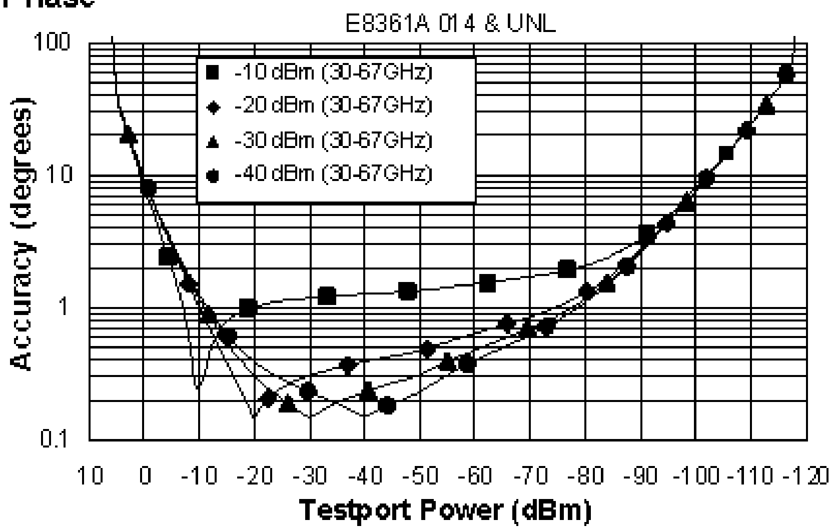
Phase



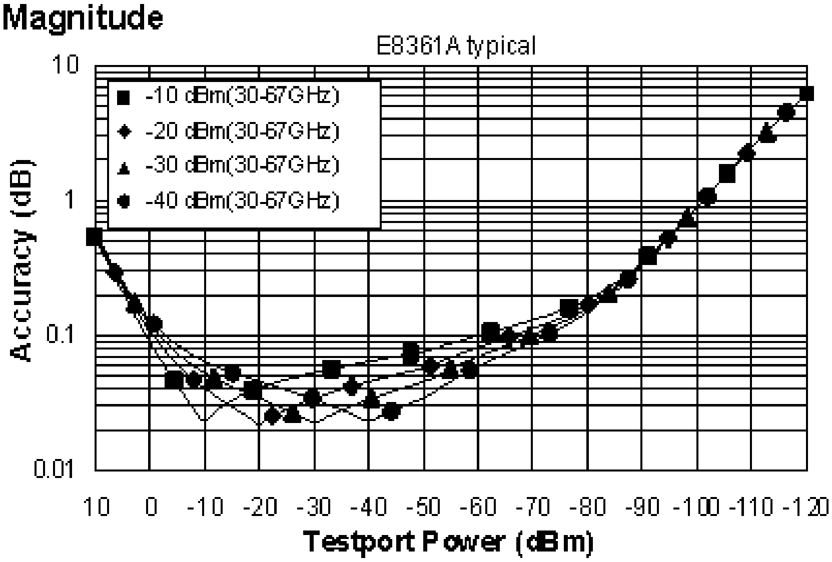
Magnitude



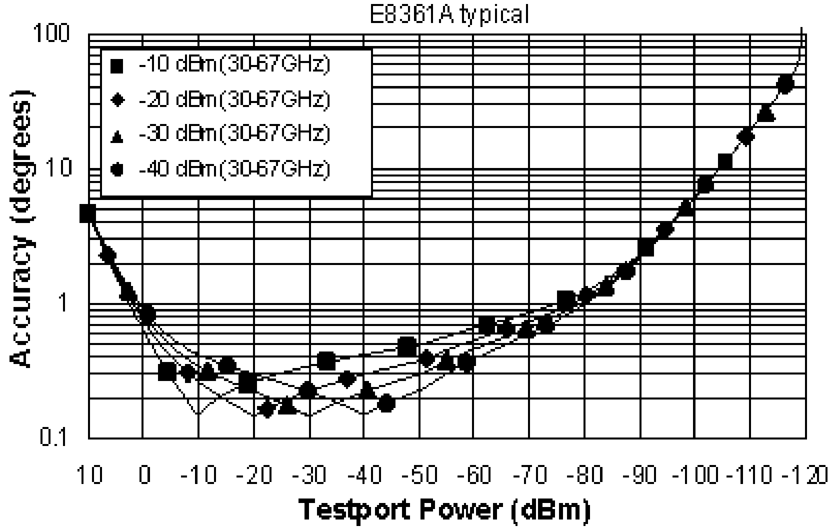
Phase



Magnitude

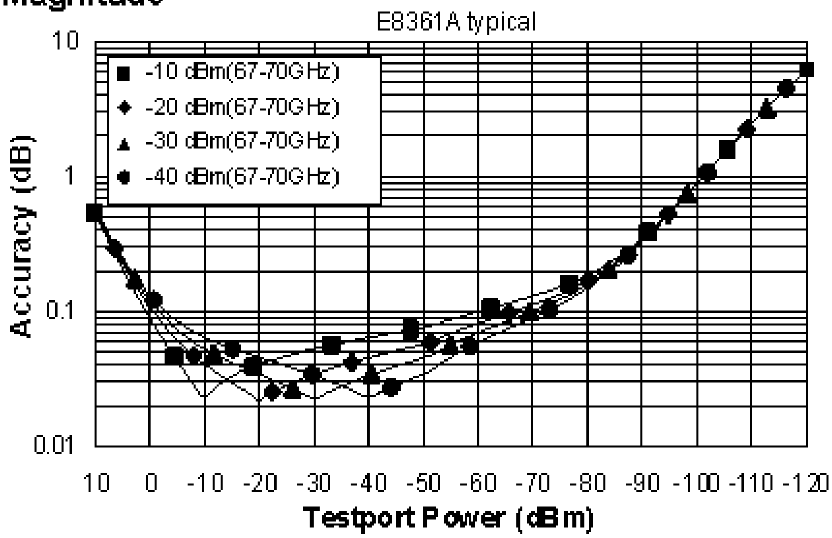


Phase

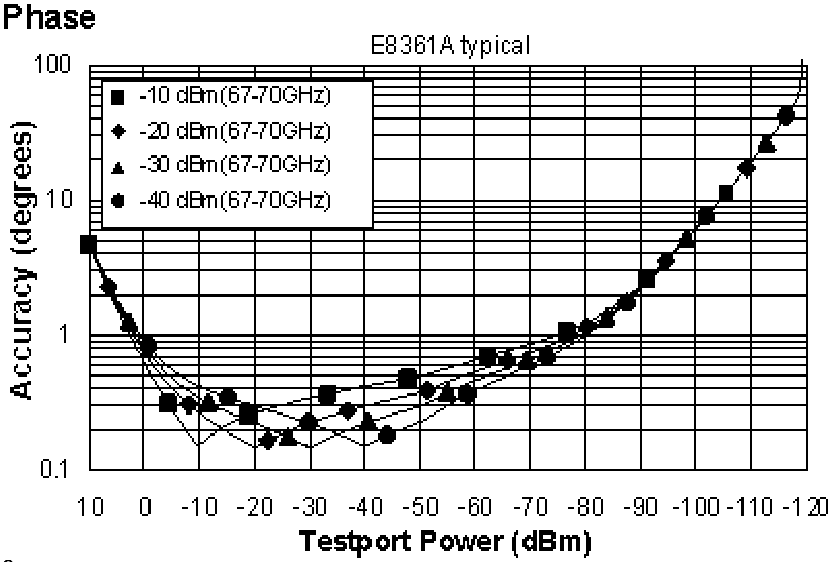


Dynamic Accuracy, 67 - 70 GHz

Magnitude



Phase



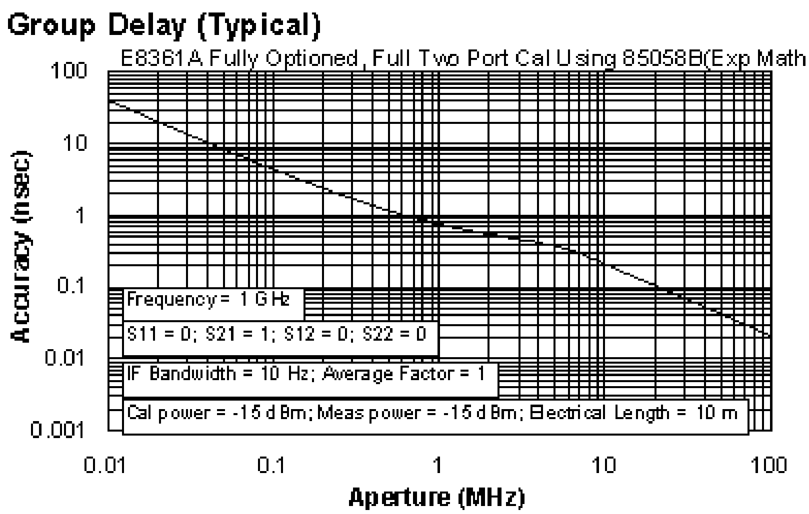
[a] Dynamic accuracy is verified with the following measurements:

- compression over frequency
- IF linearity at a single frequency of 1.195 GHz and a reference level of -20 dBm for an input power range of 0 to -120 dBm.

Table 14. Test Port Input (Group Delay)[a]

Description	Specification	Supplemental Information (typ.)
Aperture (selectable)		(frequency span)/(number of points -1)
Maximum Aperture		20% of frequency span
Range		0.5 x (1/minimum aperture)
Maximum Delay		Limited to measuring no more than 180° of phase change within the minimum aperture.)
Accuracy		See graph below. Char.

The following graph shows characteristic group delay accuracy with full 2-port calibration and a 10 Hz IF bandwidth. Insertion loss is assumed to be < 2 dB and electrical length to be ten meters.



In general, the following formula can be used to determine the accuracy, in seconds, of specific group delay measurement:

±Phase Accuracy (deg)/[360 × Aperture (Hz)]

Depending on the aperture and device length, the phase accuracy used is either incremental phase accuracy or worst case phase accuracy.

a Group delay is computed by measuring the phase change within a specified frequency step (determined by the frequency span and the number of points per sweep).

General Information

Miscellaneous Information

Front Panel

Rear Panel

Environment and Dimensions

Table 15. Miscellaneous Information

Description	Specification	Supplemental Information
System IF Bandwidth Range		1 Hz to 40 kHz, nominal
CPU		Intel® 500 MHz Pentium® III

Table 16. Front Panel Information

Description	Supplemental Information
RF Connectors	
E8361A	
Type	1.85 mm (male), 50 ohm, (nominal)
Center Pin Recession	0.002 in. (characteristic)
Display	
Size	21.3 cm (8.4 in) diagonal color active matrix LCD; 640 (horizontal) X 480 (vertical) resolution
Refresh Rate	Vertical 59.83 Hz; Horizontal 31.41 kHz
Display Range	
Magnitude	±200 dB (at 20 dB/div), max
Phase	±500°, max
Polar	10 pUnits, min 1000 Units, max
Display Resolution	
Magnitude	0.001 dB/div, min
Phase	0.01°/div, min
Marker Resolution	
Magnitude	0.001 dB, min
Phase	0.01°, min
Polar	0.01 mUnit, min; 0.01°,min

Table 17. Rear Panel Information

Description	Supplemental Information
10 MHz Reference In	
Connector	BNC, female
Input Frequency	10 MHz ± 10 ppm, typical
Input Level	-15 dBm to +20 dBm, typical

Input Impedance	200 Ω, nom.
10 MHz Reference Out	
Connector	BNC, female
Output Frequency	10 MHz ± 1 ppm, typical
Signal Type	Sine Wave, typical
Output Level	+10 dBm ± 4 dB into 50 Ω, typical
Output Impedance	50 Ω, nominal
Harmonics	<-40 dBc, typical
Option H08 & H11 Rear Panel Connectors (Typical)	
IF Connectors	A, R1, R2, B (BNC Connectors)
IF Connector Input Frequency	8 1/3 MHz
Nominal Input Impedance at IF Inputs	50 Ω
RF Damage Level to IF Connector Inputs	-20.0 dBm
DC Damage Level to IF Connector Inputs	25 volts
0.1 dB Compression Point at IF Inputs	-27.0 dBm
Pulse Input Connectors[1]	A, R1, R2, B (BNC Connectors)
Nominal Input Impedance at Pulse Inputs	1 Kohm
Minimum IF Gate Width	20 ns for less than 1 dB deviation from theoretical performance[2].
DC Damage Level to Pulse Connector Inputs	5.5 volts
Drive Voltage	TTL (0, +5.0) Volts
Rear Panel LO Power - Test Port Frequency	
1.7 GHz to 20 GHz	-7 to –16 dBm
Rear Panel RF Power - Test Port Frequencies	
1.7 GHz to 20 GHz	-2 to –12 dBm (at –5 dBm test port power[3])

10 GHz to 16 GHz	0 to –8 dBm (at –5 dBm test port power[3])
16 GHz to 20 GHz	+5 to –1 dBm (at –5 dBm test port power[3])
VGA Video Output	
Connector	15-pin mini D-Sub; Drives VGA compatible monitors
Devices Supported:	
	Resolutions:
Flat Panel (TFT)	1024 X 768, 800 X 600, 640 X 480
Flat Panel (DSTN)	800 X 600, 640 X 480
CRT Monitor	1280 X 1024, 1024 X 768, 800 X 600, 640 X 480
	Simultaneous operation of the internal and external displays is allowed, but with 640 X 480 resolution only. If you change resolution, you can only view the external display (internal display will "white out").
Bias Input Connectors (Option UNL)	
Bias current	500 mA, maximum
Bias voltage	40 Volts, maximum
Test Set IO	
	25-pin D-Sub connector, available for external test set control
Aux IO	
	25-pin D-Sub connector, male, analog and digital IO
Handler IO	
	36-pin parallel I/O port; all input/output signals are default set to negative logic; can be reset to positive logic via GPIB command
GPIB	
	24-pin D-sub (Type D-24), female; compatible with IEEE-488.
Parallel Port (LPT1)	
	25-pin D-Sub miniature connector, female; provides connection to printers or any other parallel port peripherals
Serial Port (COM 1)	
	9-pin D-Sub, male; compatible with RS-232
USB Port	

	One port on front panel and five ports on rear panel. Universal Serial Bus jack, Type A configuration (4 contacts inline, contact 1 on left); female
Contact 1	Vcc: 4.75 to 5.25 VDC, 500 mA, maximum
Contact 2	-Data
Contact 3	+Data
Contact 4	Ground
LAN	
	10/100BaseT Ethernet, 8-pin configuration; auto selects between the two data rates
Line Power (A third-wire ground is required.)	
Frequency	50/60/400 Hz
Voltage	120/240 VAC (Power supply is auto switching.)
Max	500 Watts

1 Pulse input connectors are operational only with Option H08 (Pulse Measurement Capability) enabled.

2 Based on deviation from signal reduction equation: Signal Reduction (dB) = $20\log_{10}$(Duty_cycle) = $20\log_{10}$(pulse_width/period). Measured at Pulse Repetition Frequency (PFR) of 1 MHz.

3 Test port power has to be at a high enough level such that the “Drop Cal” does not occur. If Drop Cal occurs then the power out of the rear panel RF connector will drop by about 15 dB.

Table 18. Analyzer Environment and Dimensions

Description	**Supplemental Information**
General Environmental	
RFI/EMI Susceptibility	Defined by CISPR Pub. 11, Group 1, Class A, and IEC 50082-1
ESD	Minimize using static-safe work procedures and an antistatic bench mat
Dust	Minimize for optimum reliability
Operating Environment	
Temperature	0 °C to +40 °C Instrument powers up, phase locks, and displays no error messages within this temperature range (except for "source unleveled" error message that may occur at temperatures outside the specified performance temperature range of 25 +/-5°C).
Error-Corrected Temperature Range	23°C ± 3°C with less than 1°C deviation from calibration temp.
Humidity	5% to 95% at +40 °C
Altitude	0 to 4500 m (14,760 ft.)
Non-Operating Storage Environment	

Temperature	-40 °C to +70 °C
Humidity	0% to 90% at +65 °C (non-condensing)
Altitude	0 to 15,240 m (50,000 ft.)

Cabinet Dimensions			
	Height	**Width**	**Depth**
Excluding front and rear panel hardware and feet	267 mm 10.5 in	426 mm 16.75 in	427 mm 16.8 in
As shipped - includes front panel connectors, rear panel bumpers, and feet.	280 mm 11.0 in	435 mm 17.1 in	470 mm 18.5 in
As shipped plus handles	280 mm 11.0 in	458 mm 18 in	501 mm 19.7 in
As shipped plus rack-mount flanges	280 mm 11.0 in	483 mm 19 in	470 mm 18.5 in
As shipped plus handles and rack-mount flanges	280 mm 11.0 in	483 mm 19 in	501 mm 19.70 in

Weight	
Net	
E8361A	29 kg (64 lb), nominal
Shipping	
E8361A	36.3 kg (80 lb), nominal

Measurement Throughput Summary

- **Typical Cycle Time for Measurement Completion**
- **Cycle Time vs IF Bandwidth**
- **Cycle Time vs Number of Points**
- **Data Transfer Time**

Table 19 Typical Cycle Time[a,b] (ms) for Measurement Completion

	Number of Points			
	201	**401**	**1601**	**16,001**
Start 28 GHz, Stop 30 GHz, 35 kHz IF bandwidth				
Uncorrected, 1-port cal	12	19	55	503
2-Port cal	29	44	124	1112
Start 10 MHz, Stop 10 GHz, 35 kHz IF bandwidth				
Uncorrected, 1-port cal	86	93	121	583
2-Port cal	179	199	267	1301
Start 10 MHz, Stop 20 GHz, 35 kHz IF bandwidth				
Uncorrected, 1-port cal	126	130	153	597
2-Port cal	264	275	335	1321
Start 10 MHz, Stop 40 GHz, 35 kHz IF bandwidth				
Uncorrected, 1-port cal	185	190	213	621
2-Port cal	382	401	459	1374
Start 10 MHz, Stop 50 GHz, 35 kHz IF bandwidth				
Uncorrected, 1-port cal	210	216	243	643
2-Port cal	436	450	522	1405
Start 10 MHz, Stop 67 GHz, 35 kHz IF bandwidth				
Uncorrected 1-port cal	244	254	300	645
2-Port cal	502	524	591	1423

a Typical performance.

b Includes sweep time, retrace time and band-crossing time. Analyzer display turned off with DISPLAY:ENABLE OFF. Add 21 ms for display on. Data for one trace (S11) measurement.

Table 20. Cycle Time vs IF Bandwidth[a]

Applies to the Preset condition (201 points, correction off) except for the following changes:

- CF = 28 GHz

- Span = 100 MHz
- Display off (add 21 ms for display on)

IF Bandwidth (Hz)	Cycle Time (ms)[b]	Cycle Time (ms) Option 080 enabled
40,000	11	100
35,000	12	101
30,000	13	102
20,000	16	106
10,000	30	127
7000	38	138
5000	50	152
3000	74	182
1000	274	326
300	694	782
100	1905	2054
30	6091	6355
10	17916	18372

a Typical performance.
b Cycle time includes sweep and retrace time.

Table 21. Cycle Time vs Number of Points[a]

Applies to the Preset condition (35 kHz IF bandwidth, correction off) except for the following changes:

- CF = 28 GHz
- Span = 100 MHz
- Display off (add 21 ms for display on)

Number of Points	Cycle Time (ms)[b]
3	6
11	6
51	7
101	9
201	12
401	18
801	30
1601	55
16,001	497

a Typical performance.

b Cycle time includes sweep and retrace time.

Table 22. Data Transfer Time (ms)[a]

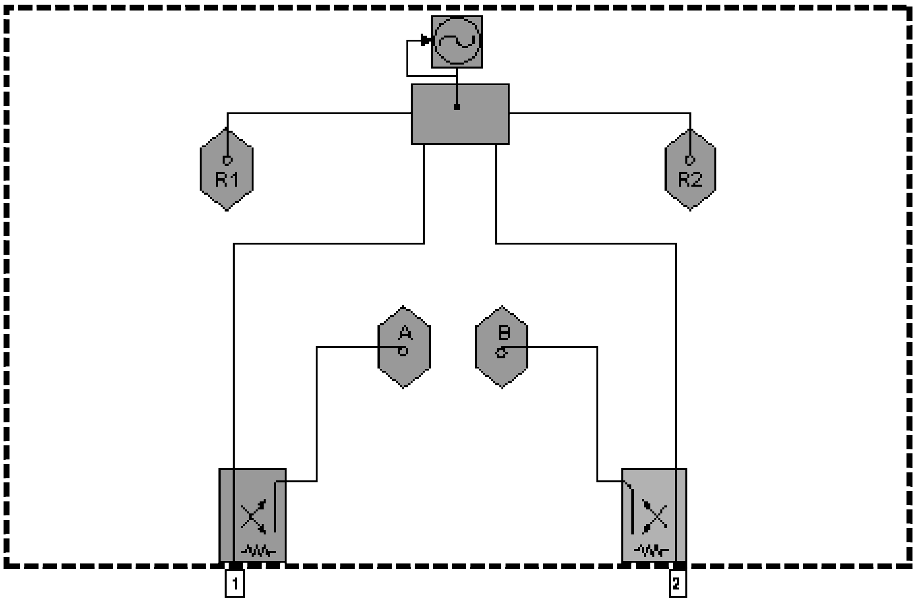
	Number of Points			
	201	401	1601	16,001
SCPI over GPIB (program executed on external PC)				
32-bit floating point	7	12	43	435
64-bit floating point	12	22	84	856
ASCII	64	124	489	5054
SCPI (program executed in the analyzer)				
32-bit floating point	1	2	3	30
64-bit floating point	2	2	4	40
ASCII	29	56	222	2220
COM (program executed in the analyzer)				
32-bit floating point	<0.4	0.4	0.5	1.9
Variant type	0.7	1	3	32
DCOM over LAN (program executed on external PC)				
32-bit floating point	<0.8	1	1.5	7.1
Variant type	1.8	2.7	8.5	80

a Typical performance

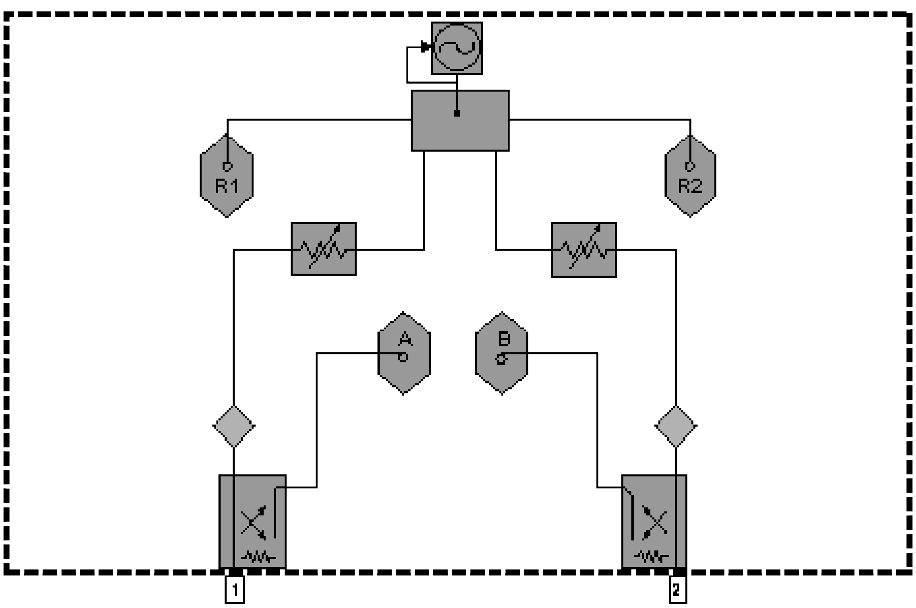
Note: Specifications for Recall & Sweep Speed are not provided for the E8361A analyzers.

Test Set Block Diagrams

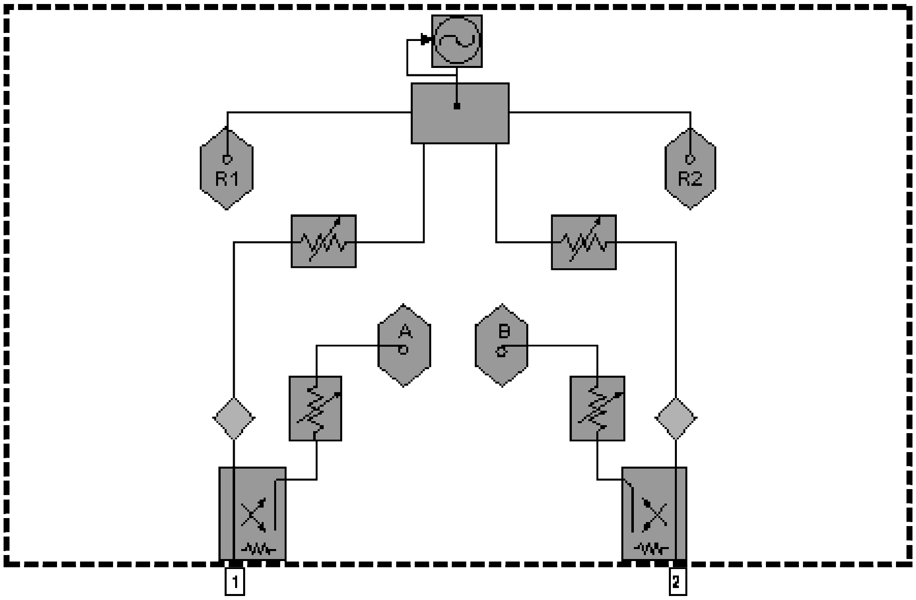
E8361A - Standard Configuration and Standard Power Range



E8361A - Option UNL Standard Configuration with Extended Power Range and Bias - Tees

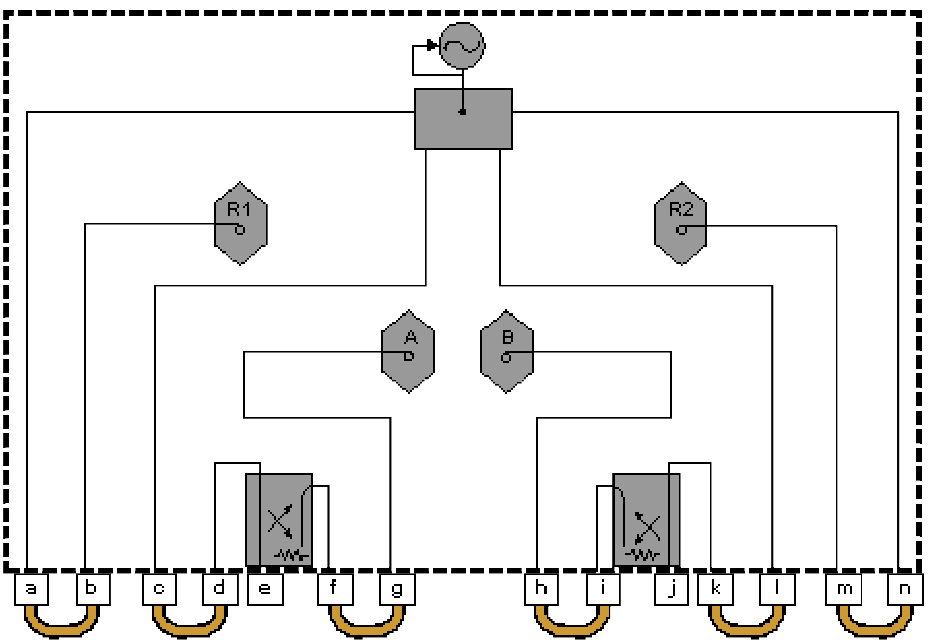


E8361A - Option UNL Standard Configuration with Extended Power Range and Bias - Tees, and Option 016, Receiver Attenuators



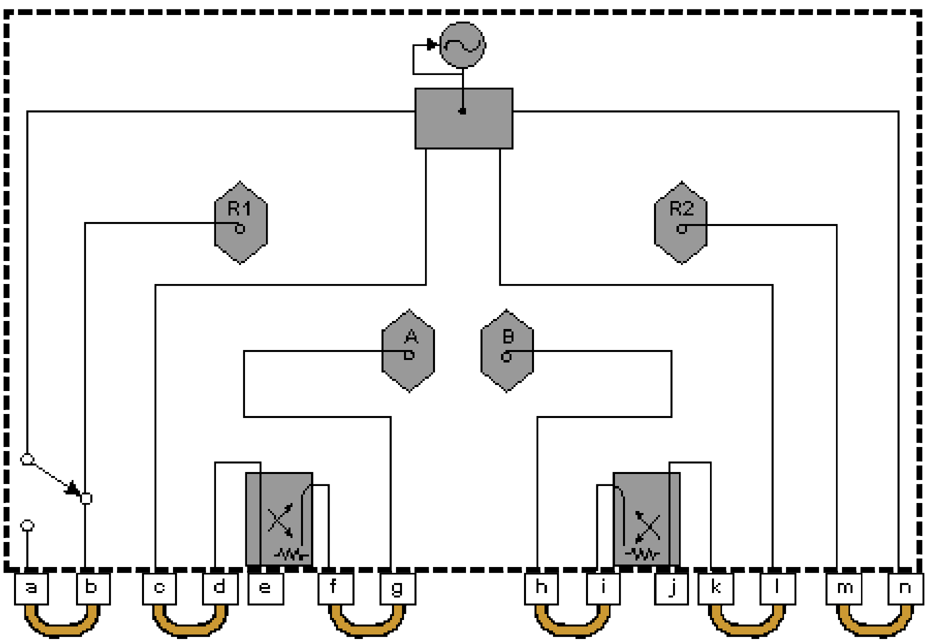
Test Set with Option 014 Block Diagrams

E8361A - Option 014 – Configurable Test Set and Standard Power Range



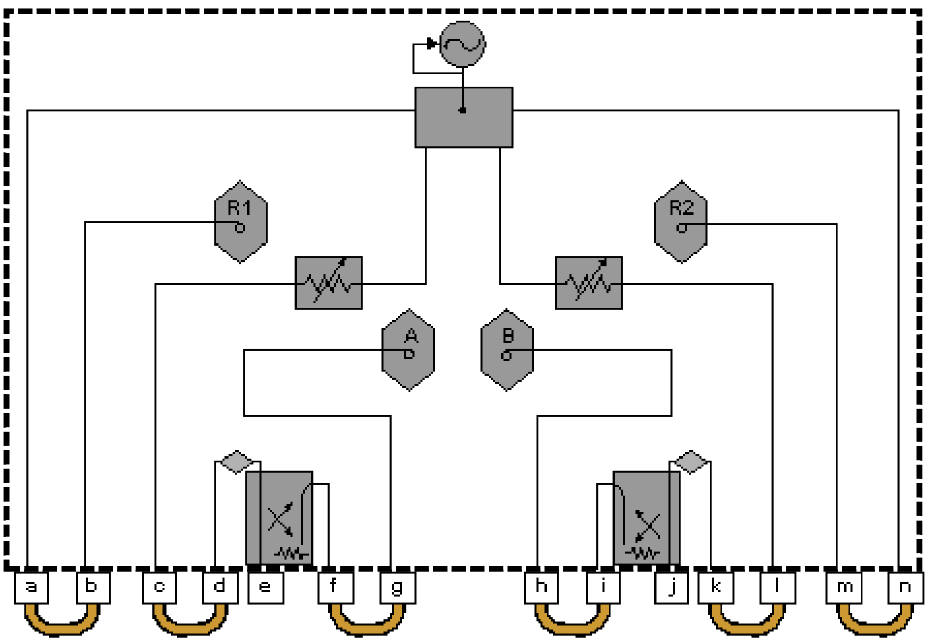
Item	Description	Item	Description
a	SOURCE OUT	h	RCVR B IN
b	RCVR R1 IN	i	CPLR ARM
c	SOURCE OUT	j	PORT 2
d	CPLR THRU	k	CPLR THRU
e	PORT 1	l	SOURCE OUT
f	CPLR ARM	m	RCVR R2 IN
g	RCVR A IN	n	SOURCE OUT

E8361A - Option 014 – Configurable Test Set and Standard Power Range, and Option 081 Reference Channel Transfer Switch



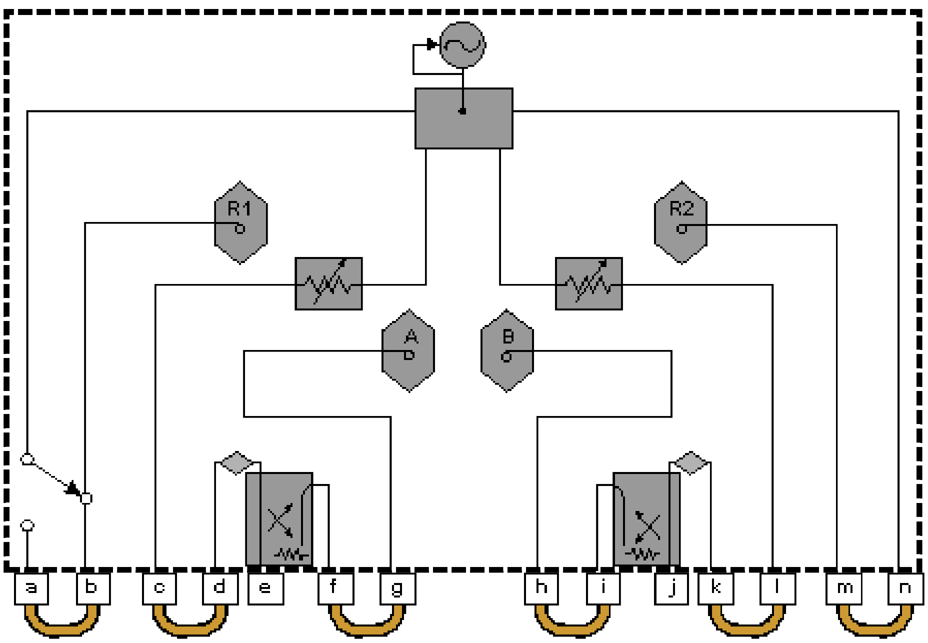
Item	Description	Item	Description
a	SOURCE OUT	h	RCVR B IN
b	RCVR R1 IN	i	CPLR ARM
c	SOURCE OUT	j	PORT 2
d	CPLR THRU	k	CPLR THRU
e	PORT 1	l	SOURCE OUT
f	CPLR ARM	m	RCVR R2 IN
g	RCVR A IN	n	SOURCE OUT

E8361A - Option 014 Configurable Test Set, and Option UNL Extended Power Range and Bias - Tees



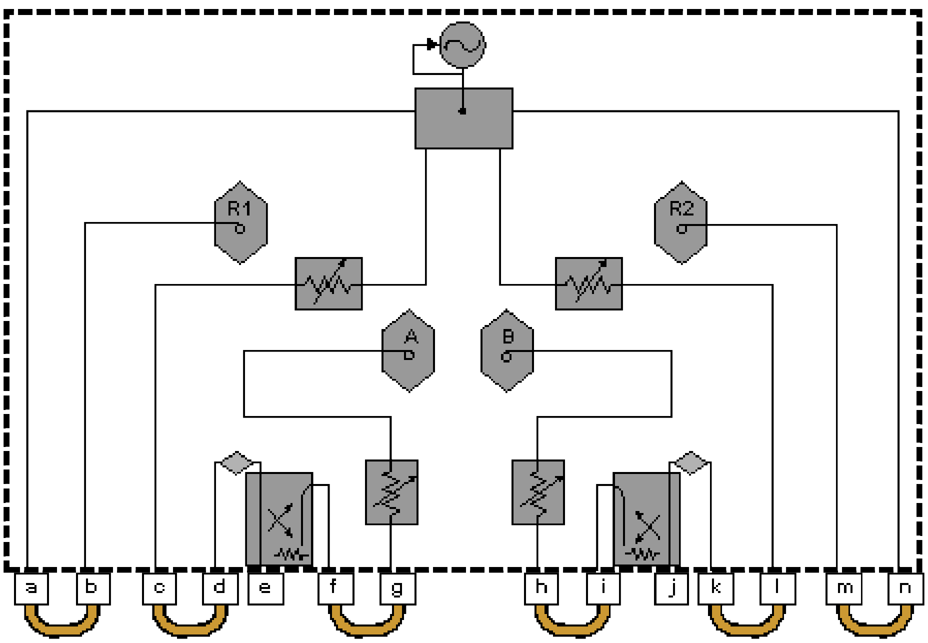
Item	Description	Item	Description
a	SOURCE OUT	h	RCVR B IN
b	RCVR R1 IN	i	CPLR ARM
c	SOURCE OUT	j	PORT 2
d	CPLR THRU	k	CPLR THRU
e	PORT 1	l	SOURCE OUT
f	CPLR ARM	m	RCVR R2 IN
g	RCVR A IN	n	SOURCE OUT

E8361A - Option 014 Configurable Test Set, and Option UNL Extended Power Range and Bias - Tees, and Option 081 Reference Channel Transfer Switch



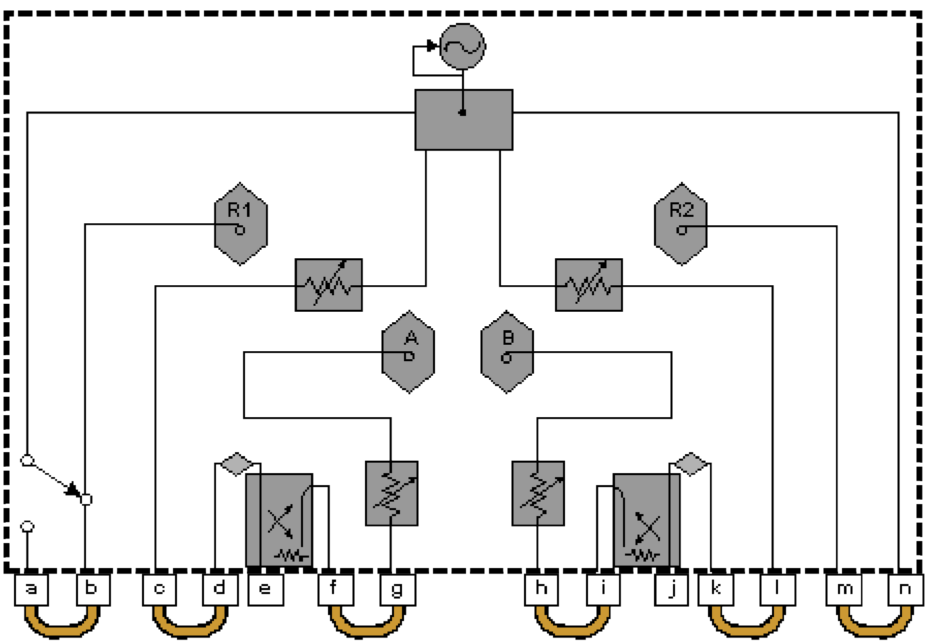
Item	Description	Item	Description
a	SOURCE OUT	h	RCVR B IN
b	RCVR R1 IN	i	CPLR ARM
c	SOURCE OUT	j	PORT 2
d	CPLR THRU	k	CPLR THRU
e	PORT 1	l	SOURCE OUT
f	CPLR ARM	m	RCVR R2 IN
g	RCVR A IN	n	SOURCE OUT

E8361A - Option 014 Configurable Test Set and Option UNL, Extended Power Range and Bias - Tees and Option 016 Receiver Attenuators



Item	Description	Item	Description
a	SOURCE OUT	h	RCVR B IN
b	RCVR R1 IN	i	CPLR ARM
c	SOURCE OUT	j	PORT 2
d	CPLR THRU	k	CPLR THRU
e	PORT 1	l	SOURCE OUT
f	CPLR ARM	m	RCVR R2 IN
g	RCVR A IN	n	SOURCE OUT

E8361A - Option 014 Configurable Test Set, and Option UNL Extended Power Range and Bias - Tees, and Option 016 Receiver Attenuators, and Option 081 Reference Channel Transfer Switch



Item	Description	Item	Description
a	SOURCE OUT	h	RCVR B IN
b	RCVR R1 IN	i	CPLR ARM
c	SOURCE OUT	j	PORT 2
d	CPLR THRU	k	CPLR THRU
e	PORT 1	l	SOURCE OUT
f	CPLR ARM	m	RCVR R2 IN
g	RCVR A IN	n	SOURCE OUT

Technical Specifications for the E8362A, E8363A, E8364A

Because the E8362A, E8363A, and E8364A network analyzer is no longer produced, the technical specifications are stored only on the Internet. To view or print the .pdf version of the specifications, visit our web site at http://www.agilent.com/find/pna, and search for "E836xA Specifications"

The uncertainty curves contained in this document apply only to the setup conditions listed. Please download our free Uncertainty Calculator from http://www.agilent.com/find/na_calculator to generate the curves for your PNA setup. View the equations used to generate the uncertainty curves.

Technical Specifications for the E8362B, E8363B, E8364B

(Rev. 2005-07-12)

This is a complete list of the E8362B, E8363B, and E8364B network analyzer technical specifications.

- To optimize viewing of uncertainty curves, click the Maximize button.

- To view or print the .pdf version of the specifications, visit our web site at http://www.agilent.com/find/pna, and search for "E836xB Specifications"

- The uncertainty curves contained in this document apply only to the setup conditions listed. Please download our free Uncertainty Calculator from http://www.agilent.com/find/na_calculator to generate the curves for your PNA setup. View the equations used to generate the uncertainty curves.

Definitions

Corrected System Performance

- **System Dynamic Range**
- **Receiver Dynamic Range**
- **2.4mm Connectors**
- **2.92mm Connectors**
- **3.5mm Connectors**
- **7mm Connectors**
- **Type-N Connectors**
- **WR-28 Connectors**
- **WR-42 Connectors**
- **WR-62 Connectors**
- **WR-90 Connectors**

Uncorrected System Performance

Test Port Output

Test Port Input

Dynamic Accuracy

Group Delay

General Information

Measurement Throughput Summary

Front-panel Jumper Specs (Option 014 only)

Test Set Block Diagrams

Test Set with Option 014 Block Diagrams

See Specs for other PNA models

Definitions

All specifications and characteristics apply over a 25 °C ±5 °C range (unless otherwise stated) and 90 minutes after the instrument has been turned on.

Specification (spec.): Warranted performance. Specifications include guardbands to account for the expected statistical performance distribution, measurement uncertainties, and changes in performance due to environmental conditions.

Characteristic (char.): A performance parameter that the product is expected to meet before it leaves the factory, but that is not verified in the field and is not covered by the product warranty. A characteristic includes the same guardbands as a specification.

Typical (typ.): Expected performance of an average unit which does not include guardbands. It is not covered by the product warranty.

Nominal (nom.): A general, descriptive term that does not imply a level of performance. It is not covered by the product warranty.

Calibration: The process of measuring known standards to characterize a network analyzer's systematic (repeatable) errors.

Corrected (residual): Indicates performance after error correction (calibration). It is determined by the quality of calibration standards and how well "known" they are, plus system repeatability, stability, and noise.

Uncorrected (raw): Indicates instrument performance without error correction. The uncorrected performance affects the stability of a calibration.

Standard: When referring to the analyzer, this includes no options unless noted otherwise.

Corrected System Performance

The specifications in this section apply for measurements made with the E836xB analyzer with the following conditions:

- 10 Hz IF bandwidth
- No averaging applied to data
- Isolation calibration with an averaging factor of 8

System Dynamic Range

Receiver Dynamic Range

2.4mm Connectors

2.92mm Connectors

3.5mm Connectors

7mm-Connectors

Type-N Connectors

WR-28 Connectors

WR-42-Connectors

WR-62 Connectors

WR-90 Connectors

Table 1. System Dynamic Range[a]

Description	Specification (dB) at Test Port[b]	Typical (dB) at Direct Receiver Access Input[c]	Supplemental Information
Dynamic Range (in a 10 Hz BW)			
Standard Configuration and Standard Power Range (E836xB - Standard)			
10 MHz to 45 MHz[d]	79	NA	
45 MHz to 500 MHz[e]	94	NA	
500 MHz to 2 GHz	119	NA	
2 GHz to 10 GHz	122	NA	
10 GHz to 20 GHz	123	NA	
20 GHz to 30 GHz	114	NA	
30 GHz to 40 GHz	110	NA	
40 GHz to 45 GHz	109	NA	
45 GHz to 50 GHz	104	NA	
Configurable Test Set and Standard Power Range (E836xB - Option 014)			
10 MHz to 45 MHz[d]	79	129	Option 016 degrades performance by 2 dB.
45 MHz to 500 MHz[e]	94	132	
500 MHz to 2 GHz	119	138	
2 GHz to 10 GHz	122	137	
10 GHz to 20 GHz	121	136	
20 GHz to 30 GHz	111	123	
30 GHz to 40 GHz	107	119	
40 GHz to 45 GHz	105	116	
45 GHz to 50 GHz	100	111	

Standard Configuration and Extended Power Range & Bias-Tees (E836xB - Option UNL)			
10 MHz to 45 MHz[d]	79	NA	Option 016 degrades performance by 2 dB.
45 MHz to 500 MHz[e]	92	NA	
500 MHz to 2 GHz	117	NA	
2 GHz to 10 GHz	120	NA	
10 GHz to 20 GHz	121	NA	
20 GHz to 30 GHz	112	NA	
30 GHz to 40 GHz	108	NA	
40 GHz to 45 GHz	105	NA	
45 GHz to 50 GHz	99	NA	
Configurable Test Set and Extended Power Range & Bias-Tees (E836xB - Option 014/UNL)			
10 MHz to 45 MHz[d]	79	129	Option 016 degrades performance by 2 dB.
45 MHz to 500 MHz[e, f]	92	130	
500 MHz to 2 GHz[f]	117	136	
2 GHz to 10 GHz[f]	120	135	
10 GHz to 20 GHz[g]	119	134	
20 GHz to 30 GHz	109	121	
30 GHz to 40 GHz	105	117	
40 GHz to 45 GHz	101	112	
45 GHz to 50 GHz	95	106	

a The system dynamic range is calculated as the difference between the noise floor and the source maximum output power. System dynamic range is a specification when the source is set to Port 1, and a characteristic when the source is set to Port 2. The effective dynamic range must take measurement uncertainties and interfering signals into account as well as the insertion loss resulting from a thru cable connected between Port 1 and Port 2..

b The test port system dynamic range is calculated as the difference between the test port noise floor and the source maximum output power. The effective dynamic range must take measurement uncertainties and interfering signals into account as well as the insertion loss resulting from a thru cable connected between Port 1 and Port 2..

c The direct receiver access input system dynamic range is calculated as the difference between the receiver access input noise floor and the source maximum output power. The effective dynamic range must take measurement uncertainties and interfering signals into account. This set-up should only be used when the receiver input will never exceed its damage level. When the analyzer is in segment sweep mode, the analyzer can have predefined frequency segments which will output a higher power level when the extended dynamic range is required (i.e. devices with high insertion loss), and reduced power when receiver damage may occur (i.e. devices with low insertion loss). The extended range is only available in one-path transmission measurements.

d Typical performance.

e May be limited to 100 dB at particular frequencies below 500 MHz due to spurious receiver residuals. Methods are available to regain the full dynamic range.

f E8362B only: Option H11 decreases value by 1 dB.

g E8362B only: Option H11 decreases value by 2 dB.

Table 2. Receiver Dynamic Range[a]

Description	Specification (dB) at Test Port[b]	Typical (dB) at Direct Receiver Access Input[c]	
Dynamic Range (in a 10 Hz BW)			
Standard Configuration and Standard Power Range (E836xB - Standard) **OR** **Standard Configuration and Extended Power Range & Bias Tees (E836xB - Option UNL)**			
10 MHz to 45 MHz[d]	82	NA	
45 MHz to 500 MHz[e]	94	NA	
500 MHz to 2 GHz	119	NA	
2 GHz to 10 GHz	122	NA	
10 GHz to 20 GHz	125	NA	
20 GHz to 30 GHz	114	NA	Option 016 degrades performance by 2 dB.
30 GHz to 40 GHz	111	NA	
40 GHz to 50 GHz	111	NA	
Configurable Test Set and Standard Power Range (E836xB - Option 014) **OR** **Configurable Test Set and Extended Power Range & Bias Tees (E836xB - Option 014/UNL)**			
10 MHz to 45 MHz[d]	82	132	
45 MHz to 500 MHz[e]	94	132	
500 MHz to 2 GHz	119	138	
2 GHz to 10 GHz	122	137	
10 GHz to 20 GHz	124	139	
20 GHz to 30 GHz	113	125	Option 016 degrades performance by 2 dB.
30 GHz to 40 GHz	110	122	
40 GHz to 50 GHz	109	120	

a The receiver dynamic range is calculated as the difference between the noise floor and the receiver maximum output power. The effective dynamic range must take measurement uncertainties and interfering signals into account.
b The test port receiver dynamic range is calculated as the difference between the test port noise floor and the receiver maximum input level. The effective dynamic range must take measurement uncertainties and interfering signals into account.
c The direct receiver access input receiver dynamic range is calculated as the difference between the direct receiver access input noise floor and the receiver maximum input level. The effective dynamic range must take measurement uncertainties and interfering signals into account. This set-up should only be used when the receiver input will never exceed its compression or damage level. When the analyzer is in segment sweep mode, the analyzer can have predefined frequency segments which will output a higher power level when the extended dynamic range is required (i.e. devices with high insertion loss), and reduced power when compression or receiver damage may occur (i.e. devices with low insertion loss). The extended range is only available in one-path transmission measurements.
d Typical performance.
e May be degraded by 10 dB at particular frequencies (multiples of 5 MHz) below 500 MHz due to spurious receiver residuals. Methods are available to regain the full dynamic range.

Note: This E836xB document provides technical specifications for the following calibration kits only: 85056A, 85056D, 85056K, 85052B, 85052C, 85052D, 85050B, 85050C, 85050D, 85054B, 85054D, K11644A, P11644A, R11644A, and the X11644A.

Table 33. Uncorrected System Performance[a]

Specifications apply over environmental temperature of 23° ±3 °C, with < 1 °C deviation from the calibration temperature

Description	**Specification**	**Supplemental Information**
Directivity		
10 MHz to 45 MHz[b]	23 dB	
45 MHz to 2 GHz	24 dB	
2 GHz to 10 GHz	22 dB	
10 GHz to 20 GHz	16 dB	
20 GHz to 40 GHz	16 dB	
40 GHz to 45 GHz	15 dB	
45 GHz to 50 GHz	13 dB	
Source Match - Standard		
10 MHz to 45 MHz[b]	11 dB	
45 MHz to 2 GHz	23 dB	
2 GHz to 10 GHz	16 dB	
10 GHz to 20 GHz	14 dB	
20 GHz to 40 GHz	10 dB	
40 GHz to 45 GHz	9 dB	
45 GHz to 50 GHz	7.5 dB	
Source Match - Opt UNL, 014 or 014/UNL		
10 MHz to 45 MHz[b]	11 dB	
45 MHz to 2 GHz	18 dB	
2 GHz to 10 GHz	14 dB	
10 GHz to 20 GHz	12 dB	
20 GHz to 40 GHz	9 dB	
40 GHz to 45 GHz	8 dB	

45 GHz to 50 GHz	6 dB	
Load Match - Standard		
10 MHz to 45 MHz[b]	11 dB	
45 MHz to 2 GHz	23 dB	
2 GHz to 10 GHz	14 dB	
10 GHz to 20 GHz	10 dB	
20 GHz to 40 GHz	9 dB	
40 GHz to 45 GHz	9 dB	
45 GHz to 50 GHz	8 dB	
Load Match - Opt UNL, 014 or 014/UNL		
10 MHz to 45 MHz[b]	11 dB	
45 MHz to 2 GHz	17 dB	
2 GHz to 10 GHz	13 dB	
10 GHz to 20 GHz	10 dB	
20 GHz to 40 GHz	9 dB	
40 GHz to 45 GHz	9 dB	
45 GHz to 50 GHz	7 dB	
Reflection Tracking		
		Typical:
10 MHz to 45 MHz		±1.5 dB
45 MHz to 20 GHz		±1.5 dB
20 GHz to 40 GHz		±1.5 dB
40 GHz to 50 GHz		±2.0 dB
Transmission Tracking[c]		
		Typical:
10 MHz to 45 MHz		±3.0 dB
45 MHz to 2 GHz		±1.5 dB

2 GHz to 10 GHz		±2.0 dB
10 GHz to 20 GHz		±2.5 dB
20 GHz to 40 GHz		±3.5 dB
40 GHz to 45 GHz		±4.0 dB
45 GHz to 50 GHz		±4.5 dB
Crosstalk[d] - Standard		
10 MHz to 45 MHz[b]	-65 dB	
45 MHz to 1 GHz	-85 dB	
1 GHz to 2 GHz	-100 dB	
2 GHz to 20 GHz	-110 dB	
20 GHz to 40 GHz	-108 dB	
40 GHz to 45 GHz	-105 dB	
45 GHz to 50 GHz	-100 dB	
Crosstalk[d] - Option UNL or 014		
10 MHz to 45 MHz[b]	-65 dB	
45 MHz to 1 GHz	-85 dB	
1 GHz to 2 GHz	-100 dB	
2 GHz to 20 GHz	-109 dB	
20 GHz to 40 GHz	-106 dB	
40 GHz to 45 GHz	-103 dB	
45 GHz to 50 GHz	-98 dB	
Crosstalk[d] - Option 014/UNL		
10 MHz to 45 MHz[b]	-65 dB	
45 MHz to 1 GHz	-85 dB	
1 GHz to 2 GHz	-98 dB	
2 GHz to 10 GHz	-108 dB	
10 GHz to 20 GHz	-107 dB	

20 GHz to 40 GHz	-104 dB	
40 GHz to 45 GHz	-100 dB	
45 GHz to 50 GHz	-95 dB	
Crosstalk - Option 080 enabled[b,e]		
		Typical:
10 MHz to 45 MHz		-65
45 MHz to 1 GHz		-85
1 GHz to 2 GHz		-100
2 GHz to 10 GHz		-109
10 GHz to 20 GHz		-110
20 GHz to 40 GHz		-106
40 GHz to 45 GHz		-103
45 GHz to 50 GHz		-98

a Specifications apply over environment temperature of 23°C +/- 3°C, with less than 1°C deviation from the calibration temperature.

b Typical performance.

cTransmission tracking performance is strongly dependent on cable used. These typical specifications are based on the use of the Agilent thru cable (part number 85133-60016).

d Measurement conditions: normalized to a thru, measured with two shorts, 10 Hz IF bandwidth, averaging factor of 16, alternate mode, source power set to the lesser of the maximum power out or the maximum receiver power.

e 0 Hz offset.

Table 34. Test Port Output

Description	**Specification**				**Supplemental**
Frequency Range					
	Standard	**Opt 014**	**Opt UNL**	**Opt 014/UNL**	
E8362B	10 MHz to 20 GHz				
E8363B	10 MHz to 40 GHz				
E8364B	10 MHz to 50 GHz				
Nominal Power[c]					
E8362B	0 dBm	-5 dBm	-5 dBm	-5 dBm	

E8363/4B	-12 dBm	-17 dBm	-17 dBm	-17 dBm	
Frequency Resolution					
	1 Hz				
CW Accuracy					
	+/-1 ppm				
Frequency Stability					
					+/-0.05 ppm. -10° to 70° C, typical; +/-0.1 ppm/yr maximum, typical
Power Level Accuracy[a]					
10 MHz to 45 MHz[b]	+/-2.0 dB	+/-2.0 dB	+/-2.0 dB	+/-2.0 dB	
45 MHz to 10 GHz	+/-1.5 dB	+/-1.5 dB	+/-1.5 dB	+/-1.5 dB	Variation from nominal power in range 0 (step attenuator at 0 dB)
10 GHz to 20 GHz	+/-2.0 dB	+/-2.0 dB	+/-2.0 dB	+/-2.0 dB	
20 GHz to 40 GHz	+/-3.0 dB	+/-3.0 dB	+/-3.0 dB	+/-3.0 dB	
40 GHz to 45 GHz	+/-3.0 dB	+/-3.5 dB	+/-3.0 dB	+/-3.5 dB	
45 GHz to 50 GHz	+/-3.0 dB	+/-4.0 dB	+/-3.0 dB	+/-4.0 dB	
Power Level Linearity[d]					

10 MHz to 45 MHz[b]	+/-1.0 dB[g]				Test reference is at the nominal power level (step attenuator at 0 dB)
45 MHz to 20 GHz	+/-1.0 dB[g]				
20 GHz to 40 GHz	+/-1.0 dB[g]				
40 GHz to 50 GHz	+/-1.0 dB[g]				
Power Range[a, e, f]					
10 MHz to 45 MHz[b]	-25 to +2 dBm	-25 to +2 dBm	-87 to +2 dBm	-87 to +2 dBm	
45 MHz to 10 GHz	-25 to +5 dBm	-25 to +5 dBm	-87 to +3 dBm	-87 to +3 dBm[h]	
10 GHz to 20 GHz	-24 to +3 dBm	-25 to +2 dBm	-86 to +1 dBm	-87 to 0 dBm[i]	
20 GHz to 30 GHz	-23 to 0 dBm	-25 to -2 dBm	-85 to -2 dBm	-87 to -4 dBm	
30 GHz to 40 GHz	-23 to -4 dBm	-25 to -6 dBm	-85 to -6 dBm	-87 to -8 dBm	
40 GHz to 45 GHz	-25 to -5 dBm	-27 to -7 dBm	-87 to -9 dBm	-87 to -11 dBm	
45 GHz to 50 GHz	-25 to -10 dBm	-27 to -12 dBm	-87 to -15 dBm	-87 to -17 dBm	
Power Sweep Range (ALC)					
10 MHz to 45 MHz[b]	27 dB	27 dB	29 dB	29 dB	

45 MHz to 10 GHz	30 dB	30 dB	30 dB	30 dB[j]	ALC range starts at maximum leveled output power and decreases by power level indicated in the table.
10 GHz to 20 GHz	27 dB	27 dB	27 dB	27 dB[k]	
20 GHz to 30 GHz	23 dB	23 dB	23 dB	23 dB	
30 GHz to 40 GHz	19 dB	19 dB	19 dB	19 dB	
40 GHz to 45 GHz	20 dB	20 dB	18 dB	16 dB	
45 GHz to 50 GHz	15 dB	15 dB	12 dB	10 dB	
Power Resolution					
	0.01 dB				
Phase Noise					
1 kHz offset from center frequency, nominal power at test port					
					Typical:
10 MHz to 10 GHz					-60 dBc
10 GHz to 20 GHz					-55 dBc
20 GHz to 50 GHz					-50 dBc
1 kHz offset from center frequency, nominal power at test port - Option 080 enabled					
					Typical:
10 MHz to 10 GHz					-60 dBc
10 GHz to 20 GHz					-60 dBc
20 GHz to 50 GHz					-50 dBc
10 kHz offset from center frequency, nominal power at test port					
					Typical:
10 MHz to 45 MHz					-70 dBc
45 MHz to 10 GHz					-70 dBc
10 GHz to 20 GHz					-65 dBc

20 GHz to 40 GHz					-55 dBc
40 GHz to 50 GHz					-55 dBc
10 kHz offset from center frequency, nominal power at test port - Option 080 enabled					
					Typical:
10 MHz to 45 MHz					-70 dBc
45 MHz to 10 GHz					-70 dBc
10 GHz to 20 GHz					-65 dBc
20 GHz to 40 GHz					-55 dBc
40 GHz to 50 GHz					-55 dBc
100 kHz offset from center frequency, nominal power at test port					
					Typical:
10 MHz to 10 GHz					-60 dBc
10 GHz to 20 GHz					-55 dBc
20 GHz to 50 GHz					-50 dBc
100 kHz offset from center frequency, nominal power at test port - Option 080 enabled					
					Typical:
10 MHz to 10 GHz					-75 dBc
10 GHz to 20 GHz					-70 dBc
20 GHz to 50 GHz					-65 dBc
1 MHz offset from center frequency, nominal power at test port					
					Typical:
10 MHz to 10 GHz					-106 dBc
10 GHz to 20 GHz					-103 dBc
20 GHz to 50 GHz					-90 dBc
1 MHz offset from center frequency, nominal power at test port - Option 080 enabled					
					Typical:
10 MHz to 10 GHz					-103 dBc

10 GHz to 20 GHz					-97 dBc
20 GHz to 50 GHz					-85 dBc
Harmonics (2nd or 3rd)					
					-23 dBc typical, in power range 0
Non-Harmonic Spurious (at Nominal Output Power)					
10 MHz to 45 MHz					-50 dBc typical, for offset frequency > 1 kHz
45 MHz to 20 GHz					-50 dBc typical, for offset frequency > 1 kHz
20 GHz to 40 GHz					-30 dBc typical, for offset frequency > 1 kHz
40 GHz to 50 GHz					-30 dBc typical, for offset frequency > 1 kHz

a Test port output is a specification when the source is set to Port 1, and a characteristic when the source is set to Port 2.

b Typical performance.

c Preset power.

d Power Level Linearity is a specification when the source is set to Port 1, and a typical when the source is set to Port 2.

e Test port power is specified into nominal 50 ohms.

f Power to which the source can be set and phase lock is assured.

g +/-1.5 dB for power <= -23 dBm.

h E8362B only: Option H11 decreases maximum power level by 1 dB.

i E8362B only: Option H11 decreases maximum power level by 2 dB.

j E8362B only: Option H11 decreases power level by 1 dB.

k E8362B only: Option H11 decreases power level by 2 dB.

Table 35: Test Port Input

Description	**Specification**				**Supplemental**
	Standard	**Opt 014**	**Opt UNL**	**Opt 014/UNL**	

Test Port Noise Floor[a]					
10 Hz IF Bandwidth					
10 MHz to 45 MHz[b]	<-77 dBm	<-77 dBm	<-77 dBm	<-77 dBm	
45 MHz to 500 MHz[c]	<-89 dBm	<-89 dBm	<-89 dBm	<-89 dBm	
500 MHz to 2 GHz	<-114 dBm	<-114 dBm	<-114 dBm	<-114 dBm	
2 GHz to 10 GHz	<-117 dBm	<-117 dBm	<-117 dBm	<-117 dBm	
10 GHz to 20 GHz	<-120 dBm	<-119 dBm	<-120 dBm	<-119 dBm	
20 GHz to 40 GHz	<-114 dBm	<-113 dBm	<-114 dBm	<-113 dBm	Option 016 degrades performance by 2 dB.
40 GHz to 50 GHz	<-114 dBm	<-112 dBm	<-114 dBm	<-112 dBm	
1 KHz IF Bandwidth					
10 MHz to 45 MHz[b]	<-57 dBm	<-57 dBm	<-57 dBm	<-57 dBm	
45 MHz to 500 MHz[c]	<-69 dBm	<-69 dBm	<-69 dBm	<-69 dBm	
500 MHz to 2 GHz	<-94 dBm	<-94 dBm	<-94 dBm	<-94 dBm	
2 GHz to 10 GHz	<-97 dBm	<-97 dBm	<-97 dBm	<-97 dBm	
10 GHz to 20 GHz	<-100 dBm	<-99 dBm	<-100 dBm	<-99 dBm	
20 GHz to 40 GHz	<-94 dBm	<-93 dBm	<-94 dBm	<-93 dBm	Option 016 degrades performance by 2 dB.
40 GHz to 50 GHz	<-94 dBm	<-92 dBm	<-94 dBm	<-92 dBm	

Test Port Noise Floor[a,b] Option 080 enabled[d]					
10 Hz IF Bandwidth					
10 MHz to 45 MHz[b]	<-77 dBm	<-77 dBm	<-77 dBm	<-77 dBm	
45 MHz to 500 MHz[c]	<-88 dBm	<-88 dBm	<-88 dBm	<-88 dBm	
500 MHz to 2 GHz	<-113 dBm	<-113 dBm	<-113 dBm	<-113 dBm	
2 GHz to 10 GHz	<-116 dBm	<-116 dBm	<-116 dBm	<-116 dBm	
10 GHz to 20 GHz	<-118 dBm	<-118 dBm	<-118 dBm	<-118 dBm	
20 GHz to 40 GHz	<-112 dBm	<-112 dBm	<-112 dBm	<-112 dBm	Option 016 degrades performance by 2 dB.
40 GHz to 50 GHz	<-111 dBm	<-111 dBm	<-111 dBm	<-111 dBm	
1 KHz IF Bandwidth					
10 MHz to 45 MHz[b]	<-57 dBm	<-57 dBm	<-57 dBm	<-57 dBm	
45 MHz to 500 MHz[c]	<-68 dBm	<-68 dBm	<-68 dBm	<-68 dBm	
500 MHz to 2 GHz	<-93 dBm	<-93 dBm	<-93 dBm	<-93 dBm	
2 GHz to 10 GHz	<-96 dBm	<-96 dBm	<-96 dBm	<-96 dBm	
10 GHz to 20 GHz	<-98 dBm	<-98 dBm	<-98 dBm	<-98 dBm	
20 GHz to 40 GHz	<-92 dBm	<-92 dBm	<-92 dBm	<-92 dBm	Option 016 degrades performance by 2 dB.
40 GHz to 50 GHz	<-91 dBm	<-91 dBm	<-91 dBm	<-91 dBm	

Direct Receiver Access Input Noise Floor[a,b]					
10 Hz IF Bandwidth					
10 MHz to 45 MHz		<-127 dBm		<-127 dBm	
45 MHz to 500 MHz[c]		<-127 dBm		<-127 dBm	
500 MHz to 2 GHz		<-133 dBm		<-133 dBm	
2 GHz to 10 GHz		<-132 dBm		<-132 dBm	
10 GHz to 20 GHz		<-134 dBm		<-134 dBm	
20 GHz to 40 GHz		<-125 dBm		<-125 dBm	Option 016 degrades performance by 2 dB.
40 GHz to 50 GHz		<-123 dBm		<-123 dBm	
1 KHz IF Bandwidth					
10 MHz to 45 MHz		<-107 dBm		<-107 dBm	
45 MHz to 500 MHz[c]		<-107 dBm		<-107 dBm	
500 MHz to 2 GHz		<-113 dBm		<-113 dBm	
2 GHz to 10 GHz		<-112 dBm		<-112 dBm	
10 GHz to 20 GHz		<-114 dBm		<-114 dBm	
20 GHz to 40 GHz		<-105 dBm		<-105 dBm	Option 016 degrades performance by 2 dB.
40 GHz to 50 GHz		<-103 dBm		<-103 dBm	
Direct Receiver Access Input Noise Floor[a,b] - Option 080 enabled[d]					

10 Hz IF Bandwidth					
10 MHz to 45 MHz		<-127 dBm		<-127 dBm	
45 MHz to 500 MHz[c]		<-126 dBm		<-126 dBm	
500 MHz to 2 GHz		<-132 dBm		<-132 dBm	
2 GHz to 10 GHz		<-131 dBm		<-131 dBm	
10 GHz to 20 GHz		<-133 dBm		<-133 dBm	
20 GHz to 40 GHz		<-124 dBm		<-124 dBm	Option 016 degrades performance by 2 dB.
40 GHz to 50 GHz		<-122 dBm		<-122 dBm	
1 KHz IF Bandwidth					
10 MHz to 45 MHz		<-107 dBm		<-107 dBm	
45 MHz to 500 MHz[c]		<-106 dBm		<-106 dBm	
500 MHz to 2 GHz		<-112 dBm		<-112 dBm	
2 GHz to 10 GHz		<-111 dBm		<-111 dBm	
10 GHz to 20 GHz		<-113 dBm		<-113 dBm	
20 GHz to 40 GHz		<-104 dBm		<-104 dBm	Option 016 degrades performance by 2 dB.
40 GHz to 50 GHz		<-102 dBm		<-102 dBm	
Receiver Compression Level					
10 MHz to 20 GHz	<0.1 dB at -5 dBm[g] and <0.45 dB at +5 dBm				

20 GHz to 30 GHz	<0.1 dB at -9.5 dBm[g] and <0.45 dB at 0 dBm	
30 GHz to 40 GHz	<0.1 dB at -12.5 dBm[g] and <0.45 dB at -3 dBm	
40 GHz to 50 GHz	<0.1 dB at -12.5 dBm[g] and <0.45 dB at -3 dBm	
System Compression Level		
	maximum output power	See dynamic accuracy table
Third Order Intercept - Tone spacing from 100 kHz - 5 MHz		
		Typical
10 MHz to 150 MHz		+33 dBm
150 MHz to 300 MHz		+34 dBm
300 MHz to 500 MHz		+30 dBm
500 MHz to 20 GHz		+24 dBm
20 to 40 GHz		+18 dBm
40 to 50 GHz		+15 dBm
Third Order Intercept - Tone spacing from 5 MHz - 20 MHz		
		Typical
10 MHz to 500 MHz		+20 dBm
500 MHz to 20 GHz		+20 dBm
20 to 40 GHz		+16 dBm
40 to 50 GHz		+15 dBm
Third Order Intercept - Tone spacing from 20 MHz - 50 MHz		
		Typical

10 MHz to 500 MHz		+26 dBm
500 MHz to 20 GHz		+26 dBm
20 to 40 GHz		+20 dBm
40 to 50 GHz		+19 dBm
Trace Noise Magnitude 1 kHz IF bandwidth. Ratio measurement, nominal power at test port.		
10 MHz to 45 MHz[b]	<0.050 dB rms	
45 MHz to 500 MHz[e]	<0.010 dB rms	
500 MHz to 20 GHz	<0.006 dB rms	
20 GHz to 40 GHz	<0.006 dB rms	
40 GHz to 50 GHz	<0.006 dB rms	
Trace Noise Magnitude - Option 080 enabled[b,d] 1 kHz IF bandwidth. Ratio measurement, nominal power at test port.		
10 MHz to 45 MHz[b]	<0.060 dB rms	
45 MHz to 500 MHz[e]	<0.010 dB rms	
500 MHz to 20 GHz	<0.006 dB rms	
20 GHz to 40 GHz	<0.007 dB rms	
40 GHz to 50 GHz	<0.008 dB rms	
Trace Noise Phase 1 kHz IF bandwidth. Ratio measurement, nominal power at test port.		

10 MHz to 45 MHz[b]	<0.350° rms	
45 MHz to 500 MHz	<0.100° rms	
500 MHz to 20 GHz	<0.060° rms	
20 GHz to 40 GHz	<0.100° rms	
40 GHz to 50 GHz	<0.100° rms	
Trace Noise Phase - Option 080 enabled[b,d] 1 kHz IF bandwidth. Ratio measurement, nominal power at test port.		
10 MHz to 45 MHz	<0.350° rms	
45 MHz to 500 MHz[e]	<0.100° rms	
500 MHz to 20 GHz	<0.060° rms	
20 GHz to 40 GHz	<0.100° rms	
40 GHz to 50 GHz	<0.100° rms	
Reference Level Magnitude		
Range	+/-200 dB	
Resolution	0.001 dB	
Reference Level Phase		
Range	+/-500°	
Resolution	0.01°	
Stability Magnitude[d]		
Typical ratio measurement, made at the test port.		

10 MHz to 45 MHz					+/-0.05 dB/°C
45 MHz to 20 GHz					+/-0.02 dB/°C
20 GHz to 40 GHz					+/-0.03 dB/°C
40 GHz to 50 GHz					+/-0.04 dB/°C
Stability Phase[d]					
Typical ratio measurement, measured at the test port.					
10 MHz to 45 MHz					+/-0.5°/°C
45 MHz to 20 GHz					+/-0.2°/°C
20 GHz to 40 GHz					+/-0.5°/°C
40 GHz to 50 GHz					+/-0.8°/°C
Damage Input Level					
Test Port 1 and 2					+30 dBm or +/-40 VDC, typical
R1, R2 in					+15 dBm or +/-15 VDC, typical
A, B in					+15 dBm or +/-15 VDC, typical
Coupler Thru (Option 014 or UNL/014)					+30 dBm or +/-40 VDC, typical
Coupler Arm (Option 014 or UNL/014)					+30 dBm or +/-7 VDC, typical

aTotal average (rms) noise power calculated as the mean value of a linear magnitude trace expressed in dBm.

bTypical performance.

cNoise floor may be degraded by 10 dB at particular frequencies (multiples of 5 MHz) due to spurious receiver residuals.

d0 Hz offset

eTrace noise magnitude may be degraded to 20 mdB rms at harmonic frequencies of the first IF (8.33 MHz) below 80 MHz.

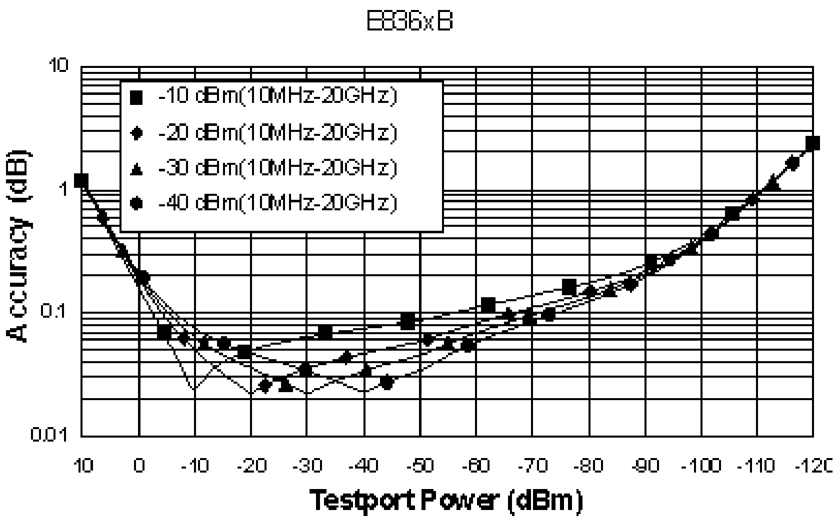
fStability is defined as a ratio measurement made at the test port.

g This compression level comes from the dynamic accuracy curve with -30 dBm reference test port power.

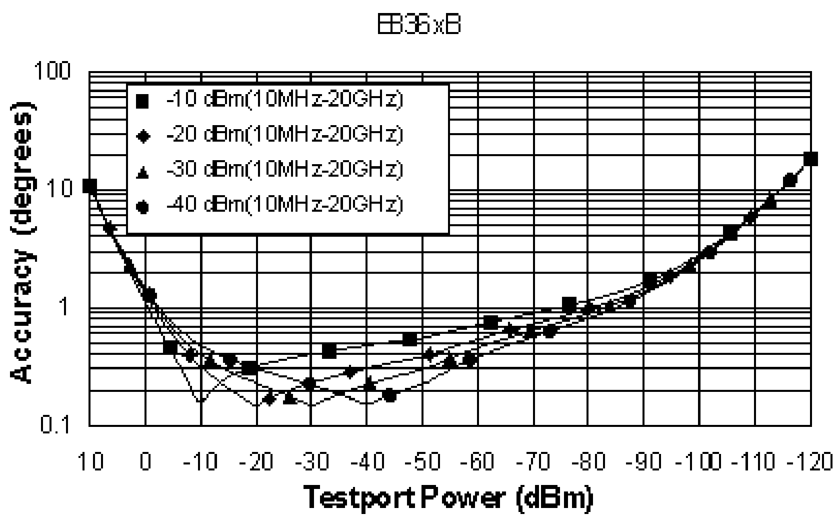
Table 36. Dynamic Accuracy (Specification[a])

Accuracy of the test port input power reading relative to the reference input power level.

Magnitude*

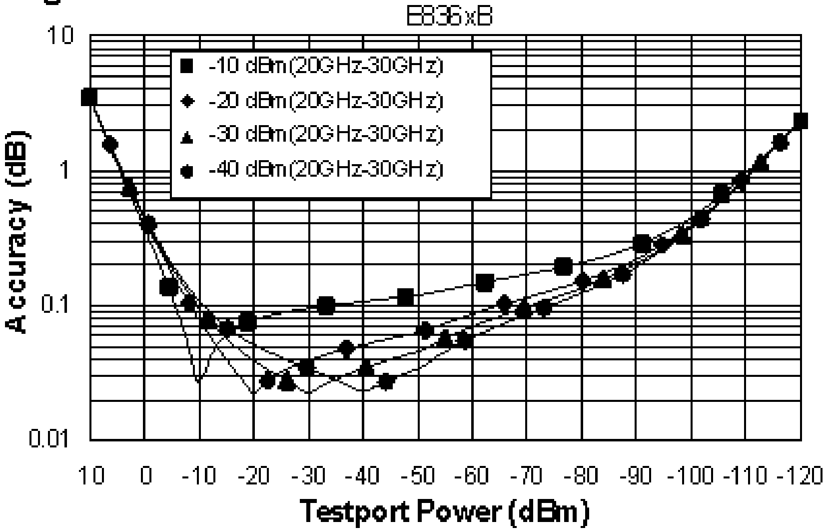


Phase*

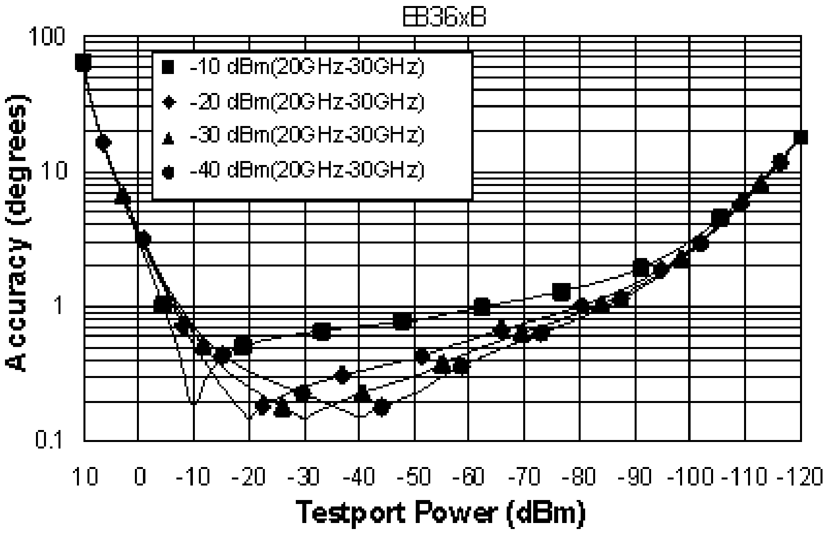


*Below 800 MHz the coupling factor rolls off 20 dB per decade causing a shift in the dynamic accuracy curves. Please see the Uncertainty Calculator (http://www.agilent.com/find/na_calculator) for detailed compression values.

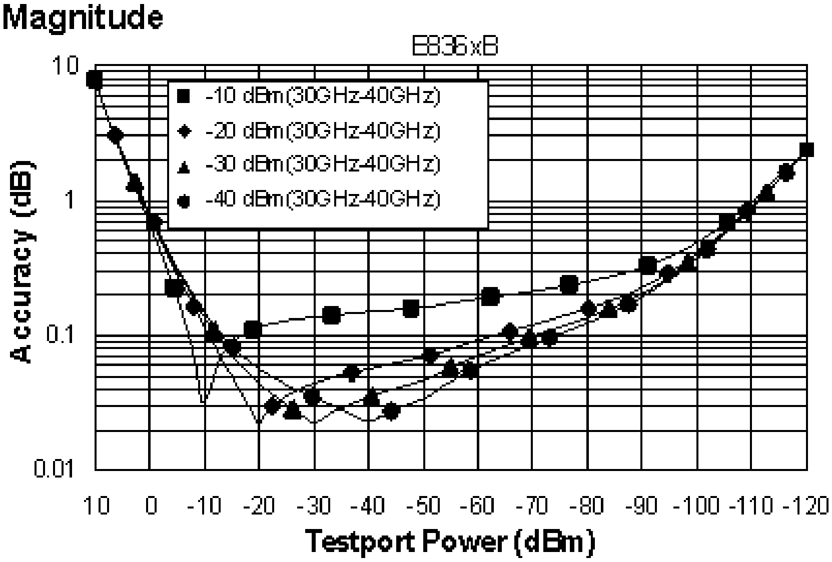
Magnitude



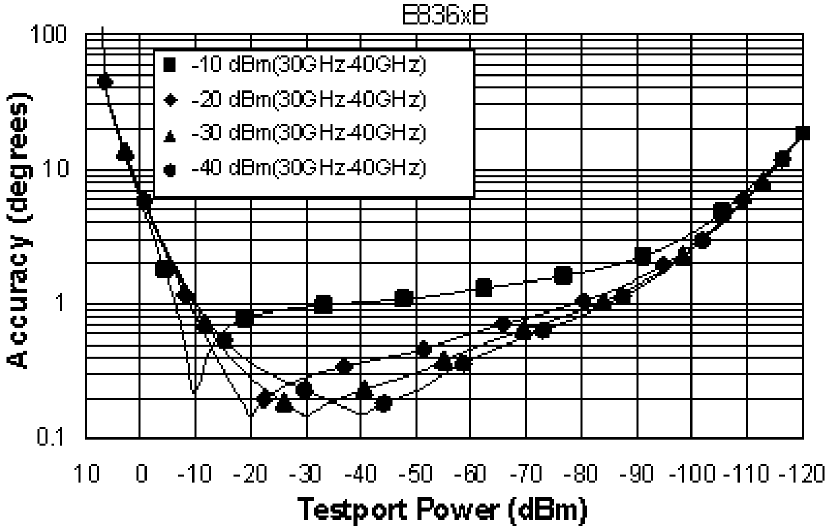
Phase



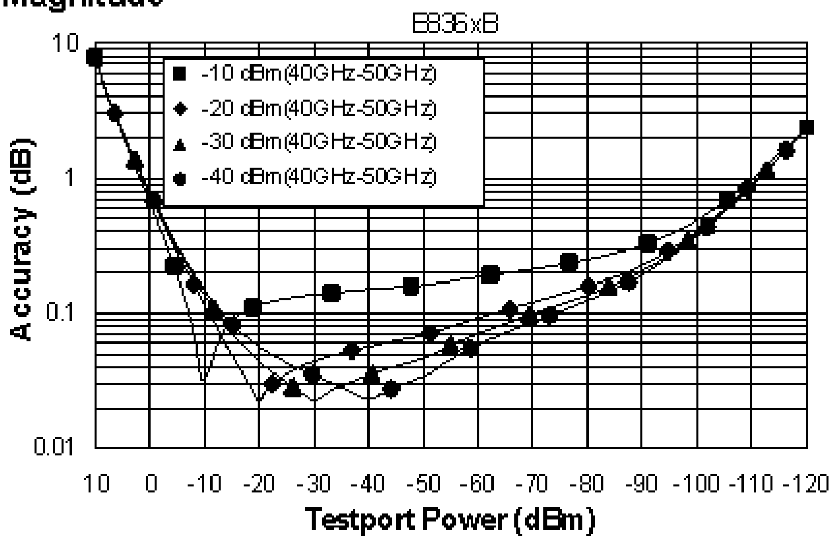
Magnitude



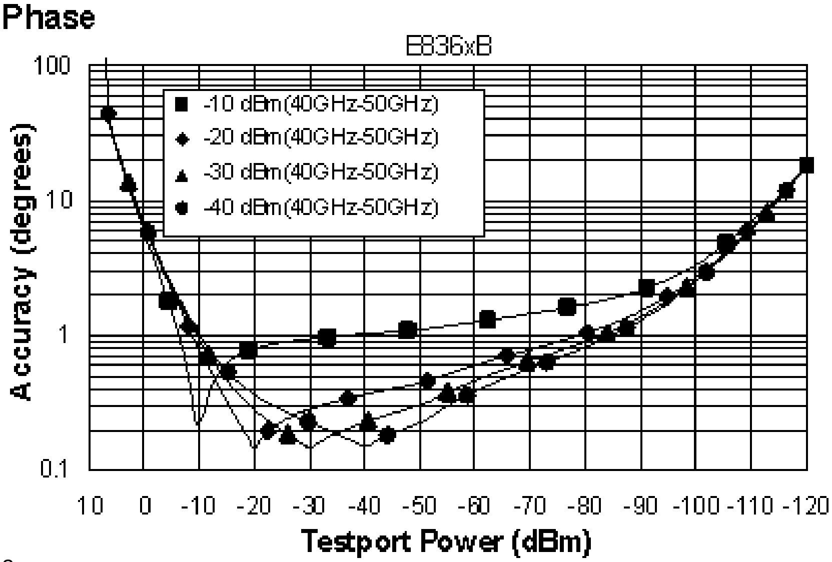
Phase



Magnitude



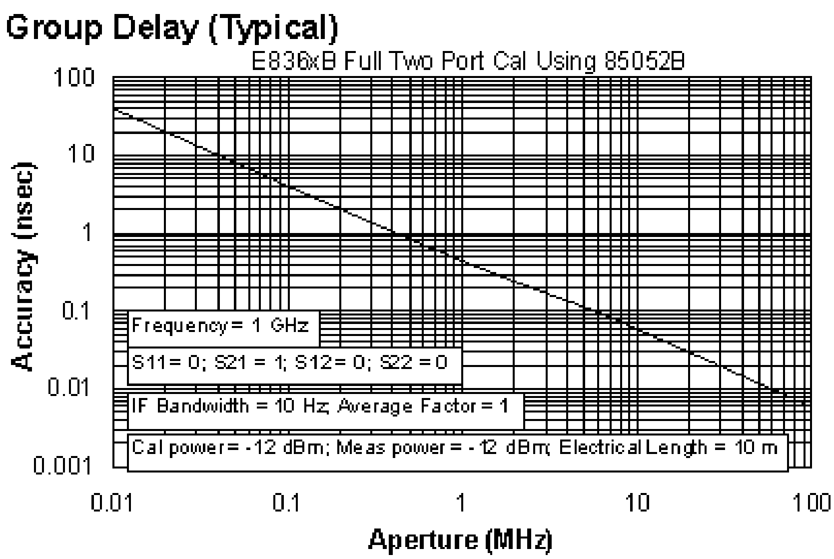
Phase



[a] Dynamic accuracy is verified with the following measurements:

- Compression over frequency.
- IF linearity at a single frequency of 1.195 GHz and a reference level of -20 dBm for an input power range of 0 to -120 dBm.

Table 37. Test Port Input (Group Delay)[a]

Description	**Specification**	**Supplemental Information (typ.)**
Aperture (selectable)		(frequency span)/(number of points -1)
Maximum Aperture		20% of frequency span
Range		0.5 x (1/minimum aperture)
Maximum Delay		Limited to measuring no more than 180° of phase change within the minimum aperture.)
Accuracy		See graph below. Char.
The following graph shows characteristic group delay accuracy with full 2-port calibration and a 10 Hz IF bandwidth. Insertion loss is assumed to be < 2 dB and electrical length to be ten meters. 		
In general, the following formula can be used to determine the accuracy, in seconds, of specific group delay measurement: ±Phase Accuracy (deg)/[360 × Aperture (Hz)] Depending on the aperture and device length, the phase accuracy used is either incremental phase accuracy or worst case phase accuracy.		

a Group delay is computed by measuring the phase change within a specified frequency step (determined by the frequency span and the number of points per sweep).

General Information

- **Miscellaneous Information**
- **Front Panel**
- **Rear Panel**
- **Environment and Dimensions**

Table 38. Miscellaneous Information

Description	Specification	Supplemental Information
System IF Bandwidth Range		1 Hz to 40 kHz, nominal
CPU		Intel® 500 MHz Pentium® III

Table 39. Front Panel Information

Description	Supplemental Information
RF Connectors	
E8362B	
Type	3.5 mm (male), 50 ohm, (nominal)
Center Pin Recession	0.002 in. (characteristic)
E8363/4B	
Type	2.4 mm (male), 50 ohm, (nominal)
Center Pin Recession	0.002 in. (characteristic)
Display	
Size	21.3 cm (8.4 in) diagonal color active matrix LCD; 640 (horizontal) X 480 (vertical) resolution; 59.83 Hz vertical refresh rate; 31.41 Hz horizontal refresh rate
Refresh Rate	Vertical 59.83 Hz; Horizontal 31.41 kHz
Display Range	
Magnitude	±200 dB (at 20 dB/div), max
Phase	±500°, max
Polar	10 pUnits, min 1000 Units, max
Display Resolution	
Magnitude	0.001 dB/div, min
Phase	0.01°/div, min
Marker Resolution	
Magnitude	0.001 dB, min
Phase	0.01°, min
Polar	0.01 mUnit, min; 0.01°,min

Table 40. Rear Panel Information

Description	Supplemental Information
10 MHz Reference In	

Connector	BNC, female
Input Frequency	10 MHz ± 10 ppm, typical
Input Level	-15 dBm to +20 dBm, typical
Input Impedance	200 Ω, nom.
10 MHz Reference Out	
Connector	BNC, female
Output Frequency	10 MHz ± 1 ppm, typical
Signal Type	Sine Wave, typical
Output Level	+10 dBm ± 4 dB into 50 Ω, typical
Output Impedance	50 Ω, nominal
Harmonics	<-40 dBc, typical
Option H08 & H11 Rear Panel Connectors (typical)	
IF Connectors	A, R1, R2, B (BNC Connectors)
IF Connector Input Frequency	8 1/3 MHz
Nominal Input Impedance at IF Inputs	50 Ω
RF Damage Level to IF Connector Inputs	-20.0 dBm
DC Damage Level to IF Connector Inputs	25 volts
0.1 dB Compression Point at IF Inputs	-27.0 dBm
Pulse Input Connectors[1]	A, R1, R2, B (BNC Connectors)
Nominal Input Impedance at Pulse Inputs	1 Kohm
Minimum IF Gate Width	20 ns for less than 1 dB deviation from theoretical performance[2]
DC Damage Level to Pulse Connector Inputs	5.5 volts
Drive Voltage	TTL (0, +5.0) Volts
Rear Panel LO Power (Typical)	

1.7 GHz- 20 GHz	-7 to -16 dBm
Rear Panel RF Power 8362B (Typical)	
1.7 GHz to 20 GHz	-5 to 16 dBm (at 5 dBm test port power[3])
Rear Panel RF Power 8363B/8364B (Typical)	
1.7 GHz to 10 GHz	-2 to 12 dBm (at 5 dBm test port power[3])
10 GHz to 16 GHz	0 to 8 dBm (at 5 dBm test port power[3])
16 GHz to 20 GHz	+5 to 1 dBm (at 5 dBm test port power[3])
VGA Video Output	
Connector	15-pin mini D-Sub; Drives VGA compatible monitors
Devices Supported:	
	Resolutions:
Flat Panel (TFT)	1024 X 768, 800 X 600, 640 X 480
Flat Panel (DSTN)	800 X 600, 640 X 480
CRT Monitor	1280 X 1024, 1024 X 768, 800 X 600, 640 X 480
	Simultaneous operation of the internal and external displays is allowed, but with 640 X 480 resolution only. If you change resolution, you can only view the external display (internal display will "white out").
Bias Input Connectors (Option UNL)	
Bias current	500 mA, maximum
Bias voltage	40 Volts, maximum
Test Set IO	
	25-pin D-Sub connector, available for external test set control.
Aux IO	
	25-pin D-Sub connector, male, analog and digital IO.
Handler IO	
	36-pin parallel I/O port; all input/output signals are default set to negative logic; can be reset to positive logic via GPIB command.
GPIB	
	24-pin D-sub (Type D-24), female; compatible with IEEE-488.
Parallel Port (LPT1)	

	25-pin D-Sub miniature connector, female; provides connection to printers or any other parallel port peripherals
Serial Port (COM 1)	
	9-pin D-Sub, male; compatible with RS-232
USB Port	
	One port on front panel and five ports on rear panel. Universal Serial Bus jack, Type A configuration (4 contacts inline, contact 1 on left); female
Contact 1	Vcc: 4.75 to 5.25 VDC, 500 mA, maximum
Contact 2	-Data
Contact 3	+Data
Contact 4	Ground
LAN	
	10/100BaseT Ethernet, 8-pin configuration; auto selects between the two data rates
Line Power (A third-wire ground is required.)	
Frequency	50/60/400 Hz
Voltages	120/240 VAC (Power supply is auto switching.)
Max	500 Watts

1 Pulse input connectors are operational only with Option H08 (Pulse Measurement Capability) enabled.

2 Based on deviation from signal reduction equation: Signal Reduction (dB) = $20\log_{10}$(Duty_cycle) = $20\log_{10}$(pulse_width/period). Measured at Pulse Repetition Frequency (PFR) of 1 MHz.

3 Test port power has to be at a high enough level such that the Drop Cal does not occur. If Drop Cal occurs then the power out of the rear panel RF connector will drop by about 15 dB.

Table 41. Analyzer Environment and Dimensions

Description	Supplemental Information
General Environmental	
RFI/EMI Susceptibility	Defined by CISPR Pub. 11, Group 1, Class A, and IEC 50082-1
ESD	Minimize using static-safe work procedures and an antistatic bench mat
Dust	Minimize for optimum reliability
Operating Environment	
Temperature	0 °C to +40 °C Instrument powers up, phase locks, and displays no error messages within this temperature range (except for "source unleveled" error message that may occur at temperature extremes).
Error-Corrected Temperature Range	23°C ± 3°C with less than 1°C deviation from calibration temp.
Humidity	5% to 95% at +40 °C
Altitude	0 to 4500 m (14,760 ft.)
Non-Operating Storage Environment	
Temperature	-40 °C to +70 °C
Humidity	0% to 90% at +65 °C (non-condensing)
Altitude	0 to 15,240 m (50,000 ft.)

Cabinet Dimensions			
	Height	**Width**	**Depth**
Excluding front and rear panel hardware and feet	267 mm 10.5 in	426 mm 16.75 in	427 mm 16.8 in
As shipped - includes front panel connectors, rear panel bumpers, and feet.	280 mm 11.0 in	435 mm 17.1 in	470 mm 18.5 in
As shipped plus handles	280 mm 11.0 in	458 mm 18 in	501 mm 19.70 in
As shipped plus rack-mount flanges	280 mm 11.0 in	483 mm 19 in	470 mm 18.5 in
As shipped plus handles and flanges	280 mm 11.0 in	483 mm 19 in	501 mm 19.70 in

Weight	
Net	
E8362B	28.6 kg (63.5 lb), nominal
E8363/4B	29 kg (64 lb), nominal
Shipping	

E8362B	35.8 kg (79.5 lb), nominal
E8363/4B	36.3 kg 80 lb), nominal

Measurement Throughput Summary

- **Typical Cycle Time for Measurement Completion**
- **Cycle Time vs IF Bandwidth**
- **Cycle Time vs Number of Points**
- **Data Transfer Time**

Table 42 Typical Cycle Time[a,b] (ms) for Measurement Completion

	Number of Points			
	201	**401**	**1601**	**16,001**
Start 28 GHz, Stop 30 GHz, 35 kHz IF bandwidth				
Uncorrected, 1-port cal	12	19	55	503
2-Port cal	29	44	124	1112
Start 10 MHz, Stop 10 GHz, 35 kHz IF bandwidth				
Uncorrected, 1-port cal	86	93	121	583
2-Port cal	179	199	267	1301
Start 10 MHz, Stop 20 GHz, 35 kHz IF bandwidth				
Uncorrected, 1-port cal	126	130	153	597
2-Port cal	264	275	335	1321
Start 10 MHz, Stop 40 GHz, 35 kHz IF bandwidth				
Uncorrected, 1-port cal	185	190	213	621
2-Port cal	382	401	459	1374
Start 10 MHz, Stop 50 GHz, 35 kHz IF bandwidth				
Uncorrected, 1-port cal	210	216	243	643
2-Port cal	436	450	522	1405

Start 10 MHz, Stop 67 GHz, 35 kHz IF bandwidth				
Uncorrected 1-Port cal	244	254	300	645
2-Port cal	502	524	591	1423

a Typical performance.
b Includes sweep time, retrace time and band-crossing time. Analyzer display turned off with DISPLAY:ENABLE OFF. Add 21 ms for display on. Data for one trace (S11) measurement.

Table 43. Cycle Time vs IF Bandwidth[a]

Applies to the Preset condition (201 points, correction off) except for the following changes:

- CF = 28 GHz
- Span = 100 MHz
- Display off (add 21 ms for display on)

IF Bandwidth (Hz)	Cycle Time (ms)[b]	Cycle Time (ms) Option 080 enabled
40,000	11	100
35,000	12	101
30,000	13	102
20,000	16	106
10,000	30	127
7000	38	138
5000	50	152
3000	74	182
1000	274	326
300	694	782
100	1905	2054
30	6091	6355
10	17916	18372

a Typical performance.
b Cycle time includes sweep and retrace time.

Table 44. Cycle Time vs Number of Points[a]

Applies to the Preset condition (35 kHz IF bandwidth, correction off) except for the following changes:

- CF = 28 GHz
- Span = 100 MHz
- Display off (add 21 ms for display on)

Number of Points	**Cycle Time (ms)**[b]
3	6
11	6
51	7
101	9
201	12
401	18
801	30
1601	55
16,001	497

a Typical performance.
b Cycle time includes sweep and retrace time.

Table 45. Data Transfer Time (ms)[a]

	Number of Points			
	201	401	1601	16,001
SCPI over GPIB (program executed on external PC)				
32-bit floating point	7	12	43	435
64-bit floating point	12	22	84	856
ASCII	64	124	489	5054
SCPI (program executed in the analyzer)				
32-bit floating point	1	2	3	30
64-bit floating point	2	2	4	40
ASCII	29	56	222	2220
COM (program executed in the analyzer)				
32-bit floating point	1	1	1	6
Variant type	1	2	6	68

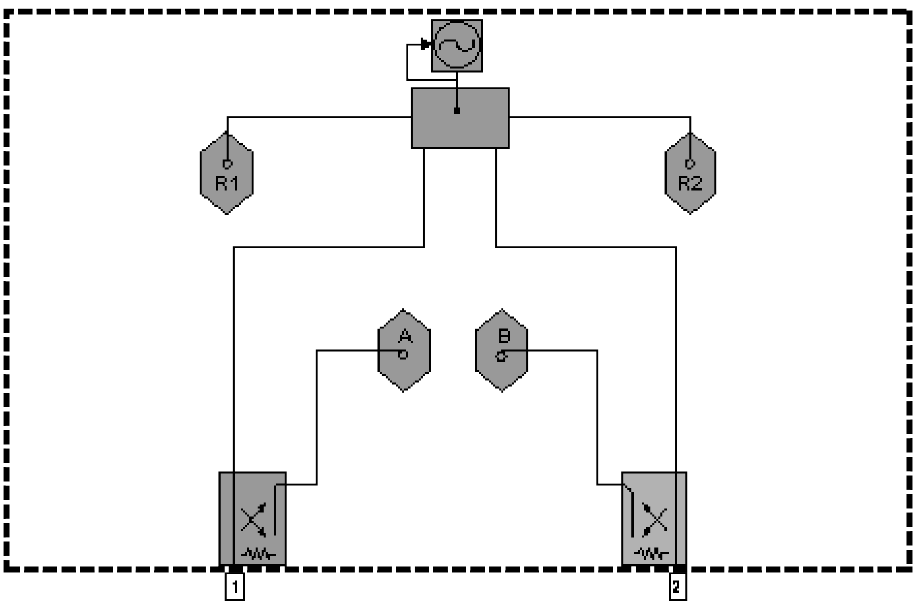
DCOM over LAN (program executed on external PC)				
32-bit floating point	1	1	2	121
Variant type	3	6	19	939

a Typical performance

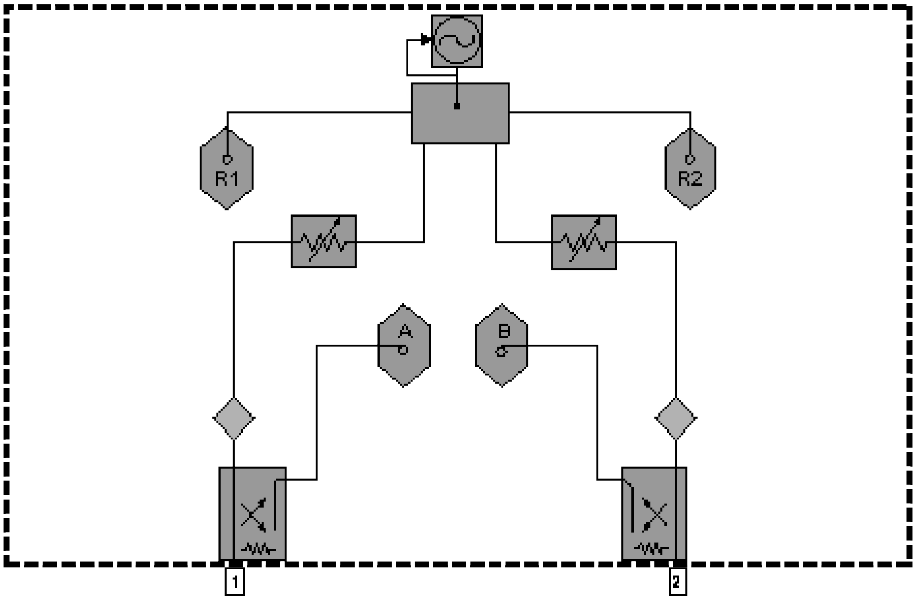
Note: Specifications for Recall & Sweep Speed are not provided for the E836xB analyzers.

Test Set Block Diagrams

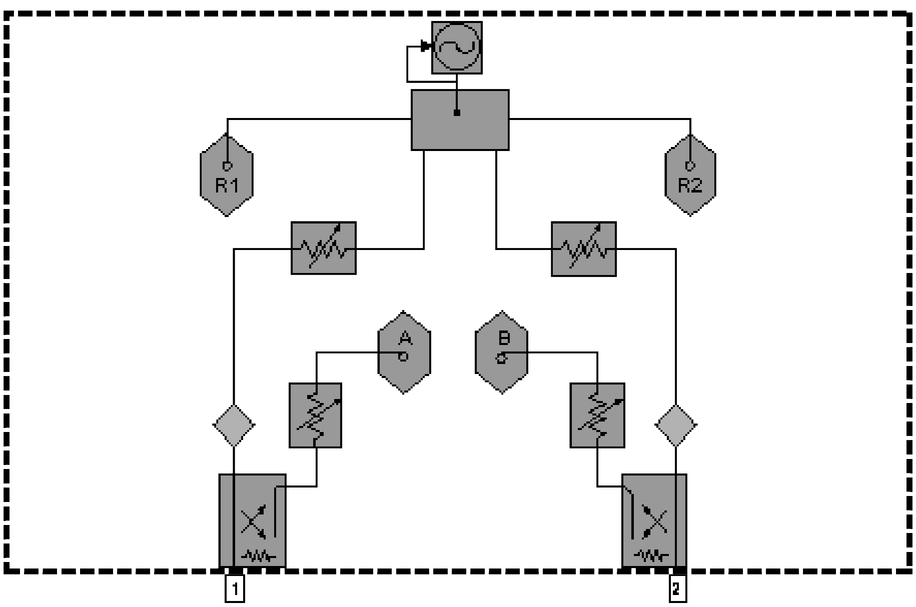
E836xB - Standard Configuration and Standard Power Range



E836xB - Option UNL Standard Configuration with Extended Power Range and Bias - Tees

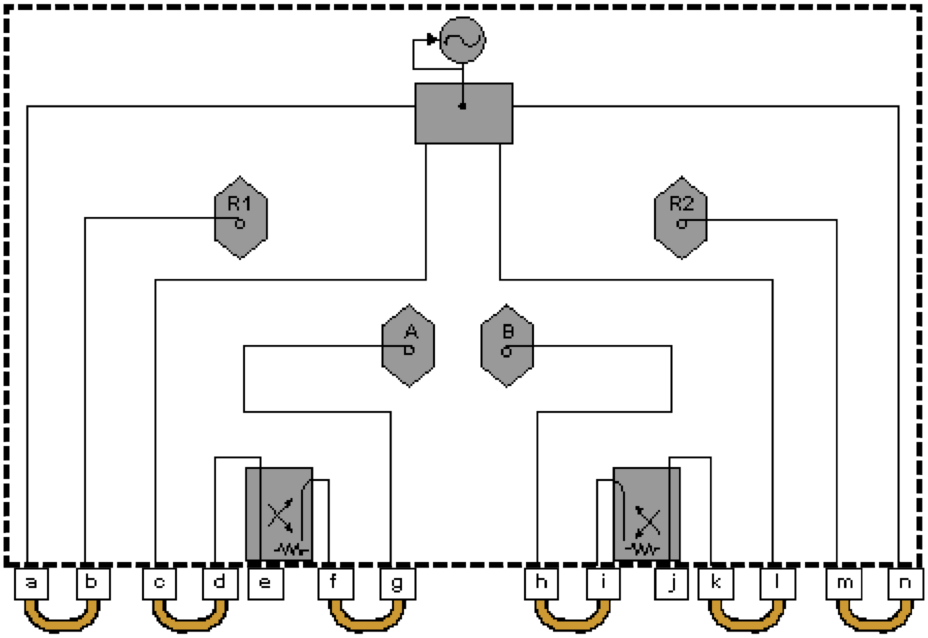


E836xB - Option UNL Standard Configuration with Extended Power Range and Bias - Tees, and Option 016, Receiver Attenuators



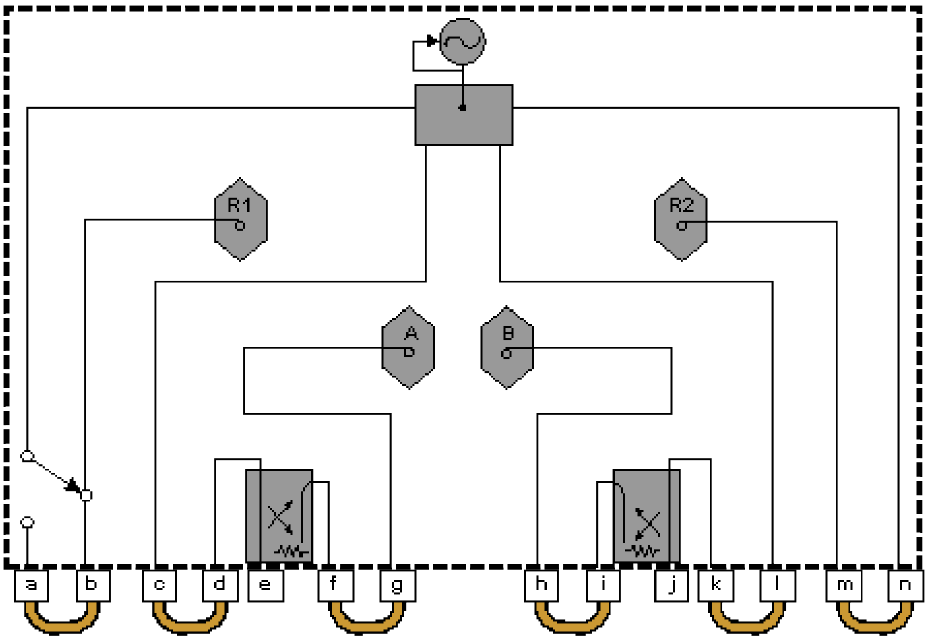
Test Set with Option 014 Block Diagrams

E836xB - Option 014 – Configurable Test Set and Standard Power Range



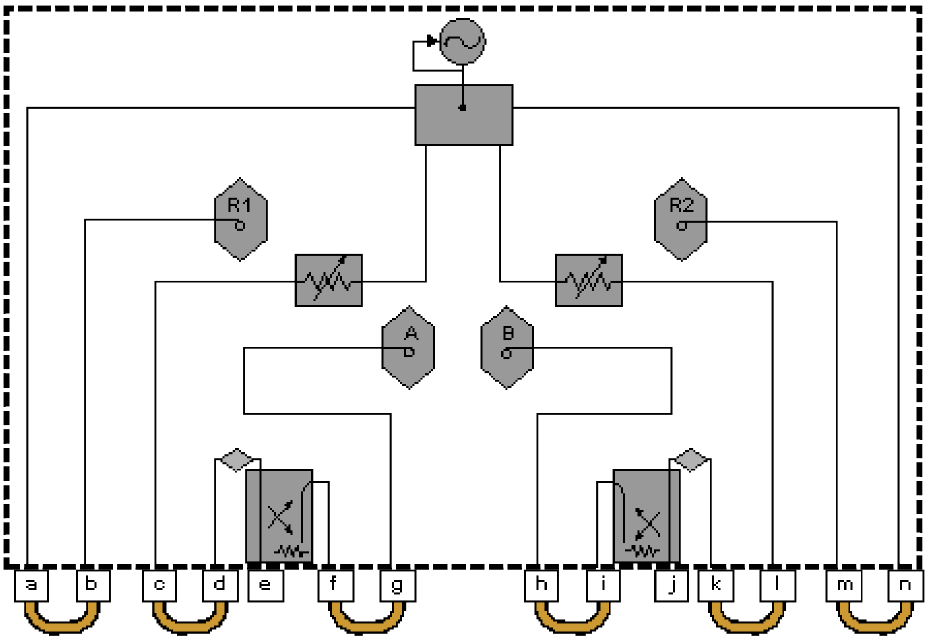
Item	Description	Item	Description
a	SOURCE OUT	h	RCVR B IN
b	RCVR R1 IN	i	CPLR ARM
c	SOURCE OUT	j	PORT 2
d	CPLR THRU	k	CPLR THRU
e	PORT 1	l	SOURCE OUT
f	CPLR ARM	m	RCVR R2 IN
g	RCVR A IN	n	SOURCE OUT

E836xB - Option 014 – Configurable Test Set and Standard Power Range, and Option 081 Reference Channel Transfer Switch



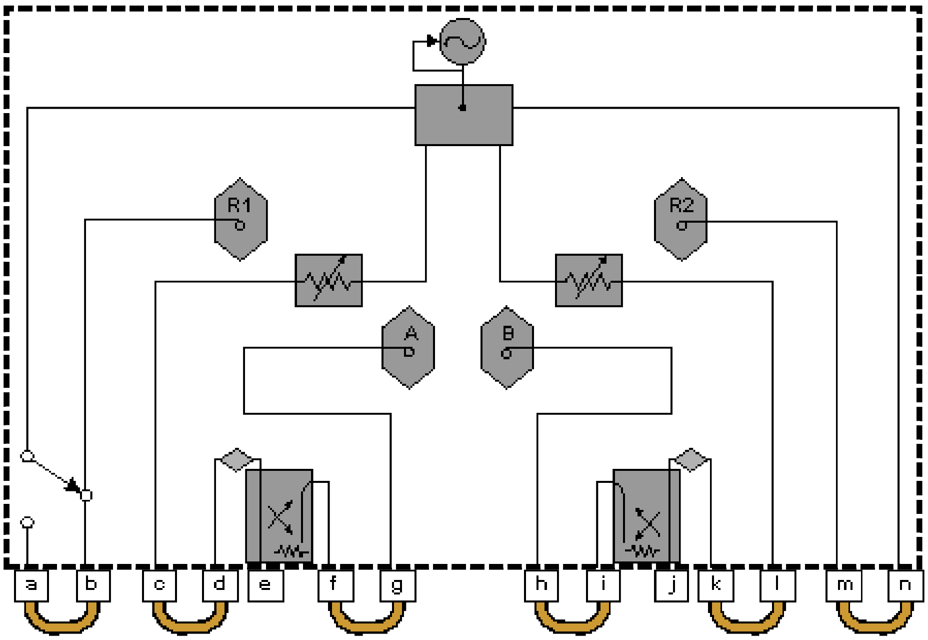
Item	Description	Item	Description
a	SOURCE OUT	**h**	RCVR B IN
b	RCVR R1 IN	**i**	CPLR ARM
c	SOURCE OUT	**j**	PORT 2
d	CPLR THRU	**k**	CPLR THRU
e	PORT 1	**l**	SOURCE OUT
f	CPLR ARM	**m**	RCVR R2 IN
g	RCVR A IN	**n**	SOURCE OUT

E836xB - Option 014 Configurable Test Set, and Option UNL Extended Power Range and Bias - Tees



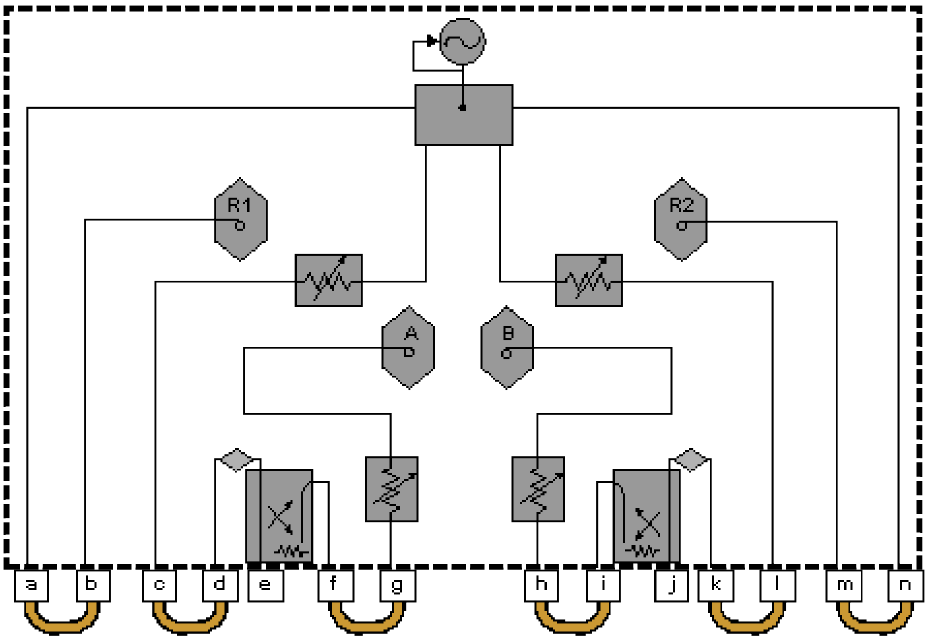
Item	Description	Item	Description
a	SOURCE OUT	h	RCVR B IN
b	RCVR R1 IN	i	CPLR ARM
c	SOURCE OUT	j	PORT 2
d	CPLR THRU	k	CPLR THRU
e	PORT 1	l	SOURCE OUT
f	CPLR ARM	m	RCVR R2 IN
g	RCVR A IN	n	SOURCE OUT

E836xB - Option 014 Configurable Test Set, and Option UNL Extended Power Range and Bias - Tees, and Option 081 Reference Channel Transfer Switch



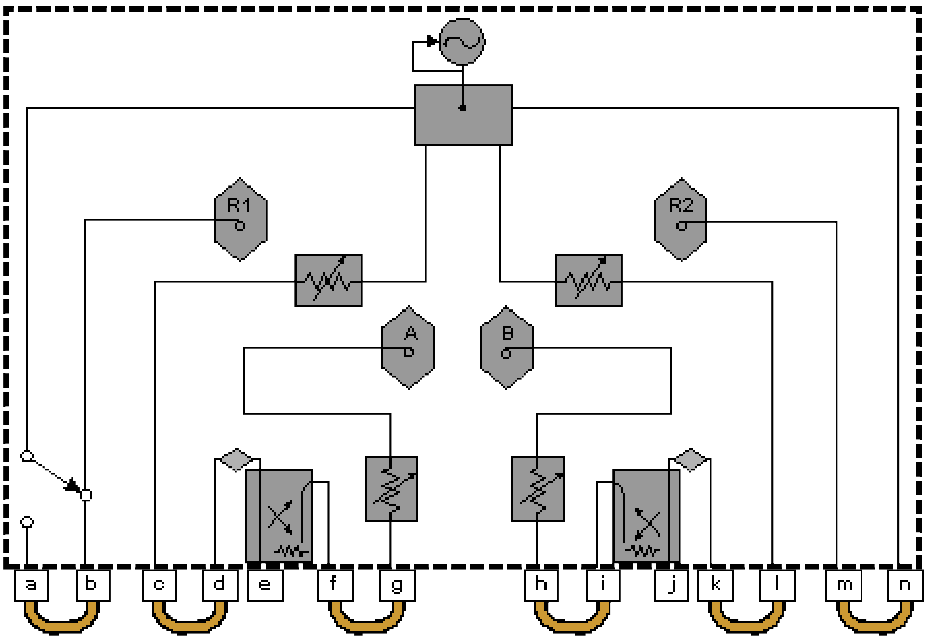
Item	Description	Item	Description
a	SOURCE OUT	h	RCVR B IN
b	RCVR R1 IN	i	CPLR ARM
c	SOURCE OUT	j	PORT 2
d	CPLR THRU	k	CPLR THRU
e	PORT 1	l	SOURCE OUT
f	CPLR ARM	m	RCVR R2 IN
g	RCVR A IN	n	SOURCE OUT

E836xB - Option 014 Configurable Test Set and Option UNL, Extended Power Range and Bias - Tees and Option 016 Receiver Attenuators



Item	Description	Item	Description
a	SOURCE OUT	h	RCVR B IN
b	RCVR R1 IN	i	CPLR ARM
c	SOURCE OUT	j	PORT 2
d	CPLR THRU	k	CPLR THRU
e	PORT 1	l	SOURCE OUT
f	CPLR ARM	m	RCVR R2 IN
g	RCVR A IN	n	SOURCE OUT

E836xB - Option 014 Configurable Test Set, and Option UNL Extended Power Range and Bias - Tees, and Option 016 Receiver Attenuators, and Option 081 Reference Channel Transfer Switch



Item	Description	Item	Description
a	SOURCE OUT	h	RCVR B IN
b	RCVR R1 IN	i	CPLR ARM
c	SOURCE OUT	j	PORT 2
d	CPLR THRU	k	CPLR THRU
e	PORT 1	l	SOURCE OUT
f	CPLR ARM	m	RCVR R2 IN
g	RCVR A IN	n	SOURCE OUT

Technical Specifications for the N5230A

Options 020/025, 120/125, 220/225, 420/425, or 520/525 (2-Port PNA)

(Rev. 2005-07-26)

This is a complete list of the N5230A Options 020, 025, 120, 125, 220, 225, 420, 425, 520, 525 network analyzer technical specifications.

- To optimize viewing of uncertainty curves, click the Maximize button.

- To view or print the .pdf version of the specifications, visit our web site at http://www.agilent.com/find/pna, and search for "N5230A Specifications"
- This N5230A document provides technical specifications for the 85056A 2.4 mm, 85052B 3.5 mm, and 85032B Type-N calibration kits and the N4691A, and N4693A ECal modules. Please download our free Uncertainty Calculator from http://www.agilent.com/find/na_calculator to generate the curves for your calibration kit and PNA setup.

- Definitions
- Corrected System Performance
  - System Dynamic Range
  - Extended Dynamic Range
  - 3.5mm Connectors
  - 2.4mm Connectors
  - Type-N Connectors
- Uncorrected System Performance
- Test Port Output
- Test Port Input
- Dynamic Accuracy
- Group Delay
- General Information
- Measurement Throughput Summary
- Front-panel Jumper Specs (Options 025, 125, 225, 425, 525)
- Option 020, or 120, or 220, or 420, or 520 (Standard Test Set and Standard Power Range) Analyzer Block Diagram
- Option 025, or 125, or 225, or 425, or 525 (Configurable Test Set and Extended Power Range) Analyzer Block Diagram

See Specs for other PNA models

Definitions

All specifications and characteristics apply over a 25 °C ±5 °C range (unless otherwise stated) and 90 minutes after the instrument has been turned on.

Specification (spec.): Warranted performance. Specifications include guardbands to account for the expected statistical performance distribution, measurement uncertainties, and changes in performance due to environmental conditions.

Characteristic (char.): A performance parameter that the product is expected to meet before it leaves the factory, but that is not verified in the field and is not covered by the product warranty. A characteristic includes the same guardbands as a specification.

Typical (typ.): Expected performance of an average unit which does not include guardbands. It is not covered by the product warranty.

Nominal (nom.): A general, descriptive term that does not imply a level of performance. It is not covered by the product warranty.

Calibration: The process of measuring known standards to characterize a network analyzer's systematic (repeatable) errors.

Corrected (residual): Indicates performance after error correction (calibration). It is determined by the quality of calibration standards and how well "known" they are, plus system repeatability, stability, and noise.

Uncorrected (raw): Indicates instrument performance without error correction. The uncorrected performance affects the stability of a calibration.

Standard: When referring to the analyzer, this includes no options unless noted otherwise.

Corrected System Performance

The specifications in this section apply for measurements made with the N5230A analyzer with the following conditions:

- 10 Hz IF bandwidth
- No averaging applied to data
- Isolation calibration with an averaging factor of 8

System Dynamic Range

Extended Dynamic Range

3.5mm Connectors

2.4mm Connectors

Type-N Connectors

Table 1. System Dynamic Range[1]

Description	Specification (dB) at Test Port				Typical (dB) at Test Port			
	Options 020, 120	**Option 220**	**Option 420**	**Option 520**	**Options 020, 120**	**Option 220**	**Option 420**	**Option 520**

Standard Configuration and Standard Power Range								
300 kHz to 1 MHz	103	--	--	--	--	--	--	--
1 MHz to 10 MHz	113	--	--	--	--	--	--	--
10 MHz to 45 MHz	122	--	--	--	--	103	89	89
45 MHz to 70 MHz[2]	122	101	90	90	--	--	--	--
70 MHz to 500 MHz[2]	122	105	90	90	--	--	--	--
500 MHz to 2 GHz	122	110	110	110	--	--	--	--
2 GHz to 6 GHz	122	110	110	110	--	--	--	--
6 GHz to 8 GHz	120	110	110	110	--	--	--	--
8 GHz to 9 GHz	120	110	100	100	--	--	--	--
9 GHz to 10.5 GHz	116	110	100	100	--	--	--	--
10.5 GHz to 12.5 GHz	111	110	100	100	--	--	--	--
12.5 GHz to 13.5 GHz	109	108	100	100	--	--	--	--
13.5 GHz to 20 GHz	--	108	100	100	--	--	--	--
20 GHz to 31.25 GHz	--	--	95	95	--	--	--	--
31.25 GHz to 40 GHz	--	--	90	90	--	--	--	--
40 GHz to 50 GHz	--	--	--	79	--	--	--	--

Table 1. System Dynamic Range[1] (Continued)

Description	Specification (dB) at Test Port				Typical (dB) at Test Port			
	Options 025, 125	**Option 225**	**Option 425**	**Option 525**	**Options 025, 125**	**Option 225**	**Option 425**	**Option 525**
Configurable Test Set and Extended Power Range								

300 KHz to 1 MHz	102	--	--	--	--	--	--	--
1 MHz to 10 MHz	112	--	--	--	--	--	--	--
10 MHz to 45 MHz	121	--	--	--	--	103	88	88
45 MHz to 70 MHz[2]	121	101	90	90	--	--	--	--
70 MHz to 500 MHz[2]	121	105	90	90	--	--	--	--
500 MHz to 2 GHz	121	110	110	110	--	--	--	--
2 GHz to 6 GHz	121	110	110	110	--	--	--	--
6 GHz to 8 GHz	120	110	110	110	--	--	--	--
8 GHz to 9 GHz	120	110	100	100	--	--	--	--
9 GHz to 10.5 GHz	116	110	100	100	--			
10.5 GHz to 12.5 GHz	111	110	100	100	--	--	--	--
12.5 GHz to 13.5 GHz	108	108	100	100	--	--	--	--
13.5 GHz to 20 GHz	--	108	100	100	--	--	--	--
20 GHz to 31.25 GHz	--	--	92	92	--	--	--	--
31.25 GHz to 40 GHz	--	--	87	87	--	--	--	--
40 GHz to 50 GHz	--	--	--	75	--	--	--	--

1 The system dynamic range is calculated as the difference between the noise floor and the specified source maximum output power. The effective dynamic range must take measurement uncertainties and interfering signals into account.
2 May be degraded by 10 dB at particular frequencies (multiples of 5 MHz) below 500 MHz due to spurious receiver residuals. Methods are available to regain the full dynamic range.

Receiver Dynamic Range technical specifications are not provided in this N5230A specs document.

Table 2. Extended Dynamic Range[1]

Description	Specification (dB) at Direct Receiver Access Input				Typical (dB) at Direct Receiver Access Input			
	Option 025, 125	Option 225	Option 425	Option 525	Option 025, 125	Option 225	Option 425	Option 525
Configurable Test Set and Extended Power Range								
300 kHz to 1 MHz	118	--	--	--	--	--	--	--
1 MHz to 10 MHz	128	--	--	--	--	--	--	--
10 MHz to 45 MHz	137	--	--	--	--	115	109	109
45 MHz to 70 MHz[2]	137	113	111	111	--	--	--	--
70 MHz to 500 MHz[2]	137	117	111	111	--	--	--	--
500 MHz to 2 GHz	137	122	122	122	--	--	--	--
2 GHz to 6 GHz	137							
6 GHz to 8 GHz	136	122	122	122	--	--	--	--
8 GHz to 9 GHz	136							
9 GHz to 10.5 GHz	132	122	112	112	--	--	--	--
10.5 GHz to 12.5 GHz	127	122	112	112	--	--	--	--
12.5 GHz to 13.5 GHz	124	120	112	112	--	--	--	--
13.5 GHz to 20 GHz	--	120	112	112	--	--	--	--
20 GHz to 31.25 GHz	--	--	103	103	--	--	--	--
31.25 GHz to 40 GHz	--	--	98	98	--	--	--	--
40 GHz to 50 GHz	--	--	--	83	--	--	--	--

1 The direct receiver access input extended dynamic range is calculated as the difference between the direct receiver access input noise floor and the source maximum output power. The effective dynamic range must take measurement uncertainties and interfering signals into account. This set-up should only be used when the receiver input will never exceed its compression or damage level. When the analyzer is in segment sweep mode, it can have predefined frequency segments which will output a higher power level when the extended dynamic range is required (i.e. devices with high insertion loss), and reduced power when receiver compression or damage may occur (i.e. devices with low insertion loss). The extended range is only available in one-path transmission measurements.
2 May be degraded by 10 dB at particular frequencies (multiples of 5 MHz) below 500 MHz due to spurious receiver residuals. Methods are available to regain the full dynamic range.

Corrected System Performance with 3.5mm Connectors (Tables 3 - 8)

Corrected System Performance with 2.4mm Connectors (Tables 9 - 12)

Corrected System Performance with Type-N Connectors (Tables 13 - 14)

Table 15. Uncorrected System Performance

Description	Specifications				Typical			
	Options **020, 025, 120, 125**	**Options** **220, 225**	**Options** **420, 425**	**Options** **520, 525**	**Options** **020, 025, 120, 125**	**Options** **220, 225**	**Options** **420, 425**	**Options** **520, 525**
Directivity								
300 kHz to 10 MHz	16 dB	--	--	--	--	23 dB	--	
10 MHz to 45 MHz	28 dB	--	--	--	--		20 dB	20 dB
45 MHz to 500 MHz	28 dB	24 dB	23 dB	23 dB	--	--	--	--
500 MHz to 1 GHz	28 dB	27 dB	23 dB	23 dB	--	--	--	--
1 GHz to 2 GHz	25 dB	27 dB	23 dB	23 dB	--	--	--	--
2 GHz to 3 GHz	25 dB	21 dB	21 dB	21 dB	--	--	--	--
3 GHz to 5 GHz	20 dB	21 dB	21 dB	21 dB	--	--	--	--
5 GHz to 8 GHz	17 dB	21 dB	21 dB	21 dB	--	--	--	--
8 GHz to 11.5 GHz	17 dB	16 dB	16 dB	16 dB	--	--	--	--
11.5 GHz to 13.5 GHz	15 dB	16 dB	16 dB	16 dB	--	--	--	--
13.5 GHz to 20 GHz	--	16 dB	16 dB	16 dB	--	--	--	--
20 GHz to 40 GHz	--	--	15 dB	15 dB	--	--	--	--
45 GHz to 50 GHz	--	--	--	13 dB	--	--	--	--
Source Match								

300 kHz to 10 MHz	18 dB	--	--	--	--	--	--	--
10 MHz to 45 MHz	25 dB	--	--	--	-	12 dB	11 dB	11 dB
45 MHz to 500 MHz	25 dB	20 dB	17 dB	17 dB	--	--	--	--
500 MHz to 2 GHz	21 dB	17 dB	17 dB	17 dB	--	--	--	--
2 GHz to 3 GHz	19 dB	12 dB	12 dB	12 dB	--	--	--	
3 GHz to 8 GHz	12 dB	12 dB	12 dB	12 dB	--	--	--	--
8 GHz to 9 GHz	12 dB	11 dB	11 dB	11 dB	--	--	--	
9 GHz to 12.5 GHz	10 dB	11 dB	11 dB	11 dB	--	--	--	--
12.5 GHz to 13.5 GHz	8 dB	10 dB	11 dB	11 dB	--	--	--	--
13.5 GHz to 20 GHz	--	10 dB	11 dB	11 dB	--	--	--	--
20 GHz to 40 GHz	--	--	7 dB	7 dB	--	--	--	--
40 GHz to 50 GHz	--	--	--	6 dB	--	--	--	--

Table 15. Uncorrected System Performance (Continued)

Description	Specifications				Typical			
	Options 020, 025, 120, 125	**Options** 220, 225	**Options** 420, 425	**Options** 520, 525	**Options** 020, 025, 120, 125	**Options** 220, 225	**Options** 420, 425	**Options** 520, 525
Load Match								
300 kHz to 10 MHz	17 dB	--	--	--	--	--	--	--

10 MHz to 45 MHz	22 dB	--	--	--	--	15 dB	13 dB	13 dB
45 MHz to 500 MHz	22 dB	22 dB	18 dB	18 dB	--	--	--	--
500 MHz to 2 GHz	17 dB	20 dB	18 dB	18 dB	--	--	--	--
2 GHz to 3 GHz	14 dB	12 dB	14 dB	14 dB	--	--	--	--
3 GHz to 8 GHz	10 dB	12 dB	14 dB	14 dB	--	--	--	--
8 GHz to 9 GHz	9 dB	10 dB	12 dB	12 dB	--	--	--	--
9 GHz to 12.5 GHz	9 dB	10 dB	12 dB	12 dB	--	--	--	--
12.5 GHz to 13.5 GHz	7 dB	9 dB	9 dB	9.5 dB	--	--	--	--
13.5 GHz to 20 GHz	--	9 dB	9 dB	9.5 dB	--	--	--	--
20 GHz to 40 GHz	--	--	8 dB	8.5 dB	--	--	--	--
40 GHz to 50 GHz	--	--	--	5 dB	--	--	--	--

Table 15. Uncorrected System Performance (Continued)

Description	**Specifications**				**Typical**			
	Options 020, 025, 120, 125	**Options 220, 225**	**Options 420, 425**	**Options 520, 525**	**Options 020, 025, 120, 125**	**Options 220, 225**	**Options 420, 425**	**Options 520, 525**
Crosstalk[1]								
300 kHz to 10 MHz	--	--	--	--	80 dB	--	--	--
10 MHz to 45 MHz	--	--	--	--	115 dB	88 dB	88 dB	88 dB

45 MHz to 500 MHz	--	--	--	--	122 dB	95 dB	94 dB	94 dB
500 MHz to 2 GHz	--	--	--	--	122 dB	96 dB	95 dB	95 dB
2 GHz to 8 GHz	--	--	--	--	122 dB	110 dB	108 dB	108 dB
8 GHz to 10.5 GHz	--	--	--	--	120 dB	116 dB	113 dB	113 dB
10.5 GHz to 12.5 GHz	--	--	--	--	115 dB	116 dB	113 dB	113 dB
12.5 GHz to 13.5 GHz	--	--	--	--	109 dB	115 dB	112 dB	112 dB
13.5 GHz to 20 GHz	--	--	--	--	--	115 dB	112 dB	112 dB
20 GHz to 40 GHz	--	--	--	--	--	--	97 dB	97 dB
40 GHz to 50 GHz	--	--	--	--	--	--	--	89 dB

1 Measurement conditions: normalized to a thru, measured with two shorts, 10 Hz IF bandwidth, averaging factor of 8, alternate mode, source power set to the specified maximum power output or the minimum receiver input power specified by the 0.1 dB compression power.

Table 16. Test Port Output[1]

Description	Specifications					Typicals			
	Options 020, 025	**Options 120, 125**	**Options 220, 225**	**Options 420, 425**	**Options 520, 525**	**Options 020, 025, 120, 125**	**Options 220, 225**	**Options 420, 425**	**Options 520, 525**
Frequency Range									
N5230A	300kHz to 6 GHz	300kHz to 13.5 GHz	10 MHz to 20 GHz	10 MHz to 40 GHz	10 MHz to 50 GHz	--			
Nominal Power									
Preset power; attenuator switch point 10 dB below nominal power									
	0 dBm	0 dBm	-5 dBm	-10 dBm	-15 dBm	--			
Frequency Resolution									

	1 Hz	--
CW Accuracy		
	+/-1 ppm	--
Frequency Stability		
	--	+/-0.05 ppm. -10° to 70° C +/-0.1 ppm/yr maximum

Table 16. Test Port Output[1] (Continued)

Description	Specifications					Typical		
	Options 020, 025, 120, 125	**Options 220, 225**	**Options 420, 425**	**Option 520**	**Option 525**	**Options 020, 025, 120, 125**	**Option 220, 225**	**Option 420, 425, 520, 525**
Power Level Accuracy								
Variation from nominal power in range 0								
300 kHz to 10 MHz	+/-1.0 dB	--	--	--	--	--	--	--
10 MHz to 45 MHz	+/-1.0 dB	--	--	--	--	--	+/-0.5 dB	+/-0.5 dB
45 MHz to 6 GHz	+/-1.0 dB	+/-1.0 dB	+/-1.0 dB	+/-1.0 dB	+/-1.0 dB	--	--	--
6 GHz to 8 GHz	+/-1.5 dB	+/-1.0 dB	+/-1.0 dB	+/-1.0 dB	+/-1.0 dB	--	--	--
8 GHz to 9 GHz	+/-1.5 dB	+/-1.0 dB	+/-1.5 dB	+/-1.5 dB	+/-1.5 dB	--	--	--
9 GHz to 10.5 GHz	+/-1.5 dB	+/-1.0 dB	+/-1.5 dB	+/-1.5 dB	+/-1.5 dB	--	--	--

10.5 GHz to 13.5 GHz	+/-2.0 dB	+/-1.0 dB	+/-1.5 dB	+/-1.5 dB	+/-1.5 dB	--	--	--
13.5 to 20 GHz	--	+/-1.0 dB	+/-1.5 dB	+/-1.5 dB	+/-1.5 dB	--	--	--
20 GHz to 40 GHz	--	--	+/-2.5 dB	+/-2.5 dB	+/-2.5 dB	--	--	--
40 GHz to 50 GHz	--	--	--	+/-3.5 dB	+/-3.5 dB	--	--	--

Table 16. Test Port Output[1] (Continued)

	Specifications							Typical
	Options 020, 120	**Options 025, 125**	**Options 220, 225**	**Option 420**	**Option 425**	**Option 520**	**Option 525**	**Options as Indicated**
Max Leveled Power								
300 kHz to 10 MHz	10 dBm	9 dBm	--	--	--	--	--	--
10 MHz to 45 MHz	10 dBm	9 dBm	--	--	--	--	--	--
45 MHz to 6 GHz	10 dBm	9 dBm	5 dBm	0 dBm	0 dBm	0 dBm	0 dBm	--
6 GHz to 9 GHz	8 dBm	8 dBm	5 dBm	0 dBm	0 dBm	0 dBm	0 dBm	--
9 GHz to 12.5 GHz	4 dBm	4 dBm	5 dBm	0 dBm	0 dBm	0 dBm	0 dBm	--
12.5 GHz to 13.5 GHz	2 dBm	1 dBm	3 dBm	0 dBm	0 dBm	0 dBm	0 dBm	--
13.5 GHz to 20 GHz	--	--	3 dBm	0 dBm	0 dBm	0 dBm	0 dBm	--
20 GHz to 40 GHz	--	--	--	-5 dBm	-8 dBm	-5 dBm	-8 dBm	--

40 GHz to 50 GHz	--	--	--	--	--	-11 dBm	-15 dBm	--
Power Level Linearity[2]								
Test reference is at the nominal power level								
300 kHz to 1 MHz	+/-4.5 dB	+/-4.5 dB	--	--	--	--	--	+/-2.0 dB (Opt 020, 025, 120, 125)
1 MHz to 10 MHz	+/-1.0 dB	+/-1.0 dB	--	--	--	--	--	--
10 MHz to 45 MHz	+/-2.0 dB	+/-2.0 dB	--	--	--	--	--	+/-0.35 dB (Opt 220 & 225) +/-0.40 dB (Opt 420, 425, 520, 525)
45 MHz to 1 GHz	+/-2.0 dB	+/-2.0 dB	+/-1.0 dB	+/-1.0 dB	+/-1.0 dB	+/-1.0 dB	+/-1.0 dB	--
1 GHz to 12.5 GHz	+/-1.5 dB	+/-1.5 dB	+/-1.0 dB	+/-1.0 dB	+/-1.0 dB	+/-1.0 dB	+/-1.0 dB	--
12.5 GHz to 13.5 GHz	+/-1.5 dB	+/-1.5 dB	+/-1.0 dB	+/-1.0 dB	+/-1.0 dB	+/-1.0 dB	+/-1.0 dB	--
13.5 GHz to 20 GHz	--	--	+/-1.0 dB	+/-1.0 dB	+/-1.0 dB	+/-1.0 dB	+/-1.0 dB	--
20 GHz to 40 GHz	--	--	--	+/-1.0 dB	+/-1.0 dB	+/-1.0 dB	+/-1.0 dB	--
40 GHz to 50 GHz	--	--	--	--	--	+/-1.0 dB	+/-1.0 dB	--
Power Sweep Range (ALC)[3]								
300 kHz to 10 MHz	37 dB	36 dB	--	--	--	--	--	--

10 MHz to 45 MHz	37 dB	36 dB	--	--	--	--	--	--
45 MHz to 6 GHz	37 dB	36 dB	25 dB	25 dB	25 dB	25 dB	25 dB	--
6 GHz to 9 GHz	35 dB	35 dB	25 dB	25 dB	25 dB	25 dB	25 dB	--
9 GHz to 12.5 GHz	31 dB	31 dB	25 dB	25 dB	25 dB	25 dB	25 dB	--
12.5 GHz to 13.5 GHz	29 dB	28 dB	23 dB	25 dB	25 dB	25 dB	25 dB	--
13.5 GHz to 20 GHz	--	--	23 dB	25 dB	25 dB	25 dB	25 dB	--
20 GHz to 40 GHz	--	--	--	20 dB	17 dB	20 dB	17 dB	--
40 GHz to 50 GHz	--	--	--	--	--	14 dB	10 dB	--
Power Resolution								
								0.01 dB

Table 16. Test Port Output[1] (Continued)

Description	Specifications	Typical					
		Option 020, 120	**Option 025, 125**	**Option 220**	**Option 225**	**Option 420, 520**	**Option 425, 525**
Power Range							
300 kHz to 10 MHz	--	-30 to +10 dBm	-90 to +9 dBm	--	--	--	--
10 MHz to 45 MHz	--	-30 to +10 dBm	-90 to +9 dBm	-27 to +14 dBm	-87 to +12 dBm	-27 to +9 dBm	-87 to +8 dBm
45 MHz to 6 GHz	--	-30 to +10 dBm	-90 to +9 dBm	-27 to +14 dBm	-87 to +12 dBm	-27 to +8 dBm	-87 to +8 dBm

6 GHz to 9 GHz	--	-30 to +8 dBm	-90 to +8 dBm	-27 to +14 dBm	-87 to +12 dBm	-27 to +8 dBm	-87 to +8 dBm
9 GHz to 12.5 GHz	--	-30 to +4 dBm	-90 to +4 dBm	-27 to +14 dBm	-87 to +12 dBm	-27 to +8 dBm	-87 to +8 dBm
12.5 GHz to 13.5 GHz	--	-30 to +2 dBm	-90 to +1 dBm	-27 to +10 dBm	-87 to +7 dBm	-27 to +5 dBm	-87 to +4 dBm
13.5 GHz to 20 GHz	--	--	--	-27 to +10 dBm	-87 to +7 dBm	-27 to +5 dBm	-87 to +4 dBm
20 GHz to 40 GHz	--	--	--	--	--	-27 to +1 dBm	-87 to -2 dBm
40 GHz to 50 GHz	--	--	--	--	--	-27 to -5 dBm	-87 to -9 dBm
Power Settings							
Minimum Power Setting	--	-33 dBm	-93 dBm	-30 dBm	-90 dBm	-30 dBm	-90 dBm
Maximum Power Setting	--	+20 dBm					

Table 16. Test Port Output[1] (Continued)

Description	Specifications	Typical					
	--	**Options 020, 025, 120, 125**			**Options 220, 225, 420, 425, 520, 525**		
Phase Noise (Nominal power at test port)							
	--	**10 kHz Offset**	**100 kHz Offset**	**1 MHz Offset**	**10 kHz Offset**	**100 kHz Offset**	**1 MHz Offset**
300 kHz to 10 MHz	--	-86 dBc/Hz	-86 dBc/Hz	-95 dBc/Hz	-77 dBc/Hz	-77 dBc/Hz	-40 dBc/Hz
10 MHz to 1.5 GHz	--	-86 dBc/Hz	-91 dBc/Hz	-95 dBc/Hz	-77 dBc/Hz	-77 dBc/Hz	-89 dBc/Hz

1.5 GHz to 3.125 GHz	--	-83 dBc/Hz	-91 dBc/Hz	-95 dBc/Hz	-83 dBc/Hz	-91 dBc/Hz	-95 dBc/Hz
3.125 GHz to 6.25 GHz	--	-77 dBc/Hz	-85 dBc/Hz	-89 dBc/Hz	-77 dBc/Hz	-85 dBc/Hz	-89 dBc/Hz
6.25 GHz to 12.5 GHz	--	-71 dBc/Hz	-79 dBc/Hz	-83 dBc/Hz	-71 dBc/Hz	-79 dBc/Hz	-83 dBc/Hz
12.5 GHz to 13.5 GHz	--	-65 dBc/Hz	-73 dBc/Hz	-77 dBc/Hz	-65 dBc/Hz	-73 dBc/Hz	-77 dBc/Hz
13.5 GHz to 20 GHz	--	--	--	--	-65 dBc/Hz	-73 dBc/Hz	-77 dBc/Hz
20 GHz to 40 GHz	--	--	--	--	-59 dBc/Hz	-67 dBc/Hz	-71 dBc/Hz
40 GHz to 50 GHz	--	--	--	--	-59 dBc/Hz	-67 dBc/Hz	-71 dBc/Hz

Table 16. Test Port Output[1] (Continued)

Description	Specifications	Typicals
	--	**Options 020, 025, 120, 125, 220, 225, 420, 520, 425, 525**
Non-Harmonic Spurious (at Nominal Output Power)		
300 kHz to 10 MHz	--	-50 dBc for offset frequency > 1 kHz
10 MHz to 13.5 GHz	--	
13.5 GHz to 20 GHz	--	
20 GHz to 40 GHz	--	-30 dBc for offset frequency > 1 kHz
40 GHz to 50 GHz	--	

Table 16. Test Port Output[1] (Continued)

Description	Specifications	Typical			
	--	**Option 020, 025, 120, 125**	**Option 220, 225**	**Option 420, 520**	**Option 425, 525**
Harmonics (2nd or 3rd) at Maximum Output Power					
300 kHz to 10 MHz	--	-17 dBc	--	--	--
10 MHz to 500 MHz	--	-17 dBc	-22 dBc	-15 dBc	-15 dBc
500 MHz to 1 GHz	--	-17 dBc	-22 dBc	-15 dBc	-15 dBc
1 GHz to 13.5 GHz	--	-20 dBc	22 dBc	-20 dBc	-20 dBc
13.5 GHz to 20 GHz	--	--	-22 dBc	-20 dBc	-20 dBc
20 GHz to 40 GHz	--	--	--	-22 dBc	-22 dBc
40 GHz to 50 GHz	--	--	--	--	-22 dBc

1Performance specified on Port 1 only. Port 2 performance is a characteristic.

2Power level linearity specified on Port 1 only. Port 2 performance is Typical. Test reference is at the nominal power level.

3ALC range starts at maximum leveled power and decreases in power level by the dB amount specified here.

Table 17. Test Port Input

Description	Specification				Typical			
	Options 020, 025, 120, 125	**Options 220, 225**	**Options 420, 425**	**Options 520, 525**	**Options 020, 025, 120, 125**	**Options 220, 225**	**Options 420, 425**	**Options 520, 525**
Test Port Noise Floor[1]								
10 Hz IF Bandwidth								
300 kHz to 1 MHz	<-93 dBm	--	--	--	<-103 dBm	--	--	--
1 MHz to 10 MHz	<-103 dBm	--	--	--	<-110 dBm	--	--	--

10 MHz to 45 MHz	<-112 dBm	--	--	--	<-116 dBm	<-89 dBm	<-80 dBm	<-80 dBm
45 MHz to 70 MHz	<-112 dBm	<-96 dBm	<-90 dBm	<-90 dBm	<-116 dBm	--	--	--
70 MHz to 500 MHz	<-112 dBm	<-100 dBm	<-90 dBm	<-90 dBm	<-116 dBm	--	--	--
500 MHz to 2 GHz	<-112 dBm	<-105 dBm	<-110 dBm	<-110 dBm	<-120 dBm	--	--	--
2 GHz to 4 GHz	<-112 dBm	<-105 dBm	<-110 dBm	<-110 dBm	<-120 dBm	--	--	--
4 GHz to 8 GHz	<-112 dBm	<-105 dBm	<-110 dBm	<-110 dBm	<-119 dBm	--	--	--
8 GHz to 10.5 GHz	<-112 dBm	<-105 dBm	<-100 dBm	<-100 dBm	<-119 dBm	--	--	--
10.5 GHz to 13.5 GHz	<-107 dBm	<-105 dBm	<-100 dBm	<-100 dBm	<-114 dBm	--	--	--
10.5 GHz to 13.5 GHz	--	<-105 dBm	<-100 dBm	<-100 dBm	--	--	--	--
13.5 GHz to 20 GHz	--	<-105 dBm	<-100 dBm	<-100 dBm	--	--	--	--
20 GHz to 31.25 GHz	--	--	<-100 dBm	<-100 dBm	--	--	--	--
31.25 GHz to 40 GHz	--	--	<-95 dBm	<-95 dBm	--	--	--	--
40 GHz to 50 GHz	--	--	--	<-90 dBm	--	--	--	--
1 KHz IF Bandwidth								
3000 kHz to 1 MHz	<-73 dBm	--	--	--	<-83 dBm	--	--	--
1 MHz to 10 MHz	<-83 dBm	--	--	--	<-90 dBm	--	--	--
10 MHz to 45 MHz	<-92 dBm	--	--	--	<-96 dBm	<-69 dBm	<-60 dBm	<-60 dBm
45 MHz to 70 MHz	<-92 dBm	<-76 dBm	<-70 dBm	<-70 dBm	<-96 dBm	--	--	--

70 MHz to 500 MHz	<-92 dBm	<-80 dBm	<-70 dBm	<-70 dBm	<-96 dBm	--	--	--
500 MHz to 2 GHz	<-92 dBm	<-85 dBm	<-90 dBm	<-90 dBm	<-100 dBm	--	--	--
2 GHz to 4 GHz	<-92 dBm	<-85 dBm	<-90 dBm	<-90 dBm	<-100 dBm	--	--	--
4 GHz to 8 GHz	<-92 dBm	<-85 dBm	<-90 dBm	<-90 dBm	<-99 dBm	--	--	--
8 GHz to 10.5 GHz	<-92 dBm	<-85 dBm	<-80 dBm	<-80 dBm	<-99 dBm	--	--	--
10.5 GHz to 13.5 GHz	<-87 dBm	<-85 dBm	<-80 dBm	<-80 dBm	<-94 dBm	--	--	--
13.5 GHz to 20 GHz	--	<-85 dBm	<-80 dBm	<-80 dBm	--	--	--	--
20 GHz to 31.25 GHz	--	--	<-80 dBm	<-80 dBm	--	--	--	--
31.25 GHz to 40 GHz	--	--	<-75 dBm	<-75 dBm	--	--	--	--
40 GHz to 50 GHz	--	--	--	<-70 dBm	--	--	--	--

Table 17. Test Port Input (Continued)

Description	**Specification**				**Typical**			
	Options 025, 125	**Options 225**	**Options 425**	**Options 525**	**Options 025, 125**	**Options 225**	**Options 425**	**Options 525**
Direct Receiver Access Input Noise Floor[1] (Options 025, 125, 225, 425, and 525 only)								
10 Hz IF Bandwidth								
300 kHz to 1 MHz	<-109 dBm	--	--	--	--	--	--	--
1 MHz to 10 MHz	<-119 dBm	--	--	--	--	--	--	--
10 MHz to 45 MHz	<-128 dBm	--	--	--	--	<-120 dBm	<-126 dBm	<-126 dBm

45 MHz to 70 MHz	<-128 dBm	<-108 dBm	<-111 dBm	<-111 dBm	--	--	--	--
70 MHz to 500 MHz	<-128 dBm	<-112 dBm	<-111 dBm	<-111 dBm	--	--	--	--
500 MHz to 2 GHz	<-128 dBm	<-117 dBm	<-122 dBm	<-122 dBm	--	--	--	--
2 GHz to 8 GHz	<-128 dBm	<-117 dBm	<-122 dBm	<-122 dBm	--	--	--	--
8 GHz to 10.5 GHz	<-128 dBm	<-117 dBm	<-112 dBm	<-112 dBm	--	--	--	--
10.5 GHz to 13.5 GHz	<-128 dBm	<-117 dBm	<-112 dBm	<-112 dBm	--	--	--	--
13.5 GHz to 20 GHz	<-123 dBm	<-117 dBm	<-112 dBm	<-112 dBm	--	--	--	--
20 GHz to 31.25 GHz	--	--	<-111 dBm	<-111 dBm	--	--	--	--
31.25 GHz to 40 GHz	--	--	<-106 dBm	<-106 dBm	--	--	--	--
40 GHz to 50 GHz	--	--	--	<-98 dBm	--	--	--	--
1 KHz IF Bandwidth								
300 kHz to 1 MHz	<-89 dBm	--	--	--	--	--	--	--
1 MHz to 10 MHz	<-99 dBm	--	--	--	--	--	--	--
10 MHz to 45 MHz	<-108 dBm	--	--	--	--	<-100 dBm	<-106 dBm	<-106 dBm
45 MHz to 70 MHz	<-108 dBm	<-88 dBm	<-91 dBm	<-91 dBm	--	--	--	--
70 MHz to 500 MHz	<-108 dBm	<-92 dBm	<-91 dBm	<-91 dBm	--	--	--	--
500 MHz to 2 GHz	<-108 dBm	<-97 dBm	<-102 dBm	<-102 dBm	--	--	--	--
2 GHz to 8 GHz	<-108 dBm	<-97 dBm	<-102 dBm	<-102 dBm	--	--	--	--

8 GHz to 10.5 GHz	<-108 dBm	<-97 dBm	<-92 dBm	<-92 dBm	--	--	--	--
10.5 GHz to 13.5 GHz	<-108 dBm	<-97 dBm	<-92 dBm	<-92 dBm	--			
13.5 GHz to 20 GHz	<-103 dBm	<-97 dBm	<-92 dBm	<-92 dBm	--	--	--	--
20 GHz to 31.25 GHz	--	--	<-91 dBm	<-91 dBm	--	--	--	--
31.25 GHz to 40 GHz	--	--	<-86 dBm	<-86 dBm	--	--	--	--
40 GHz to 50 GHz	--	--	--	<-78 dBm	--	--	--	--

Table 17. Test Port Input (Continued)

Description	Specification						Typical		
	Options 220, 225		Options 420, 520		Options 425, 525		--		
Compression Level									
	Power	**Com-pression**	**Power**	**Com-pression**	**Power**	**Com-pression**	--	--	--
10 MHz to 45 MHz[2]	--	--	--	--	--	--	--	--	--
45 MHz to 500 MHz	+5 dBm	0.10 dB	+5 dBm	0.40 dB	+5 dBm	0.40 dB	--	--	--
500 MHz to 2 GHz	+5 dBm	0.15 dB	+5 dBm	0.77 dB	+5 dBm	0.67 dB	--	--	--
2 GHz to 8 GHz	+5 dBm	0.21 dB	+5 dBm	0.75 dB	+5 dBm	0.55 dB	--	--	--
8 GHz to 12.5 GHz	+5 dBm	0.21 dB	+5 dBm	0.56 dB	+5 dBm	0.51 dB	--	--	--
12.5 GHz to 20 GHz	+3 dBm	0.20 dB	+5 dBm	0.79 dB	+5 dBm	0.69 dB	--	--	--

20 GHz to 31.25 GHz	--	--	0 dBm	0.60 dB	0 dBm	0.50 dB	--	--	--
31.25 GHz to 40 GHz	--	--	-3 dBm	0.55 dB	-3 dBm	0.60 dB	--	--	--
40 GHz to 50 GHz	--	--	-3 dBm	0.66 dB	-3 dBm	0.71 dB	--	--	--

Table 17. Test Port Input (Continued)

Description	Specification		Typical		
	Options 020, 120, 025, 125		**--**		
Compression Level (continued)					
	Power	**Com-pression**	--	--	--
300 kHz to 10 MHz	+8 dBm	1.0 dB (0.1 dB at +5 dBm typ.)	--	--	--
10 MHz to 50 MHz	+8 dBm	0.35 dB	--	--	--
50 MHz to 1GHz	+8 dBm	0.35 dB	--	--	--
1 GHz to 6 GHz	+8 dBm	0.25 dB	--	--	--
6 GHz to 8 GHz	+8 dBm	0.25 dB	--	--	--
8 GHz to 12.5 GHz	+8 dBm	0.30 dB	--	--	--
12.5 GHz to 13.5 GHz	+8 dBm	0.40 dB	--	--	--

Table 17. Test Port Input (Continued)

Description	Specification		Typical	
	--		**Options 020, 120, 025, 125**	
Test Port Compression - 0.1 dB				
	--	--	**Power**	**Compression**
300 kHz to 10 MHz	--	--	+5 dBm	0.1 dB
10 MHz to 1 GHz	--	--	+9 dBm	0.1 dB
1 GHz to 12.5 GHz	--	--	+10 dBm	0.1 dB
12.5 GHz to 13.5 GHz	--	--	+9 dBm	0.1 dB

Table 17. Test Port Input (Continued)

Description	Specification			Typical		
	Options 220, 225	**Options 420, 520**	**Options 425, 525**	**Options 220, 225**	**Options 420, 520**	**Options 425, 525**
Test Port Compression - 0.1 dB (continued)						
300 kHz to 10 MHz	--	--	--	--	--	--
10 MHz to 45 MHz[2]	--	--	--	negligible	negligible	negligible
45 MHz to 500 MHz	--	--	--	+10 dBm	0.0 dBm	+1.0 dBm
500 MHz to 2 GHz	--	--	--	+9 dBm	0.0 dBm	+1.0 dBm
2 GHz to 12.5 GHz	--	--	--	+6 dBm	0.0 dBm	+1.5 dBm

12.5 GHz to 13.5 GHz	--	--	--	+6 dBm	-1.0 dBm	0.0 dBm
13.5 GHz to 20 GHz	--	--	--	+6 dBm	-1.0 dBm	0.0 dBm
20 GHz to 31.25 GHz	--	--	--	--	-5.5 dBm	-3.0 dBm
31.25 GHz to 40 GHz	--	--	--	--	-8.5 dBm	-7.5 dBm
40 GHz to 50 GHz	--	--	--	--	-11.5 dBm	-10.0 dBm
Trace Noise Magnitude[3] 1 kHz IF bandwidth, ratioed measurement, nominal power at test port.						
300 kHz to 10 MHz	--	--	--	--	--	--
10 MHz to 45 MHz	--	--	--	0.004 dB rms	0.015 dB rms	0.015 dB rms
45 MHz to 500 MHz	0.004 dB rms	0.010 dB rms	0.010 dB rms	--	--	--
500 MHz to 2 GHz	0.004 dB rms	0.006 dB rms	0.006 dB rms	--	--	--
2 GHz to 10.5 GHz	0.004 dB rms	0.006 dB rms	0.006 dB rms	--	--	--
10.5 GHz to 13.5 GHz	0.006 dB rms	0.010 dB rms	0.010 dB rms	--	--	--
13.5 GHz to 20 GHz	0.006 dB rms	0.010 dB rms	0.010 dB rms	--	--	--
20 GHz to 31.25 GHz	--	0.010 dB rms	0.010 dB rms	--	--	--
31.25 GHz to 40 GHz	--	0.020 dB rms	0.020 dB rms	--	--	--
40 GHz to 50 GHz	--	--	0.020 dB rms	--	--	--

Table 17. Test Port Input (Continued)

Description	Specification	Typical
	Options 020, 120, 025, 125	**Options 020, 120, 025, 125**
Trace Noise Magnitude[3] (continued)		
100 kHz IF bandwidth, ratioed measurement, nominal power at test port		
300 kHz to 10 MHz	12 mdB	--
10 MHz to 6 GHz	4 mdB	--
6 GHz to 10.5 GHz	4 mdB	--
10.5 GHz to 13.5 GHz	8 mdB	--
600 kHz IF bandwidth, ratioed measurement, nominal power at test port		
300 kHz to 10 MHz	--	20 mdB
10 MHz to 6 GHz	--	8 mdB
6 GHz to 10.5 GHz	--	8 mdB
10.5 GHz to 13.5 GHz	--	10 mdB

Table 17. Test Port Input (Continued)

Description	Specification			Typical		
	Options 220, 225	**Options 420, 520**	**Options 425, 525**	**Options 220, 225**	**Options 420, 520**	**Options 425, 525**
Trace Noise Phase[3] 1 kHz IF bandwidth, ratioed measurement, nominal power at test port.						
300 kHz to 10 MHz						
10 MHz to 45 MHz	--	--	--	0.025° rms	0.100° rms	0.100° rms

45 MHz to 500 MHz	0.060° rms	0.100° rms	0.100° rms	--	--	--
500 MHz to 2 GHz	0.060° rms	0.060° rms	0.060° rms	--	--	--
2 GHz to 10.5 GHz	0.060° rms	0.060° rms	0.060° rms	--	--	--
10.5 GHz to 13.5 GHz	0.060° rms	0.100° rms	0.100° rms			
13.5 GHz to 20 GHz	0.060° rms	0.100° rms	0.100° rms	--	--	--
20 GHz to 31.25 GHz	--	0.100° rms	0.100° rms	--	--	--
31.25 GHz to 40 GHz	--	0.200° rms	0.200° rms	--	--	--
40 GHz to 50 GHz	--	--	0.200° rms	--	--	--

Table 17. Test Port Input (Continued)

Description	Specification	Technical
	Options 020, 120, 025, 125	**Options 020, 120, 025, 125**
Trace Noise Phase[3] (continued)		
100 kHz IF bandwidth, ratioed measurement, nominal power at test port		
300 kHz to 10 MHz	80 mdeg	--
10 MHz to 6 GHz	30 mdeg	--
6 GHz to 10.5 GHz	30 mdeg	--
10.5 GHz to 13.5 GHz	60 mdeg	--
600 kHz IF bandwidth, ratioed measurement, nominal power at test port		
300 kHz to 10 MHz	--	100 mdeg

10 MHz to 6 GHz	--	60 mdeg
6 GHz to 10.5 GHz	--	60 mdeg
10.5 GHz to 13.5 GHz	--	80 mdeg

Table 17. Test Port Input (Continued)

Description	Specification	Typical
	Options 020, 025, 120, 125, 220, 225, 420, 425, 520, 525	**--**
Reference Level Magnitude		
Range	+/-200 dB	
Resolution	0.001dB	
Reference Level Phase		
Range	+/-500°	
Resolution	0.01°	

Table 17. Test Port Input (Continued)

Description	Specification				Typical		
	--	--	--	--	**Option 020, 025, 120, 125**	**Option 220, 225**	**Options 420, 425, 520, 525**
Stability Magnitude[4]							
300 kHz to 10 MHz	--	--	--	--	+/-0.015 dB/°C	--	--
10 MHz to 45 MHz	--	--	--	--	+/-0.010 dB/°C	+/-0.015 dB/°C	+/-0.015 dB/°C
45 MHz to 500 MHz	--	--	--	--	+/-0.010 dB/°C	+/-0.010 dB/°C	+/-0.010 dB/°C

500 MHz to 2 GHz	--	--	--	--	+/-0.010 dB/°C	+/-0.010 dB/°C	+/-0.010 dB/°C
2 GHz to 4 GHz	--	--	--	--	+/-0.015 dB/°C	+/-0.020 dB/°C	+/-0.010 dB/°C
4 GHz to 8 GHz	--	--	--	--	+/-0.020 dB/°C	+/-0.020 dB/°C	+/-0.010 dB/°C
4 GHz to 13.5 GHz	--	--	--	--	+/-0.020 dB/°C	+/-0.030 dB/°C	+/-0.015 dB/°C
13.5 GHz to 20 GHz	--	--	--	--	--	+/-0.030 dB/°C	+/-0.015 dB/°C
20 GHz to 40 GHz	--	--	--	--	--	--	+/-0.040 dB/°C
40 GHz to 50 GHz	--	--	--	--	--	--	+/-0.060 dB/°C
Stability Phase[4]							
300 kHz to 10 MHz	--	--	--	--	+/-0.30°/°C	--	--
10 MHz to 45 MHz	--	--	--	--	+/-0.025°/°C	+/-0.25°/°C	+/-0.25°/°C
45 MHz to 500 MHz	--	--	--	--	+/-0.035°/°C	+/-0.20°/°C	+/-0.22°/°C
500 MHz to 2 GHz	--	--	--	--	+/-0.050°/°C	+/-0.15°/°C	+/-0.22°/°C
2 GHz to 4 GHz	--	--	--	--	+/-0.10°/°C	+/-0.15°/°C	+/-0.10°/°C
4 GHz to 8 GHz	--	--	--	--	+/-0.15°/°C	+/-0.15°/°C	+/-0.10°/°C
8 GHz to 13.5 GHz	--	--	--	--	+/-0.30°/°C	+/-0.45°/°C	+/-0.15°/°C
13.5 GHz to 20 GHz	--	--	--	--	--	+/-0.45°/°C	+/-0.15°/°C
20 GHz to 40 GHz	--	--	--	--	--	--	+/-0.40°/°C
40 GHz to 50 GHz	--	--	--	--	--	--	+/-0.40°/°C

Table 17. Test Port Input (Continued)

Description	Specifications	Typical			
	--	**Options 020, 120**	**Option 220, 420, 520**	**Options 025, 125**	**Options 225, 425 , 525**
Damage Input Level					
Test Port 1 and 2	--	+27 dBm or +/- 16 VDC	+30 dBm or +/- 40 VDC	+27 dBm or +/- 16 VDC	+27 dBm or +/- 16 VDC
R1, R2 in	--	--	--	+15 dBm or +/- 16 VDC	+15 dBm or +/-7 VDC
A, B in	--	--	--	+15 dBm or +/- 16 VDC	+15 dBm or +/-7 VDC
Coupler Thru	--	--	--	+27 dBm or +/- 16 VDC	+30 dBm or +/-40 VDC
Coupler Arm	--	--	--	+15 dBm or +/- 0 VDC	+30 dBm or +/-7 VDC

1Total average (rms) noise power calculated as the mean value of a linear magnitude trace expressed in dBm.

2Coupler roll-off will reduce compression to a negligible level below 45 MHz.

31 kHz IF BW, ratioed measurement, nominal power at the test port.

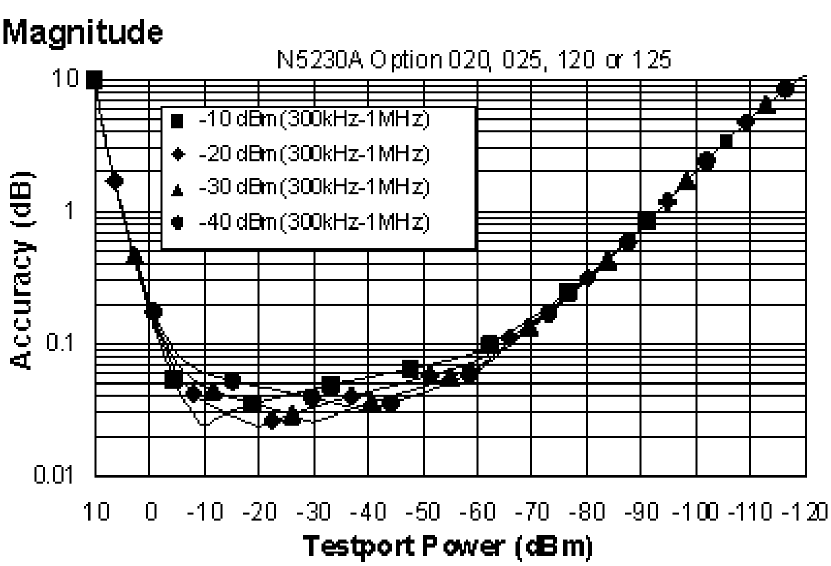
4Stability is defined as a ratio measurement made at the test port.

Table 18. Dynamic Accuracy (Specification[a])

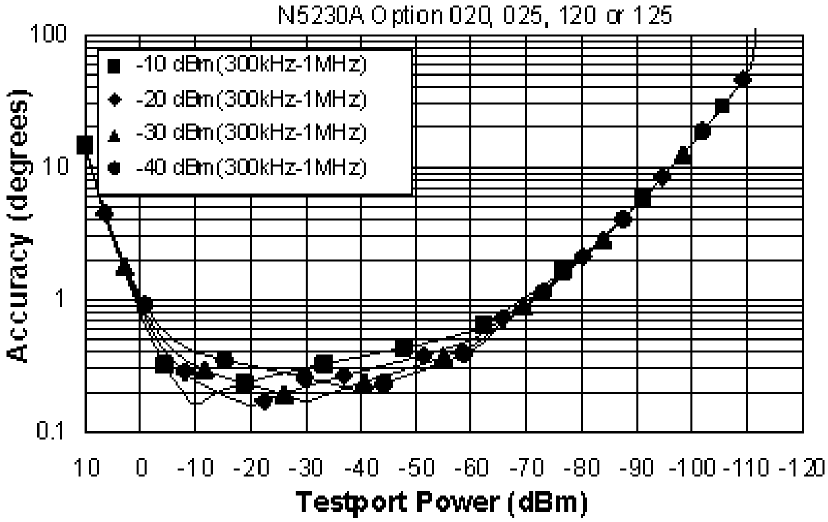
Accuracy of the test port input power reading relative to the reference input power level.

Options 020, 025, 120, 125

Dynamic Accuracy, 300 kHz - 1 MHz, Option 020, 025, 120, or 125

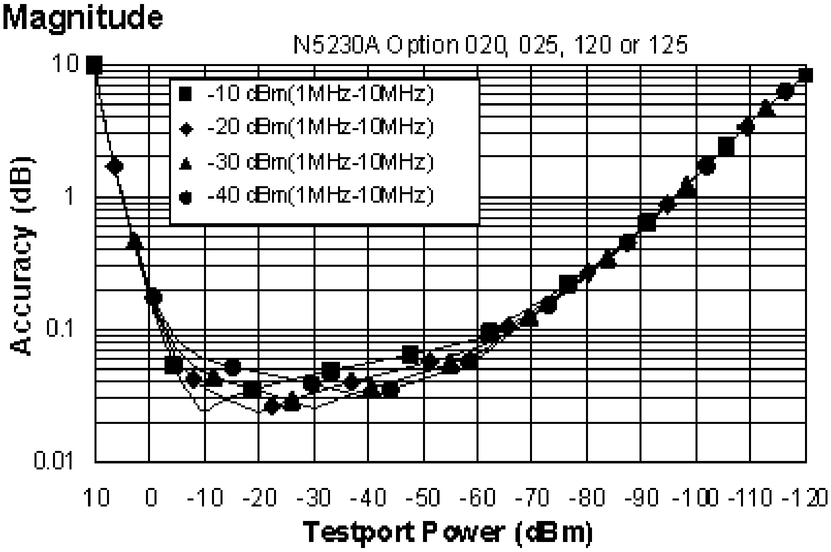


Phase

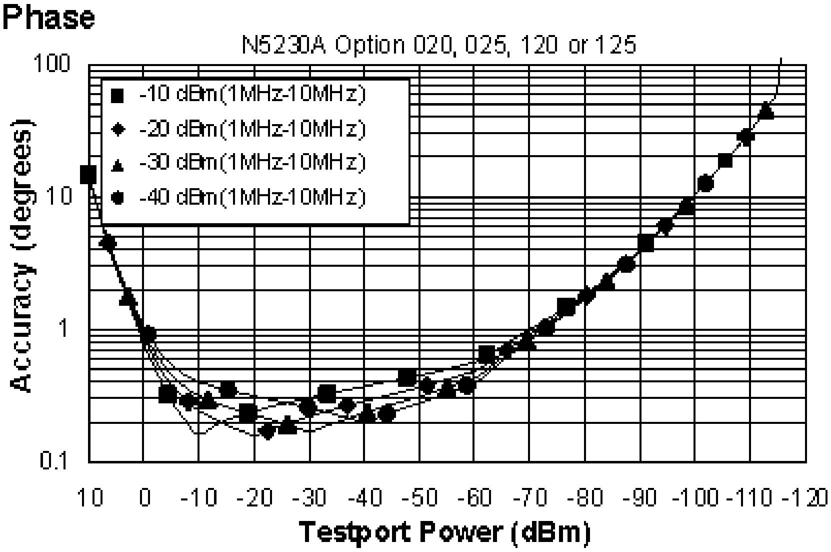


Dynamic Accuracy, 1 MHz- 10 MHz, Option 020, 025, 120, or 125

Magnitude

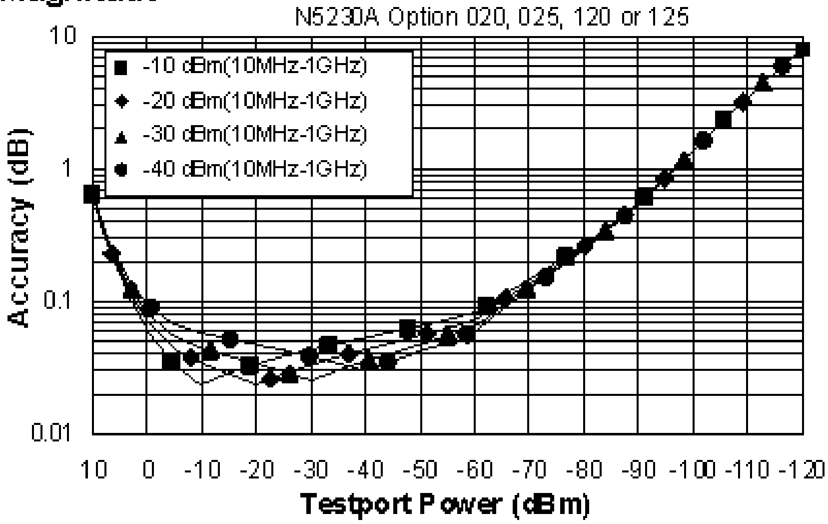


Phase

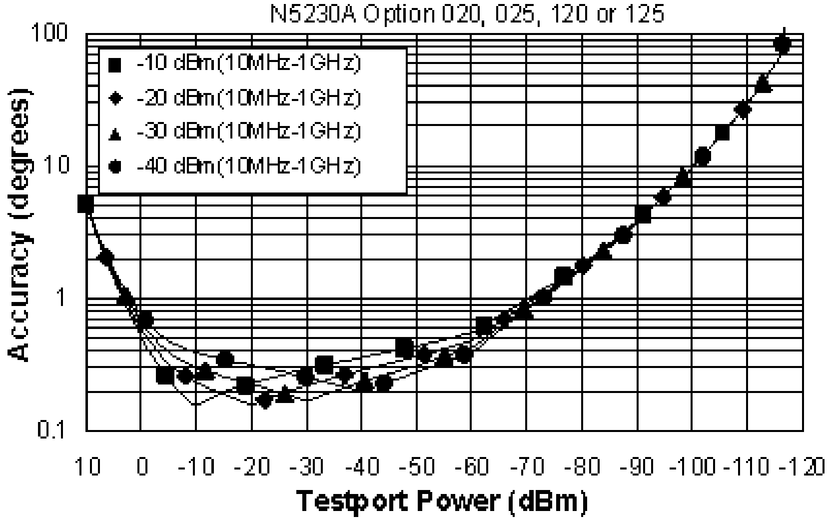


Dynamic Accuracy, 10 MHz- 1 GHz, Option 020, 025, 120, or 125

Magnitude

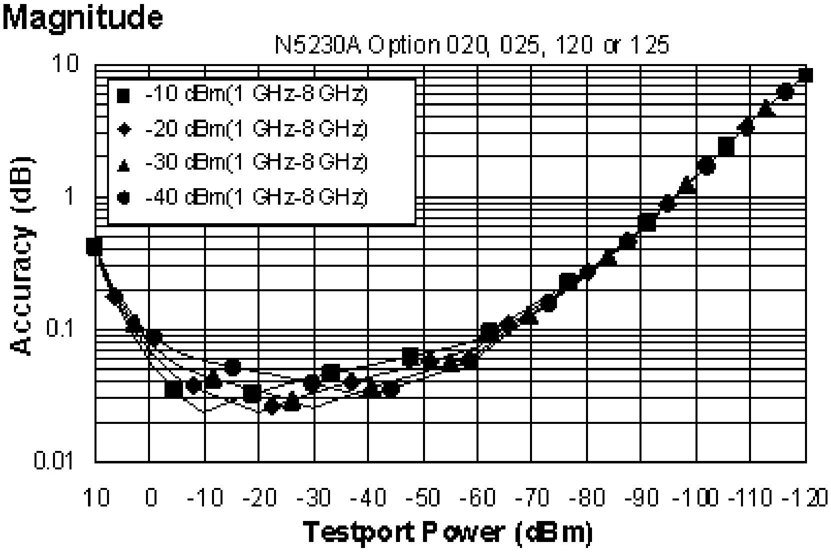


Phase

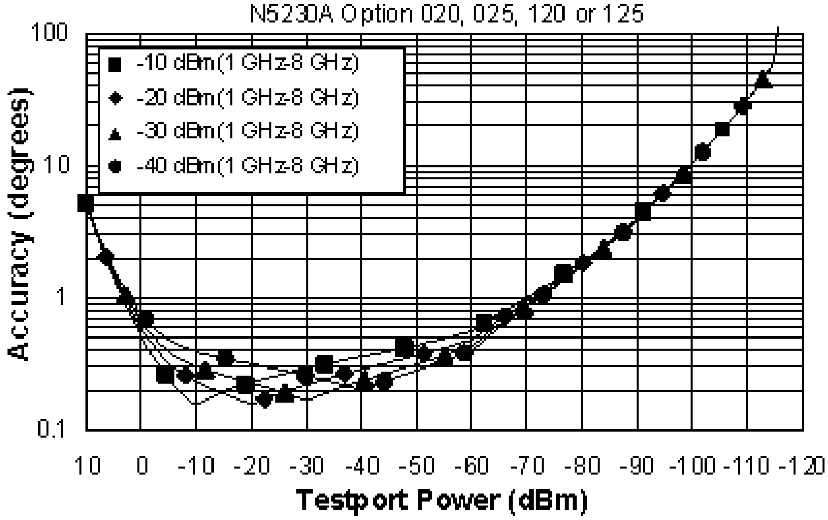


Dynamic Accuracy, 1 - 8 GHz, Option 020, 025, 120, or 125

Magnitude

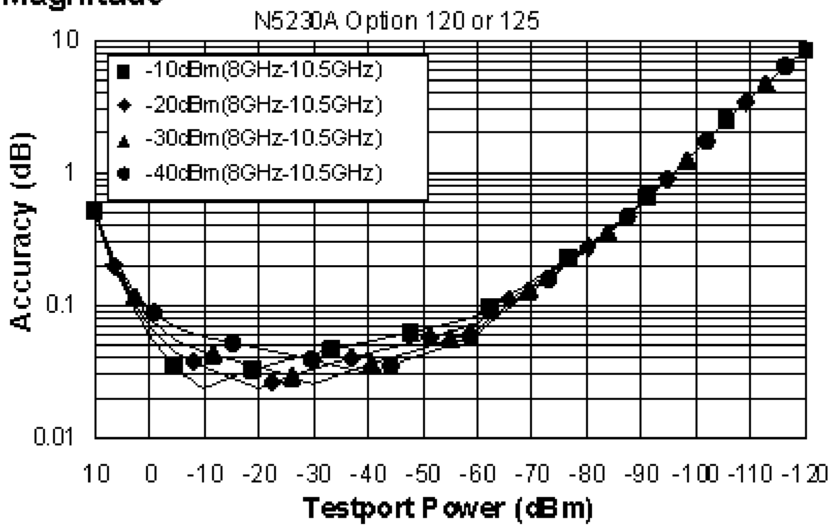


Phase

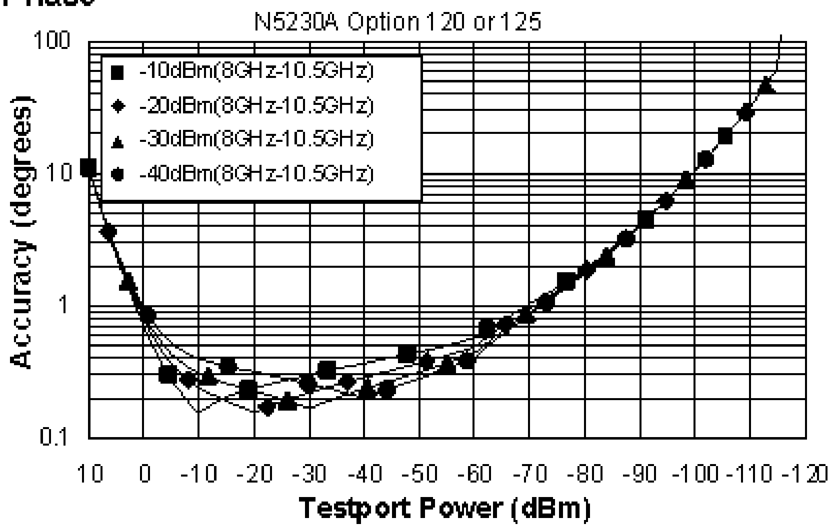


Dynamic Accuracy, 8 - 10.5 GHz, Option 120 or 125

Magnitude

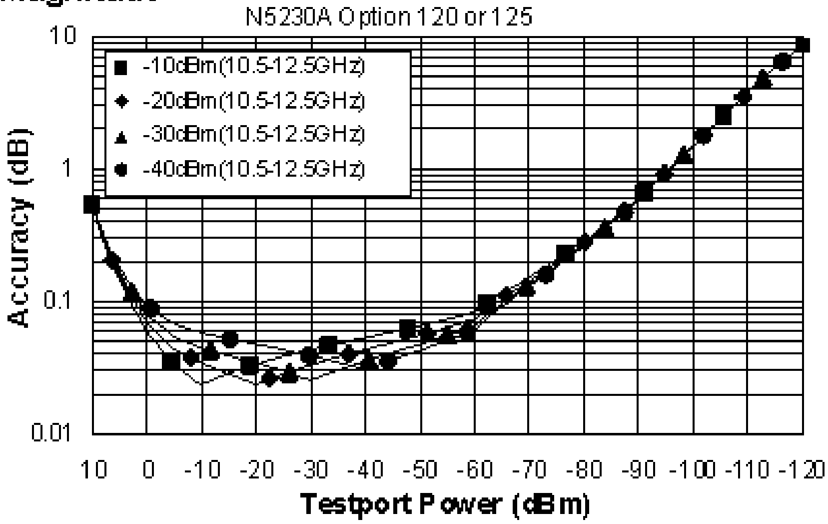


Phase

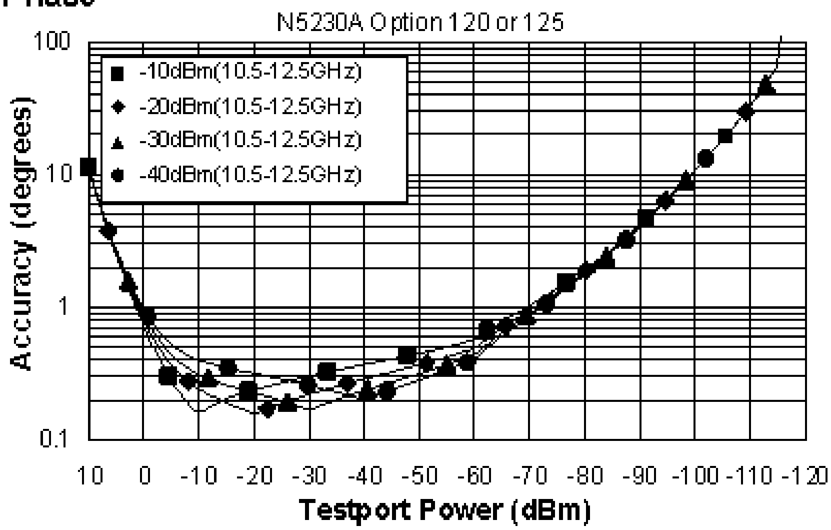


Dynamic Accuracy, 10.5 - 12.5 GHz, Option 120 or 125

Magnitude

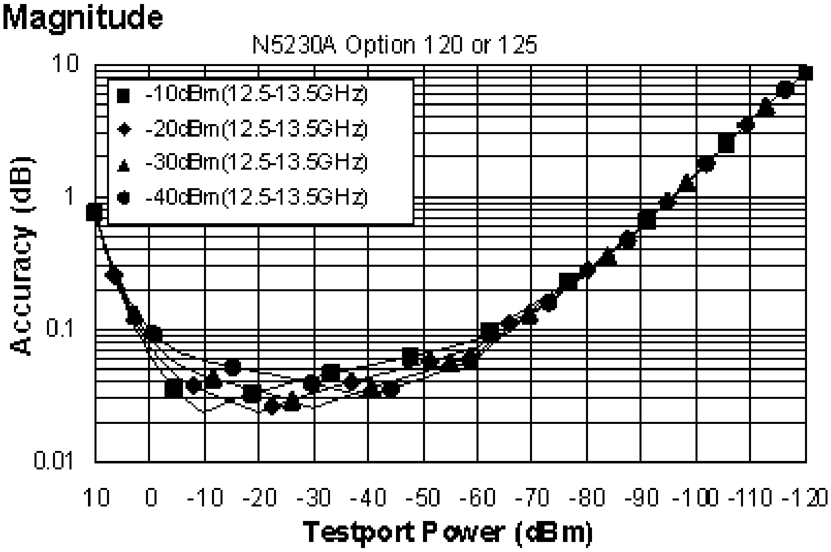


Phase

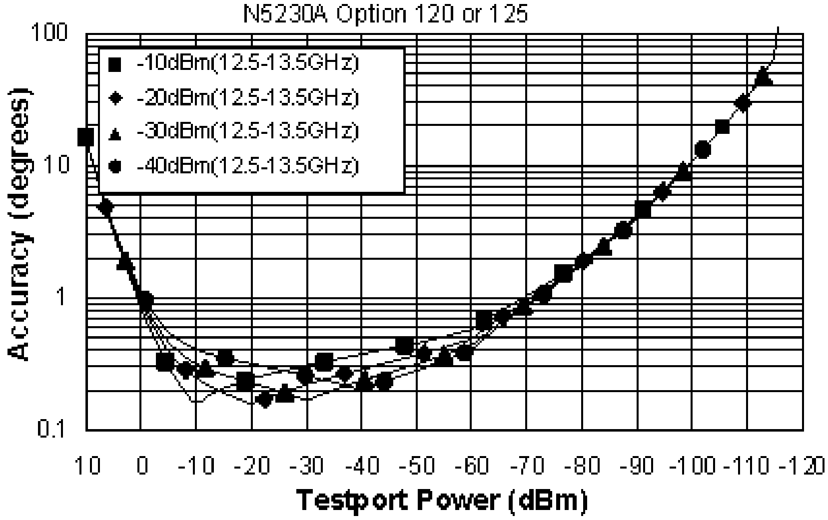


Dynamic Accuracy, 12.5 - 13.5 GHz, Option 120 or 125

Magnitude



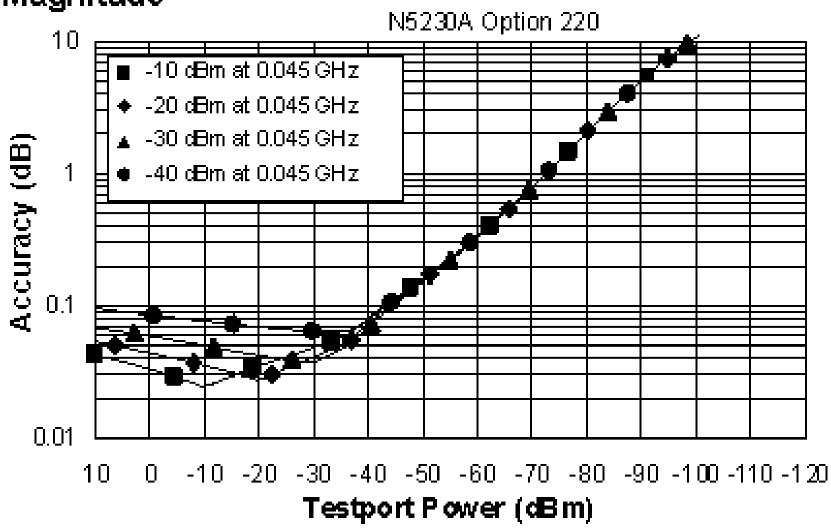
Phase



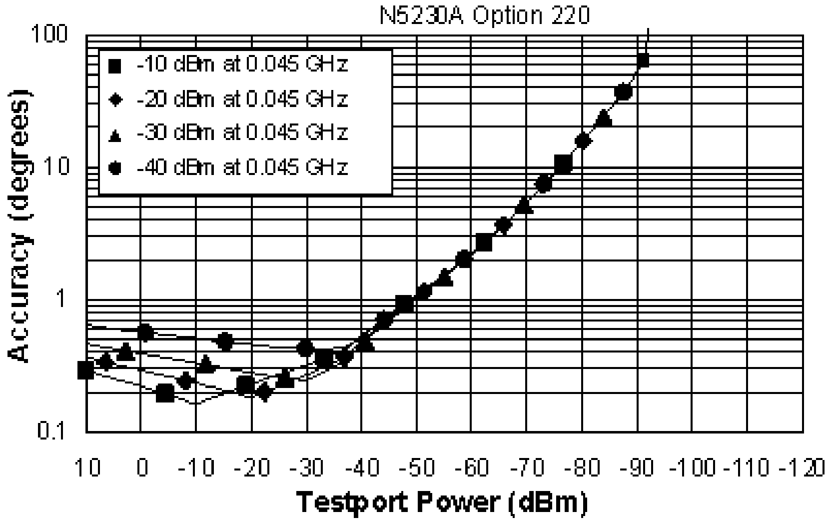
Options 220, 225

Dynamic Accuracy, 0.045 GHz, Option 220 or 225

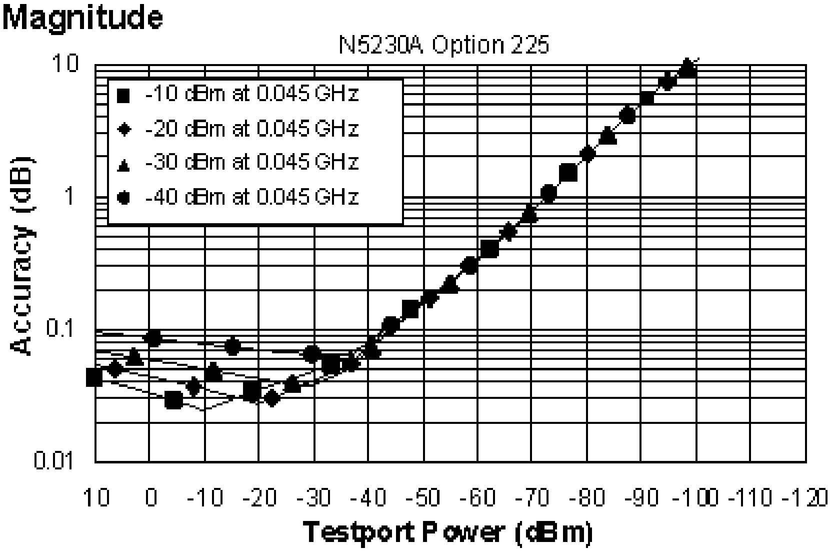
Magnitude



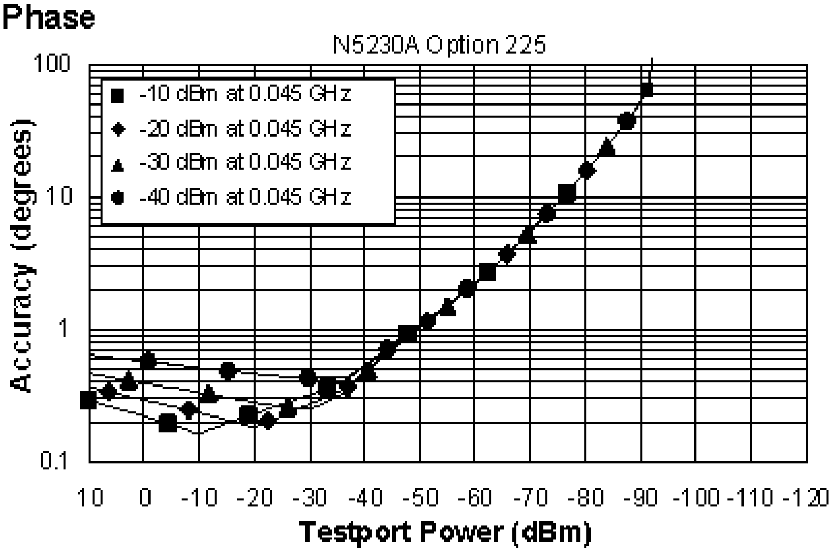
Phase



Magnitude

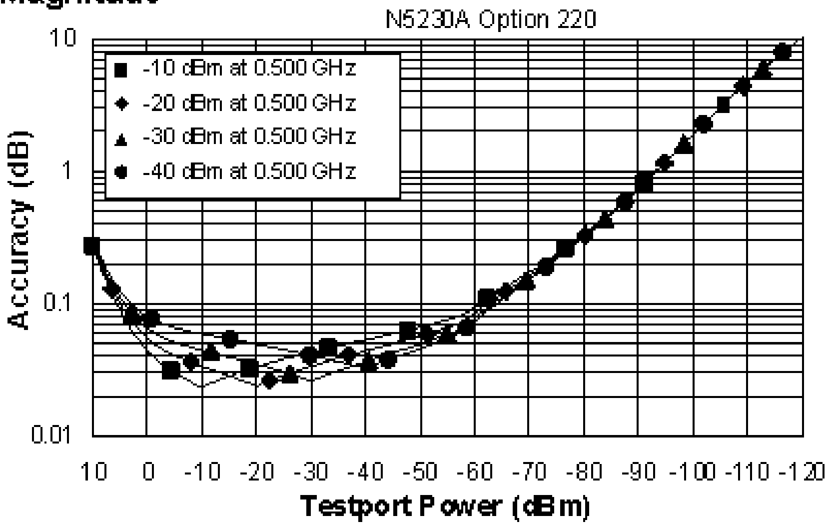


Phase

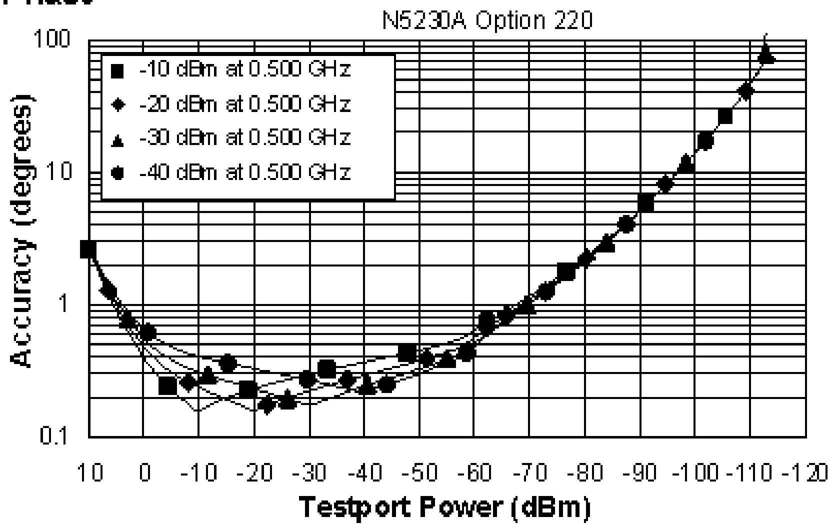


Dynamic Accuracy, 0.500 GHz, Option 220 or 225

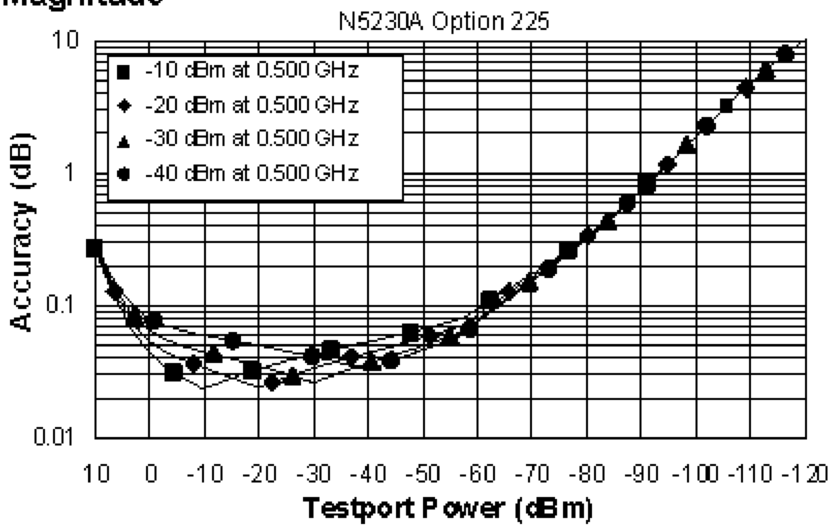
Magnitude



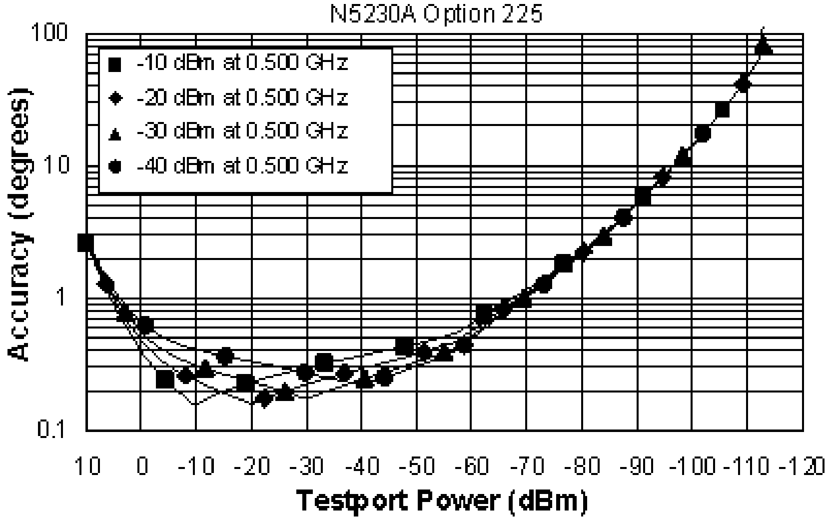
Phase



Magnitude

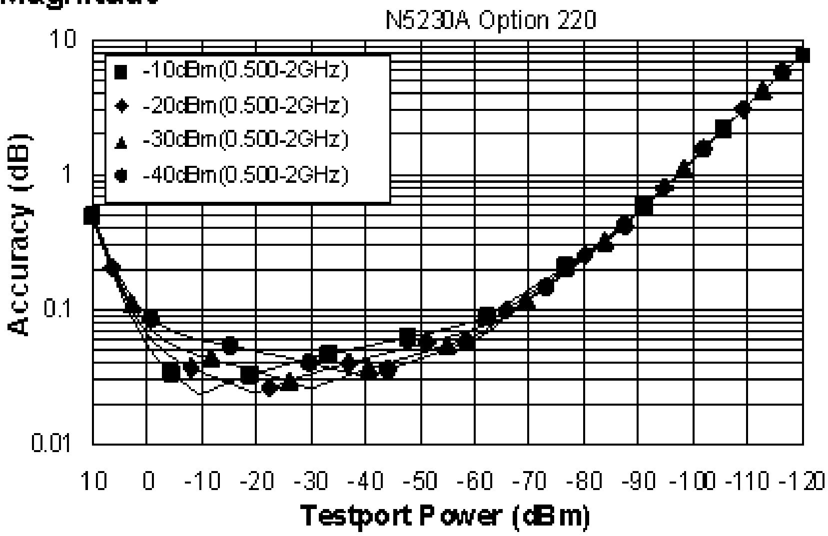


Phase

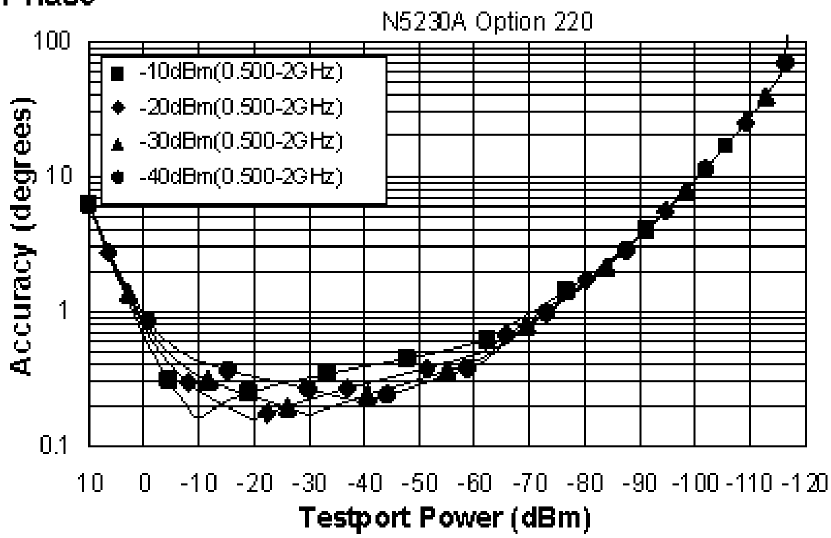


Dynamic Accuracy, 0.500 - 2 GHz, Option 220 or 225

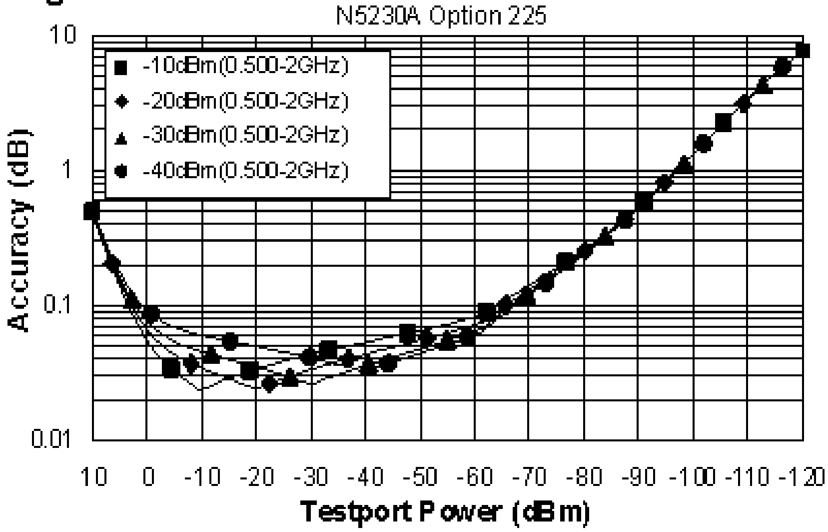
Magnitude



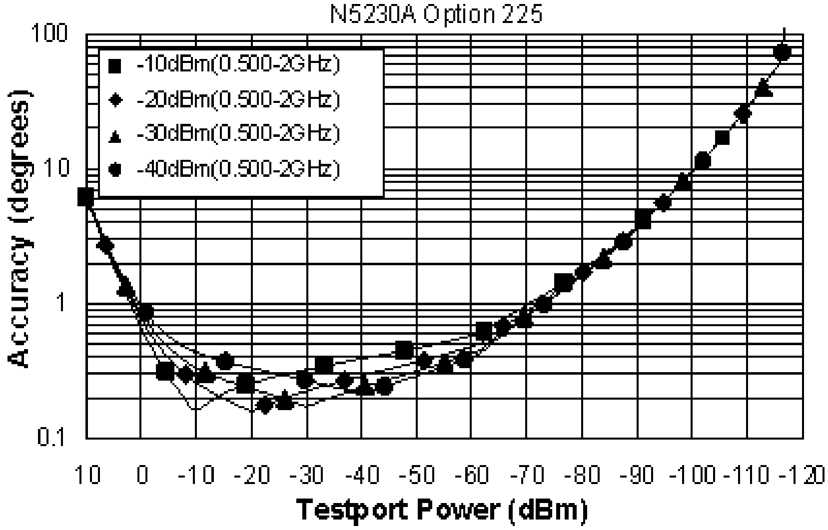
Phase



Magnitude

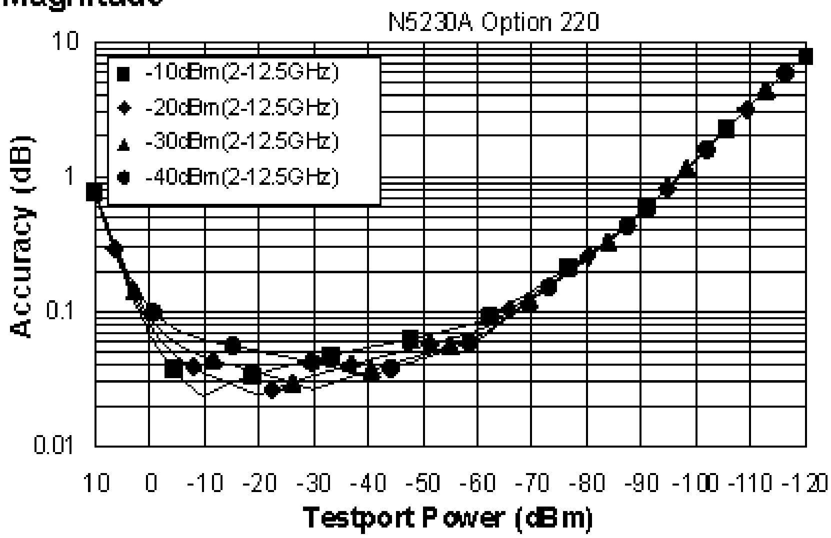


Phase

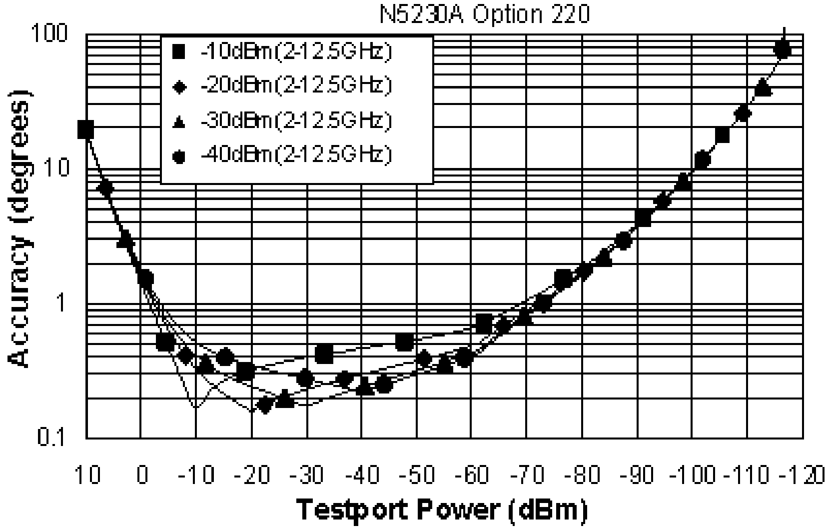


Dynamic Accuracy, 2 - 12.5 GHz, Option 220 or 225

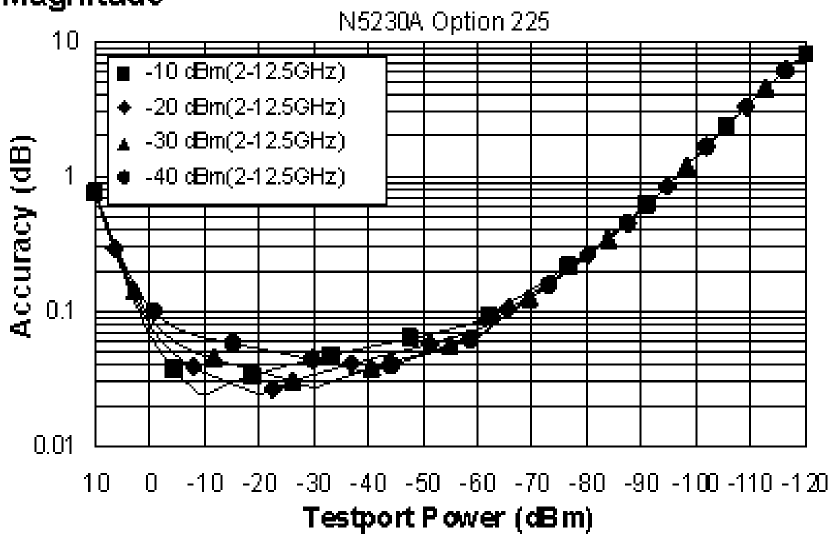
Magnitude



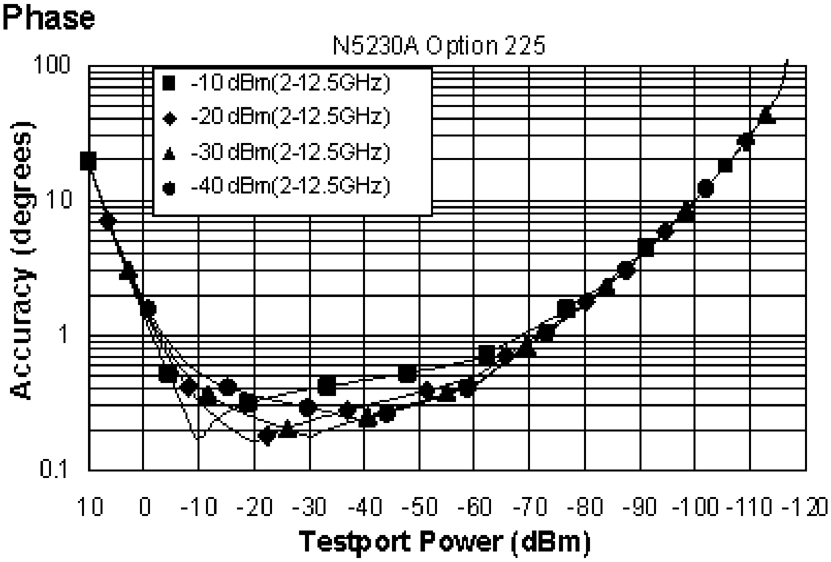
Phase



Magnitude

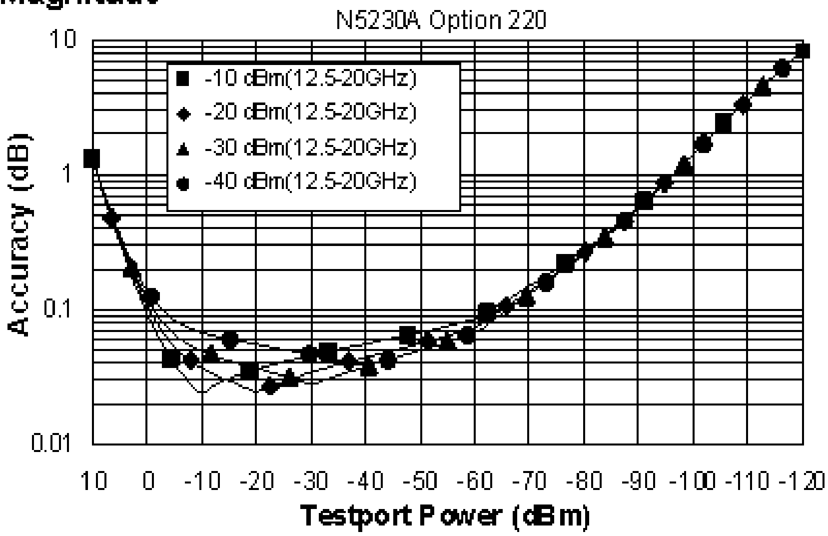


Phase

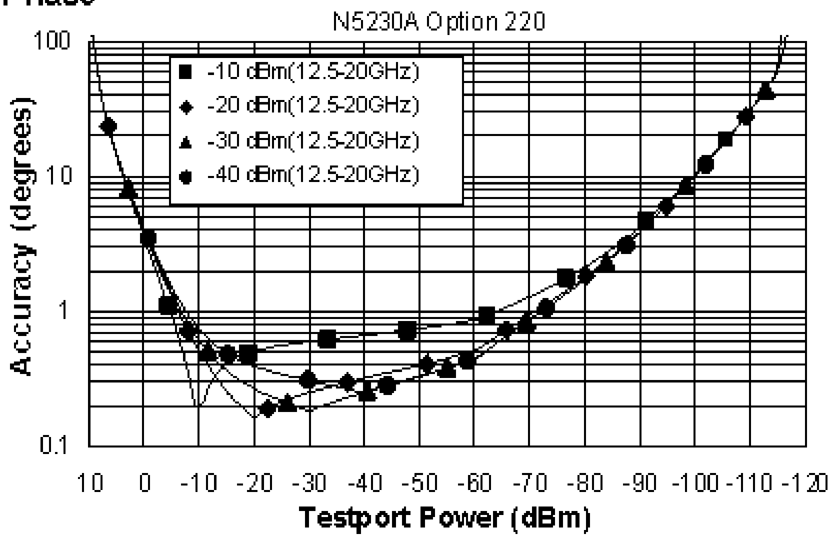


Dynamic Accuracy, 12.5 - 20 GHz, Option 220 or 225

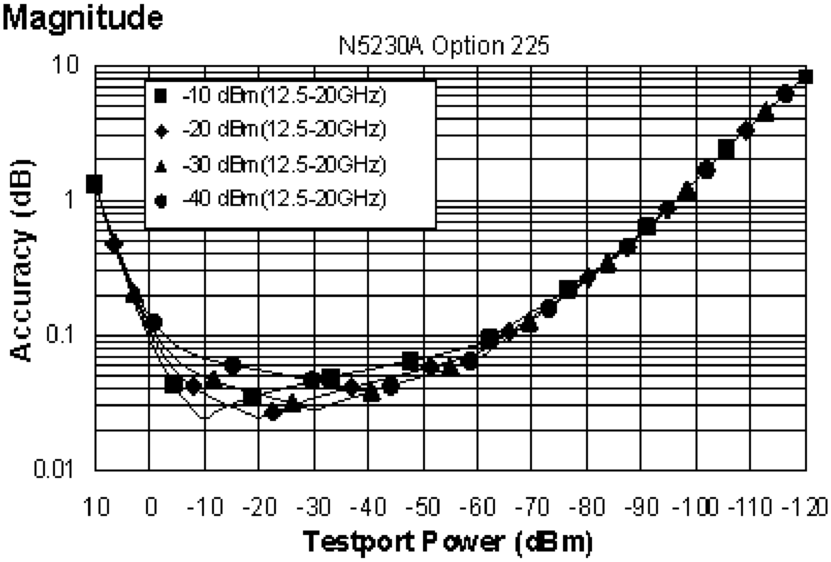
Magnitude



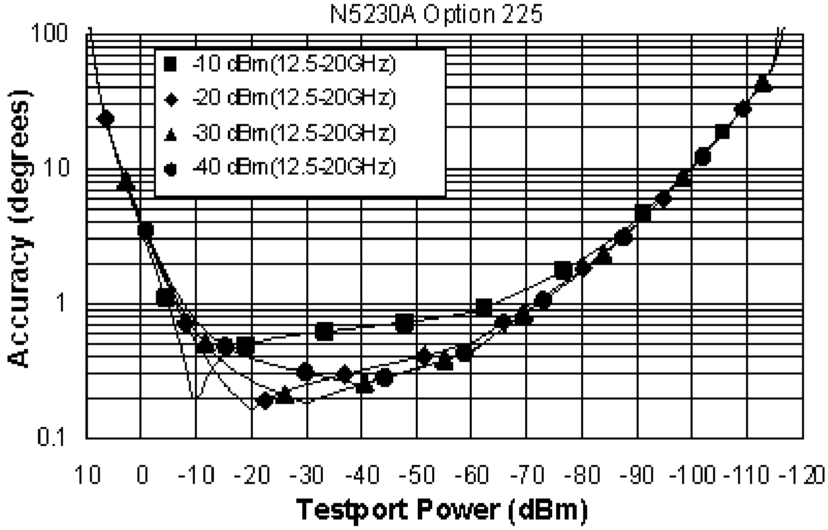
Phase



Magnitude



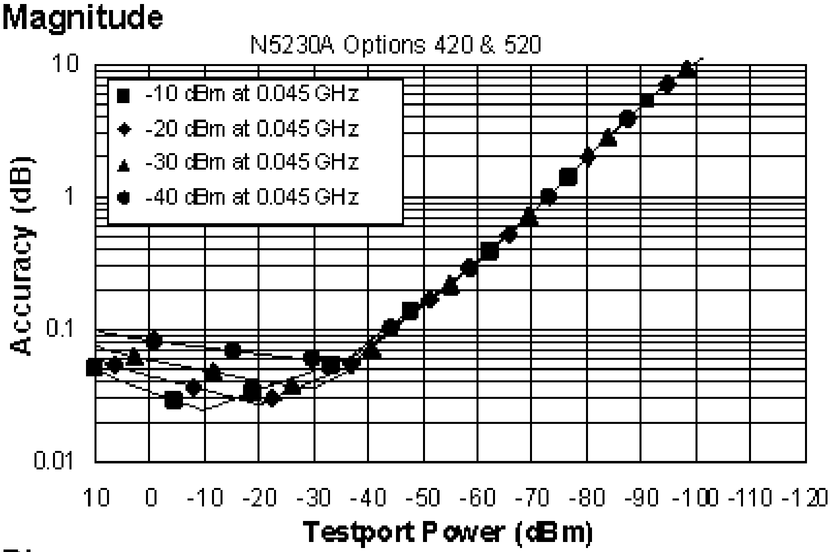
Phase



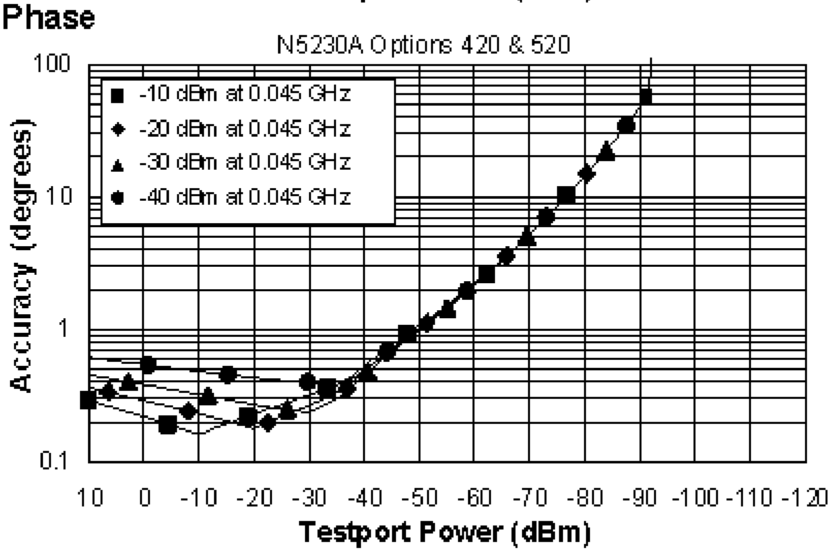
Options 420, 425, 520, 525

Dynamic Accuracy, 0.045 GHz, Option 420, 425, 520, or 525

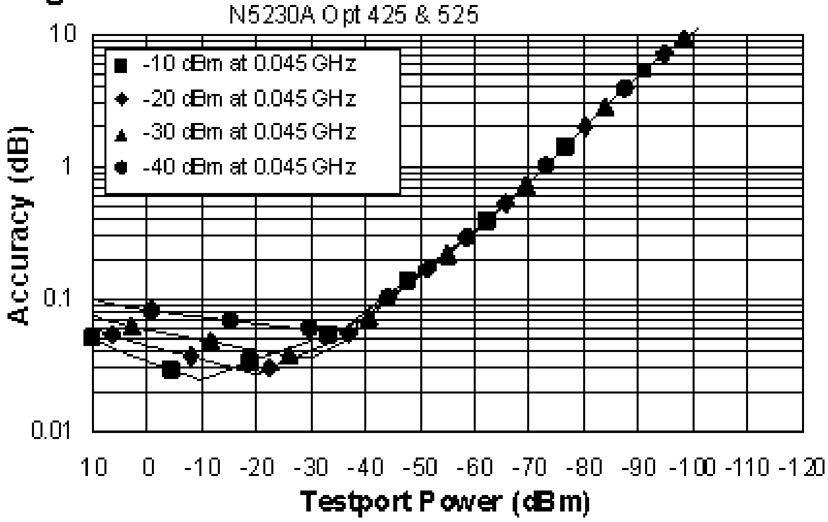
Magnitude



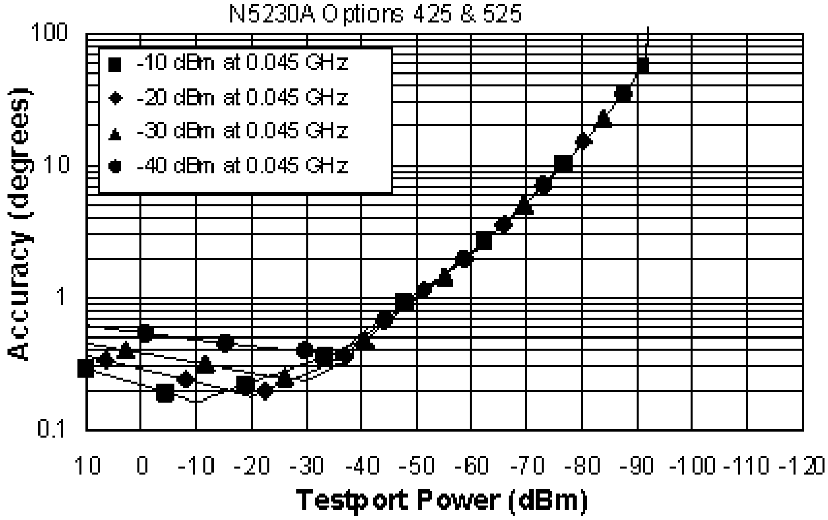
Phase



Magnitude

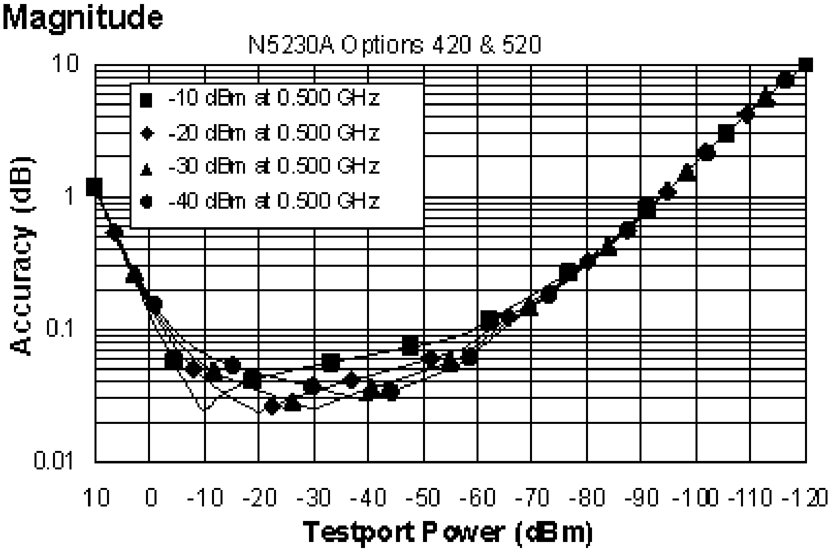


Phase

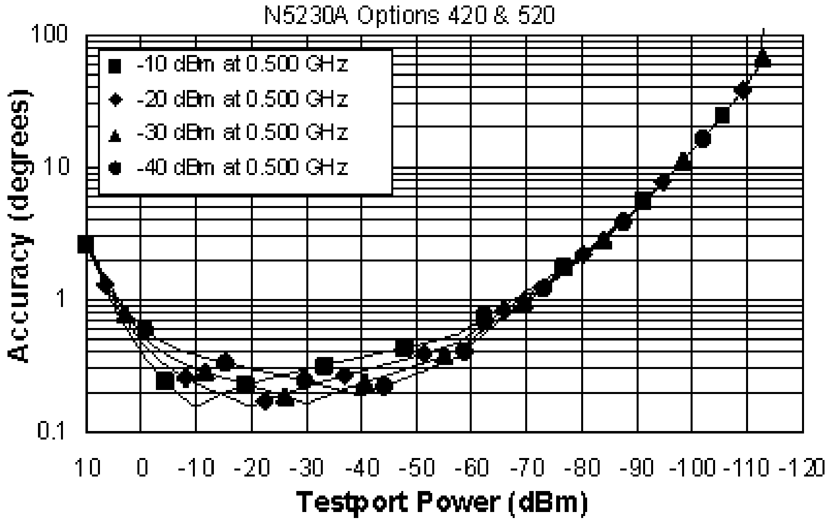


Dynamic Accuracy, 0.500 GHz, Option 420, 425, 520, or 525

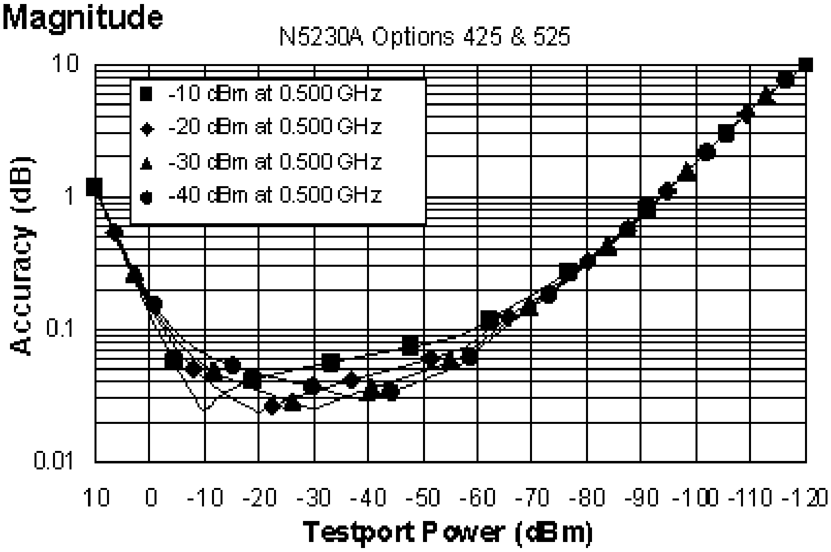
Magnitude



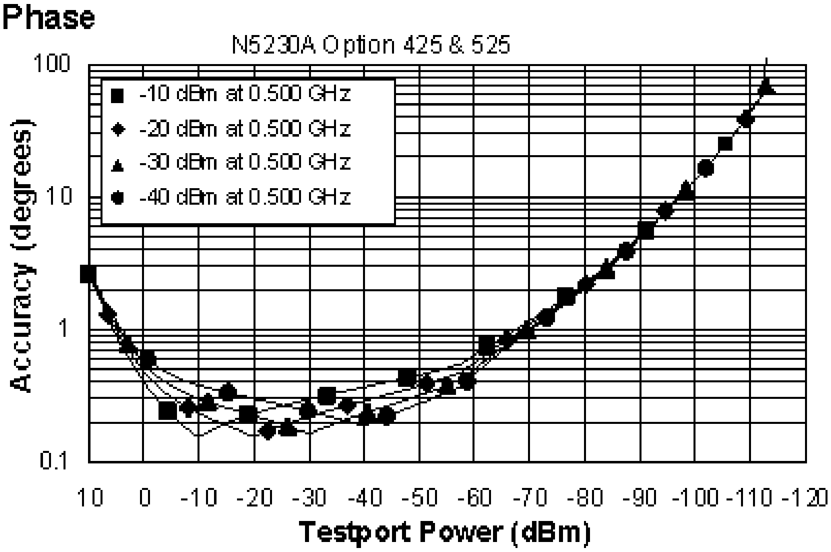
Phase



Magnitude

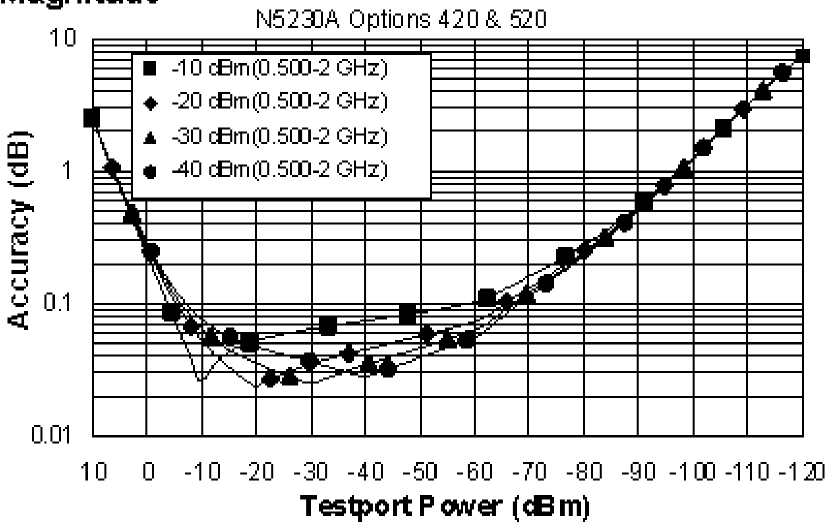


Phase

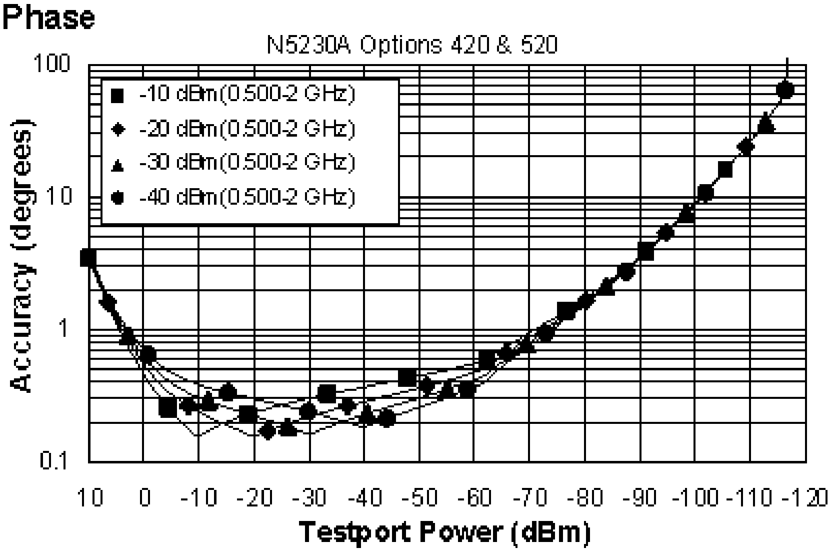


Dynamic Accuracy, 0.500 - 2 GHz, Option 420, 425, 520, or 525

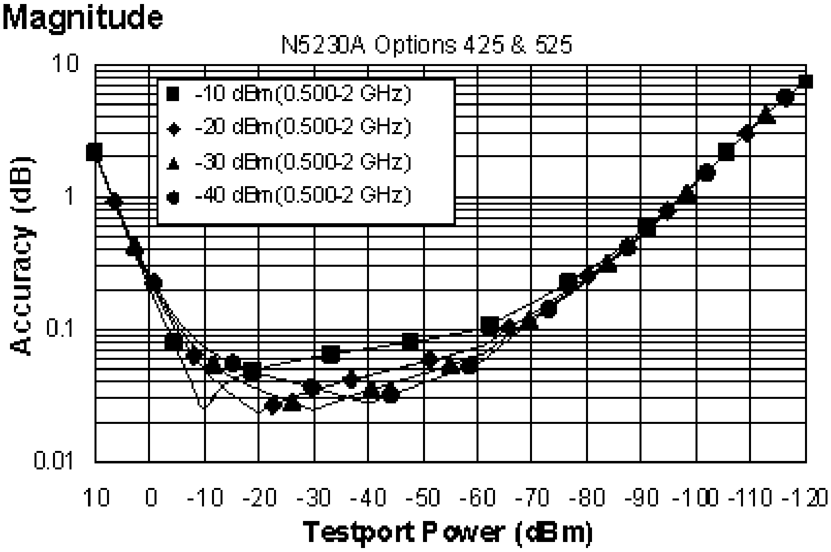
Magnitude



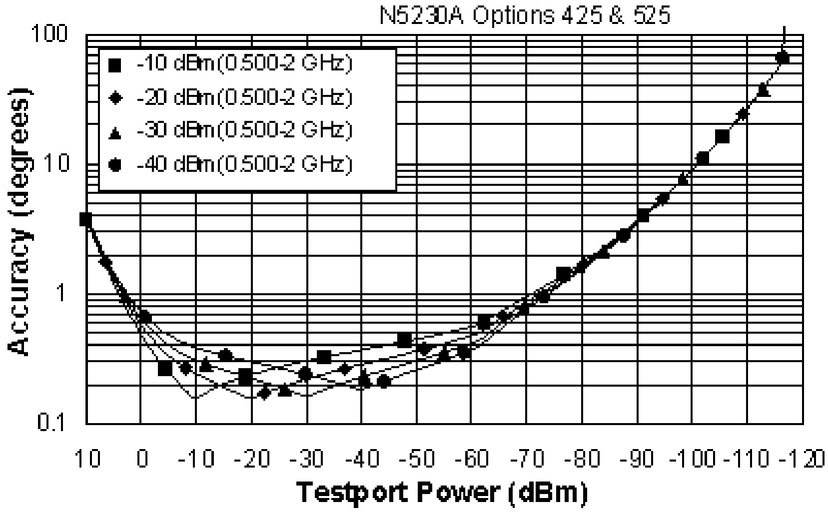
Phase



Magnitude

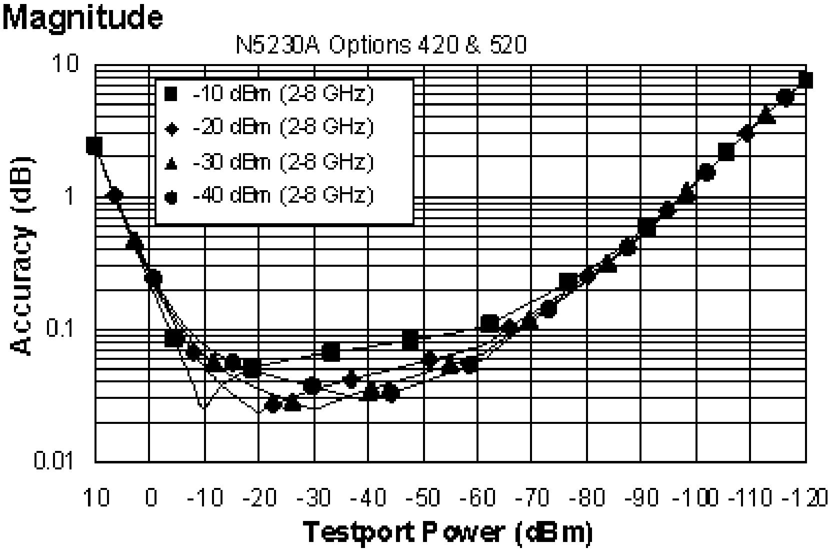


Phase

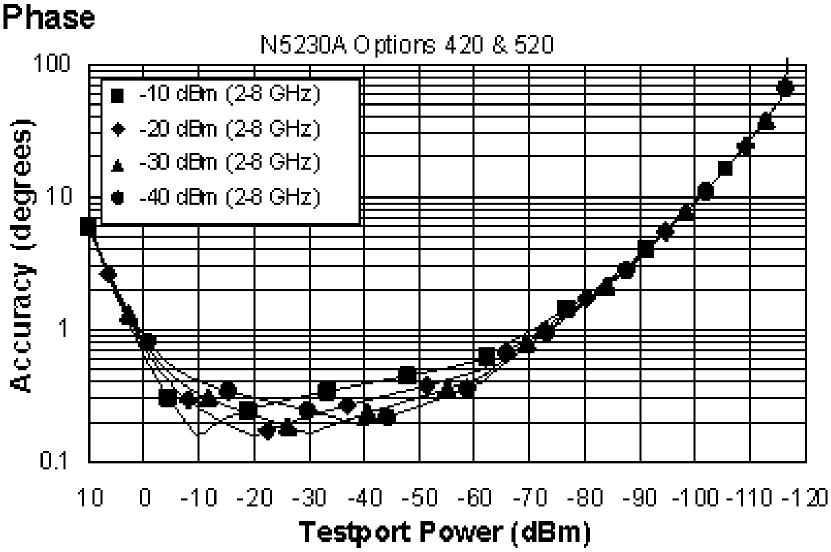


Dynamic Accuracy, 2 - 8 GHz, Option 420, 425, 520, or 525

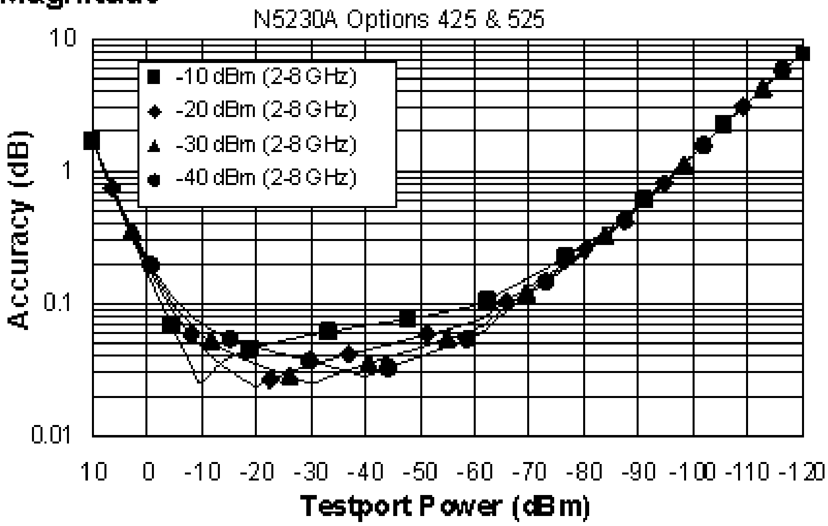
Magnitude



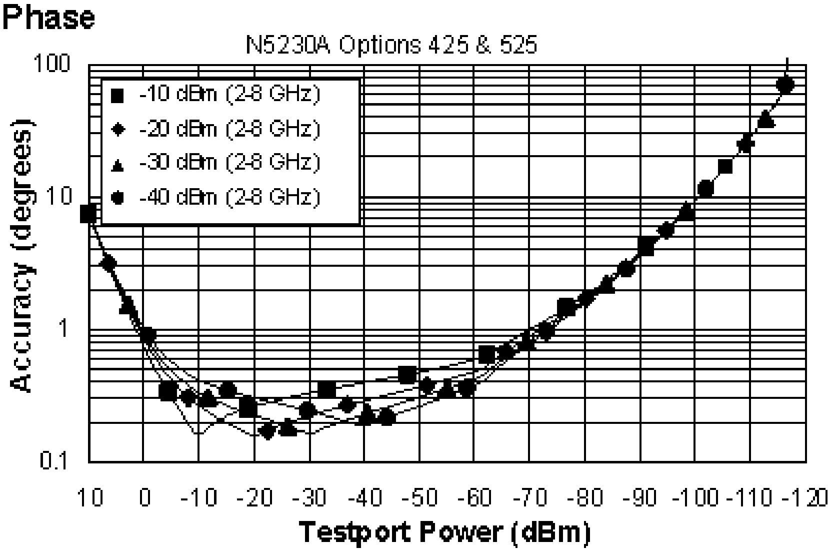
Phase



Magnitude

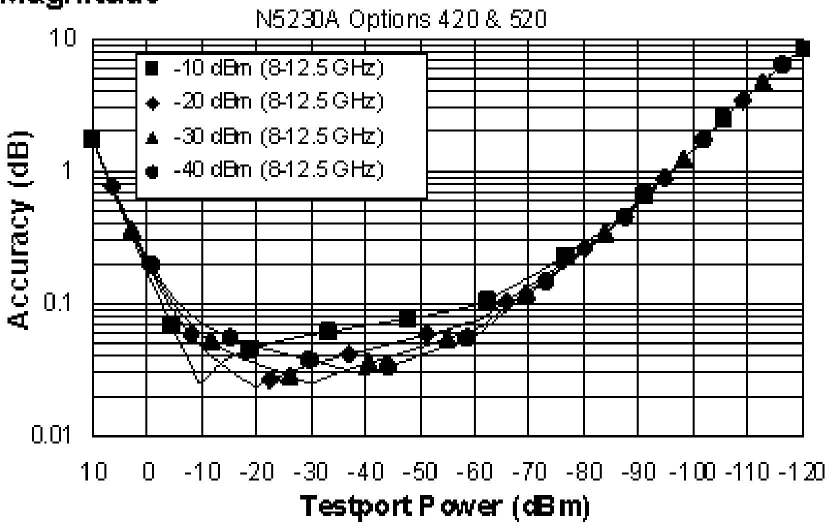


Phase

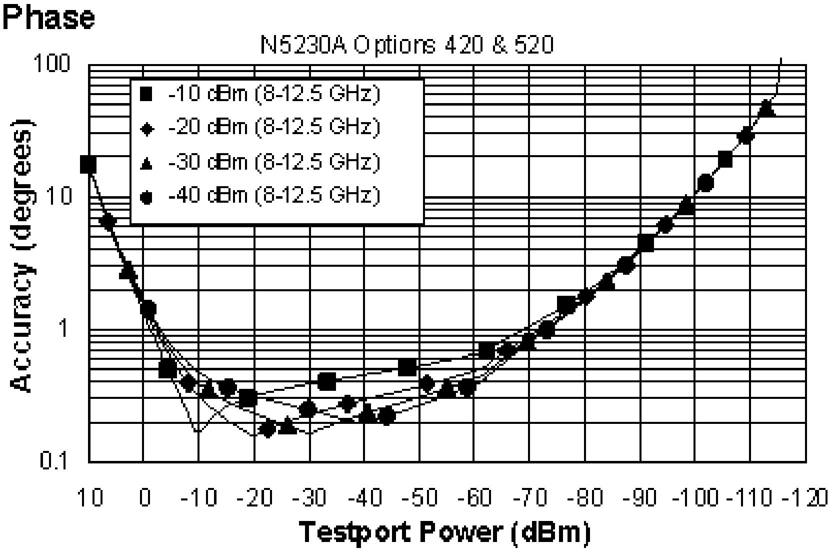


Dynamic Accuracy, 8 - 12.5 GHz, Option 420, 425, 520, or 525

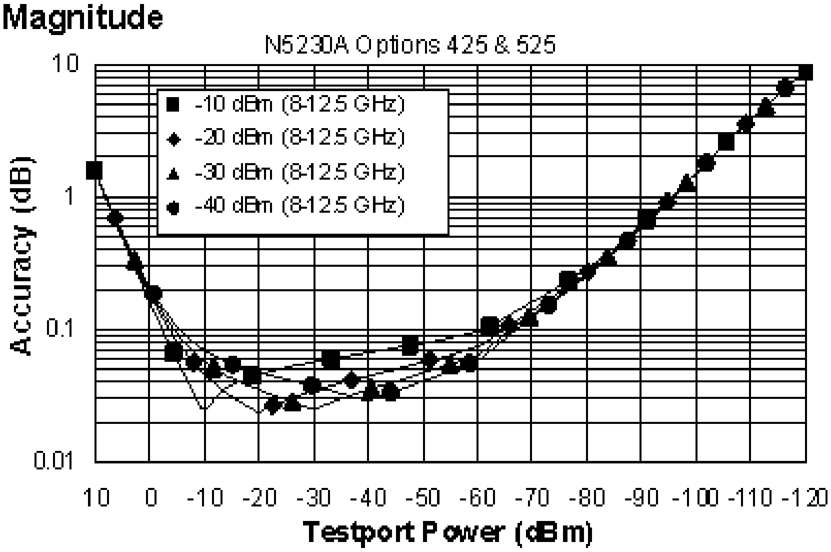
Magnitude



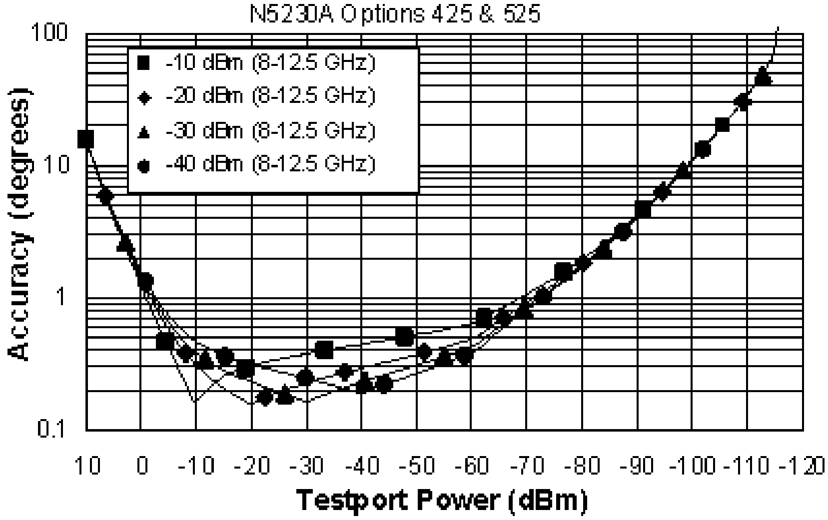
Phase



Magnitude

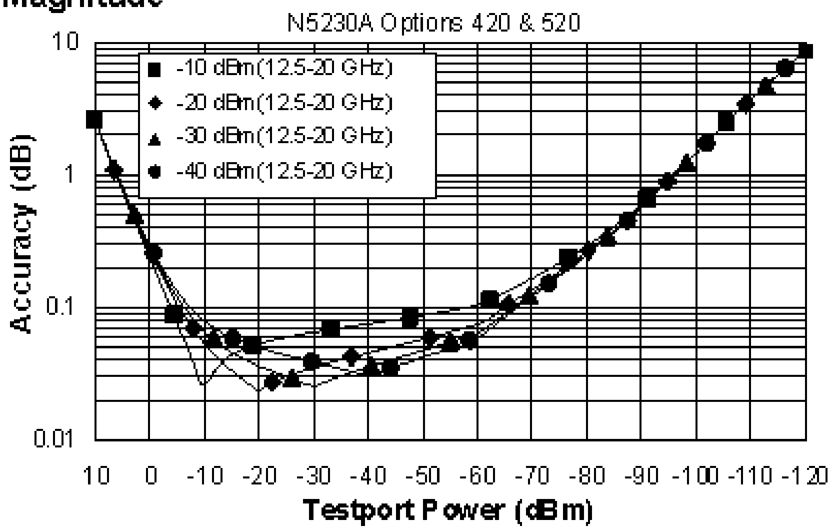


Phase

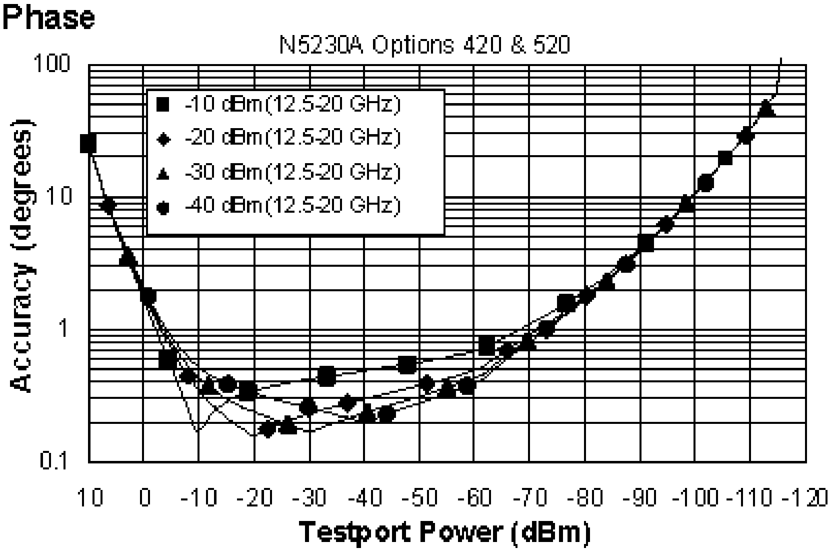


Dynamic Accuracy, 12.5 - 20 GHz, Option 420, 425, 520, or 525

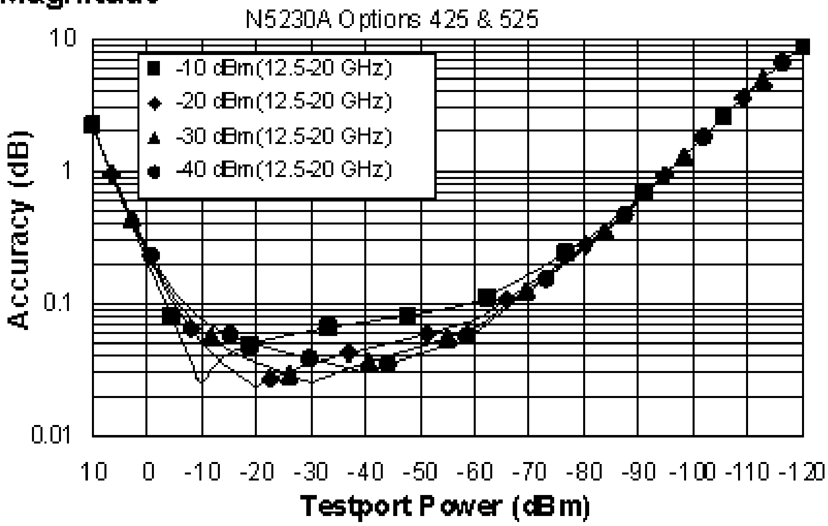
Magnitude



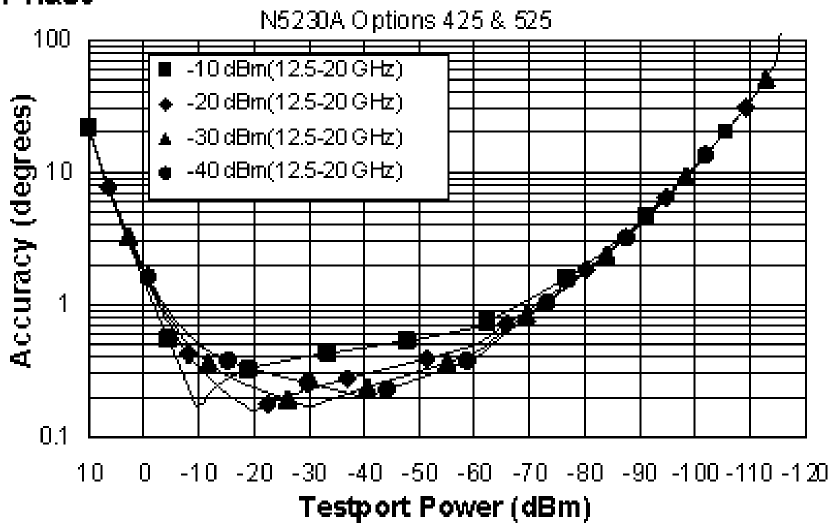
Phase



Magnitude

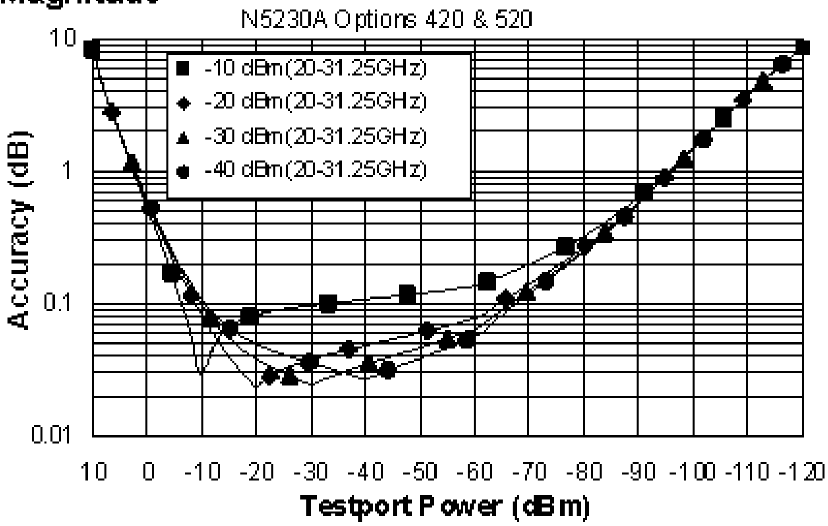


Phase

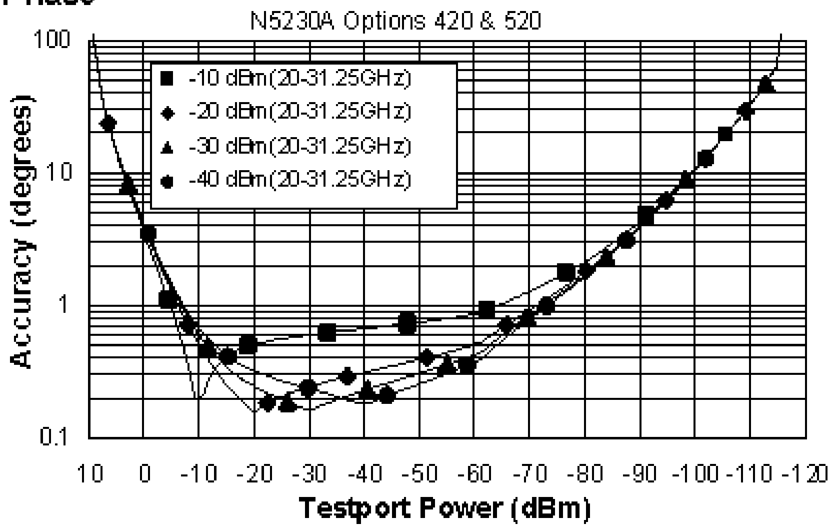


Dynamic Accuracy, 20 - 31.25 GHz, Option 420, 425, 520, or 525

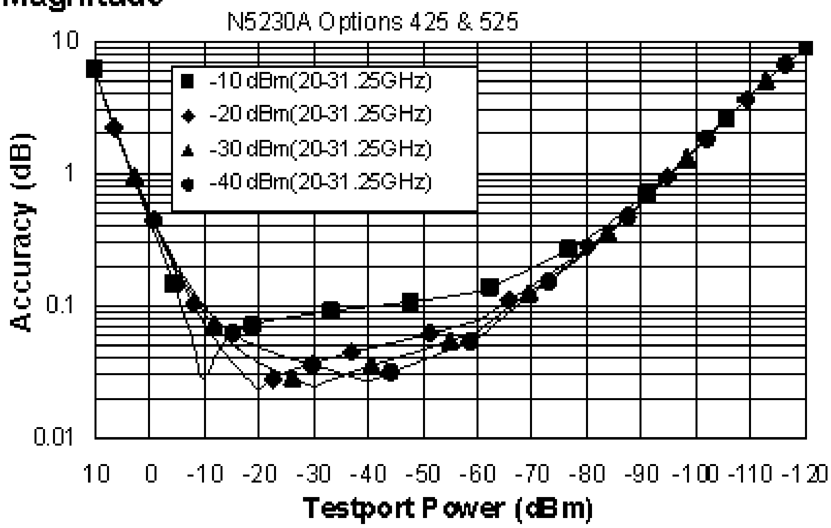
Magnitude



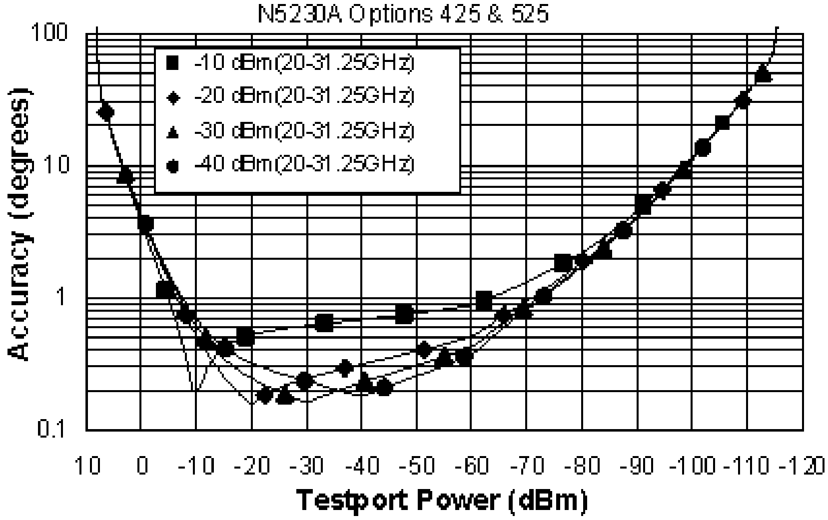
Phase



Magnitude

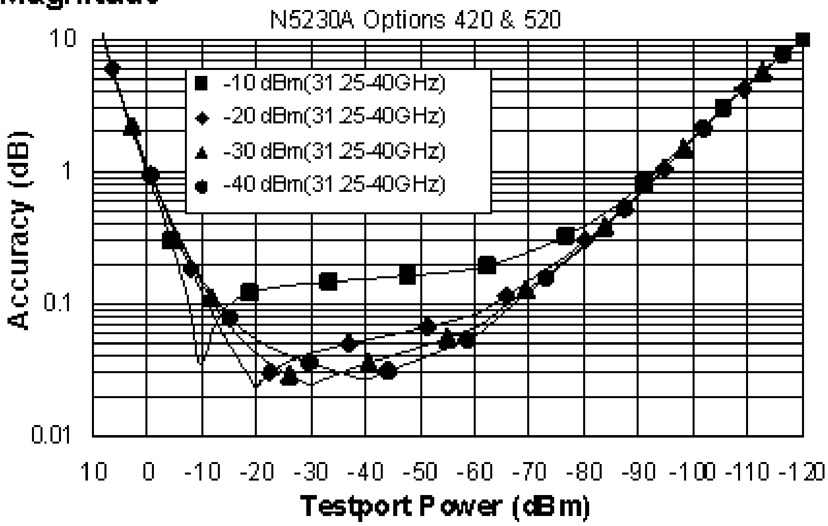


Phase

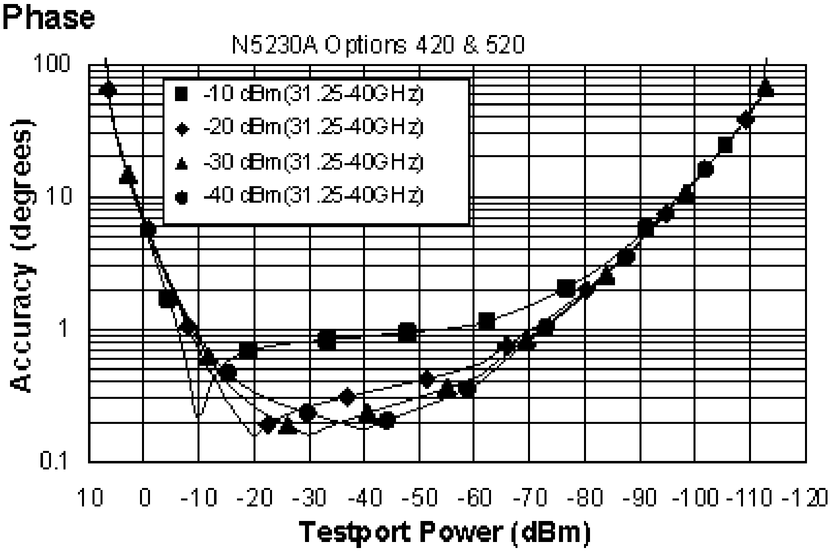


Dynamic Accuracy, 31.25 - 40 GHz, Option 420, 425, 520, or 525

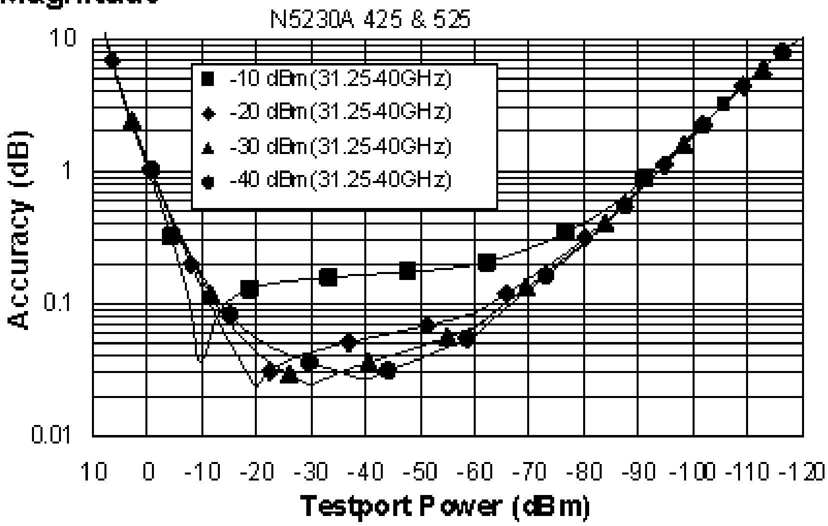
Magnitude



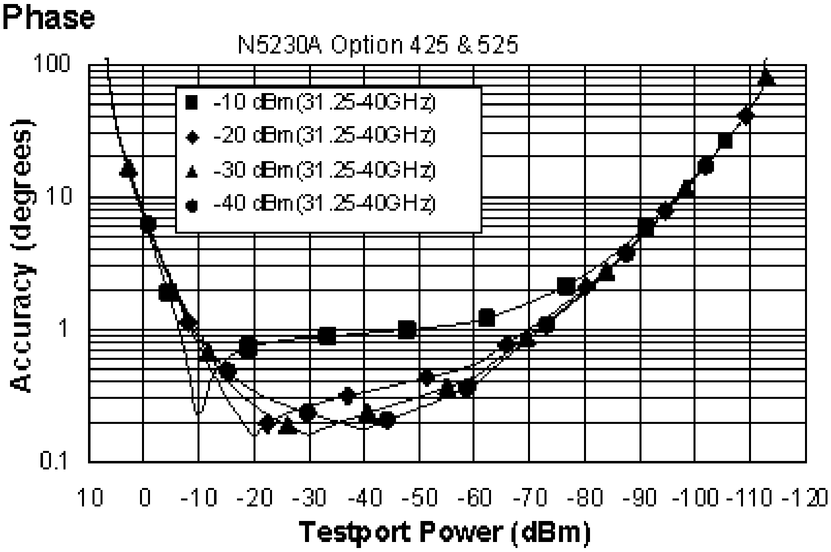
Phase



Magnitude

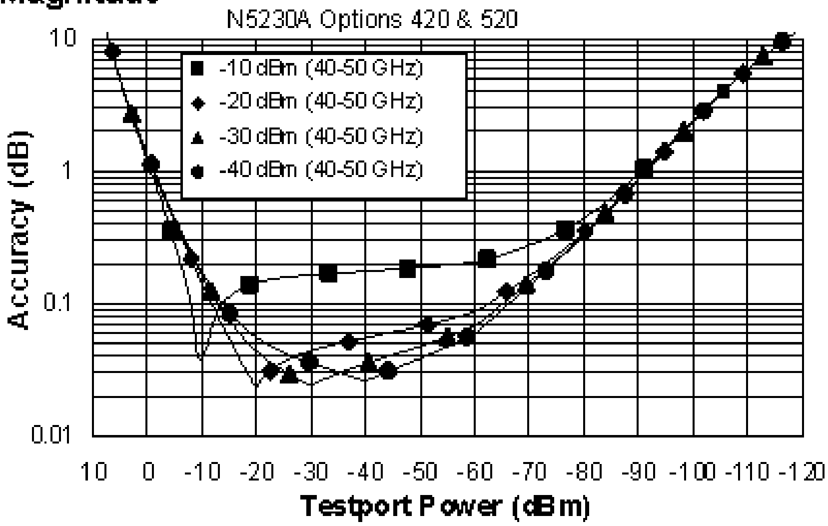


Phase

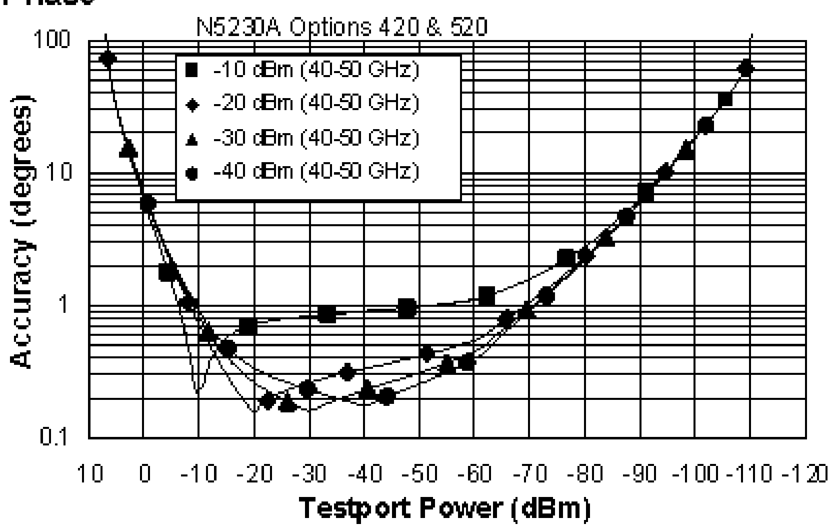


Dynamic Accuracy, 40 - 50 GHz, Option 520 or 525

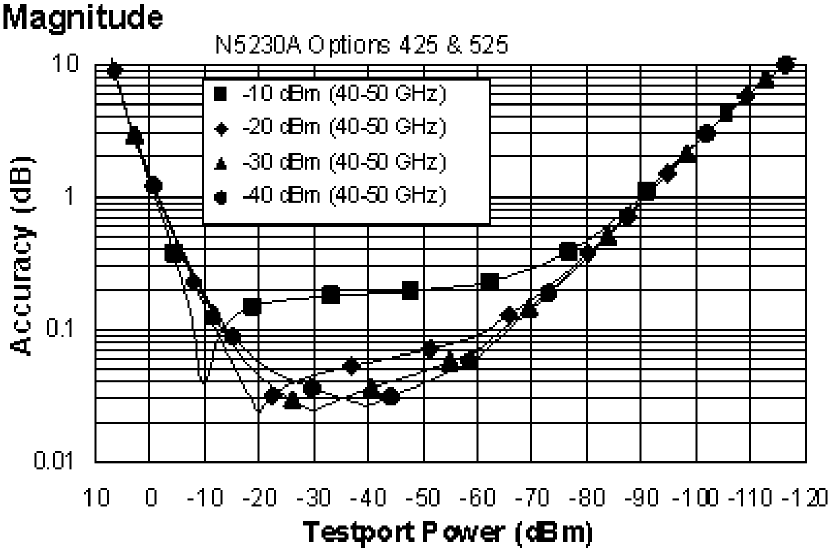
Magnitude

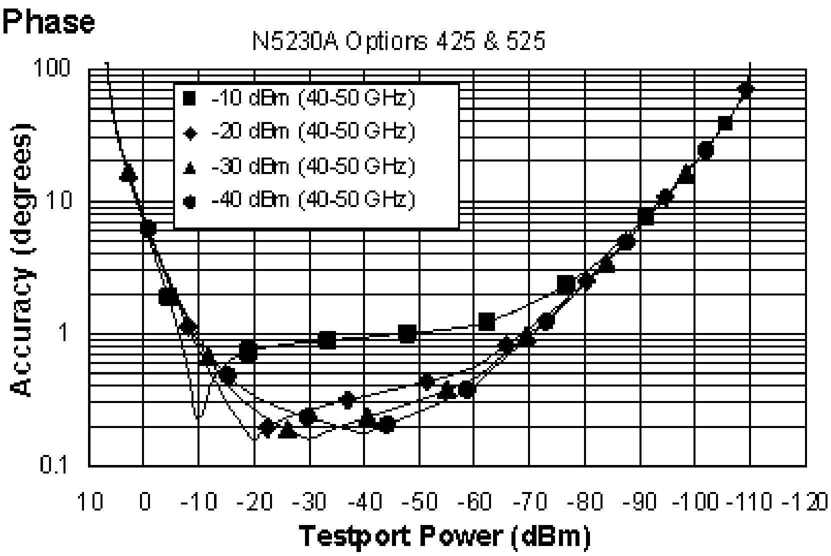


Phase



Magnitude





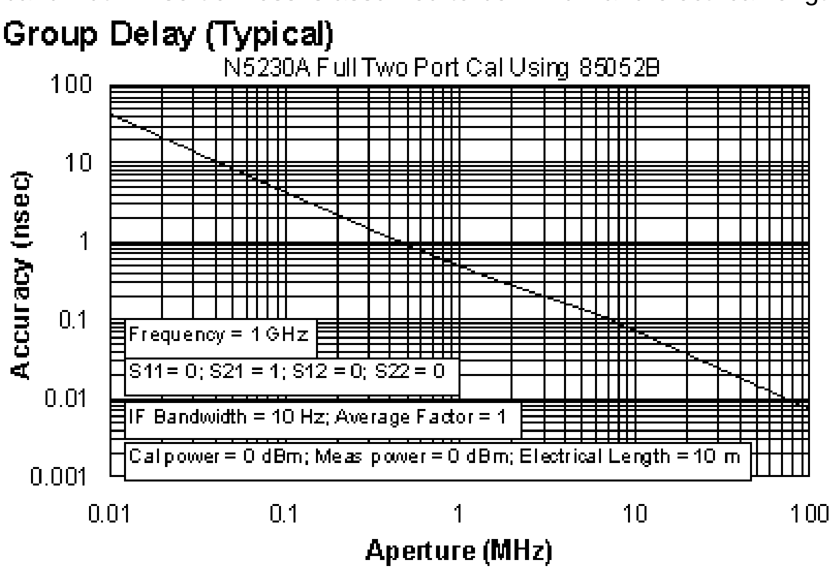
a Dynamic accuracy is verified with the following measurements:

- compression over frequency
- IF linearity at a single frequency of 1.195 GHz using a reference level of -20 dBm for an input power range of 0 to -110 dBm.

Table 19. Test Port Input (Group Delay)[a]

Description	Specification	Supplemental Information (typ.)
Aperture (selectable)		(frequency span)/(number of points -1)
Maximum Aperture		20% of frequency span
Range		0.5 x (1/minimum aperture)
Maximum Delay		Limited to measuring no more than 180° of phase change within the minimum aperture.)
Accuracy		See graph below. Char.

The following graph shows characteristic group delay accuracy with full 2-port calibration and a 10 Hz IF bandwidth. Insertion loss is assumed to be < 2 dB and electrical length to be ten meters.



In general, the following formula can be used to determine the accuracy, in seconds, of specific group delay measurement: ±Phase Accuracy (deg)/[360 × Aperture (Hz)] Depending on the aperture and device length, the phase accuracy used is either incremental phase accuracy or worst case phase accuracy.

a Group delay is computed by measuring the phase change within a specified frequency step (determined by the frequency span and the number of points per sweep).

General Information

- Miscellaneous Information
- Front Panel
- Rear Panel
- Environment and Dimensions

Table 20. Miscellaneous Information

Description	Specification	Supplemental Information
System IF Bandwidth Range		1 Hz to 250 kHz, nominal
CPU		Intel® 500 MHz Pentium® III

Table 21. Front Panel Information

Description	Supplemental Information
RF Connectors	
N5230A	
Type	Option 220 or 225: 3.5 mm (male), 50 ohm, (nominal) Option 420, 425, 520, or 525: 2.4 mm (male), 50 ohm, (nominal)
Center Pin Recession	0.002 in. (characteristic)
Display	
Size	21.3 cm (8.4 in) diagonal color active matrix LCD; 640 (horizontal) X 480 (vertical) resolution
Refresh Rate	Vertical 59.83 Hz; Horizontal 31.41 kHz

Pixels	When running the analyzer's built-in "Display Test[1]," one or more of the following symptoms indicate a faulty display assembly: • A complete row or column of "stuck on" or "dark" pixels. • More than six "stuck on" pixels (but not more than three green) • More than twelve "dark" pixels (but not more than seven of the same color) • Two or more consecutive "stuck on" pixels or three or more consecutive "dark" pixels (but no more than one set of two consecutive dark pixels") • "Stuck on" or "dark" pixels less than 6.5 mm apart (excluding consecutive pixels)
Display Range	
Magnitude	±500 dB (at 20 dB/div), max
Phase	±500°, max
Polar	10 pUnits, min 1000 Units, max
Display Resolution	
Magnitude	0.001 dB/div, min
Phase	0.01°/div, min
Marker Resolution	
Magnitude	0.001 dB, min
Phase	0.01°, min
Polar	0.01 mUnit, min; 0.01°,min

1 Any PNA Service Guide contains instructions for running the analyzer's built-in "Display Test." You can view a PDF file of a Service Guide on the CD-ROM that you received with your PNA, or on the Web at http://www.agilent.com. After opening the PDF file, look in the left-side column and the click on the "Index" bookmark. From the index, click on "Display Test".

Table 22. Rear Panel Information

Description	Supplemental Information
10 MHz Reference In	
Connector	BNC, female
Input Frequency	10 MHz ± 10 ppm, Typical
Input Level	-15 dBm to +20 dBm, Typical
Input Impedance	200 Ω, nom.
10 MHz Reference Out	
Connector	BNC, female

Output Frequency	10 MHz ± 1 ppm, Typical
Signal Type	Sine Wave, Typical
Output Level	+10 dBm ± 4 dB into 50 Ω, Typical
Output Impedance	50 Ω, nominal
Harmonics	<-40 dBc, Typical
VGA Video Output	
Connector	15-pin mini D-Sub; Drives VGA compatible monitors
Devices Supported:	
	Resolutions:
Flat Panel (TFT)	1024 X 768, 800 X 600, 640 X 480
Flat Panel (DSTN)	800 X 600, 640 X 480
CRT Monitor	1280 X 1024, 1024 X 768, 800 X 600, 640 X 480
	Simultaneous operation of the internal and external displays is allowed, but with 640 X 480 resolution only. If you change resolution, you can only view the external display (internal display will "white out").
Test Set IO	
	25-pin D-Sub connector, available for external test set control
Aux IO	
	25-pin D-Sub connector, male, analog and digital IO
Handler IO	
	36-pin parallel I/O port; all input/output signals are default set to negative logic; can be reset to positive logic via GPIB command
GPIB	
	24-pin D-sub (Type D-24), female; compatible with IEEE-488.
Parallel Port (LPT1)	
	25-pin D-Sub miniature connector, female; provides connection to printers or any other parallel port peripherals
Serial Port (COM 1)	
	9-pin D-Sub, male; compatible with RS-232
USB Port	
	One port on front panel and five ports on rear panel. Universal Serial Bus jack, Type A configuration (4 contacts inline, contact 1 on left); female

Contact 1	Vcc: 4.75 to 5.25 VDC, 500 mA, maximum
Contact 2	-Data
Contact 3	+Data
Contact 4	Ground

Table 22. Rear Panel Information (Continued)

Description	**Supplemental Information**
LAN	
	10/100BaseT Ethernet, 8-pin configuration; auto selects between the two data rates
Line Power (A third-wire ground is required.	
Frequency	50/60/400 Hz
Voltage	120/240 VAC (Power supply is auto switching.)
Max	500 Watts

Note: Option H08 and Option H11 are not available with the N5230A

Table 23. Analyzer Environment and Dimensions

Description	**Supplemental Information**
General Environmental	
RFI/EMI Susceptibility	Defined by CISPR Pub. 11, Group 1, Class A, and IEC 50082-1
ESD	Minimize using static-safe work procedures and an antistatic bench mat
Dust	Minimize for optimum reliability
Operating Environment	
Temperature	0 °C to +40 °C Instrument powers up and displays no error messages within this temperature range (except for "source unleveled" error message that may occur at temperatures outside the specified performance temperature range of 25 +/- 5°C).
Error-Corrected Temperature Range	23°C ± 3°C with less than 1°C deviation from calibration temp.
Humidity	5% to 95% at +40 °C
Altitude	0 to 4500 m (14,760 ft.)
Non-Operating Storage Environment	

Temperature	-40 °C to +70 °C
Humidity	0% to 90% at +65 °C (non-condensing)
Altitude	0 to 15,240 m (50,000 ft.)

Cabinet Dimensions			
	Height	**Width**	**Depth**
Excluding front and rear panel hardware and feet	267 mm 10.5 in	426 mm 16.75 in	427 mm 16.8 in
As shipped - includes front panel connectors, rear panel bumpers, and feet.	280 mm 11.0 in	435 mm 17.1 in	470 mm 18.5 in
As shipped plus handles	280 mm 11.0 in	458 mm 18 in	501 mm 19.7 in
As shipped plus rack-mount flanges	280 mm 11.0 in	483 mm 19 in	470 mm 18.5 in
As shipped plus handles and rack-mount flanges	280 mm 11.0 in	483 mm 19 in	501 mm 19.70 in

Weight	
Net	
N5230A	24.9 kg (55 lb), nominal
Shipping	
N5230A	36.3 kg (80 lb), nominal

Measurement Throughput Summary

- Typical Cycle Time for Measurement Completion
- Cycle Time vs IF Bandwidth
- Cycle Time vs Number of Points
- Data Transfer Time

Table 24. Typical Cycle Time[a] (ms) for Measurement Completion

Description	Typical				
	Number of Points				
	201	**401**	**801**	**1601**	**16,001**
Start 8 GHz, Stop 18 GHz, 30 kHz IF bandwidth					
Uncorrected	97.5	102.7	103.8	108.2	683.9
2-Port cal	203.7	213.5	218.5	234.6	1504.3
Start 10 MHz, Stop 10 GHz, 30 kHz IF bandwidth					
Uncorrected	112.6	120.6	124.8	138.2	738.4
2-Port cal	232.8	251.8	265.2	304.3	1623.4
Start 10 MHz, Stop 20 GHz, 30 kHz IF bandwidth					
Uncorrected	146	199.3	210.9	217.2	753.9
2-Port cal	302.3	410.5	438.7	462.5	1660.5
Start 8 GHz, Stop 18 GHz, 50 kHz IF bandwidth					
Uncorrected	79.1	81	81.7	86.6	482
2-Port cal	164.5	170.3	175.3	193.5	1104.7
Start 10 MHz, Stop 10 GHz, 50 kHz IF bandwidth					
Uncorrected	96.8	101.7	108.8	122.2	524.6
2-Port cal	202.1	215.6	236.7	276.7	1198.8
Start 10 MHz, Stop 20 GHz, 50 kHz IF bandwidth					
Uncorrected	141.6	163.9	170.7	179.7	546.5
2-Port cal	293.6	341	360	389.5	1248.8

a Includes sweep time, retrace time and band-crossing time. Analyzer display turned off with DISPLAY:ENABLE OFF. Add 21 ms for display on. Data for one trace (S_{11}) measurement.

Table 25. *(Options 020/025, 120/125, only)* Cycle Time vs IF Bandwidth

Applies to the Preset condition (201 points, correction off) except for the following changes:

- CF = 10 GHz
- Span = 100 MHz
- Display off (add 21 ms for display on)

Description	Typical Performance

IF Bandwidth (Hz)	Cycle Time (ms)[a]	Trace Noise (dB rms)
600,000	7	0.0035
360,000	7	0.0026
280,000	7	0.0022
200,000	7	0.0021
150,000	7	0.0016
100,000	7	0.0012
70,000	7	0.0011
50,000	9	0.0009
30,000	11	0.0008
20,000	14	0.0006
15,000	17	0.0005
10,000	28	0.0004
7000	37	0.0004
5000	48	0.0003
3000	72	0.0003
2000	102	0.0002
1500	130	0.0001
1000	218	0.0001
700	294	0.0001
500	399	0.0001
300	636	0.0001
200	932	0
100	1826	0
30	6004	0

10	17903	0
1	178398	0

a Cycle time includes sweep and retrace time.

Table 26. ***(Options 220/225, 420/425, 520/525 only)*** Cycle Time vs IF Bandwidth

Applies to the Preset condition (201 points, correction off) except for the following changes:

- CF = 10 GHz
- Span = 100 MHz
- Display off (add 21 ms for display on)

Description	**Typical**	
IF Bandwidth (Hz)	**Cycle Time (ms)**[a]	**Cycle Time (ms)** **Option 080 enabled**
250,000	8.9	37.9
200,000	9.3	39.3
150,000	9.9	40.1
100,000	10.5	41.8
70,000	11.5	43.6
50,000	12.8	45.4
30,000	15.4	50
20,000	18.3	53.9
15,000	21	57.5
10,000	27	65.8
7000	34	75.4
5000	48.5	93
3000	72.8	124
2000	108.8	169
1500	126.8	187.1

1000	272.5
700	357.7
500	460
300	697.7
200	1003.5
150	1307.8
100	1917.6
30	6173.8
10	18214.8
1	181699.2

a Cycle time includes sweep and retrace time.

Table 27. ***(Options 020/025, 120/125, only)*** Cycle Time vs Number of Points

Applies to the Preset condition (correction off) except for the following changes:

- CF = 10 GHz
- Span = 100 MHz
- Display off (add 21 ms for display on)

Description	**Typical Performance**	
IF Bandwidth (Hz)	**Number of Points**	**Cycle Time (ms)**[a]
30,000	3	6.7
	11	7.4
	51	6.9
	101	7.8
	201	11.2
	401	18.3
	801	32.4
	1,601	59.4
	6,401	224.7
	16,001	556.9

100,000	3	6.7
	11	6.6
	51	6.8
	101	7
	201	7.5
	401	9
	801	13.5
	1,601	22.9
	6,401	75.3
	16,001	180.3
600,000	3	6.5
	11	6.6
	51	6.8
	101	6.9
	201	7.3
	401	8.1
	801	9.4
	1,601	12
	6,401	27.7
	16,001	59.3

a Cycle time includes sweep and retrace time.

Table 28. ***(Options 220/225, 420/425, 520/525 only)*** Cycle Time vs Number of Points

Applies to the Preset condition (correction off) except for the following changes:

- CF = 10 GHz
- Span = 100 MHz
- Display off (add 21 ms for display on)

Description	Typical	
IF Bandwidth (Hz)	Number of Points	Cycle Time (ms)[a]
30,000	3	8
	11	8
	51	9.38
	101	11.4
	201	15.5
	401	23.6
	801	39.9
	1,601	71.6
	6,401	265.4
	16,001	650.8
50,000	3	7.7
	11	7.7
	51	8.7
	101	10.1
	201	13
	401	18.6
	801	29.8
	1,601	52.3
	6,401	184.5
	16,001	448.8
250,000	101	8.7
	201	9.05
	401	10.85
	801	14.42
	1,601	21.63
	6,401	61.1
	16,001	147.7

a Cycle time includes sweep and retrace time.

Table 29. Data Transfer Time (ms)

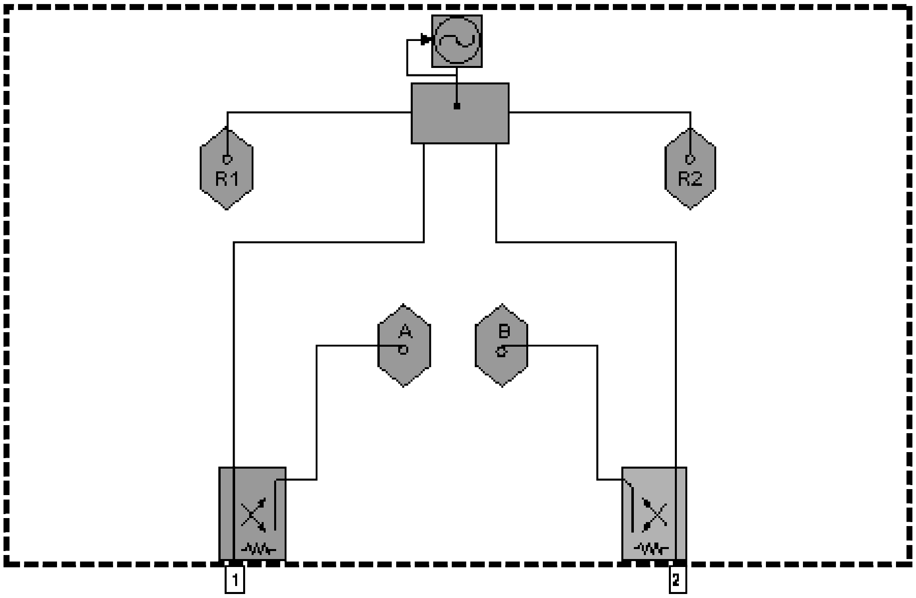
Description	Typical			
	Number of Points			
	201	401	1601	16,001
SCPI over GPIB (program executed on external PC)				
32-bit floating point	7	12	43	435
64-bit floating point	12	22	84	856
ASCII	64	124	489	5054
SCPI (program executed in the analyzer)				
32-bit floating point	1	2	3	30
64-bit floating point	2	2	4	40
ASCII	29	56	222	2220
COM (program executed in the analyzer)				
32-bit floating point	<0.4	0.4	0.5	1.9
Variant type	0.7	1	3	32
DCOM over LAN (program executed on external PC)				
32-bit floating point	<0.8	1	1.5	7.1
Variant type	1.8	2.7	8.5	80

Note: Specifications for Recall & Sweep Speed are not provided for the N5230A analyzers.

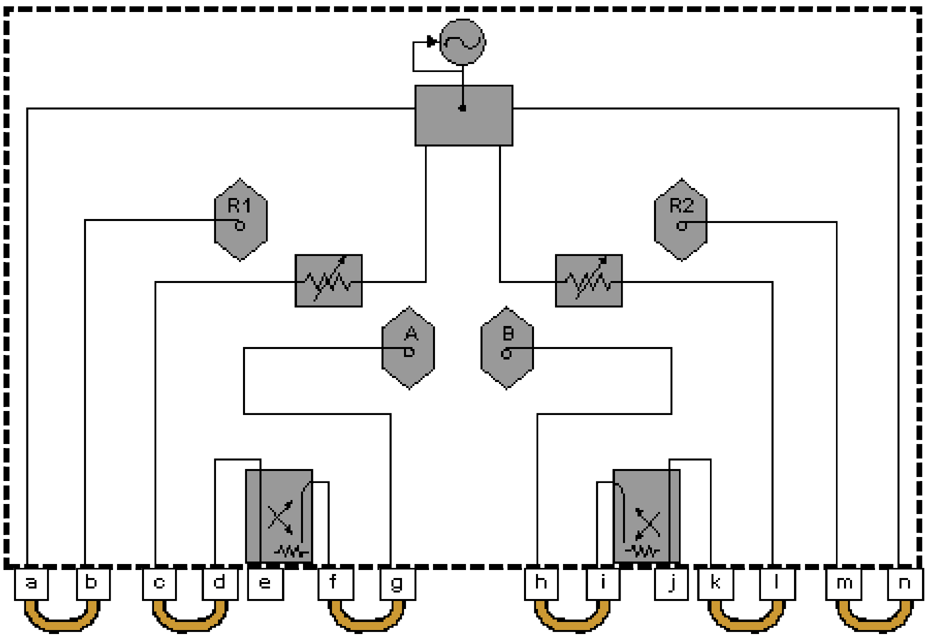
Tables 30 - 35 Front-panel Jumper Specs (Options 025, 125, 225, 425, 525)

Test Set Block Diagrams

N5230A Option 020, or 120, or 220, or 420, or 520 (Standard Test Set and Standard Power Range)



N5230A Option 025, or 125, or 225, or 425, or 525 (Configurable Test Set and Extended Power Range)



Item	Description	Item	Description
a	SOURCE OUT	**h**	RCVR B IN
b	RCVR R1 IN	**i**	CPLR ARM
c	SOURCE OUT	**j**	PORT 2
d	CPLR THRU	**k**	CPLR THRU
e	PORT 1	**l**	SOURCE OUT
f	CPLR ARM	**m**	RCVR R2 IN
g	RCVR A IN	**n**	SOURCE OUT

Typical System Performance for the N5250A

(Rev. 2004-08-02)

- This is a complete list of the N5250A network analyzer typical system performance.
- To view or print the .pdf version of this document, visit our web site at http://www.agilent.com, type 5988-9620EN in the Quick Search box, then click GO.

See Specs for other PNA models

Definitions

Typical : Expected performance of an average unit which does not include guardbands. It is not covered by the product warranty.

Standard: When referring to the analyzer, this includes no options unless noted otherwise.

Unlike the lengthy specifications documents for other PNA models, this document presents typical system performance for the following categories only:

- System Dynamic Range
- Test Port Power
- Noise Floor
- Test Port Damage Level
- Option H11 Rear Panel Connectors

Typical System Performance

Table 1. System Dynamic Range

Frequency	1.0 mm Test Port	1.85 mm PNA Test Port	Waveguide Port
10 MHz to 45 MHz	63 dB	65 dB	
45 MHz to 500 MHz	94 dB	97 dB	
500 MHz to 2 GHz	120 dB	123 dB	
2 GHz to 10 GHz	116 dB	123 dB	
10 GHz to 24 GHz	111 dB	121 dB	
24 GHz to 30 GHz	100 dB	112 dB	
30 GHz to 40 GHz	92 dB	107 dB	
40 GHz to 45 GHz	84 dB	101 dB	

45 GHz to 50 GHz	85 dB	103 dB	
50 GHz to 60 GHz	80 dB	100 dB	
60 GHz to 67 GHz	75 dB	95 dB	
67 GHz to 70 GHz	68 dB		82 dB
70 GHz to 75 GHz	74 dB		87 dB
75 GHz to 80 GHz	85 dB		98 dB
80 GHz to 100 GHz	89 dB		101 dB
100 GHz to 110 GHz	87 dB		98 dB

Table 2. Test Port Power

Frequency	1.0 mm Test Port (Std Configuration[a] or Opt 017[b])	1.85 mm PNA Port	Waveguide Port
10 MHz to 45 MHz	-8 dBm	-7 dBm	
45 MHz to 500 MHz	-3 dBm	-1 dBm	
500 MHz to 2 GHz	0 dBm	2 dBm	
2 GHz to 10 GHz	-2 dBm	2 dBm	
10 GHz to 24 GHz	-5 dBm	0 dBm	
24 GHz to 30 GHz	-7 dBm	0 dBm	
30 GHz to 40 GHz	-10 dBm	-1 dBm	
40 GHz to 45 GHz	-15 dBm	-5 dBm	
45 GHz to 50 GHz	-12 dBm	-1 dBm	
50 GHz to 60 GHz	-17 dBm	-4 dBm	
60 GHz to 67 GHz	-22 dBm	-8 dBm	
67 GHz to 70 GHz	-9 dBm		-2 dBm
70 GHz to 75 GHz	-7 dBm		0 dBm
75 GHz to 80 GHz	-6 dBm		+1 dBm
80 GHz to 100 GHz	-5 dBm		+1 dBm
100 GHz to 110 GHz	-8 dBm		-2 dBm

a Assumes a 30" cable from the PNA 1.85mm Test Port Out is used to provide the 10 MHz to 67 GHz source signal. The Standard configuration does not have a bias tee in the 1.0mm head.

b Assumes a 30" cable from the PNA Source Out bulkhead connector is used to provide the 10 MHz to 67 GHz source signal. Option 017 includes a bias tee in the 1.0mm head.

Table 3: Noise Floor

Frequency	1.0mm Test Port	1.85mm Test Port	Waveguide Port
10 MHz to 45 MHz	-71 dBm	-72 dBm	
45 MHz to 500 MHz	-97 dBm	-98 dBm	
500 MHz to 2 GHz	-120 dBm	-121 dBm	
2 GHz to 10 GHz	-118 dBm	-121 dBm	
10 GHz to 24 GHz	-116 dBm	-121 dBm	
24 GHz to 30 GHz	-107 dBm	-112 dBm	
30 GHz to 40 GHz	-102 dBm	-108 dBm	
40 GHz to 45 GHz	-99 dBm	-106 dBm	
45 GHz to 50 GHz	-97 dBm	-104 dBm	
50 GHz to 60 GHz	-97 dBm	-104 dBm	
60 GHz to 67 GHz	-92 dBm	-103 dBm	
67 GHz to 70 GHz	-77 dBm		-84 dBm
70 GHz to 75 GHz	-81 dBm		-87 dBm
75 GHz to 80 GHz	-91 dBm		-97 dBm
80 GHz to 100 GHz	-94 dBm		-100 dBm
100 GHz to 110 GHz	-95 dBm		-100 dBm

Table 4. Test Port Damage Level

Frequency	1.0mm Test Port	1.85mm Test Port	Waveguide Port
10 MHz to 110 GHz	27 dBm	27 dBm	27 dBm

Table 5 Option H08 & H11 Rear Panel Connectors (typical)

IF Connectors	A, R1, R2, B (BNC Connectors)
IF Connector Input Frequency	8 1/3 MHz
Nominal Input Impedance at IF Inputs	50 Ω
RF Damage Level to IF Connector Inputs	-20.0 dBm
DC Damage Level to IF Connector Inputs	25 volts
0.1 dB Compression Point at IF Inputs	-27.0 dBm
Pulse Input Connectors[1]	A, R1, R2, B (BNC Connectors)
Nominal Input Impedance at Pulse Inputs	1 Kohm
Minimum IF Gate Width	20 ns for less than 1 dB deviation from theoretical performance[2].
DC Damage Level to Pulse Connector Inputs	5.5 volts
Drive Voltage	TTL (0, +5.0) Volts
Rear Panel LO Power – Test Port Frequency (see 836x H11 Specs for Test Port Frequencies up to 67 GHz)	
67 GHz to 110 GHz[3]	-7 to –13 dBm
Rear Panel RF Power - Test Port Frequencies (see 836x H11 Specs for Test Port Frequencies up to 67 GHz)	
67 GHz to 76 GHz2[4]	-4 to –10 dBm
76 GHz to 96 GHz[4]	+1 to –5 dBm
96 GHz to 110 GHz[4]	+5 to –1 dBm

1 Pulse input connectors are operational only with Option H08 (Pulse Measurement Capability) enabled.

2 Based on deviation from signal reduction equation: Signal Reduction (dB) = $20\log_{10}$(Duty_cycle) = $20\log_{10}$(pulse_width/period). Measured at Pulse Repetition Frequency (PFR) of 1 MHz.

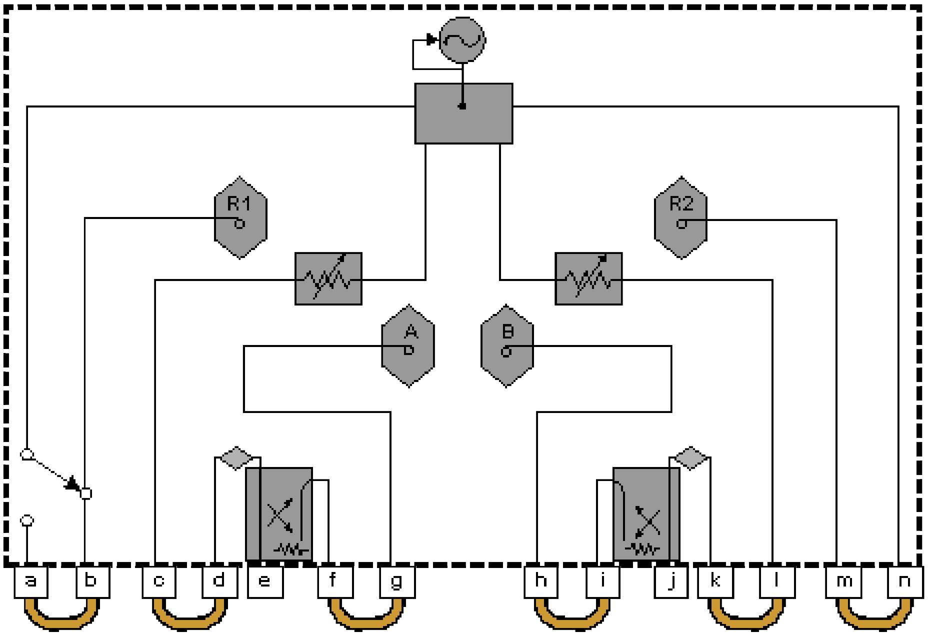
3 For rear panel LO port frequency, divide by 8

4 For rear panel RF port frequency, divide by 6

Note: Typical system performance for front panel jumpers is not provided for the N5250A.

Test Set Block Diagram

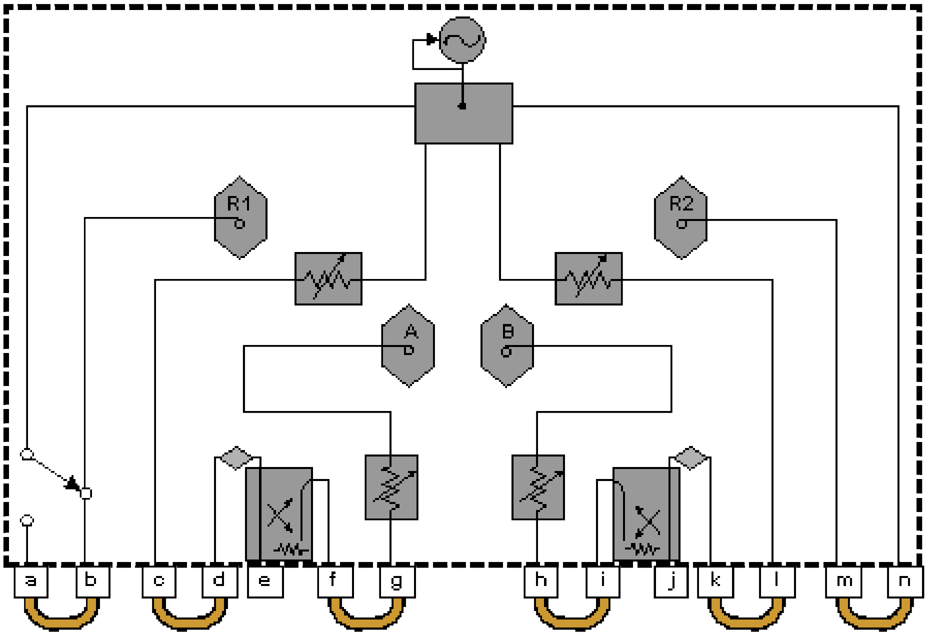
N5250A - Standard Network Analyzer



Item	Description	Item	Description
a	SOURCE OUT	h	RCVR B IN
b	RCVR R1 IN	i	CPLR ARM
c	SOURCE OUT	j	PORT 2
d	CPLR THRU	k	CPLR THRU
e	PORT 1	l	SOURCE OUT
f	CPLR ARM	m	RCVR R2 IN
g	RCVR A IN	n	SOURCE OUT

Test Set with Option 016 Block Diagram

N5250A - Option 016 Receiver Attenuators Network Analyzer



Item	Description	Item	Description
a	SOURCE OUT	h	RCVR B IN
b	RCVR R1 IN	i	CPLR ARM
c	SOURCE OUT	j	PORT 2
d	CPLR THRU	k	CPLR THRU
e	PORT 1	l	SOURCE OUT
f	CPLR ARM	m	RCVR R2 IN
g	RCVR A IN	n	SOURCE OUT

Glossary

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

12-Term Error Correction See Error Correction, 12-Term.

1-Port Device A device with a single connector or path to the device's circuitry. Examples include an oscillator and a load.

2-Port Calibration, Full See Error Correction, 12-Term.

2-Port Device A device with two connectors or other paths to the device's circuitry. Examples include filters, SAW devices, attenuators, matching pads, and amplifiers.

3-Term Error Correction See Error Correction, 3-Term.

A

Active Channel The highlighted channel affected by front panel functions.

Active Function Readout The area of a display screen where the active function and its state are displayed. The active function is the one that was completed by the last key selection or remote programming command.

Active Marker The marker on a trace that can be repositioned either by front panel controls or by programming commands.

Active Trace A trace that is being swept (updated) with incoming signal information.

ADC Analog to Digital Converter

Address The identification (represented by a name, label, or number) for a register, location in storage, or any other data source or destination. Examples are the location of a station in a communications network, or a device on the GP-IB.

ADM Add-Drop Multiplexer

Admittance (Y) The inverse of an impedance (i.e. the ratio of current to voltage). Complex admittances take the form Y = G + jB(t).

ALC Automatic Level Control. See Automatic Gain Control.

AM Amplitude Modulation

AM Group Delay A technique for the measurement of group delay through a device which utilizes an amplitude modulated (AM) source. Note: The actual delay of the modulation envelope is measured directly with an external scalar detector. Devices that distort the amplitude of a signal cannot be measured. These include amplifiers with automatic gain control (AGC) and devices subject to saturation or power limiting.

Amplitude Modulation The process, or result of the process, of varying the amplitude of a carrier signal. The resulting modulated carrier contains information that can be recovered by demodulation. See also Modulation.

Analog The general class of devices or circuits in which the output varies as a continuous function of the input.

Annotation The labeling of specific information on the display (such as frequency or power).

ANSI American National Standards Institute: A national membership organization (open to manufacturers, organizations, users, and communications carriers) that approves standards, accredits standards development groups and certificate programs, and represents and coordinates US interests in non-treaty and non-government standards bodies.

Aperture The frequency span of the network analyzer used for calculating group delay. The narrower the aperture, the finer the resolution of the group delay variations, but noise is reduced by increasing the aperture.

Array A set of numbers or characters that represents any given function.

ASCII American Standard Code for Information Interchange

Attenuation Denotes a reduction in signal amplitude. The difference between transmitted and received power due to loss through equipment, lines, or other transmission devices; usually expressed in decibels.

Attenuator An RF or microwave device used to reduce the power level of a signal by precise, incremental amounts over its entire frequency range.

Automatic Calibration System AutoCal: Feature offered on Rohde&Schwarz network analyzers.

Automatic Gain Control (AGC) A circuit used in amplifiers and other active devices to keep its RF power level constant as other parameters change, such as frequency. Synonym: Automatic Leveling Control (ALC)

Autoscale An analyzer feature that evaluates waveforms and adjusts controls to stable and enhance the display.

AUX Auxiliary; refers to rear-panel input connector.

Averaging A noise reduction technique that computes each data point based on consecutive sweeps and weighted by a user-specified averaging factor. Each new sweep is averaged into the trace until the total number of sweeps is equal to the averaging factor.

B

B/R The ratio of data sampled at B to the data sampled at R.

Band Pass A range of frequencies that are passed through a device, such as a filter. Frequencies not within the band pass are limited or attenuated. See also Cutoff Frequency.

Bandwidth (BW) The difference between the frequencies of a continuous frequency band within which performance of a device falls within specifications.

Bandwidth Limit The condition prevailing when the system bandwidth is exceeded and signal distortion occurs beyond specifications.

Bandwidth Selectivity A measure of a filter's ability to resolve signals unequal in amplitude. It is the ratio of the 60 dB bandwidth to the 3 dB bandwidth for a given resolution filter (IF). Bandwidth selectivity tells us how steep the filter skirts are. Bandwidth selectivity is sometimes called shape factor.

Binary A method of representing numbers in a scale of two (on or off, high-level or low-level, one or zero). A compact, fast format used to transfer information to and from the analyzer.

BMP Bit-Mapped

Brightness See Color Brightness.

Broadband Device A device that operates over a very wide frequency range and exhibits only small variations in response over that range.

Buffer A storage device used when transmitting information to compensate for a difference in the rate of flow of information between two devices.

Burst Carrier A carrier that is periodically turned off and on. A burst carrier may or may not be modulated.

BUS Basic Utility System

Bus One or more conductors used as a path to deliver transmitted information from any of several sources to any of several destinations.

BW Bandwidth

Byte Eight bits of data representing one character processed as a unit.

C

CAD Computer Aided Design

CAE Computer Aided Engineering

Calibration In HP instrumentation, the process of periodically (usually annually) verifying an instrument is performing to specifications. A calibration certificate is awarded after verification.

In network analyzers, the process of removing systematic errors from measurements. See Error Correction.

Calibration Kit Hardware and software required to perform error correction on a network analyzer for a specific measurement and/or test set.

Calibration, 2-Port See Error Correction,12-Term.

Calibration, Blackburn Calibrations of transmission path with corrected source match involving 15 calibration terms. Synonym: 15-term error correction

Calibration, Frequency Response The simplest error correction procedure to perform, but only corrects for a few of the twelve possible systematic error terms. Frequency response corrections can be made for reflection measurements, transmission measurements, and isolation measurements.

Calibration, Interpolation A user selectable network analyzer feature that calculates (interpolates) new error correction terms from existing terms when there is a change in network analyzer parameters, such as IF bandwidth, power, or sweep time. The resulting error correction is not as accurate as completing a full 2-port calibration.

Calibration, Port Extension See Port Extension.

Calibration, Reference Plane See Reference Plane.

Calibration, Set Z Sets the system impedance, usually 50 or 75 ohms.

Calibration, SOLT A calibration using four known standards: Short-Open-Load-Through. Also known as a full two-port calibration and 12-term error correction. See also Error Correction.

Calibration, TRL and LRM A calibration used in environments where the DUT cannot be connected directly to the network analyzer ports, (MMIC, microstrip, beam-lead diodes etc.). Thru-Reflect-Line (TRL) and M (Match) standards are fabricated and used because known high-quality standards are not readily available. The requirements for characterizing these standards are less stringent, but the calibration is not as accurate as the traditional full two-port calibration using S-O-L-T standards. The terms are used interchangeably (TRL, LRL, LRM etc.) but they all refer to the same basic calibration method.

Characteristic Impedance The impedance looking into the end of an infinitely long lossless transmission line.

Color Brightness A measure of the intensity (brightness) of a color.

Command A set of instructions that are translated into instrument actions. The actions are usually made up of individual steps that together can execute an operation.

Continuous Sweep Mode The analyzer condition where traces are automatically updated each time trigger conditions are met.

Controller A device capable of specifying the talker and listeners for an information transfer. An external computer connected to an instrument to control its operation.

Corrected Measurements made after performing error correction.

Coupler See Directional Coupler.

CPU Central Processing Unit

Crosstalk The occurrence of a signal at one port of a device being affected by a signal in any other path. Isolation is the measurement of crosstalk.

Cursor An electronically generated pointer that moves across the display to manipulate controls.

Cutoff Frequency In filters, the frequency at which attenuation is 3dB below the band pass signal level, known as the 3dB points.

CW Continuous wave: A single frequency (rather than a swept frequency).

D

DAC Digital to Analog Converter

dB Decibel: a relative unit of measure. The ratio in dB is given by: 10 $\log_{10}$ (P_1/P_2) where P_1 and P_2 are the measured powers. The dB is preferred instead of arithmetic ratios or percentages because when components are connected in series, their effect on power, expressed in dB, may be arithmetically added and subtracted. For example, if a 3dB attenuator is connected to a 10dB amplifier, the net gain of the two components is (-3dB + 10dB = +7dB).

dBm Absolute unit of measure in decibels: 0dBm = 1 mW. The conventions of the dB (adding and subtracting) continue to apply.

DBMS Database Management System

DC Direct Current

Default A known set of conditions used in the absence of user-defined conditions.

Delay See Group Delay.

Demodulation The process of recovering from a modulated carrier, information in the form of a signal having essentially the same characteristics as the original modulating signal. Recovery of the modulating signal accomplished by signal detection.

Detection The process of demodulating signal carriers. There are two basic ways of providing signal detection in network analyzers: Diode detectors (used in broadband applications) and heterodyning, (used in narrowband applications).

Detector, Diode A device used to convert a RF signal to a proportional DC level. If the signal is amplitude modulated, the diode strips the RF carrier signal from the modulation. Many sources used with scalar analyzers are amplitude modulated with a 27.778 kHz signal and then detected in the network analyzer. Phase information on the signal carrier is lost in diode detection.

Deviation from Linear Phase Linear phase refers to the nature of the phase shift of a signal through a device. The phase is linear if a plot of phase shift versus frequency is a straight line using linear scales. Deviation from linear phase causes signal distortion.

Digital Pertaining to the class of devices or circuits in which the output varies in discrete steps.

Digital Demodulation Describes a technique of extracting the information used to modulate a signal. Digital signal processing algorithms are used on the signal after it has been converted from an analog to a digital form (digitized).

Dimension To specify the size of an array. The number of array rows or columns.

Directivity In a 3-port directional coupler, the ratio of the power present at the auxiliary port when the signal is traveling in the forward direction to the power present at the auxiliary port when the same signal is traveling in the reverse direction.

Directional Coupler A 3-port device typically used for separately sampling the backward (reflected) wave in a transmission line.

Disk A circular, magnetic storage medium.

Display Noun: See Screen.

Verb: To show annotation and measurement data on the display.

Display Detector Mode The manner in which analog, video information is processed prior to being digitized and stored in memory.

Display Dynamic Accuracy The amplitude uncertainty, usually in dB, over the display dynamic range.

Display Dynamic Range The amplitude range, in dB, over which the display dynamic accuracy applies.

Display Formats Graphical formats for displaying measurement data. These include single channel, overlay (multiple traces on one graticule), split (each trace on separate graticules).

Display Modes The ways in which measurement data can be presented graphically. On a network analyzer, the choices are Cartesian/rectilinear (XY plot with log or linear magnitude, phase, group delay, SWR, real and imaginary, and dBV, dBmV and dBuV), polar (magnitude and angle), magnitude and phase, and Smith chart. Not all display modes are available on all network analyzers. In addition, displays can present this information in various combinations of traces. Common modes are dual, (the ability to display more than one trace, usually over the same frequency range), and alternate, (the ability to display more than one trace, each with different frequency range and type).

Display Phase Dynamic Accuracy The phase measurement uncertainty, usually in degrees, for measurements whose units are in degrees.

Display Points The total number of measurement points made in a single measurement. The points can be in units of frequency, power, or time. The number of points often dictates measurement speed, resolution, and aperture.

Display Trace Noise, Magnitude The amplitude uncertainty of the trace, in dB, due to random noise in the test system.

Display Trace Noise, Phase The phase uncertainty of the trace, in degrees, due to random noise in the test system.

Display Type The type of display screen built into the analyzer. Data can be displayed as a raster drawing (a computer-like dot map) or as a vector drawing (lines drawn on the display). Color and display standard can also be specified as monochrome (single color), or color (two or more colors). The format standard may also be specified, such as VGA or SVGA, for IBM-compatible personal computers.

Distortion Deterioration of a signal's quality due to the nonlinear characteristics of a device or system transfer function. Distortion is measured as a combination of the changes in amplitude, frequency and phase of signal at the output of a device or system as compared to the signal at the input.

Drift The slow change in signal frequency.

DSP Digital Signal Processing

DUT Device Under Test

DVM Digital Volt Meter

Dynamic Range In a receiver, the range of signal levels, from minimum to maximum, that can be reliably measured simultaneously. Dynamic range allows small signals to be measured in the presence of large signals. Source power and receiver compression usually limits the maximum boundary to dynamic range. Receiver residual responses and noise floor usually limit the minimum power boundary.

E

ECal See Electronic Calibration.

Electrical Delay A simulated variable length of lossless transmission line, added to or subtracted from a receiver input, to compensate for interconnecting cables. The firmware equivalent of mechanical or analog "line stretchers" in other network analyzers.

Electronic Calibration (ECal) A calibration system for electronic calibration of RF and microwave vector network analyzers. The electronic calibration system creates a twelve-term, two-port error model and then provides a confidence check of the calibration. The Ecal system consists of a repeatable, variable-impedance, solid-state calibration standard and a mainframe control unit which interfaces with the 8510, 8720 series, and the 8753 network analyzers or a USB module which interfaces with the PNA series network analyzers.

EMC Electro-Magnetic Compatibility

EMI Electro-Magnetic Interference: Unintentional interfering signals generated within or external to electronic equipment. Typical sources could be power-line transients, noise from switching-type power supplies and/or spurious radiation from oscillators. EMI is suppressed with power-line filtering, shielding, etc.

Engage To activate a function.

Enter The process of inputting information.

EPROM Electronically Programmable, Read-Only Memory

Error Correction In network analyzers, a process that removes or reduces systematic (repeatable) measurement errors by measuring known standards from a calibration kit. Synonym: measurement calibration

Error Correction, 3-Term Used to remove systematic measurement errors on a device with one port, such as a load.

Error Correction, 12-Term Correction for a two port device using six parameters:
Directivity
Source match
Load match
Reflection frequency response
Transmission frequency response
Isolation

To completely characterize a two-port device, these six parameters must be characterized in the forward and reverse directions, making a total of 12 terms. The user usually has the option of omitting isolation from the correction process. Synonym: Full two-port error correction

Error Correction, 1-Port Corrects a test set for port 1 or port 2 directivity, frequency response, and source match errors. The process requires three known standard terminations, for example, open, short, and load.

Error Message A message on a display that indicates an error condition. Missing or failed hardware, improper user operation, or other conditions that require additional attention can cause an error condition. Generally, the requested action or operation cannot be completed until the condition is resolved.

ESD Electro Static Discharge

Ethernet A network that adheres to the IEEE 802.3 Local Area Network standard.

Ethernet address A hexadecimal number which is used to identify a machine on a network. Each analyzer is assigned a unique Ethernet address at the factory and it is stored in the analyzer's ROM.

External trigger signal A TTL signal that is input to an analyzer and initiates a measurement sweep or similar event, making the measurements synchronous with the external triggering source.

F

Filter A passive device that allows some frequencies to pass and attenuates others, depending on the type and specifications. A high-pass filter passes frequencies above the cutoff frequency, a low-pass filter passes frequencies below the cutoff frequency, and a band-pass filter passes frequencies between two specific frequencies.

Firmware An assembly made up of hardware and instruction code. The hardware and instruction code is integrated and forms a functional set that cannot be altered during normal operation. The instruction code, permanently installed in the circuitry of the instrument, is classified as ROM (read only memory). The firmware determines the operating characteristics of the instrument or equipment.

Flatness The amplitude and phase response of a device under test (DUT), a signal source, a receiver, or a combination of these. See also Frequency Response.

FM Frequency Modulation

Frequency The number of periodic oscillations, vibrations, or waves per unit of time, usually expressed in cycles per second, or Hertz (Hz).

Frequency Accuracy The uncertainty with which the frequency of a signal or spectral component is indicated, either in an absolute sense or relative to another signal or spectral component. Absolute and relative frequency accuracies are specified independently.

Frequency Range The range of frequencies over which a device or instrument performance is specified.

Frequency Resolution The ability of a network analyzer to measure device characteristics at closely spaced frequencies and display them separately. Resolution of equal amplitude responses is determined by IF bandwidth. Resolution of unequal amplitude responses is determined by IF bandwidth and bandwidth selectivity.

Frequency Response The peak-to-peak variation in the displayed amplitude response over a specified center frequency range. Frequency response is typically specified in terms of dB, relative to the value midway between the extremes.

Frequency Span The magnitude of the displayed frequency component. Span is represented by the horizontal axis of the display. Generally, frequency span is given as the total span across the full display. Some analyzers represent frequency span (scan width) as a per-division value.

Frequency Stability The ability of a frequency component to remain unchanged in frequency or amplitude over short and long-term periods of time. Stability refers to an oscillator's ability to remain fixed at a particular frequency over time.

Front Panel Key Keys that are located on the front panel of an instrument. The key labels identify the function the key activities. Numeric keys and step keys are two examples of front panel keys.

Full 2-Port Calibration See Error Correction, 12-Term.

Function The action or purpose that a specific item is intended to perform or serve. The network analyzer contains functions that can be executed via front panel key selections, or through programming commands. The characteristics of these functions are determined by the firmware in the instrument. In some cases, a DLP (downloadable program) execution of a function allows you to execute the function from front panel key selections.

Fundamental Frequency In any waveform, the lowest frequency component; all other components are harmonics. A pure sinusoid has only one component, the fundamental.

G

Gb Gigabit

GB Gigabyte

GHz Gigahertz

GIF Graphics Interchange Format - Standard graphic format to store bitmapped graphics files.

Giga Prefix for one billion.

GP I/0 General Purpose Input / Output; a connector usually on the back of an instrument that allows communication with other test equipment, external test sets, switches, and computers that enable the instrument to be triggered or to trigger external equipment. An example is a foot switch that continues or cycles a measurement, allowing the operator to use both hands on the test hardware.

GPIB General Purpose Interface Bus - IEEE 488 bus is interconnect bus and protocol, allows linking of instruments and computer.

Graticule (or Grid) Enclosed area where waveform is displayed on instrument. Tick marks, on frame or axis, are a scaling aid for making visual measurements.

Group Delay A measure of the transit time of a signal through a DUT versus frequency. Group delay can be calculated by differentiating the DUT's insertion-phase response with respect to frequency. See also AM Group Delay and Deviation from Linear Phase.

GUI Graphical User Interface

H

Hardcopy Paper copy of data.

Hardkey A front-panel key, which engages a single analyzer function or presents a single menu of softkeys.

Horizontal Reference See Reference Level.

Horizontal Resolution The analyzer's ability to take closely spaced horizontal data points over the full sweep.

Host Computer A computer or device on a network that provides end users with services such as computation and database access and that usually performs network control functions.

Host Name A unique name that is used to identify each host machine on a network. The host name is directly linked to, and can usually be used in place of, the IP address. The user or the system administrator usually creates the host name.

HP Hewlett-Packard Company

HPGL Hewlett-Packard Graphics Language

HP-IB Hewlett-Packard Interface Bus. A parallel interface that allows "daisy chaining" of more than one device to a port on a computer or instrument. Interface protocol is defined in IEEE 488.2; equivalent to the industry standard GPIB.

HTTP HyperText Transfer Protocol: Used to carry World Wide Web (WWW) traffic.

Hue The dimension of color referred to a scale of perceptions ranging from red through yellow, green, and blue, and back to red. A particular gradation of color, tint, shade.

I

I/O Input/Output

I/O Path Input/Output Path

IEEE Institute of Electrical and Electronic Engineers

IF Intermediate Frequency: the frequency at which a signal is processed after mixing.

Impedance The ratio of voltage to current at a port of a circuit, expressed in ohms.

Initialize The process that assigns information locations to a disk to prepare the magnetic media to accept files.

Input A path intended for putting a signal into an instrument.

Most network analyzers have either 3 (labeled A, B, and R) or 4 inputs (labeled A, B, R1, and R2). Inputs are not the same as channels.

Input Attenuator An attenuator between the input connector and the first mixer of a spectrum analyzer (also called an RF attenuator). The input attenuator is used to adjust the signal level incident to the first mixer, and to prevent gain compression due to high-level or broadband signals. It is also used to set the dynamic range by controlling the degree of internally-generated distortion. For some analyzers, changing the input attenuator settings changes the vertical position of the signal on the display, which then changes the reference level accordingly. In Agilent microprocessor-controlled analyzers, the IF gain is changed to compensate for changes in input attenuator settings. Because of this, the signals remain stationary on the display, and the reference level is not changed.

Insertion Loss The difference between the power measured before and after the insertion of a device. The attenuation between the input and output of a device.

Intensity Brightness; emitting or reflecting light; luminosity.

Interface A connection that allows a common communication link between two or more instruments.

Intermodulation Distortion Undesired frequency components resulting from the interaction of two or more spectral components passing through a device having nonlinear behavior, such as a mixer or an amplifier. The undesired components are related to the fundamental components by sums and differences of the fundamentals and various harmonics. The algorithm is: f1 ±f2, 2xf1 ±f2, 2xf2±f1, 3xf1 ±2x f2, and so on.

Internet The connection of two or more distinct networks. Often a gateway or router is used to make the connection.

Interpolate To determine a value of a signal between to adjacent points by a procedure or algorithm.

IP Internet Protocol

IP Address Internet protocol address: a unique number that is assigned to each device which is to be connected to a TCP/IP network. Before using an analyzer on a network, your network administrator will need to assign an IP address. An IP address consists of a 32-bit value presented in decimal dot notation: 4 octets (bytes) separated by a dot.

ISDN Integrated Services Digital Network: A standard digital service capability that features one or more circuit-switched communication channels capable of carrying digital voice, data, or image signals, a packet-switched channel for out-of-band signaling and control. In addition, ISDN provides a collection of standard and optional features that support information productivity for the user, providing higher-speed Internet access than analog systems.

ISO International Standards Organization

Isolation A specification or measure of the immunity that one signal has to being affected by another adjacent signal. The occurrence is known as crosstalk.

Isolator An RF device used for providing isolation between paths and components. Made from a 3-port circulator, the third port being terminated in a 50ohm load.

J

K

Kilo Prefix for one thousand.

KB Kilobyte

Kb/s Kilobytes per second

L

LAN Local Area Network

LANS Local Area Network System

LCD Liquid Crystal Display

LED Light Emitting Diode

LIF Logical Interchange Format (used for older HP disk drives/computers)

Limit Lines Lines input by the user that overlay the analyzer's measurement data to allow automatic detection of data that is out of the acceptable range. Pass/Fail annotation, audio alarms, or electronic output can be triggered to notify the operator or on-line computer program of the over-limit condition.

Limit-Line File The user-memory file that contains the limit-line table entries.

Limit-Line Table The line segments of a limit line are stored in the limit-line table. The table can be recalled to edit the line segments, then restored in the limit-line file.

Linear Device A device in which the output is continuously proportional to the input.

LO Local Oscillator. In a superheterodyne system, the LO is mixed with the received signal to produce a sum or difference equal to the intermediate frequency (IF) of the receiver.

LO Feedthrough The response that in a superheterodyne system when the first local oscillator frequency is equal to the first IF.

Load A one port microwave device used to terminate a path in its characteristic impedance.

Load Match A measure of how close the device's terminating load impedance is to the ideal transmission line impedance. Match is usually measured as return loss or standing wave ratio (SWR) of the load.

Local Lock Out A condition or command that prevents analyzer front-panel entries (and disables the Local key).

Local Operation To operate manually from the front panel.

Log Logarithm

Log Display The display mode in which vertical deflection is a logarithmic function of the input signal amplitude. Log display is also called logarithmic display. The display calibration is set by selecting the value of the reference level position and scale factor in dB per division.

LRM Line-Reflect-Match. See Calibration, TRL, and LRM.

M

Magnitude The amplitude of a signal measured in its characteristic impedance without regard to phase. See also Scalar.

Marker A graphical symbol along a display trace that is annotated with measurement characteristics of that specific data point.

Marker Functions Mathematical or statistical computation on the data of one or more markers to provide the operator more information. For example, the marker delta function calculates and displays the difference between two markers.

Maximum Input Level The maximum signal power that may be safely applied to the input of an analyzer. The maximum input level is typically 1 W (+30 dBm) for Agilent spectrum analyzers.

MB Megabyte

Measurement Uncertainty The quantified amount of error in a measurement situation. Calibrations are intended to reduce the amount of uncertainty. The following are sources of measurement errors that lead to uncertainty:

- Systematic errors (imperfections in calibration standards, connectors, cables, and instrumentation)
- Random errors (noise, connector repeatability)
- Drift (source and instrumentation)

Mega Prefix for one million.

Memory A storage medium, device, or recording medium into which data can be stored and held until some later time, and from which the entire original data may be retrieved.

Memory Card A small memory device shaped like a credit card that can store data or programs.

Menu The analyzer functions that appear on the display and are selected by pressing front panel keys. These selections may invoke a series of other related functions that establish groups called menus.

MHz Megahertz

milli Prefix for one-thousandth.

Modem Modulator/Demodulator

Modulation The process, or the result of the process, of varying a characteristic of a carrier signal with an information-bearing signal, causing the carrier to contain the information. See AM and FM.

Monitor Any external display.

Monochrome Having only one color (chromaticity).

ms Millisecond

mW Milliwatt: one thousandth of a watt

Multisync A type of monitor that can synchronize its horizontal sweep to various frequencies within a specified range.

N

Narrowband In network analysis, the frequency resolution of the analyzer's receiver that is sufficiently narrow to resolve the magnitude and phase characteristics of narrowband devices. The reduced receiver bandwidth usually decreases the noise floor of the receiver, providing more measurement amplitude range.

Narrowband Device A device whose transfer characteristics are intended to operate over a very narrow frequency range and are designed to provide well-defined amplitude responses in that range, such as a band pass filter.

Network Analysis The characterization of a device, circuit, or system derived by comparing a signal input going into the device to a signal or signals coming out from the device.

NIST National Institute of Standards and Technology

Nit The unit of luminance (photometric brightness) equal to one candela per square meter.

Noise Random variations of unwanted or disturbing energy in a communications system from man-made and natural sources that affects or distorts the information carried by the signal. See also Signal-to-Noise Ratio.

Noise Figure (F): For a two-port device, a measure of how the noise generated inside the device degrades the signal-to-noise ratio of a signal passing through the device at 290 degrees, usually expressed in dB.

Noise Floor The analyzer's internal displayed noise. The noise level often limits how small a signal magnitude can be measured. In network analysis, noise floor is measured with the test ports terminated in loads, full two-port error correction, 10 Hz IF bandwidth, maximum test port power, and no averaging during the test.

Non-Insertable Devices In measurement calibration, a device that cannot be substituted for a Zero-Length Through Path. It has the same type and sex connectors on each port, or a different type of connector on each port.

Nonvolatile Memory Memory data that is retained in the absence of an ac power source. This memory is typically retained with a battery. Refer also to battery-backed RAM.

Normalize To subtract one trace from another to eliminate calibration data errors or to obtain relative information.

O

Offset To move or set off a determined amount. Used in instruments for offsetting frequencies, limits, delay, loss, impedance, etc.

Output Attenuation The ability to attenuate the signal, the source, in order to control its power level.

P

PC Personal Computer

PDF Portable Document Format (used on the Web)

Parser, Command Reads program messages from the input queue of a device in the order they were received from the controller. The parser determines what actions the analyzer should take. One of the most important functions of the command parser is to determine the position of a program message in the analyzer SCPI command tree. When the command parser is reset, the next element it receives is expected to arise from the base of the analyzer command tree.

Peak Search A function on an analyzer that searches for the largest response and places a marker on it.

Phase The fractional part of a cycle through which an oscillation has advanced, measured from an arbitrary starting point; usually measured in radians or degrees. In network analysis, the phase response of the device under test is the change in phase as a function of frequency between the input stimulus and the measured response.

Port The physical input or output connection of an instrument or device.

Port Extension Redefining the reference plane to other than that established at calibration. A new reference plane is defined in seconds of delay from the test set port.

Positive Peak The maximum, instantaneous value of an incoming signal.

Postscript (.ps files) Stores bitmapped graphics files in an encapsulated format for direct use by postscript printers.

Power, Max Input The upper limit to input power for which the specifications apply. Some specifications may have different levels of maximum inputs. For example, compression power maximum is usually higher than the harmonic distortion maximum.

Power, Safe Input The input power, usually in dBm, allowed without damaging the instrument.

Preset A pre-defined instrument state (that also runs an analyzer self-test). The action of pushing the Preset key.

Protocol A set of conventions that specify how information will be formatted and transmitted on a network, and how machines on a network will communicate.

Q

Q or Q Factor The ratio of energy stored to energy lost in a resonant circuit. High Q indicates a sharp resonance response over frequency.

Query Any analyzer programming command having the distinct function of returning a response. These commands may end with a question mark (?). Queried commands return information to the computer.

R

r + jx Expression for complex impedance, where r represents the resistive portion and x represents the reactive portion.

R Channel Reference Channel

RAM Random Access Memory, or read-write memory: A storage area allowing access to any of its storage locations. Data can be written to or retrieved from RAM, but data storage is only temporary. When the power is removed, the information disappears. User-generated information appearing on a display is RAM data.

ROM Read Only Memory: A storage area that can be read only; it cannot be written to or altered by the user. In instruments, the storage area that contains the "brains" or operational programming; the firmware.

Receiver A circuit or system designed for the reception and/or measurement of signals in a specified frequency spectrum.

Receiver Dynamic Range See Dynamic Range.

Reference Level An instrument function that allows the user to set the amplitude value at the reference position. On network analyzers, the reference position is also selectable. On some spectrum analyzers, the reference position is fixed at the top of the display.

Reference Plane The electrical location at which a network analyzer assumes the system connectors and fixturing ends and the DUT begins. The reference plane is set by using calibration standards with known electrical length. The closer the reference plane is to the device under test (DUT), the better the characterization of the device because of the elimination of test system uncertainties.

Reflection The phenomenon in which a traveling wave strikes a discontinuity and returns to the original medium.

Reflection Coefficient The ratio of the reflected voltage to the incident voltage into a transmission line or circuit. If a transmission line is terminated in its characteristic impedance, the reflection coefficient is zero. If the line is shorted or open the coefficient is 1. See also Return Loss and SWR.

Reflection Measurements Measurements that characterize the input and /or output behavior of the device under test (DUT). Measured as the ratio of the reflected signal to the incident signal as a function of frequency. Parameters are called return loss, reflection coefficient, impedance, and standing wave ratio (SWR), all as a function of frequency. See also S-Parameters.

Remote A mode of operation where another device (or computer) controls an instrument via the HP-IB. In this mode, the instrument front panel keys are disabled. Front panel operation is called local operation.

Remote Programming The automatic operation of an instrument by a computer, usually through a HP-IB, LAN, or RS-232 link.

Resolution The ability of a receiver to resolve two signals.

Resolution Bandwidth The ability of a spectrum analyzer to display adjacent responses discretely (Hertz, Hertz decibel down). This term is used to identify the width of the resolution bandwidth filter of a spectrum analyzer at some level below the minimum insertion loss point (maximum deflection' point on the display). Typically, it is the 3 dB resolution bandwidth that is specified, but in some cases the 6 dB resolution bandwidth is specified.

Return Loss The amount of dB that the reflected signal is below the incident signal. If zero signal is reflected, the impedance of the device is equal to the characteristic impedance of the transmission system, and return loss is infinite. If the entire incident signal is reflected, the return loss is zero. See also S-Parameters, Reflection Coefficient, and SWR.

Reverse Measurement The measurement of a device from output to input.

RF Radio Frequency (from approximately 50 kHz to approximately 3 GHz). Usually referred to whenever a signal is radiated through the air.

ROM Read Only Memory

S

S/N Signal-to-Noise Ratio

Sampler An electronic component that captures the signal level and phase across a known impedance at a uniform rate. In Network Analyzers, this sampling rate must be sufficiently high and precisely timed to make accurate measurements. Network analyzers typically have three or four samplers or mixers.

Sampler Bounce The leakage or crosstalk between a network analyzer's samplers. Delay in this crosstalk caused by leakage transmission propagation, give the interference its "bounce" appearance. Sampler bounce causes an increase in the noise level of the affected channel, reducing the sensitivity of the analyzer.

Saturation The degree of color purity, on a scale from white to pure color.

Scalar A quantity that has magnitude but no phase. A network analyzer capable of measuring only magnitude.

Scale Factor The display vertical axis calibration in terms of units per division.

SCPI Standard Commands for Programmable Instruments

Screen The physical surface of the CRT or flat panel upon which the measurement results, setup information, softkey definitions, and other instrument communication is presented.

Self-Test A group of tests performed at power-up (or at preset) that verify proper instrument operation.

Sensitivity The minimum input signal required to produce a specified output signal having a specified signal-to-noise ratio, or other specified criteria.

On a spectrum analyzer, the level of the smallest sinusoid that can be observed, usually under optimized conditions of minimum resolution bandwidth, 0 dB input attenuation, and minimum video bandwidth.

The normalized change in YIG component's center frequency resulting from a change in tuning coil current, specified in MHz/mA.

Serial Prefix The five-character prefix that begins an instrument serial number; used to represent versions of firmware or hardware changes that have occurred.

Server A device that is configured to provide a service to other devices on a network, such as shared access to a file system or printer.

Signal-to-Noise Ratio SNR: The ratio of the amplitude of the desired signal to the amplitude of noise signals, usually expressed in dB and in terms of peak values for impulse noise and root-mean-square values for random noise.

Single Sweep Mode The spectrum analyzer sweeps once when trigger conditions are met. Each sweep is initiated by pressing an appropriate front panel key, or by sending a programming command.

Small Signal Gain Compression A situation when the input signal's measured amplitude is less than its actual level due to overloading of the network analyzer's input mixer; the analyzer is operating nonlinearly. For broadband analyzer detectors, a signal other than the one under test can put the analyzer into this gain compressed mode, thereby making even lower level signals appear at a lower level than actual. The broadband mode measures all the power incident to the analyzer, not just the signals at the frequency of interest.

Smith Chart A graphical mapping of the complex reflection coefficient into normalized complex impedance. Circles on the chart represent constant resistance and radiating lines orthogonal to the circles represent constant reactance. The center of the chart represents the characteristic impedance of the transmission system. Any point on the chart defines a single complex impedance. A line on the chart represents changing impedance over frequency.

SOLT Short-Open-Load-Through calibration. See also Calibration, SOLT.

Source A device that supplies signal power; a sweep oscillator or synthesized sweeper.

Source Amplitude Accuracy The amplitude uncertainty, in dB, of the source power readout.

Source Amplitude Flatness The amplitude flatness, in dB, of the source power over the frequency range specified.

Source Frequency Resolution The smallest unit of frequency which can be set and/or measured, in Hz.

Source Frequency Time Base Accuracy A measure of the analyzer's frequency stability measured in parts per million (ppm. or 1 part in 10E6). For example, a stability of ±5.0 ppm means that an analyzer will measure 1 MHz to an accuracy of ±5 x 10-6 x 10E6 Hz = +5 Hz.

Source Frequency Time Base Stability A measure of the analyzer's time base accuracy over time and temperature. Typically the time base accuracy will be specified for 1 year. A typical temperature frequency stability is ±10 ppm for 250 C± 50 C.

Source Harmonics The level of harmonics generated by the analyzer's signal source, in dBc from the fundamental.

Source Match A measure of how close the signal source impedance is to the ideal transmission line impedance of the test system. Match is usually measured as return loss or standing wave ratio (SWR) of the source.

Span The stop frequency minus the start frequency. The span setting determines the horizontal-axis scale of the analyzer display.

Span Accuracy The uncertainty of the indicated frequency separation of any two signals on the display.

S-Parameters (Scattering Parameters) A convention used to characterize the way a device modifies signal flow using a network analyzer. A two port device has four S-parameters: forward transmission (S21), reverse transmission (S12), forward reflection (S11), and reverse reflection (S22).

Stop/Start Frequency Terms used in association with the stop and start points of the frequency measurement range. Together they determine the span of the measurement range.

Storage States The number of settings, programs, traces, and other parameters available to be saved, cataloged, and recalled at any one time.

Storage, Disk An internal or external digital storage disk for saving test data, instrument settings, IBASIC programs, and other measurement parameters. Storage formats include MS-DOS (R) and HPs standard LIF with binary, PCX, HP-GL, or ASCII data formats.

Structural Return Loss Poor return loss in cable due to a periodic fault such as a periodic dent caused by dropping the cable spool or by the cable pulling process during manufacture.

Supplemental Characteristics Typical but non-warranted performance parameters, denoted as "typical", "nominal" or "approximate".

Sweep The ability of the source to provide a specified signal level over a specified frequency range in a specified time period. Also see Sweep Mode and Sweep Type.

In data processing mode, a series of consecutive data point measurements, taken over a sequence of stimulus values.

Sweep Mode The way in which a sweep is initiated or selected, e.g., single, continuous, alternate, or chopped.

Sweep Type The method of sweeping the source, e.g., linear, log, or frequency step.

Sweeper A signal source that outputs a signal that varies continuously in frequency.

SWR Standing Wave Ratio, calculated as $(1 + \pi) / (1 -\pi)$ where π is the reflection coefficient.

Sync Synchronization, or Synchronized

Syntax The grammar rules that specify how commands must be structured for an operating system, programming language, or applications.

System Dynamic Range The difference between the maximum receiver input level and the receiver's noise floor. System dynamic range applies to transmission measurements only, since reflection measurements are limited by directivity.

T

T/R See Transmission/Reflection.

Termination A load connected to a transmission line or other device.

Test Limit The acceptable result levels for any given measurement.

Test Port See Port.

Test Set The arrangement of hardware (switches, couplers, connectors and cables) that connect a test device input and output to the network analyzer's source and receiver to make s-parameter measurements.

Third Order Intercept TOI: The power input to a non-linear device that would cause third order distortion at the same power level. TOI is a measurement to determine the distortion characteristics of a mixer or receiver. The higher the value, the more immune the receiver to internal distortion.

Thru Through line: A calibration standard. See Calibration, SOLT.

Tint A shade of color; hue.

Toggle To switch states, usually to change a function from on to off, or off to on.

TOM Thru-Open-Match: A Rohde&Schwarz term to describe a calibration method.

Trace A series of data points containing frequency and response information. The series of data points is often called an array. The number of traces is specific to the instrument.

Tracking The ability of the analyzer's receiver to tune to the source frequency over the measurement frequency range. Poor tracking results in amplitude and phase errors due to the receiver IF circuits attenuating and delaying the device under test output.

Transfer Function The ratio of the output signal to the stimulus signal, both as a function of frequency.

Transmission See Transmission Measurements.

Transmission Intermodulation Spurious A measure of the capability of the transmitter to inhibit the generation of intermodulation distortion products. Intermodulation spurious is sometimes called intermodulation attenuation.

Transmission Measurements The characterization of the transfer function of a device, that is, the ratio of the output signal to the incident signal. Most common measurements include gain, insertion loss, transmission coefficient, insertion phase, and group delay, all measured over frequency. See also S-Parameters.

Transmission/Reflection (T/R) Refers to the suite of measurements made by a scalar or vector network analyzer to characterize a device's behavior over frequency. See also S-Parameters.

Transparent Something that is not visible to the user. Usually a procedure that occurs without the user's initiation or knowledge.

Trigger A signal that causes the instrument to make a measurement. The user can select several options for triggering, such as manual, continuous, or external (for synchronizing measurements to an external source).

TRL Through-Reflect-Line. See Calibration, TRL and LRM.

TTL Transistor-Transistor Logic

Two-Port Error Correction See Error Correction, 12-Term.

U

Uncorrected Measurements made without performing error correction.

Uncoupled Channels Stimulus or receiver settings allowed to be set independently for each channel.

UNI User-Network Interface: The point at which users connect to the network.

Units Dimensions on the measured quantities. Units usually refer to amplitude quantities because they can be changed. In analyzers with microprocessors, available units are dBm (dB relative to 1 mW dissipated in the nominal input impedance), dBmV (dB relative to 1 mV), dBW (dB relative to 1 1W), V (volts), W (watts).

V

Variable A symbol, the value of which changes either from one iteration of a program to the next, or within each iteration of a program.

Vector A quantity that has both magnitude and phase.

A network analyzer capable of measuring both magnitude and phase.

VEE Visual Engineering Environment (Agilent software product)

Velocity Factor A numerical value related the speed of energy through transmission lines with different dielectrics (.66 for polyethylene). Used in making time domain measurements.

Vertical Resolution The degree to which an instrument can differentiate amplitude between two signals.

Video An electrical signal containing timing, intensity, and often color information that, when displayed, gives a visual image.

Video Bandwidth In spectrum analyzers, the cutoff frequency (3 dB point) of an adjustable low-pass filter in the video circuit. When the video bandwidth is equal to or less than the resolution bandwidth, the video circuit cannot fully respond to the more rapid fluctuations of the output of the envelope detector. The result is a smoothing of the trace, or a reduction in the peak-to-peak excursion, of broadband signals such as noise and pulsed RF when viewed in broadband mode. The degree of averaging or smoothing is a function of the ratio of the video bandwidth to the resolution bandwidth.

Video Filter In spectrum analyzers, a post-detection, low-pass filter that determines the bandwidth of the video amplifier. It is used to average or smooth a trace. Refer also to Video Bandwidth.

VNA Vector Network Analyzer

W

Waveform A representation of a signal plotting amplitude versus time.

Wireless A term that refers to a broad range of technologies that provide mobile communications for home or office, and "in-building wireless" for extended mobility around the work area, campus, or business complex. It is also used to mean "cellular" for in-or out-of-building mobility services.

WWW World Wide Web

X

Y

Z

Zero-Length Through Path In a measurement calibration, when the two test cables mate together directly without using adapters or a thru-line. See also Non-Insertable Devices.