ユーザおよびプログラミング情報

(プリント版ヘルプ)

Agilent Technologies
PNAシリーズRFネットワーク・アナライザ
E8356A、E8357A、E8358A

製造パーツ番号: E8356-90028
Printed in USA
2000年11月

2000年10月iiの後継文書

ご注意

本書に記載した内容は、予告なしに変更することがあります。

本書に記載した内容は、商品性や特定の目的に対する適合性を保証するものではなく、当社はそれらに関して責任を負いません。 また、本書の記載の誤り、あるいは本書の配布、内容、利用にともなって生じる偶発的、結果的損害に関して責任を負いません。

© Copyright 2000 Agilent Technologies, Inc.

PNAシリーズRFネットワーク・アナライザ
プリント版ヘルプ

PNAシリーズ・ネットワーク・アナライザの主要なドキュメンテーションは、ヘルプ・システムとして機器に組み込まれています。

- 本器の**Help**メニューで、**Network Analyzer Hel**pをクリックします。
- 本器フロントパネルの**COMMAND**ブロックで**Hel**pを押します。

ヘルプ・システムは、ハイパーテキストとハイパーリンクによって本器に関する情報にアクセスできるように構成されており、随所にアニメーションが用意されています。本書はヘルプ・システムをプリントしたものであり、オンライン・ヘルプ・システムのこのような特長は得られません。

PNAシリーズに関して知りたいことがあるときは、まずアナライザまたは製品に付属するCD-ROM上のヘルプを参照してください。ヘルプの情報は下記のセクションに分かれています。

- **入門編**では、本器の操作とインタフェースについて説明します。本器の使用準備のために最初に行う作業についても記載されています。
- **測定の実行**では、測定を実行するための主な手順について説明します。
    - **ステップ1 アナライザの設定**では、既知の初期設定から出発して測定を実行する方法を説明します。
    - **ステップ2 測定の最適化**では、最善の測定結果を得るためにどのような設定を変更すればよいかについて説明します。
    - **ステップ3 測定校正の実行**では、校正の目的と、さまざまな校正タイプの選択と実行の方法について説明します。
    - ステップ4 データの解析では、測定結果のデータをマーカ、トレース演算、リミット・テストを使って解析する方法を説明します。
    - ステップ5 データのプリント、セーブ、リコールでは、測定データの出力方法について説明します。

- **チュートリアル**では、具体的なデバイス特性の測定についてさまざまな観点から紹介し、測定の実行方法を説明します。
- **プログラミング**では、SCPIおよびCOMコマンドを使った測定の自動化について説明し、すべてのコマンドのリストを記載します。
- **技術仕様の理解**では、PNAシリーズに関するすべての仕様の一覧を記載します。
- **製品サポート**では、アップグレード、アクセサリ、技術的アドバイスなど、アナライザのサポートに関する事項を説明します。

PNAシリーズRFネットワーク・アナライザ 3
測定前の準備 15
フロントパネル・ツアー 16
補助出力(ポート1) 16
補助出力(ポート2) 18
チャネル設定キー 20
コマンド・キー 21
ディスプレイ・キー 22
表示画面 23
入力キー 25
3.5インチ・フロッピー・ディスク・ドライブ 26
ナビゲーション・キー 26
電源スイッチ 27
プローブ電源 27
テスト・ポート 27
トレース設定キー 28
USBコネクタ 29
Utilityキー 29
リアパネル・ツアー 31
バイアス入力コネクタ 31
外部AM入力コネクタ 32
外部ディテクタ入力コネクタ 32
補助入出力コネクタ 32
10 MHz基準コネクタ 33
GPIB(General Purpose Interface Bus)コネクタ 34
LAN(ローカルエリア・ネットワーク)コネクタ 34
パラレル・インタフェース・ポート(LPT1)コネクタ 34
シリアル・インタフェース・ポート(COM1)コネクタ 34
USB(ユニバーサル・シリアル・バス)コネクタ 35
VGA(ビデオ・グラフィックス・アダプタ)出力コネクタ 35
アナライザの動作モデル 36
基本測定シーケンス 38
ユーザ・インタフェース 39
フロントパネル・インタフェース: 基本機能アクセス 39
フロントパネル・インタフェース: ダイアログ・ボックス・アクセス 39
フロントパネル・インタフェース: ドロップダウン・メニュー・アクセス 40
マウス・インタフェース 40
アナライザ・ヘルプの使用法 41
ヘルプでの情報検索 41
ホーム・ページとツールバー 41
索引 42
目次 42
用語集 42
ダイアログ・ボックス 43
ナビゲーション・リンク 43

現在のヘルプ・トピックのプリント 44
ヘルプ・ウィンドウのサイズと位置の変更 44
ヘルプ・ウィンドウの縦横比を維持して縮小または拡大 44
ヘルプ・ウィンドウの位置変更 45
ヘルプ・ウィンドウのクローズ 45
ネットワーク・アナライザのユーザの追加 46
緊急用リペア・ディスク 51
測定の実行 52
ステップ1. アナライザの設定 53
アナライザの初期設定 54
アナライザ画面のカスタマイズ 56
測定ステータス 58
データ・トレース 59
メモリ・トレース 59
タイトル・バー 59
データ・フォーマットとスケール 60
周波数レンジ 67
PNAシリーズ・アナライザの周波数レンジ: 67
測定パラメータ 69
Sパラメータ 69
任意比 70
絶対パワー 71
Sパラメータ測定 72
任意比測定 72
絶対パワー測定 73
複数のトレースとチャネルの表示 74
複数のチャネル、トレース、ウィンドウ 74
構成済み測定条件設定 75
表示の配列 77
構成済み測定条件設定 80
ウィンドウ配列構成の選択 80
複数ウィンドウ構成の1つのウィンドウを最大化 81
掃引 82
掃引のタイプ 82
リニア周波数掃引 84
パワー掃引 85
CW時間掃引 85
セグメント掃引の手順 87
トリガ 90
トリガ・モデル 90
トリガ・ソース 90
トリガ・スコープ 91
チャネル・トリガ・ステート 91
トリガ設定とリスタート 93
単純なトリガ設定 94
トリガの設定 94

パワー・レベル 96
パワー・レンジ 96
ポート間のパワー結合 96
信号源減衰 97
パワー・スロープ 101
レシーバの減衰(オプション015を使用) 101
パワー・レベルの設定 102
被試験デバイスへの パワーのオフ 102
減衰の手動設定 103
パワー・スロープの設定 103
アナライザ・レシーバの減衰レベルの設定(オプション015使用時のみ) 104
ステップ2. 測定の最適化 106
測定確度の改善 107
アクセサリの影響 107
ケーブル損失の補正 108
不要応答の除去(オプション010時間領域のみ) 108
低損失2ポート・デバイスの反射確度 110
ダイナミック・レンジ 112
デバイスの入力パワーを増やす 112
レシーバのノイズ・フロアを下げる 112
デバイスの入力パワーを増やす 114
レシーバのノイズ・フロアを下げる 114
電気長の長いデバイスの測定 116
デバイス遅延によって不正確な結果が生じる理由 116
測定確度を改善する方法 116
掃引速度を下げる 117
ステップ掃引を使用 118
Rチャネルに電気長を追加 118
位相測定確度 119
電気遅延 119
ポート延長 119
位相オフセット 119
周波数ポイントの間隔 120
リニア位相シフトを補正 121
基準面を電気的に移動 122
位相測定結果にオフセットを付加 123
エリアジングのチェック 123
トレース・ノイズ 124
掃引アベレージング 124
トレース・スムージング 124
IF帯域幅 125
掃引アベレージングの適用または変更 125
トレース・スムージングの適用 126
IF帯域幅を狭める 126
レシーバ・クロストーク 127
交互掃引に設定 127

測定校正を実行 127
交互掃引に設定 128
レスポンス/アイソレーション測定校正を実行 128
ポイント数 129
測定の最適化 130
測定の安定度 131
周波数ドリフト 131
温度ドリフト 131
不正確な測定校正 131
デバイス接続 132
外部周波数基準の使用 132
室温の制御 132
校正標準温度の安定化 132
よい接続の実行 133
測定スループットの改善 134
掃引速度 134
掃引設定 134
ノイズ低減設定 134
測定校正選択 135
不要な機能 135
掃引設定の変更 135
ノイズ低減設定の変更 137
不要な機能をオフにする 137
高速な校正の選択 139
複数測定 140
スループットを上げるための測定条件設定 140
測定の切替えを自動化 143
測定の呼出しの高速化 143
測定セットの作成 144
選択的に測定をトリガ 145
データ転送速度 146
シングル掃引モードの使用 147
ステップ3. 測定校正の実行 148
測定校正の概要 149
測定校正とは何か 149
なぜ校正が必要なのか 149
校正を行った方がよい場合 150
測定誤差 151
校正標準 160
校正キット 160
校正キット・ファイルの変更 161
電子校正(ECal) 166
測定校正の選択と実行 168
測定校正方法 168
測定校正選択のまとめ 172
校正ウィザード 172

補正オン/オフ 173
校正プロパティ 173
校正データ記憶 173
現在の校正データの変更 174
測定校正のリコール 174
校正ウィザードの使用 175
補正のオン/オフ 179
校正プロパティの表示 179
正確な測定校正 180
測定基準面 180
ポート延長 180
正しくない校正標準を使用することによる影響 180
ノンインサータブル・デバイスの校正 181
2ポート校正のアイソレーション部分 183
ポート延長機能の使用 184
スルー・アダプタの特性評価 185
等価アダプタ交換校正法の実行 185
測定校正の有効性 186
補正なし(No Cor) 186
誤差補正能力の低下の可能性(C*) 186
校正確度の検証 187
クイック・チェック 187
補間のオン/オフ 188
ステップ4. データの解析 189
マーカ 190
マーカ・タイプ 190
基準マーカ 190
マーカによる検索 191
マーカ機能 - 機器設定の変更 192
離散マーカ 192
マーカ表示 193
マーカをオンにする 194
マーカによるアナライザ設定の変更 194
マーカ・データの表示 195
マーカ・データ・フォーマットの変更 196
データ検索 196
基準マーカによる相対測定の実行 197
リミットを使用したデバイスのテスト 199
リミット 199
リミット・ライン 199
リミット・テスト 199
リミット・セグメントの定義 201
リミット・テーブルの表示/非表示 202
リミット・ラインのオン 202
リミット・テストのオン 202
演算機能 203

トレース演算 203
トレース統計 204
トレース演算の選択 205
トレース統計の設定 205
ステップ5. データの出力 207
ドライブ・マッピング 208
本器から外部コンピュータへのドライブのマッピング 208
外部コンピュータから本器へのドライブのマッピング 209
ファイルのセーブ/リコール 210
ファイルの保存 210
オートセーブ 210
名前を付けて保存 210
ファイルへのプリント 212
ファイル名の編集 212
ファイルのリコール 212
ファイルの保存 213
ファイルのオートセーブ 213
別の名前または場所へのファイルの保存 213
ファイルへのプリント 213
ファイル名の編集 214
アナライザへのファイルのリコール 215
外部コンピュータへのファイルのリコール 215
表示された測定結果のプリント 216
ネットワーク上のプリンタへの接続 216
アナライザへのプリンタの接続 216
プリントアウトの生成 218
測定チュートリアル 219
位相測定リファレンス 220
位相測定とは何か 220
なぜ位相を測定するのか 220
位相フォーマットの使用 221
位相測定のタイプ 222
増幅器パラメータのリファレンス 223
利得 223
利得フラットネス 223
リバース・アイソレーション 223
利得ドリフト対時間(温度、バイアス) 223
リニア位相からの偏移 223
群遅延 224
リターン・ロス(SWR、ρ) 224
複素インピーダンス 224
利得圧縮 224
AM-PM変換係数 224
複素インピーダンス 225
複素インピーダンスとは何か 225
スミス・チャート・フォーマットによるデータの表示 225

極座標フォーマットによるデータの表示 226
確度に関する考慮事項 227
群遅延 228
群遅延とは何か 228
群遅延とリニア位相からの偏移との違い 229
アパーチャとは何か 230
確度に関する考慮事項 231
絶対出力パワー 233
絶対出力パワーとは何か 233
なぜ絶対出力パワーを測定するのか 233
確度に関する考慮事項 233
ハイパワー・コンポーネント測定 235
ハイパワー測定はなぜ難しいか 235
CDMAハイパワー基地局増幅器 235
外部レベリング 236
測定校正 239
AGCループを持つ増幅器のテスト 240
確度に関する考慮事項 241
構成選択ガイド 243
構成1(標準アナライザ) 243
構成2(標準アナライザ) 244
構成3(標準アナライザ) 245
構成4(オプション015アナライザ) 247
構成5(オプション015アナライザ) 248
構成6(オプション015アナライザ) 249
AM-PM変換 251
AM-PM変換とは何か 251
AM-PM変換を測定する理由 252
確度に関する考慮事項 252
利得圧縮 255
利得圧縮とは何か 255
なぜ利得圧縮を測定するのか 255
確度に関する考慮事項 256
掃引周波数利得圧縮測定 257
掃引パワー利得圧縮測定 257
絶対出力パワー測定 258
リニア位相からの偏移 259
リニア位相シフトとは何か 259
リニア位相からの偏移とは何か 259
なぜリニア位相からの偏移を測定するのか 259
電気長の使用 259
確度に関する考慮事項 260
リバース・アイソレーション 261
リバース・アイソレーションとは何か 261
なぜリバース・アイソレーションを測定するのか 261
確度に関する考慮事項 261

小信号利得およびフラットネス 263
利得とは何か 263
フラットネスとは何か 263
なぜ小信号利得およびフラットネスを測定するのか 263
確度に関する考慮事項 263
時間領域測定 265
時間領域の概要 265
時間領域測定とは何か 265
タイム・ドメイン・リフレクトメトリとの類似性 266
時間領域測定のシーケンス 267
測定条件設定 268
結果の最適化 268
表示データの解釈 268
時間領域の分解能とレンジ 269
応答分解能 269
表示分解能 271
測定レンジ 272
表示データに対するウィンドウ・フィルタ 277
ウィンドウ・フィルタリングの利点 277
適切なウィンドウ値 278
時間ゲーティング・フィルタ 280
ゲーティングの操作 280
ゲート設定 280
ゲート形状 281
ゲート特性 282
時間領域測定データ 284
マスキング 284
バンドパス・モードでの反射測定 285
バンドパス・モードでの伝送測定 285
ローパス・モードを使った障害位置検出 286
ローパス・モードでの反射測定 287
ローパス・モードでの伝送測定 287
バンドパス/ローパス時間領域モード 290
各モードの比較 290
被試験デバイスに応じたモードの選択 291
周波数レンジとデータ・ポイント 291
インパルス応答とステップ応答 292
有用なデータ・フォーマット 292
ローパス・モードの設定 294
ローパス・モードの設定 295
バンドパス・モードの設定 296
プログラミング 297
Programming COM Fundamentals 298
Getting a Handle to an Object 300
Collections in the Analyzer 303
COM-DCOMプログラミングの設定 304

COM Data Types 307
Working with Events 309
PNA Object Model 319
Calibrator Object 323
CalKit Object 325
CalStandard Object 326
Channel Object 328
Channels Collection 331
Gating Object 331
IArrayTransfer Object 332
Limit Test Collection 333
LimitSegment Object 334
Marker Object 335
Measurement Object 338
Measurements Collection 341
NAWindow Object 342
Port Extensions Object 343
SCPIStringParser Object 344
Segment Object 344
Segments Collection 345
Trace Object 346
Traces Collection 346
Transform Object 347
Getting Trace Data from the Analyzer 499
Limit Line Testing with COM 502
GPIB Fundamentals 503
The GPIB Hardware Components 503
The GPIB Programming Elements 504
GPIB Specifications 506
GPIB Interface Capability Codes 506
The Rules and Syntax of SCPI 508
Branches on the Command Tree 508
Command and Query 509
Multiple Commands 509
Command Abbreviation 510
Getting Data from the Analyzer 512
Understanding Command Synchronization 516
Analyzer Queues 516
Synchronizing Overlapped Commands 517
Reading the Analyzer's Status Register 522
Agilent VEE Example 525
Microsoft Visual Basic Example 525
Microsoft Visual Basic Script Example 526
Microsoft Visual C++ Example 528
Microsoft Word Example 530
Microsoft Excel Example 531
National Instruments™ LabVIEW Example 532
GPIB using Visual C++ 533
GPIB using Visual Basic 537
COM versus SCPI 539
Channels, Windows, and Measurements using GPIB 541

Create a Measurement using GPIB 541
Catalog Measurements using GPIB 542
Perform a Calibration using GPIB 542
ECALibrate Using GPIB 543
Modify a Calibration Kit using GPIB 544
Setup Sweep Parameters using GPIB 545
Setup the Display using GPIB 546
Getting and Putting Data using GPIB 546
Status Reporting using GPIB 548
SCPI Command Tree 550
Abort Command 550
IEEE 488.2 Common Commands 551
Calc:Correction Commands 554
Calc:Data Command 555
Calc:Filter Commands 556
Calc:Format Command 559
Calc:Function Commands 560
Calc:Limit Commands 562
Calc:Marker Commands 566
Calc:Math Commands 576
Calc:Parameter Commands 577
Calc:RData Command 579
Calc:Smoothing Commands 580
Calc:Transform Commands 581
DISPlay Commands 586
Format Commands 594
Hardcopy Command 595
Initiate Commands 596
Memory Commands 597
Output Command 600
Sens:Average Commands 600
Sense:Bandwidth Command 602
Sense:Correction Commands 602
Sense:Correction:Collect:CKit Commands 609
Sense:Couple Command 616
Sense:Frequency Commands 616
Sense:Power Command 619
Sense:Roscillator Command 619
Sense:Segment Commands 620
Sense:Sweep Commands 626
Source Commands 631
Status Register Commands 635
System Commands 645
Trigger Commands 646
技術仕様の理解 649
定義 650
補正済みシステム性能 651
テスト・ポート出力特性(信号源) 660
テスト・ポートおよびレシーバ入力特性 661
一般情報 666

測定スループット一覧 671
製品サポートの取得 675
その他のリソース 676
ドキュメント・リソース 676
サード・パーティ・リソース 677
アナライザ・オプション 678
機器オプション 678
オプション・アップグレード 678
ドキュメンテーションとローカライゼーション 679
サービス・サポート・オプション 679
テクニカル・サポート 680
アナライザのトラブルシューティング 682
基本チェック 682
誤差項チェック 683
サービス・ガイドのチェック 683
アクセサリ 684
校正キット 684
ベリフィケーション・キット 684
コネクタ・アダプタおよびアクセサリ・キット 685
テスト・ポート・ケーブル 685
ファームウェアとヘルプのアップグレード 685
PNAシリーズ・アナライザを使用する手順 686
外部PCを使用する手順 686

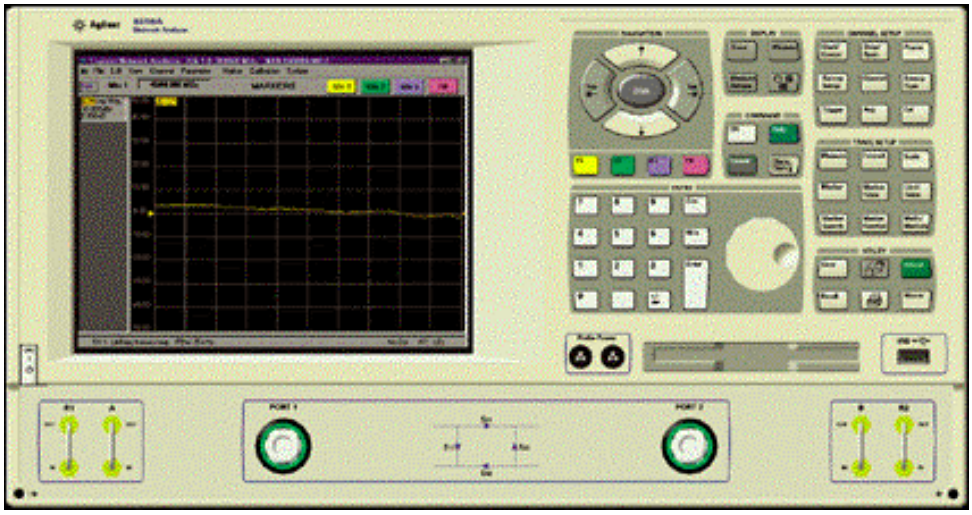
測定前の準備

ボタンをクリックすると、本器について簡単に知ることができます。

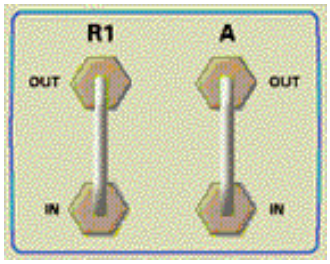
- **フロントパネル・ツアー**
- **リアパネル・ツアー**
- **ユーザ・インタフェース**
- **アナライザ動作モデル**
- **基本測定シーケンス**
- **ヘルプの使用法**
- **ネットワーク機能とアナライザの接続**
- **緊急用リペア・ディスク**

フロントパネル・ツアー

- マウスを使って図をクリックしてください。
- フロントパネルの各部分をクリックすると、説明が表示されます。



補助出力(ポート1)



これらのSMA(メス)コネクタは、カスタム測定の開発に使用します。

レシーバA直接アクセス・ジャンパ

- 上の"A"コネクタは、ポート1カップラの結合アームにつながっています。
- 下のコネクタは、レシーバ"A"の入力に直接つながります。オプション015がインストールされている場合、このパスは切替え可能なアッテネータに直接つながっており、そこからレシーバ入力につながります。

レシーバ入力について:

最大入力レベル:

*6 dBm、<0.4 dB圧縮(300 kHz～3 GHz)
*6 dBm、<0.8 dB圧縮(3 GHz～6 GHz)
*11 dBm、<0.8 dB圧縮(6 GHz～9 GHz)

ノイズ・フロア、RMS(10 Hz帯域幅):

<*133 dBm(300 kHz～3 GHz)
<*123 dBm(3 GHz～6 GHz)
<*123 dBm(6 GHz～9 GHz)

損傷レベル: +26 dBm

最大DCレベル: 40V(標準)、0V(オプション015)

リターン・ロス:

>17 dB(300 kHz～3 GHz)
>12 dB(3 GHz～6 GHz)
>7 dB(6 GHz～9 GHz)

基準チャネル･ジャンパ

- 上のコネクタは、切替えスイッチの基準1出力につながります。
- 下のコネクタは、R1レシーバ入力に直接つながります。

レシーバ入力について:

最大入力レベル:

*6 dBm、<0.4 dB圧縮(300 kHz～3 GHz)
*6 dBm、<0.8 dB圧縮(3 GHz～6 GHz)
*11 dBm、<0.8 dB圧縮(6 GHz～9 GHz)

損傷レベル: >+26 dBm

フェーズ・ロック維持可能な最小レベル

*35 dBm(300 kHz～3 GHz)
*25 dBm(3 GHz～9 GHz)

基準出力について(外部入力によって信号源をロックする場合)

出力レベル:

*5～*30 dBm(300 kHz～6 GHz)
*10～*35 dBm(6 GHz～9 GHz)

信号源整合リターン・ロス:

16 dB(300 kHz～3 GHz)
14 dB(3 GHz～9 GHz)

損傷レベル: >+26 dBm

最大DCレベル: 40 V

テスト･ポート･ジャンパ(オプション015のみ)

- 上のコネクタは、ポート1に接続された信号源につながります。
- 下のコネクタは、ポート1カップラの主入力に直接つながります。テスト・ポートのパワーを増幅するためのパワーアンプはここに挿入します。

信号源出力について:

出力レベル:

+12～*83 dBm(300 kHz～6 GHz)
+7～*88 dBm(6 GHz～9 GHz)

信号源整合: 15 dB、9 GHzで

カップラへの入力について:

テスト・ポート1への挿入損失:

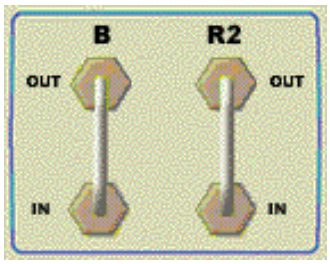
<3.5 dB、3 GHzで
<5 dB、9 GHzで

最大入力パワー: 2 W、CW

損傷レベル: 4 W、CW

フロントパネルに戻る

補助出力(ポート2)



これらのSMA(メス)コネクタは、カスタム測定の開発に使用します。

レシーバB直接アクセス・ジャンパ

- 上の"B"コネクタは、ポート2カップラの結合アームにつながっています。
- 下のコネクタは、レシーバ"B"の入力に直接つながります。オプション015がインストールされている場合、このパスは切替え可能なアッテネータに直接つながっており、そこからレシーバ入力につながります。

レシーバ入力について:

最大入力レベル:

-6 dBm、<0.4 dB圧縮(300 kHz～3 GHz)
-6 dBm、<0.8 dB圧縮(3 GHz～6 GHz)
-11 dBm、<0.8 dB圧縮(6 GHz～9 GHz)

ノイズ・フロア、RMS(10 Hz帯域幅):

<-133 dBm(300 kHz～3 GHz)
<-123 dBm(3 GHz～6 GHz)
<-123 dBm(6 GHz～9 GHz)

損傷レベル: +26 dBm

最大DCレベル: 40V(標準)、0V(オプション015)

リターン・ロス:

>17 dB(300 kHz～3 GHz)
>12 dB(3 GHz～6 GHz)
>7 dB(6 GHz～9 GHz)

基準チャネル・ジャンパ

- 上のコネクタは、切替えスイッチの基準2出力につながります。
- 下のコネクタは、R2レシーバ入力に直接つながります。

レシーバ入力について:

最大入力レベル:

-6 dBm、<0.4 dB圧縮(300 kHz～3 GHz)
-6 dBm、<0.8 dB圧縮(3 GHz～6 GHz)
-11 dBm、<0.8 dB圧縮(6 GHz～9 GHz)

損傷レベル: >+26 dBm

フェーズ・ロック維持可能な最小レベル

-35 dBm(300 kHz～3 GHz)
-25 dBm(3 GHz～9 GHz)

基準出力について(外部入力によって信号源をロックする場合)

出力レベル:

-5～-30 dBm(300 kHz～6 GHz)
-10～-35 dBm(6 GHz～9 GHz)

信号源整合リターン・ロス:

16 dB(300 kHz～3 GHz)

14 dB(3 GHz～9 GHz)

損傷レベル: >+26 dBm

最大DCレベル: 40 V

テスト・ポート・ジャンパ(オプション015のみ)

- 上の"R2"コネクタは、ポート2カップラの切替えスイッチにつながります。
- 下のコネクタは、ポート2カップラの主入力に直接つながります。アナライザのレシーバを保護するためのアッテネータはここに挿入します。

信号源出力について:

出力レベル:

+12～-83 dBm(300 kHz～6 GHz)
+7～-88 dBm(6 GHz～9 GHz)

信号源整合: 15 dB、9 GHzで

カップラへの入力について:

テスト・ポート1への挿入損失:

<3.5 dB、3 GHzで
<5 dB、9 GHzで

最大入力パワー: 2 W、CW

損傷レベル: 4 W、CW

フロントパネルに戻る

チャネル設定キー

各キーの機能を知るには、下の図を使用してください。

これらのキーには以下の機能があります。

- さまざまなスティミュラス設定によるチャネルの構成
- 測定校正の実行

フロントパネルに戻る

アベレージング・キー

測定アベレージングを使用してノイズを減らす場合に使います。指定した数の掃引に対して、複素指数平均が計算されます。

校正キー

測定校正を開始します。Menu/Dialog, Calを押すと、校正ウィザードが表示されます。それ以外の場合、Calキーを押すと校正アクティブ・エントリ・ツールバーが表示されます。

チャネル・キー

アクティブ・チャネルの選択、またはチャネルの削除に使用します。チャネル設定を変更するには、そのチャネルがアクティブになっている必要があります。

Powerキー

信号源出力レベルを指定するために使います。

Stop / Spanキー

測定レンジのストップ値またはスパンを入力するために使います。

Start / Centerキー

測定レンジのスタート値または中心を入力するために使います。

Sweep Setupキー

本器の信号源が被試験デバイスにどのようにスティミュラスを供給するかを定義するために使います。

Sweep Typeキー

本器の信号源が掃引を実行する方法と、掃引の属性を選択するために使います。

Triggerキー

測定掃引をどのように開始するかを指定するために使用します。

コマンド・キー

各キーの機能を知るには、下の図を使用してください。

これらのキーには以下の機能があります。

- ダイアログ・ボックス内のコマンドを実行します。
- メニューまたはダイアログ・ボックスを表示します。
- アナライザのヘルプを起動します。

フロントパネルに戻る

キャンセル・キー

ダイアログ・ボックスをクローズします。

Helpキー

アナライザ・ヘルプ(組込みドキュメンテーション)を起動します。

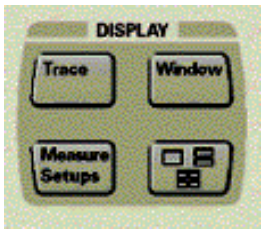
Menu / Dialogキー

ナビゲーション・キーを使ってメニューを操作するために使います。また、Menu/Dialogを押してからChannel、Trace、またはUtilityブロックのキーを押すことにより、ダイアログ・ボックスが表示されます。

OKキー

ダイアログ・ボックスをクローズし、ダイアログ・ボックスで指定した値を確定します。

ディスプレイ･キー



これらのキーには以下の機能があります。

- トレースとウィンドウの管理と作成
- 一連の測定を簡単に構成
- 現在の測定の表示を調整

フロントパネルに戻る

配置キー

ウィンドウの配置を、Overlay、Stack 2、Split 3、Quad 4の4種類から選択できます。

Windowキー

最初にキーを押すと、ウィンドウ・アクティブ・ツールバーが表示されます。 このツールバーでファンクション・キーを使い、ウィンドウを作成、選択、削除することができます。続いてこのキーを押すと、 現在セットアップされたウィンドウが順番にアクティブになります。サイズの変更は、アクティブ・ウィンドウに対してのみ実行できます。 ウィンドウ内のトレースを順番に選択する場合にも、 ウィンドウはアクティブでなければなりません。

Measure Setupsキー

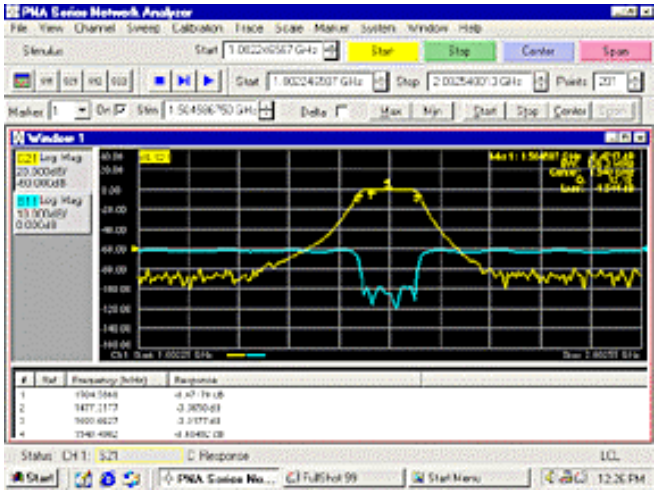
4種類の構成済み測定条件設定を選択するために使います。

Traceキー

1回押すと、Trace Activeツールバーが表示されます。さらに続けて押すと、ウィンドウ内の各測定トレースが順番にアクティブになります。変更ができるのはアクティブ機能だけです。このキーを使うと、ファンクション・キーを使ってトレースの作成や削除も可能です。

表示画面

- マウスを使って図をクリックしてください。



この表示画面の特性は以下のとおりです。

- アクティブ・マトリクス(TFT)、8.4インチLCD
- 解像度: 640×480ピクセル
- 垂直リフレッシュ・レート: 59.83 Hz
- 水平リフレッシュ・レート: 31.41 kHz
- 輝度: >250 NIT

フロントパネルに戻る

アクティブ・エントリ・ツールバー

アクティブ・エントリ・ツールバーで、フロントパネル・キーを押したときに使用可能な機能から機能を選択できます。

ウィンドウ制御ボタン

ウィンドウ制御ボタンで、ネットワーク・アナライザ・アプリケーションおよびアプリケーションに含まれるウィンドウを最小化、復元、クローズできます。

画面グリッド

画面グリッドで、測定トレースを表示できます。アナライザの画面には、最高4つのウィンドウを表示することができます。各ウィンドウには、測定トレースとメモリ・トレースをそれぞれ4つまで表示できます。

ステータス・バー

ステータス・バーで、現在のネットワーク・アナライザの情報を表示できます。情報として、アクティブ・チャネル、選択された測定、測定の補正、ローカルまたはリモート・モード、掃引アベレージ、スムージング、変換、ゲーティング、電気遅延が表示されます。

測定ツールバー

測定ツールバーで、Sパラメータ測定を追加できます。測定はアクティブ・ウィンドウに表示され、アクティブ・チャネルの設定が使用されます。

掃引制御ツールバー

掃引制御ツールバーで、現在の測定掃引を中止したり、シングル測定掃引や連続測定掃引を実行することができます。

メニュー・タイトル

メニュー・タイトルで、アナライザの機能を示すドロップダウン・メニューにアクセスできます。

マーカ・ツールバー

マーカ・ツールバーで、マーカをスティミュラス値に割り当てたり、マーカを使ってピークを検索したり、マーカの位置に基づいてスティミュラス値を設定することができます。

トレース・バー

トレース・ステータス・バーで、マウスや他のポインティング・デバイスを用い、アクティブ測定トレースを選択できます。測定に関係する設定を変更するには、測定がアクティブでなければなりません。

スティミュラス・ツールバー

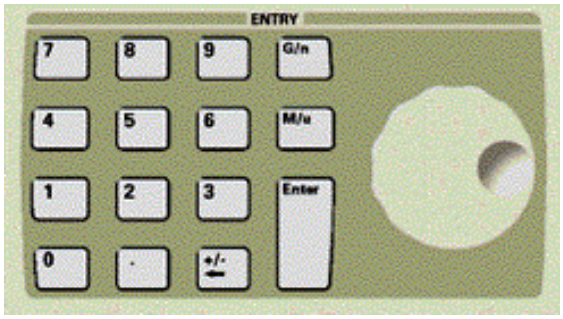
スティミュラス・ツールバーで、スティミュラス設定の値を変更できます。

テーブル

テーブル領域で、マーカ・データ、セグメント掃引の設定、リミット・テストの設定を交互に表示できます。

入力キー

下の図を使ってそれぞれのキーの機能を知ることができます。



これらのキーは、測定設定の値を入力するために使います。
フロントパネルに戻る

小数点キー

小数を表すための小数点を入力します。

Enterキー

測定設定の値を入力します。

数字キー

測定設定の値をこれらのキーで選択したあと、**Enter**または**G/n**または**M/u**を押せば、選択が完了します。

プラス - マイナス - バックスペース・キー

- 入力の最初に押した場合、入力値の正と負を切り換えます。
- カーソルを前に戻し、以前の選択値を削除します。

フロントパネル・ノブ

回すことでアクティブ・エントリの値を増減できます。

Unitsキー

測定単位を指定して値の選択を終了します。そのあと**Enter**を押すと値の入力が完了します。

3.5インチ・フロッピー・ディスク・ドライブ

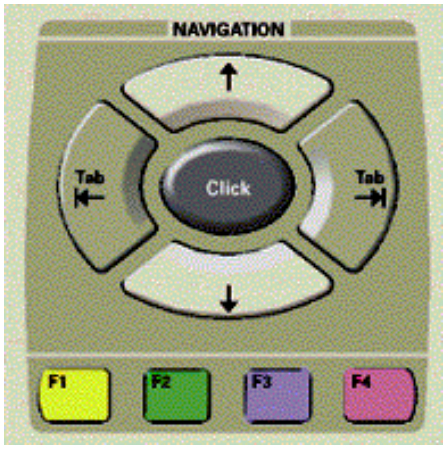
本器のハード・ドライブへのファイルのインストールや、本器のデータ・ファイルの記憶に用いられます。ディスク・ドライブへのアクセスにはNTのエクスプローラを使います。

- アンフォーマット・データ容量: 2 MB
- 高密度(2HD)および通常密度(2DD)互換
- 転送レート: 500 kビット/s

フロントパネルに戻る

ナビゲーション・キー

各キーの機能を知るには、下の図を使用してください。



これらのキーには以下の機能があります。

- アクティブ・エントリ・ツールバーの項目を選択
- メニューとダイアログ・ボックスの操作

フロントパネルに戻る

ナビゲーション・キー

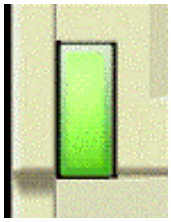
矢印キー- メニューとダイアログ・ボックスのカーソル位置を制御します。

Click- マウス・クリックと同様に選択を実行します。

F1、F2、F3、F4キー

アクティブ・エントリ・ツールバーの項目を選択します。キーの色は、アクティブ・エントリ・ツールバーの項目に対応しています。

電源スイッチ



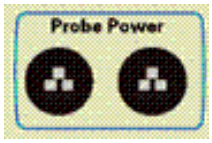
本器のオン状態とハイバネーション状態を切り換えます。

- オン状態では、ボタンが緑になります。
- ハイバネーション状態では、ボタンが橙黄色になります。
- 本器をオフにするには、電源ボタンを押します。実行中のすべてのプログラムのステートが保存され、電源がオフになります。
- もう一度ボタンを押すと、電源をオフにしたときに保存されたステートがロードされます。

このスイッチは電源には接続されていません。

フロントパネルに戻る

プローブ電源



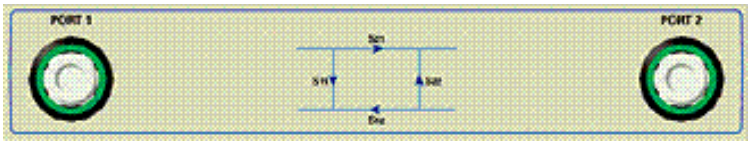
この3ピン(オス)コネクタは、アクティブ・プローブに電力を供給します。

- 上ピン: グランド
- 左ピン: +15 VDC±2%、最大400 mA、0.5 AでPTCにより保護
- 右ピン: -12.6 VDC±5%、最大300 mA、0.5 AでPTCにより保護

(PTC = 遅延動作、自動リセット可能ヒューズ)

フロントパネルに戻る

テスト・ポート



2個の50 Ω N型(メス)ポートをRF信号源とレシーバの間で切り換えることにより、デバイスの双方向測定が可能です。

- 緑のライトは、信号源ポートを示します。
- オレンジのライトは、レシーバ・ポートを示します。

入力損傷レベル:

CWレベル: +15 dBm
DC電圧レベル: ±30V

フロントパネルに戻る

トレース設定キー

各キーの機能を知るには下の図を使用してください。

これらのキーには以下の機能があります。

- 測定トレースの構成
- 結果からのデータの抽出

フロントパネルに戻る

Formatキー

測定データの表示に使用するフォーマットを選択するために使います。

Limit Tableキー

合否テストに使用するリミット・セグメントの値のテーブルを表示します。

Math / Memoryキー

測定データに対する演算/メモリ機能を選択するために使います。

Measureキー

Sパラメータ測定を選択するために使います。ダイアログ・ボックスを使うことにより、任意比測定または絶対パワー測定も選択できます。

Markerキー

マーカをアクティブにして値を設定するために使います。マーカを使うと測定データの値を読み取ることができます。

Marker Functionキー

アクティブ・マーカの位置に基づいて測定設定を変更するために使います。マーカが表示されていない場合、このキーを押すとマーカがアクティブになります。

Marker Searchキー

マーカ検索機能を使用するために使います。マーカが表示されていない場合、このキーを押すとマーカがアクティブになります。

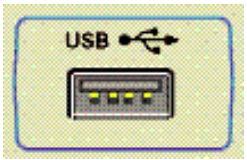
Marker Tableキー

テーブル・エリアには次の3つのテーブルが表示されます。1) マーカ・テーブル アクティブ・トレース上のすべてのマーカのデータ読み値を表示します。2) セグメント掃引テーブル 各セグメントの設定を示します。3) リミット・テーブル 各リミットの設定を示します。

Scaleキー

表示された測定応答をスケーリングするために本器が使用する値を選択するために使います。測定トレース全体が画面に表示されるようにY軸のスケールを自動的に設定させることもできます。

USBコネクタ



USB(ユニバーサル・シリアル・バス)を使用すると、キーボードやその他のポインティング・デバイスを使って アナライザを制御できるようになります。USBコネクタは、 電子校正モジュールにも適応します。

ピン1: Vcc; 4.75 ～ 5.25 VDC、500 mA 最大値
ピン2: -Data
ピン3: +Data
ピン4: Ground

フロントパネルに戻る

Utilityキー

各キーの機能を知るには、下の図を使用してください。

これらのキーには以下の機能があります。

- 測定結果のプリント
- 下記の保存とリコール
  - 機器ステート
  - 測定校正
  - 測定データ
- マクロの実行
- ウィンドウのサイズの変更
- アナライザのプリセット

フロントパネルに戻る

Macro / Localキー

- 本器が自動制御されている場合、このキーを押すとフロントパネル・キーによる操作に応答するようになります。
- 本器が通常動作している場合、このキーを押すとユーザ・マクロの組が表示されます。マクロは実行ファイルに対応します。最大12個のマクロに名前を付けて記憶できます。このキーを繰返し押すと、アクティブ・エントリ・ツールバーのタイトルが変化して、4個のタイトルの3つの組が順番に表示されます。

注記: マクロの組の実行ファイルを動作させるのに必要なアプリケーションがアナライザのハードディスクにインストールされている必要があります。

Restore Up/Downキー

選択したウィンドウを測定画面いっぱいに拡大し、また元のウィンドウ・サイズに戻します。

Presetキー

本器を初期設定(デフォルト)状態に切り換えます。

Printキー

プリント機能を開始し、プリント条件とプリンタの選択を可能にします。

Recallキー

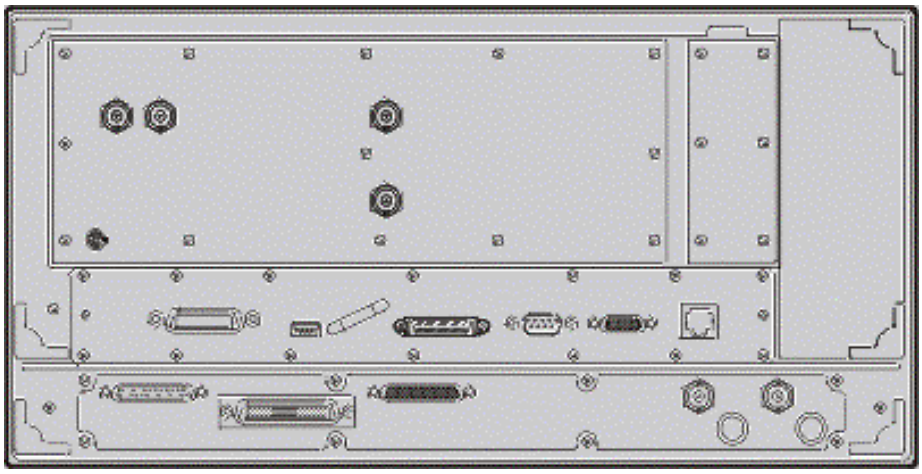
機器ステート、測定校正、測定データを記録したファイルをリコールするために使います。

Saveキー

機器ステート、校正データ、測定データをファイルに保存するために使います。

リアパネル・ツアー

- マウスで図をクリックしてみてください。
- コネクタをクリックすると情報が表示されます。



バイアス入力コネクタ

テスト・ポート・バイアス入力

このBNC(メス)コネクタは、テスト・ポートに外部バイアスを印加するために使います。

- 最大電圧: ±30 VDC、代表値
- 最大電流: ±1 A(RF仕様の低下を伴わないのは±200 mA)*

テスト・ポート・バイアス・ヒューズ:

定格1 A、125 Vのbipin即断ヒューズです。

200 mAレベル以下では、本器はすべてのRF仕様を満たします。DCバイアスの増加に伴い、特に低いRF周波数(一般には<100 MHz、主に>10 MHz)で、低周波補正信号源整合と方向性が悪化します。100 MHzより上のRFでは、バイアス1 Aまで補正信号源整合と方向性はほとんど悪化しません。

リアパネルに戻る

外部AM入力コネクタ

このBNC(メス)コネクタは、テスト・ポートの出力信号に低周波AM変調を加えるための電圧を入力するために使います。入力した電圧によってテスト・ポートの出力パワー・レベルを変化させることもできます。入力が0 Vのときはパワー・レベルは変化しません。電圧が正の時は、テスト・ポートのパワー・レベルが上がります。電圧が負のときは、パワー・レベルが下がります。

- 入力感度: 8 dB/V、代表値
- 帯域幅: 1 kHz、代表値
- 入力インピーダンス: 1 kΩ、代表値

リアパネルに戻る

外部ディテクタ入力コネクタ

このBNC(メス)コネクタは、外部の負極性ダイオード・ディテクタからのDC電圧を入力して、テスト・ポートのパワー・レベルを制御するために使います。この外部電圧を本器に受け入れさせるには、**Sweep Setup**ダイアログ・ボックスの**External ALC**ボックスをチェックしておく必要があります。

- 入力感度: -500 mVのディテクタ入力で約-3 dBmの変化
- 帯域幅: 50 kHz、代表値
- 入力インピーダンス: 1 kΩ、代表値

リアパネルに戻る

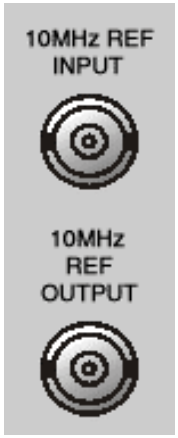
補助入出力コネクタ

外部信号を入力して本器をトリガし、測定を開始することができます。この外部トリガを本器に検出させるには、トリガ・ソースを外部トリガに設定し、TTLレベルのスロープが正か負かを指定する必要があります。

- トリガ準備出力TTLレベル(ピン18)
- 外部トリガ入力TTLレベル、正または負のスロープ(ピン19)

リアパネルに戻る

10 MHz基準コネクタ



10 MHz基準入力

このBNC(メス)コネクタは、以下の特性を持つ外部基準信号を本器で使用するために使います。

- 入力周波数: 10 MHz±1 ppm、代表値
- 入力レベル: -15〜+20 dBm、代表値
- 入力インピーダンス: 200 Ω、公称値

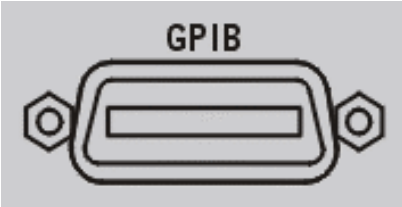
10 MHz基準出力

このBNC(メス)コネクタは、以下の特性を持つ周波数基準信号を供給します。

- 出力周波数: 10 MHz ±10ppm
- 信号タイプ: 正弦波
- 出力レベル: 10 dBm ±4 dB、50Ωに対して
- 出力インピーダンス: 50Ω
- 高調波: < -40 dBc

リアパネルに戻る

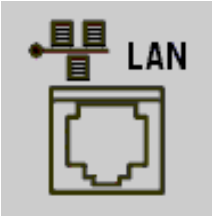
GPIB(General Purpose Interface Bus)コネクタ



この24ピン・コネクタは、IEEE-488互換のD-24メス型です。

リアパネルに戻る

LAN(ローカルエリア・ネットワーク)コネクタ



この10/100BaseTイーサネット接続は標準の8ピン構成であり、2つのデータ・レートから適切なものを自動的に選択します。

リアパネルに戻る

パラレル・インタフェース・ポート(LPT1)コネクタ

この36ピン、ミニD、1284-C、メス型コネクタは、プリンタやその他のパラレル・ポート周辺機器への接続に使用します。

このパラレル・ポートとセントロニクス・ポートとを接続するためのアダプタが本器に付属しています。

リアパネルに戻る

シリアル・インタフェース・ポート(COM1)コネクタ

この9ピン・オス型D-subコネクタは、RS-232互換です。

リアパネルに戻る

USB(ユニバーサル・シリアル・バス)コネクタ

本USBジャックは、以下に示すタイプA構成(一列に4本のピン、左端がピン1)のメス型ジャックです。

- ピン1: Vcc: 4.75 ～ 5.25 VDC、500mA 最大値
- ピン2: -Data
- ピン3: +Data
- ピン4: Ground

ピン1は、定格1 Aの自動リセット可能ヒューズによって保護されています。USBマウス、キーボード、またはその他のポインティング・デバイスを接続できます。

リアパネルに戻る

VGA(ビデオ・グラフィックス・アダプタ)出力コネクタ

この15ピン・メス型Dサブ・コネクタを使って 下記の解像度を持つ外付けVGAモニタを接続できます。 内蔵ディスプレイと外付けディスプレイにデータを同時に表示することができます。

注記:アナライザをデフォルト解像度 (640 x 480)に設定した場合、アナライザと外付けモニタの両方を使用できます。 コントロール・パネルでアナライザの解像度をデフォルト以外の設定値に変更した場合、 使用できるのは外付けモニタだけです。解像度を変更するには、**System** メニューで、**Windows Taskbar**、**Settings**、**Control Panel**をクリックします。次に、**Display** アイコンと**Settings**タブをクリックします。

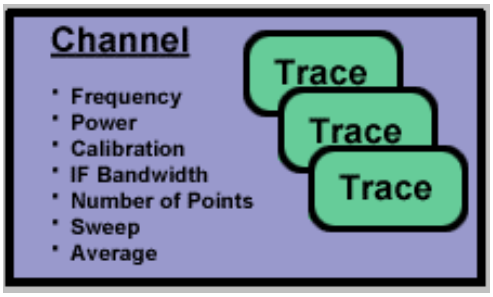
リアパネルに戻る

アナライザの動作モデル

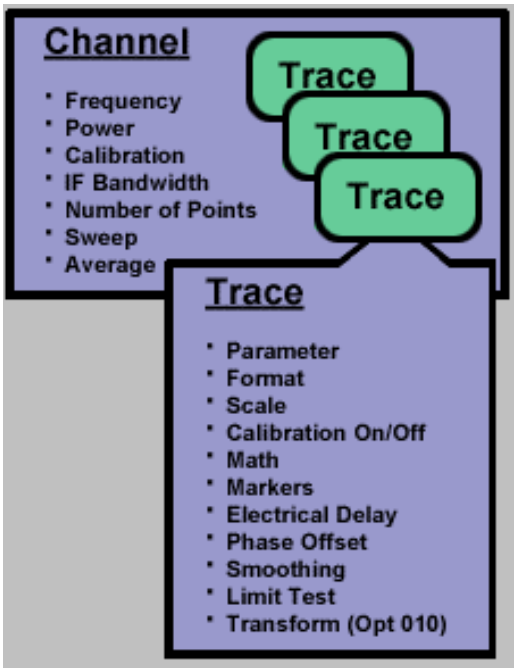
本器の測定には下記の動作モデルがあります。

- **チャネル**
- **トレース**
- **ウィンドウ**
- **動作限界**

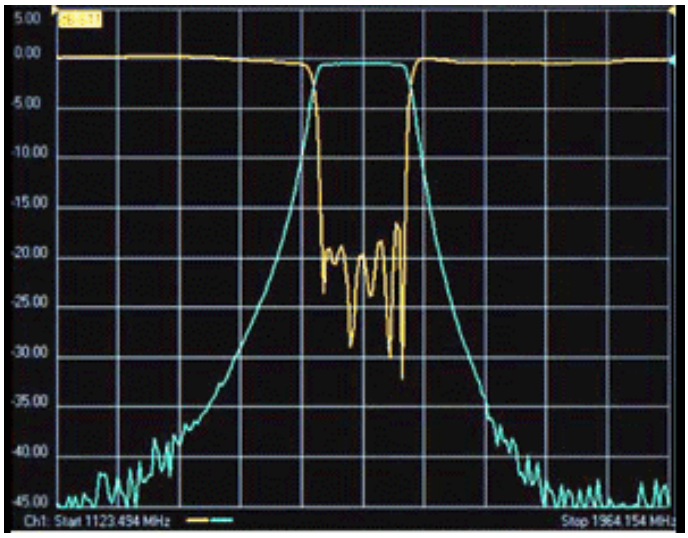
- **チャネル**は、データ収集に関連する属性の組合わせをそれぞれ独立に持っています。
- これらの属性は、そのチャネルに割り当てられたすべてのトレースに適用されます。
- 最大4つの独立したチャネルが使用できます。
- チャネルの設定を変更するには、そのチャネルがアクティブでなければなりません。



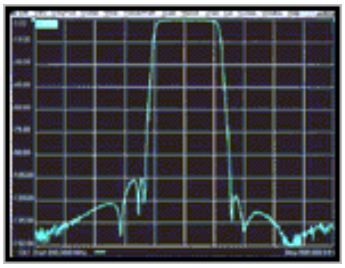
- **トレース**は、測定データの表示方法に関する設定を保持しています。
- 本器には最大16のトレースを表示できます。すべてのトレースを1つのチャネルに割り当てることも、4つのチャネルに分配することもできます。
- トレースの設定を変更するには、そのトレースがアクティブでなければなりません。



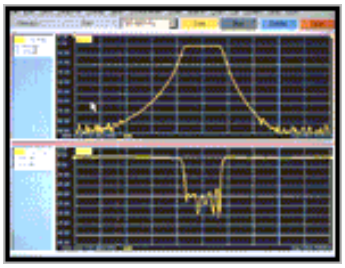
- **ウィンドウ**は、トレースを表示するために用いられます。



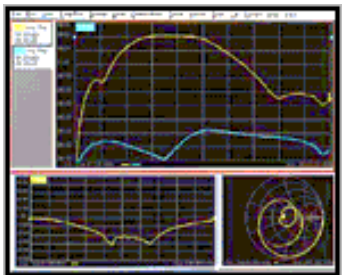
- 本器は画面上に1、2、3、4のいずれかの数のウィンドウを表示できます。



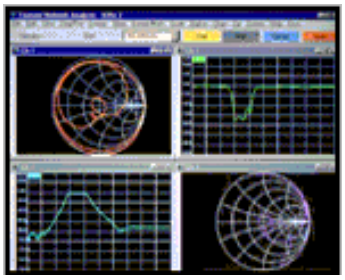
1ウィンドウ



2ウィンドウ

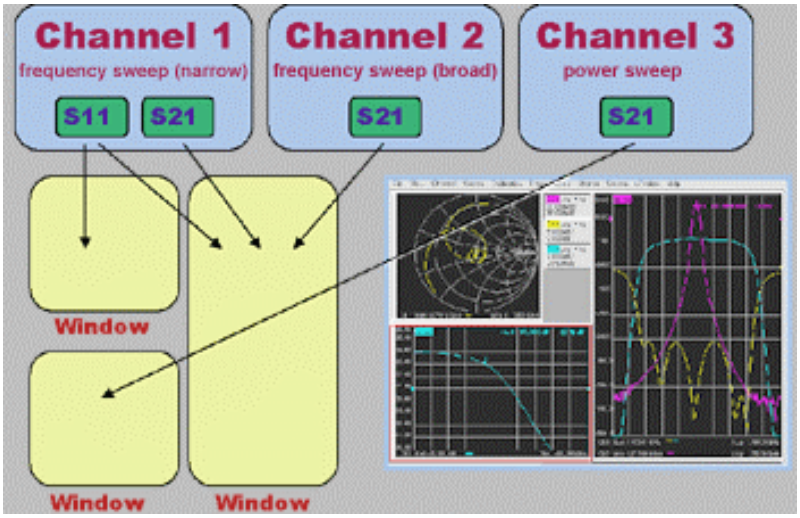


3ウィンドウ



4ウィンドウ

- 複数のチャネルのトレースを1つのウィンドウに表示
- 複数のチャネルのトレースを複数のウィンドウに表示



本器の動作限界

- 1画面あたり4つのウィンドウ
- 1ウィンドウあたり4つのデータ・トレースと4つのメモリ・トレース
- 1画面あたり16のアクティブ・トレースと1画面あたり16のメモリ・トレース
- 4つの独立した測定チャネル

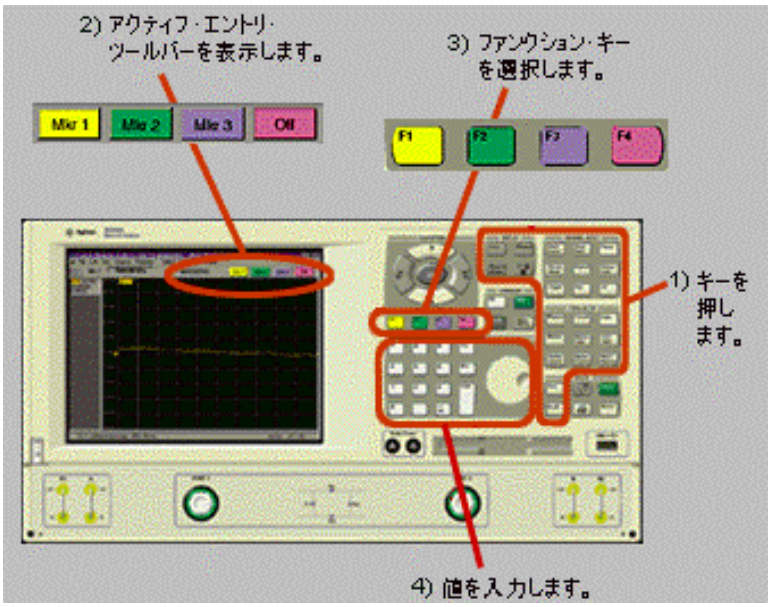
基本測定シーケンス

ステップ1 測定条件設定	本器を初期設定し、測定ステートを作成し、表示を調整します。
ステップ2 測定の最適化	さまざまな技法や機能を使って、測定の確度とスループットを改善します。
ステップ3 測定校正の実行	校正を実行して測定誤差を低減します。
ステップ4 データの解析	マーカ、演算機能、リミット・テストなどを使って測定結果を解析します。
ステップ5 データのプリント、保存、呼出し	測定データを保存またはプリントします。

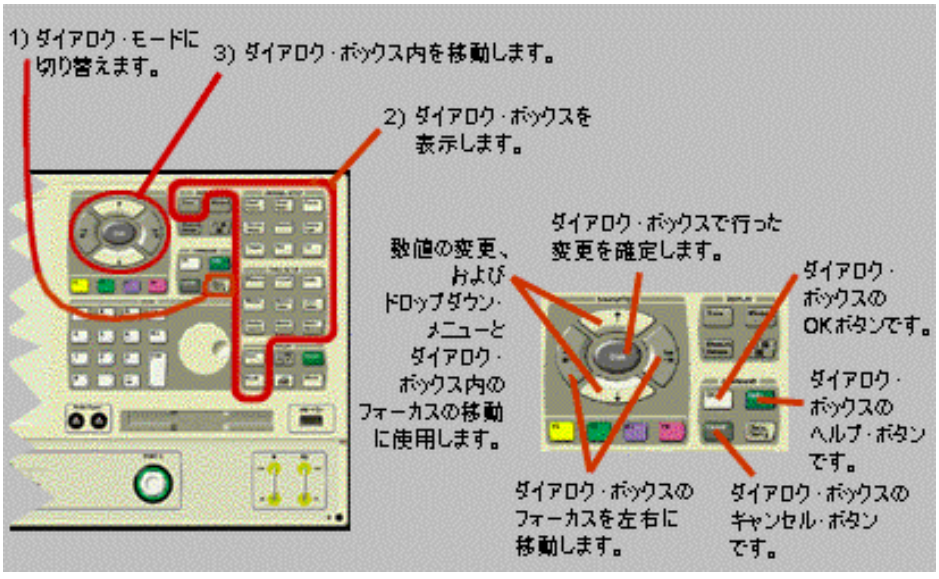
ユーザ・インタフェース

- フロントパネル・インタフェース
- マウス・インタフェース

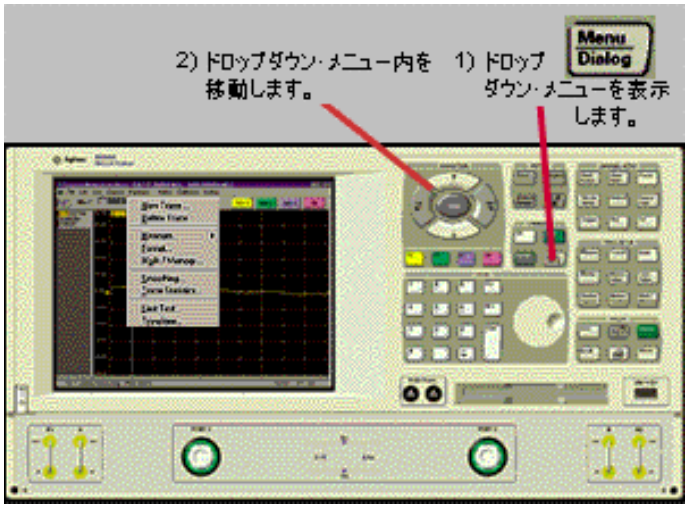
フロントパネル・インタフェース: 基本機能アクセス



フロントパネル・インタフェース: ダイアログ・ボックス・アクセス



フロントパネル・インタフェース: ドロップダウン・メニュー・アクセス



マウス・インタフェース

アナライザ・ヘルプの使用法

ヘルプには本器のすべてのドキュメンテーションが収録されています。ボタンをクリックすると、ヘルプを使って本器に関する疑問に答える方法を簡単に知ることができます。

- **ヘルプでの情報検索**
- **現在のヘルプ・トピックのプリント**
- **ヘルプ・ウィンドウのサイズと位置の変更**
- **ヘルプ・ウィンドウのクローズ**

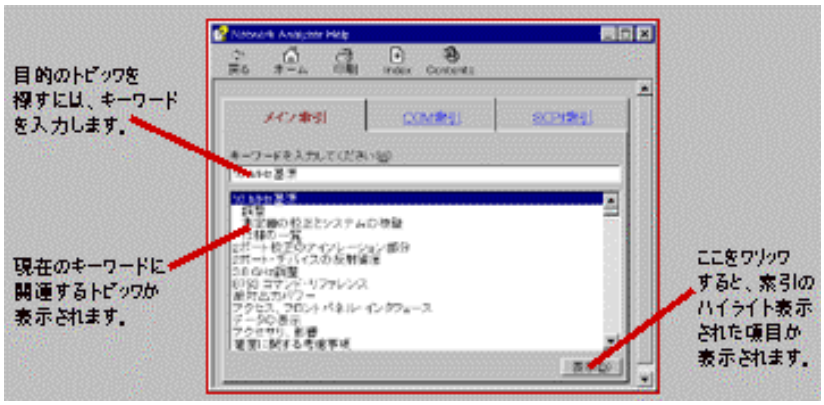
ヘルプでの情報検索

情報の検索は下記の場所から実行できます。

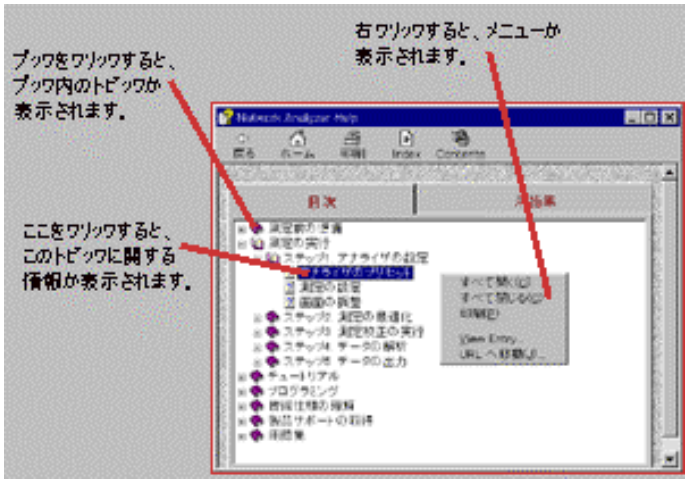
- **ホーム・ページとツールバー**
- **索引**
- **目次**
- **用語集**
- **ダイアログ・ボックス**
- **ナビゲーション・リンク**

ホーム・ページとツールバー

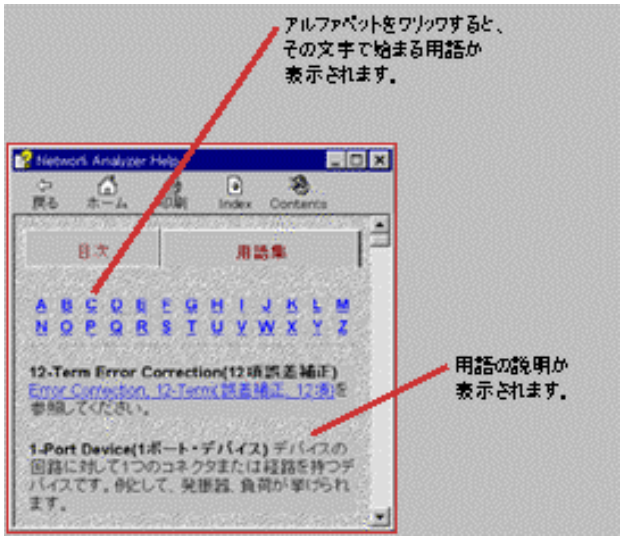
索引



目次

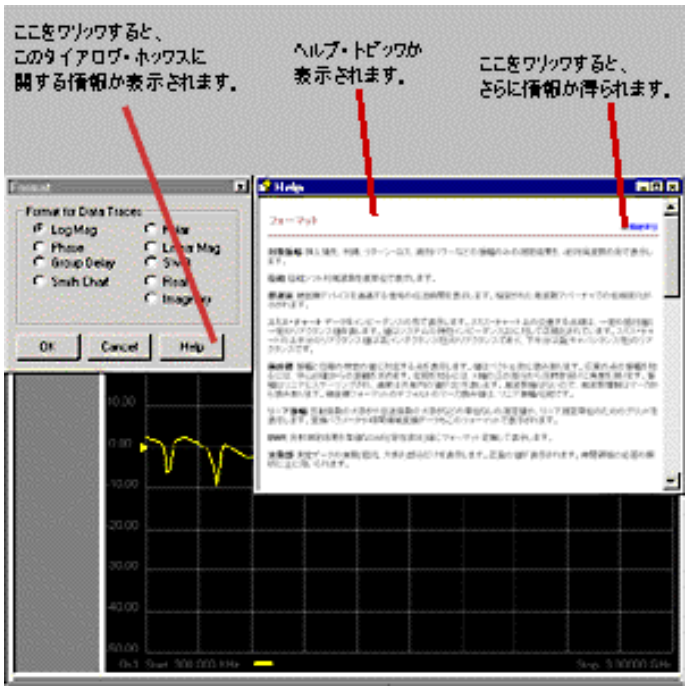


用語集

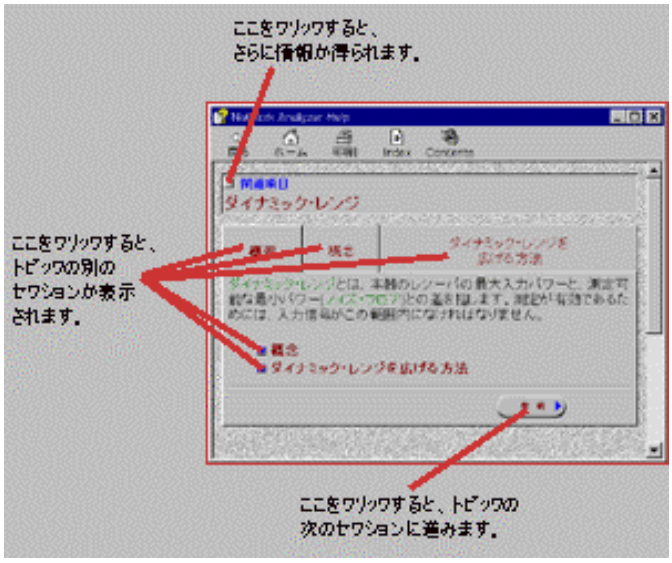


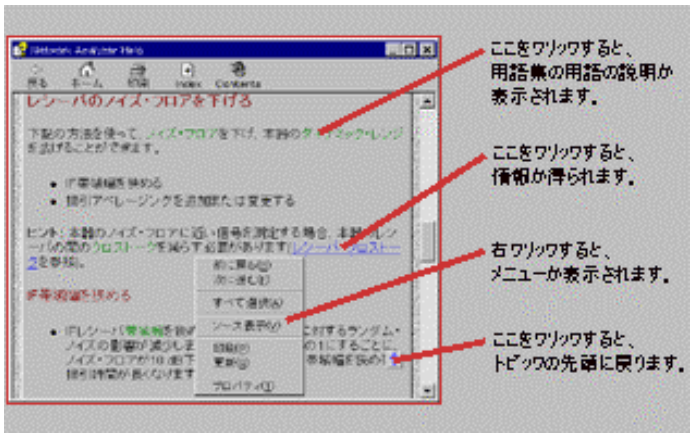
注記: 緑色の文字の語句をクリックすると、用語集の説明が表示されます。

ダイアログ・ボックス



ナビゲーション・リンク





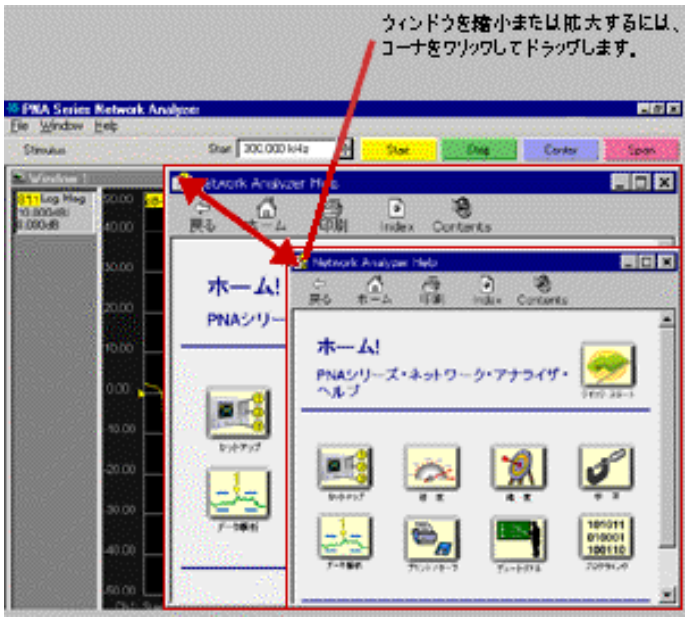
現在のヘルプ・トピックのプリント

ヘルプ・ウィンドウのサイズと位置の変更

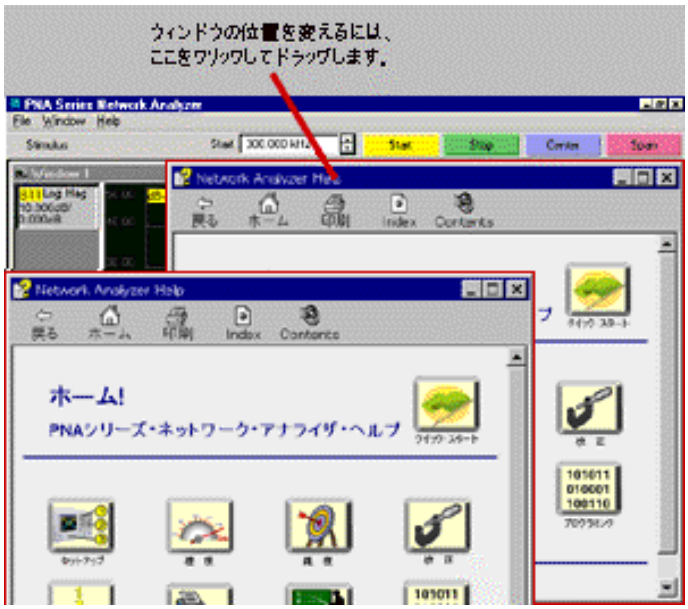
ヘルプ・ウィンドウのサイズと位置を変更するにはいくつかの方法があります。

- **縦横比を維持して縮小または拡大**
- **最大化または最小化**
- **位置変更**

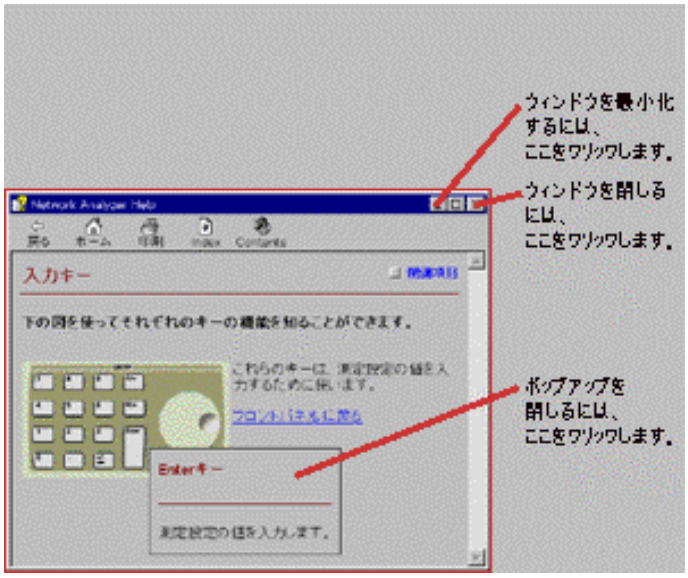
ヘルプ・ウィンドウの縦横比を維持して縮小または拡大



ヘルプ・ウィンドウの位置変更



ヘルプ・ウィンドウのクローズ



ネットワーク･アナライザのユーザの追加

アナライザの電源をオンにすると、デフォルトの管理者ユーザ名とパスワードで自動的にWindows 2000にログインします。本器が下記のような環境にある場合、管理作業の必要はほとんどありません。

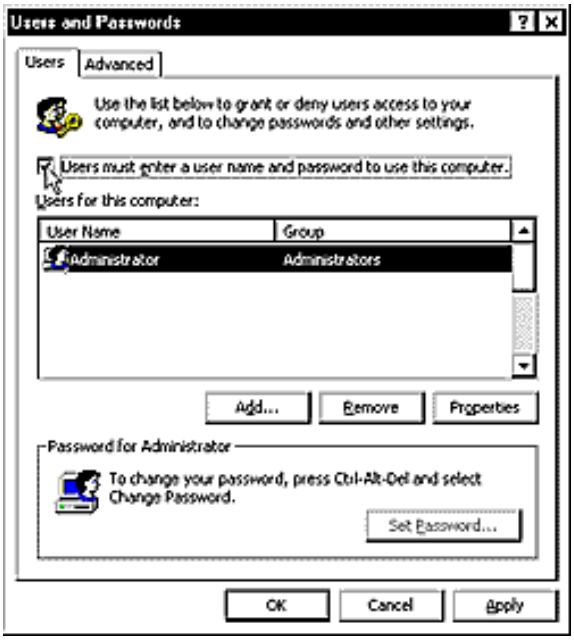
- 安全な環境にある場合
- LAN経由でCOMにより制御されない場合
- ユーザが自分のユーザ名とパスワードを使って本器のディスク・ドライブやプリンタにネットワーク接続しない場合

異なるレベルのユーザを必要とする環境に本器が存在する場合､管理者は他のユーザが使用できる本器の機能を制限することができます｡すなわち､他のユーザはアナライザにログインして使用することはできますが、一部の管理機能は実行できなくなります。

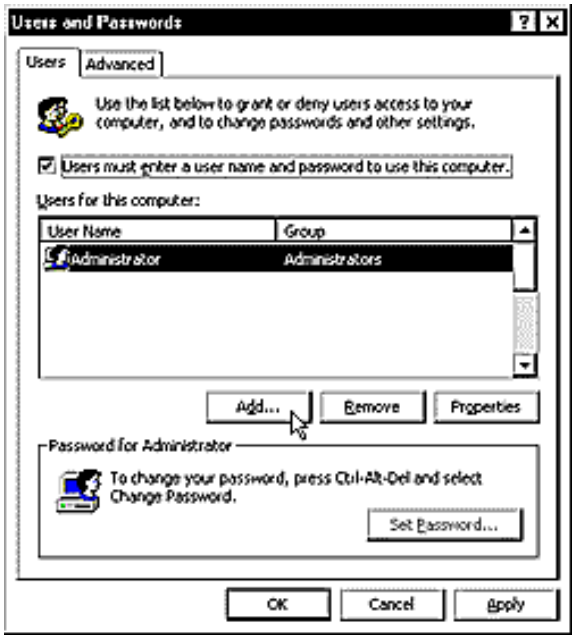
---

1. 本器の**System**メニューで**Configure**をポイントし､**Control Panel**をクリックします。
2. **Control Panel**ウィンドウを下にスクロールさせて、**Users and Passwords**を選択します。

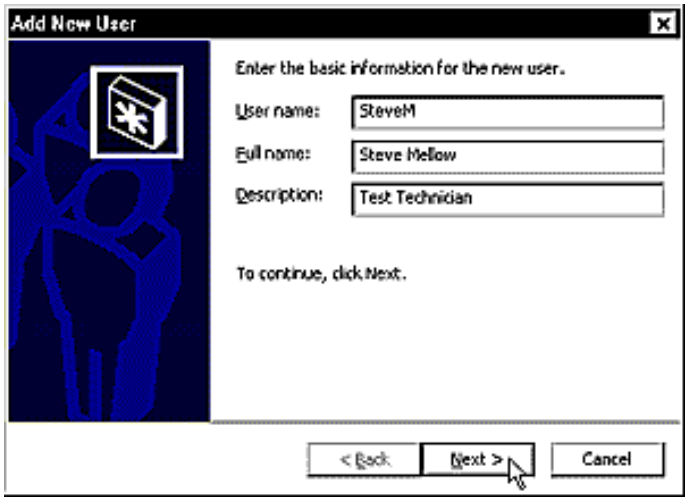
3. **Users**タブで**Add**ボタンが淡色表示されている場合、ウィンドウの上部近くにある**Users must enter a user name and password to use this computer**チェック・ボックスを選択します。



4. **Add**をクリックして新しいユーザの情報を入力します。

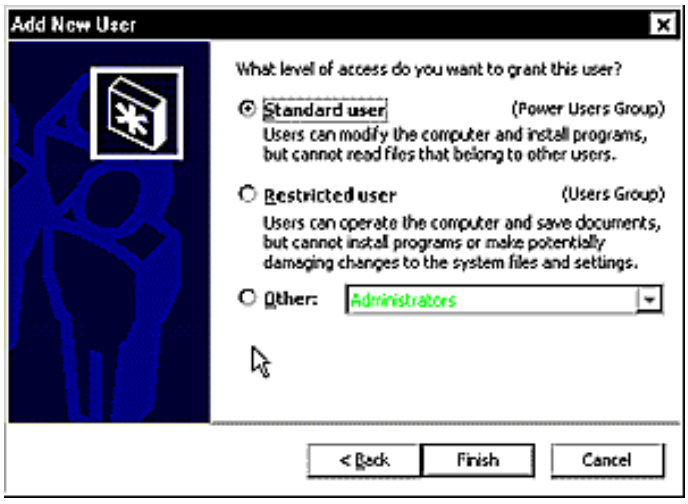


5. **User name**ボックスに新しいユーザのユーザ名を入力します。**Full name**ボックスに新しいユーザのフルネームを入力します。
6. **Description**ボックスに新しいユーザの説明を入力します。次に**Next**を押します。

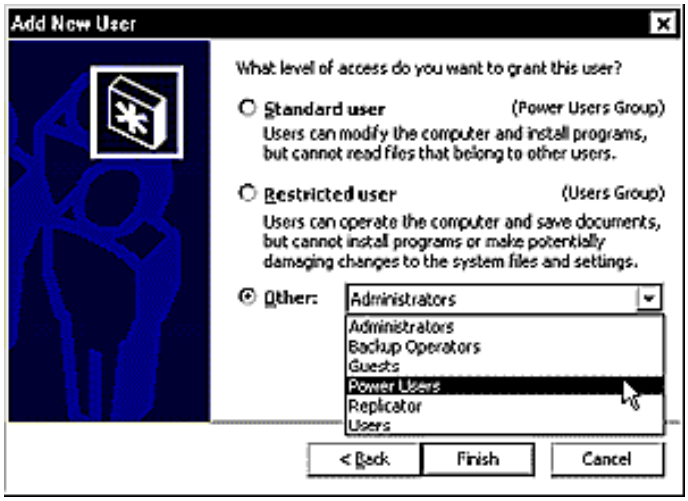


7. **Password**ボックスには、新しいユーザ自身にパスワードを入力してもらいます。**Confirm password**ボックスに同じパスワードをもう一度入力してもらいます。次に、**Next**をクリックします。

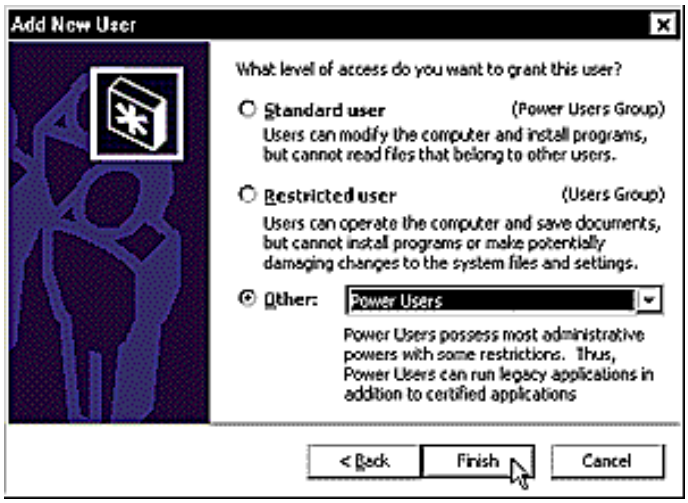
8. 新しいユーザに与えるアクセスのレベルを決めます。現時点では、**Restricted User**はサポートされていません。



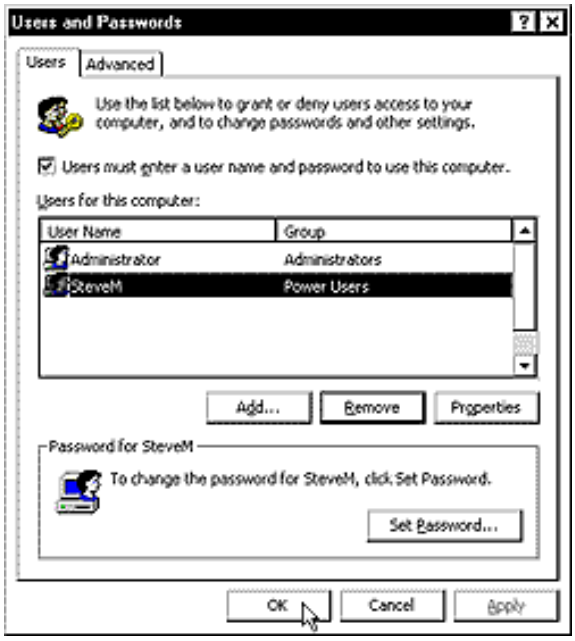
9. その他のレベルのセキュリティを**Other**リストで選択できます。**Other**で選択したレベルの説明がボックスの下に表示されます。



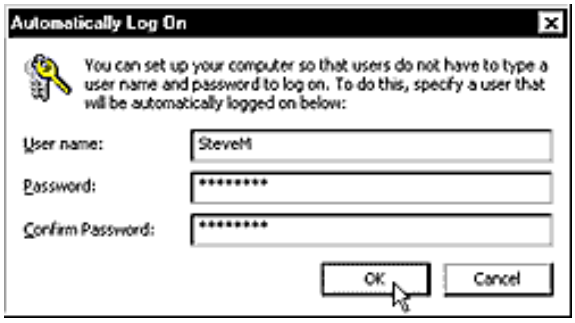
10. 新しいユーザに与えるアクセスのレベルを選択し、**Finish**をクリックします。



11. **Users for this computer**ボックスで、新しいユーザのユーザ名とセキュリティ・レベル・グループを確認します。
12. このユーザが毎回パスワードを入力しなくても本器を使用できるようにするには、**Users must enter a user name and password to use this computer**チェック・ボックスをクリアします。



12. **Automatically Log On**ウィンドウが表示されたら、新しいユーザに**Password**ボックスにパスワードを入力してもらい、**Confirm Password**ボックスに同じパスワードをもう一度入力してもらいます。次に、**OK**をクリックします。



13. **OK**をクリックして新しいユーザの追加を終了します。
14. **File**メニューで**Close**をクリックしてコントロール・パネルをクローズします。

緊急用リペア・ディスク

本器のハード・ディスクが故障した場合、緊急用リカバリ・ディスクを作成してあればハード・ディスクの内容を回復できる可能性があります。このディスクは本器を最初に構成したときに作成してください。

緊急用リペア・ディスクの作成には、1.44 MBの空のフロッピー・ディスクが必要です。

注記: 回復プロセスは、systemroot¥repairフォルダに保存されている情報に基づいて行われます。このフォルダを変更、削除しないでください。

リペア・プロセスに関する詳細は下記の方法で知ることができます。

- Windows 2000 Professionalをお使いの場合、Windows 2000 Professional Getting Startedオンライン・マニュアルまたは、Windows 2000のCDROMに付属する書籍版のWindows 2000 Professional Getting Startedを参照してください。
- Windows 2000 Serverをお使いの場合、「障害からの回復方法」のセクションを参照してください。

緊急用リペア・ディスクの作成手順

1. 本器の**System**メニューで**Windows Taskbar**をクリックします。
2. **Start**をクリックし、**Programs**、**Accessories**、**System Tools**をポイントし、**Backup**をクリックします。
3. **Tools**メニューで**Create an Emergency Repair Disk**をクリックします。
4. **Also back up the registry to the repair directory...**のチェック・ボックスをクリックします。
5. 画面に表示される指示に従います。

測定の実行

- ステップ1. アナライザの設定
- ステップ2. 測定の最適化
- ステップ3. 測定校正の実行
- ステップ4. データの解析
- ステップ5. データの出力

ステップ1. アナライザの設定

- アナライザのプリセット
- アナライザ画面のカスタマイズ
- データ・フォーマットとスケールの選択
- 周波数レンジの設定
- 測定パラメータの選択
- 複数のトレースおよびチャネルの表示
- 掃引の選択
- トリガの選択
- RFパワー・レベルの設定

アナライザの初期設定

概要　　初期設定状態　　ハイバネーション状態　　電源投入時状態

本器は下記の既知の状態に設定可能です。

フロントパネルの**Preset**キーを押すと、本器の機器ステートは下記の初期設定状態に設定されます。

周波数設定:

パワー設定:

掃引設定:

セグメント掃引設定:

トリガ設定

表示設定:

フォーマットを選択すると、下記の設定が適用されます。

応答設定:

補正設定

マーカ設定:

リミット・テスト設定:

リミット・リスト設定

時間領域(オプション010のみ):

グローバル表示設定:

オン/オフ・スイッチは、本器のオン状態とハイバネーション状態を切り換えます。ハイバネーション状態では、本器のオペレーティング・システムはスリープ状態になります。本器をオン状態に戻すと、**オフにしたときと同じ状態で**システムが立ち上がります。

オン状態ではボタンのライトが緑色に点灯します。ハイバネーション状態ではライトは黄色になります。フロントパネルのスイッチは単なるスタンバイ・スイッチであり、ライン・スイッチ(断路装置)ではありません。ハイバネーション・ステートが通常の「オフ」状態です。ハイバネーション・モードでは、わずかな量のスタンバイ電力が本器に供給されます。このスタンバイ電力は電源スイッチ回路だけに供給され、他のCPU関連の回路にはスタンバイ中に電力は供給されません。

ハイバネーションを起動すると、実行中のすべてのプログラムのステートがハード・ディスクに保存され、電源が自動的にオフになります。本器をオン状態に戻すと、オペレーティング・システムはハイバネーション動作によって保存されたステートを探し出して再ロードします。これにより、システムの立上げにかかる時間が短縮されます。

本器をハイバネーション状態にするには、電源ボタンを短時間押します。

注意: 電源スイッチを4秒間以上押し続けると、即座に電源が遮断されます。これは本器の電源プラグを抜くのと同じです。これはPCの場合と同様にファイル・システムの破損の原因となるので、できる限り避けてください。ほとんどの場合Windowsはデータの損失なしに回復できますが、安全性が保証されているわけではありません。

数秒以内にWindowsが**Ready to Hibernate**と表示し、すぐ後に**Preparing to Hibernate**という表示が現れます。本器がハイバネーション・モードになると、電源ボタンのライトが緑から黄色に変わります。

本器を再び動作させるには、電源ボタンをもう一度押します。電源ボタンのライトが黄色から緑に戻り、動作を再開しつつあることを示します。約45秒以内に本器は元の状態に戻るはずです。

特定の場合(移動や保守作業など)を除いて、本器の電源を遮断しないでください。電源を遮断した場合、電源投入時状態が適用されます。

本器の電源を遮断したあとで、もう一度電源を接続して本器のスイッチをオンにした場合、電源投入時状態が適用されます。電源投入時状態は、以下の**例外**を除いて初期設定状態と同じです。

アナライザ画面のカスタマイズ

概要　　ツールバーとテーブル　　測定ステータス　　画面設定の変更方法

いくつかの表示要素の表示/非表示を切り換えることにより、本器の画面をカスタマイズできます。

下の図をマウスでクリックすると、画面の各部分についての説明を見ることができます。

ステータス・バー

ステータス・バーは、画面のいちばん下に表示されています。

以下の情報が表示されます。

- アクティブ・チャネル
- アクティブ・トレースの測定パラメータ
- アクティブ・トレースの誤差補正ステータス
- アクティブ・チャネルのアベレージング・ステータス(オン/オフ)
- GPIBステータス(ローカル/リモート)

ツールバー

最大6種類の異なるツールバーを表示できます。ツールバーを使うと測定の設定や変更が簡単になります。

アクティブ・エントリ・ツールバー

アクティブ・エントリ・ツールバーは、画面の上の方、メニュー・バーの下に表示されます。アクティブ機能を選択するために使います。

以下のものが表示されます。

- フロントパネル・キー操作のアクティブ機能のソフト・キー選択

マーカ・ツールバー

マーカ・ツールバーは画面の最前面のウィンドウのすぐ上に表示されます。マーカの設定と変更に使います。

以下のものが表示されます。

- マーカ番号
- スティミュレーション値
- マーカ機能: デルタ; スタート/ストップ; 中心/スパン

測定ツールバー

測定ツールバーは、画面の最前面のウィンドウの上に表示されます。以下の機能があります。

- Sパラメータ測定の新規トレースを以下の場所に作成
  現在のウィンドウ
  新規ウィンドウ

掃引制御ツールバー

掃引制御ツールバーは、画面の最前面のウィンドウの上に表示されます。このツールバーのボタンは順に、アクティブ・チャネルを以下のように設定します。

- ホールド・モード
- シングル掃引の後ホールド・モード
- 連続掃引

詳細についてはトリガを参照してください。

スティミュラス・ツールバー

スティミュラス・ツールバーは、画面の最前面のウィンドウの上に表示されます。掃引スティミュラスの表示、設定、変更に使います。

以下のものが表示されます。

- **スタート値**
- **ストップ値**
- **ポイント数**

すべてオフ

1回の選択で**すべてのツールバーを非表示にできます。**

テーブル

マーカ、リミット、セグメントに関するテーブルを表示して、これらの設定を表示、作成、変更(マーカ・テーブル以外)できます。

- テーブルは選択したウィンドウの下部に表示されます。
- 1つのウィンドウに表示できるテーブルは一度に1つだけです。

マーカ・テーブル

マーカ設定のテーブルを表示できます。以下の設定が対象となります。

- マーカ番号
- マーカ基準(デルタ測定の場合)
- 周波数
- レスポンス

リミット・テーブル

リミット・テスト設定のテーブルを表示、作成、変更できます。以下のものが対象となります。

- タイプ(MIN、MAX、OFF)
- 開始/終了スティミュラス値
- 開始/終了レスポンス値

セグメント・テーブル

セグメント掃引設定のテーブルを表示、作成、変更できます。以下のものが対象となります。

- ステート(オン/オフ)
- スタート/ストップ周波数
- ポイント数
- IF帯域幅(独立レベルの場合)
- パワー・レベル(独立レベルの場合)
- 掃引時間(独立レベルの場合)

測定ステータス

6種類の測定情報の表示/非表示を切り換えられます。

- トレース・ステータス
- 周波数/スティミュラス値
- マーカ読み値
- リミット・テスト結果
- リミット・ライン
- タイトル

トレース・ステータス

トレース・ステータスは、画面上の各ウィンドウの左側に表示されます。
以下のものが表示されます。

- 測定パラメータ
- フォーマット
- スケーリング・ファクタ
- 基準レベル

周波数/スティミュラス

周波数/スティミュラス情報は、画面上の各ウィンドウの下部に表示されます。セキュリティ上の理由でこの情報を非表示にすることができます。以下のものが表示されます。

- チャネル番号
- スタート値
- ストップ値

マーカ読み値

マーカ読み値情報は、画面上の各ウィンドウの右上隅に表示されます。以下のものが表示されます。

- マーカ番号
- スティミュラス値
- マーカに対して選択されたフォーマットによるレスポンス値

リミット・テスト結果

リミット・テスト結果は、指定したウィンドウの右側に表示されます。以下のものが表示されます。

- **パス**または**フェール**

リミット・ライン

リミット・ラインは、指定したウィンドウのアクティブ・トレースに対して表示されます。リミット・ラインの位置は以下の要因で決まります。

- リミット・レベル
- フォーマット
- スケーリング
- 基準レベル

タイトル

ウィンドウのタイトルを作成して表示できます。下のTitle Entryダイアログ・ボックスが表示されます。

入力したタイトルは、指定したウィンドウの左上隅に表示されます。

データ・トレース

選択したデータ・トレースを非表示にできます。

メモリ・トレース

選択したメモリ・トレースを非表示にできます。

タイトル・バー

タイトル・バー情報は、画面上の各ウィンドウの上部に表示されます。以下のものが表示されます。

- ウィンドウ番号

タイトル・バーを非表示にすると、画面上のすべてのウィンドウのタイトル・バーがなくなり、測定結果を表示する画面領域を広げることができます。

ボタンをクリックすると各手順が表示されます。

マウス手順

1. Viewメニューで、目的の表示注釈をクリックします。

フロントパネル手順

1. COMMANDブロックでMenu/Dialogを押します。
2. 右タブを押して、Viewメニューを選択します。
3. 下矢印ボタンを押して、目的の表示注釈を選択します。
4. COMMANDブロックでOKを押します。

データ・フォーマットとスケール

概要　　概念　　フォーマットとスケールの設定方法

データ・フォーマットとは、測定データをグラフィカルに表示する方法のことです。スケール機能を使えば、測定データが見やすいように編集することができます。

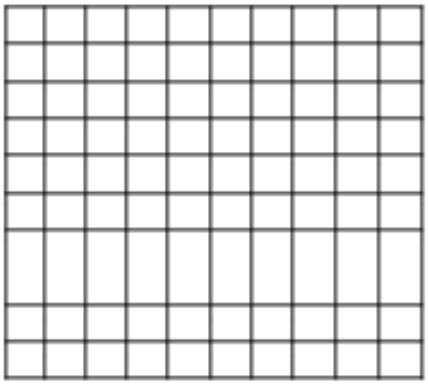
データ・フォーマットは、被試験デバイスについて知りたい情報に合わせて選択します。このトピックでは、用意されている9種類のフォーマットについて説明します。この内容を理解することで、測定データから最大限の情報を引き出すことが可能になります。また、情報を最も見やすく表示するためのスケール設定方法についても説明します。

直交座標表示フォーマット

使用可能な9つのデータ・フォーマットのうち、7つが直交座標表示によって測定データを表します。この表示は**デカルト座標**、X/Y、レクティリニアとも呼ばれます。直交座標表示には下記の性質があります。

- スティミュラス(周波数、パワー、時間のどれか)をX軸にリニア・スケールで表示します。
- 被試験デバイスの周波数応答情報を非常に明確に示します。

7つのデータ・フォーマットの違いは、Y軸データの表現方法です。



対数振幅フォーマット

- 振幅を表示します(位相なし)。
- Y軸: dB
- 代表的測定: リターン・ロス; 挿入損失または利得

位相フォーマット

- 位相を表示します(振幅なし)。
- Y軸: 位相(度)
- 代表的測定:
- リニア位相からの偏移

群遅延フォーマット

- デバイスを通る信号の伝送(伝搬)時間を表示します。
- Y軸: 時間(秒)
- 代表的測定: 群遅延

リニア位相フォーマット

- 正の値のみを表示します。
- Y軸: 単位なし
- 代表的測定: 反射/伝送係数(振幅); 時間領域伝送

SWRフォーマット

- 反射測定データを、(1+ρ)/ (1ρ)という式から計算して表示します。ここで、ρは反射係数です。
- 反射測定だけに有効。
- Y軸: 単位なし
- 代表的測定:
- SWR

実数フォーマット

- 測定された複素データの実数(抵抗)成分だけを表示します。
- 正負両方の値を表示します。
- Y軸: 単位なし
- 代表的測定: 時間領域; サービス用の補助入力電圧信号

虚数フォーマット

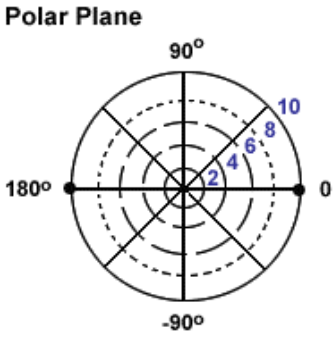
- 測定データの虚数(リアクタンス)成分だけを表示します。
- Y軸: 単位なし
- 代表的測定: 整合回路設計のためのインピーダンス

極座標フォーマット

極座標フォーマットには、振幅と位相の両方の情報が含まれます。下記のようにベクトル的に値を読み取ります。

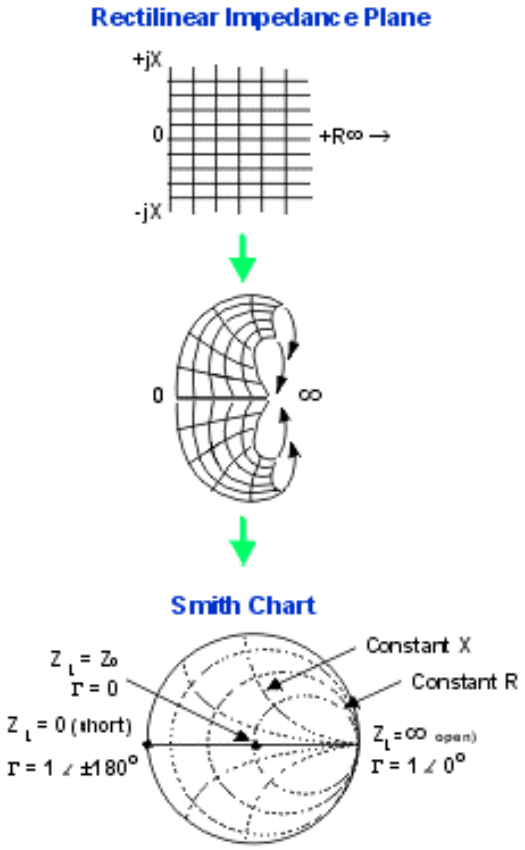
- 任意の点の振幅を知るには、中心(0値)からの距離を求めます。振幅はリニアにスケーリングされており、外周円の値が比1を表します。
- 位相を知るには、X軸の正の部分からの角度を求めます。
- 周波数軸はないので、周波数情報はマーカから読み取ります。

振幅と位相のスケールはリニアです。対数振幅と実数部/虚数部は極座標マーカ・メニューに用意されています。



スミス・チャート・フォーマット

被試験デバイスのインピーダンスを知るには、スミス・チャート・フォーマットを使って反射測定データをインピーダンス表示します。スミス・チャートはインピーダンス平面を極座標平面にマッピングしたものです。



スミス・チャートの解釈

スミス・チャート上の各点は、実数の抵抗(r)と虚数のリアクタンスから構成される複素インピーダンス(r+-jX)を表します。

破線の円は一定の抵抗を表します。中央を通る水平線は純抵抗性(リアクタンス成分なし)です。

- 右端で値は無限大Ω(オープン)になります。
- 左端の値は0Ω(ショート)です。

破線の円弧は一定のリアクタンスを表します。

- 円の上半分のリアクタンス円弧は、インダクタンス性のリアクタンスを表します。
- 円の下半分のリアクタンス円弧は、キャパシタンス性のリアクタンスを表します。

本器に表示される値は、システムの特性インピーダンスに正規化されています。図の中心がシステムの特性インピーダンスを表します。

スケール

この機能は、表示グリッドの縦軸目盛りのスケールを設定するために使います。極座標とスミス・チャートの各フォーマットでは、これにより外周部のフルスケール値が設定されます。

レンジ: 0.001dB/div～500 dB/div

オケールートス

これは、データ・トレースが画面のグリッド領域にちょうど収まるように、縦軸スケールを自動的に調整するものです。スティミュラス値は変化せず、スケールと基準値だけが変更されます。

- すべての表示データが垂直グリッドの80%に収まるような最小のスケール・ファクタが自動的に計算されます。
- 基準値は、トレースが画面中央に位置するように選ばれます。

すべてオートスケール

これはオートスケールと同様の働きをしますが、選択されたウィンドウに表示されているすべてのデータ・トレースが、画面のグリッド領域に収まるように垂直方向に自動的にスケーリングされます。

基準レベル

基準レベルは、選択されたデータ・フォーマットに応じて、直交座標フォーマットの基準ラインまたは極座標フォーマットの外周円の値を指定します。

レンジ: −500 dB～500 dB

基準位置

基準位置は、直交座標表示の基準ラインの位置を指定します。0が格子線のいちばん下のライン、10がいちばん上のラインを表します。デフォルトの位置は5です。

ボタンをクリックすると、データのフォーマット指定とスケーリングの手順が表示されます。

データ･フォーマットの選択

マウス手順

1. **Trace**メニューで**Format**をクリックします。
2. 目的のデータ・フォーマットをクリックします。
3. **OK**をクリックします。

フロントパネル・キー手順

1. **TRACE SETUP**ブロックで**Format**を押します。
2. 対数振幅フォーマットを選択するには、**F1 (LogMag)**を押します。
3. スミス・チャート・フォーマットを選択するには、**F2 (Smith)**を押します。
4. 位相フォーマットを選択するには、**F3 (Phase)**を押します。
5. 群遅延フォーマットを選択するには、**F4 (Delay)**を押します。
6. 他の5種類のデータ・フォーマットを選択するには、この手順のステップ7に進みます。
7. **COMMAND**ブロックの**Menu/Dialog**ボタンを押します。
8. **TRACE SETUP**ブロックで**Format**を押します。
9. 右タブ**NAVIGATION**ボタンを使って目的のデータ・フォーマットを選択します。
   COMMANDブロックのOKボタンを押します。

スケーリング手動選択

マウス手順

1. **Scale**メニューで**Scale**を選択します。**Scale**ダイアログ・ボックスが画面に表示されます。
2. ダイアログ・ボックスの**Scale**部分で、**Per Division**ボックスをクリックします。値を入力するか矢印ボタンを押して、スケーリング・ファクタを選択します。
3. **OK**をクリックします。

フロントパネル・キー手順

1. **TRACE SETUP**ブロックで**Scale**を押します。
2. **F2 (Scale)**を押し、値を入力するか上下の**NAVIGATION**ボタンを押して、スケーリング・ファクタを設定します。

スケーリング自動選択

マウス手順

1. **Scal**eメニューで**Autoscale**または**Autoscale All**をクリックします。

スケーリングを自動的に選択するもう1つの方法は下記の通りです。

1. **Scale**メニューで**Scale**をクリックします。
2. ダイアログ・ボックスの**Scale**部分で**AutoScale**ボタンをクリックして、選択したトレースに自動スケーリングを設定します。または、**AutoAll**ボタンをクリックして、アクティブ・ウィンドウ内のすべてのトレースに自動スケーリングを設定します。
3. **OK**をクリックします。

フロントパネル・キー手順

1. **TRACE SETUP**ブロックで**Scale**を押します。
2. **F1 (AutoScale)**を押して、選択したトレースに自動スケーリングを設定します。すべてのトレースに自動スケーリングを設定するには、この手順のステップ3に進みます。
3. **COMMAND**ブロックの**Menu/Dialog**ボタンを押します。
4. **TRACE SETUP**ブロックで**Scale**ボタンを押します。
5. 右タブ**NAVIGATION**ボタンを使って**AutoAll**ボタンを選択します。**COMMAND**ブロックの**OK**ボタンを押します。

基準ラインの選択

マウス手順

1. **Scale**メニューで**Scale**をクリックします。
2. ダイアログ・ボックスの**Reference**部分で**Level**ボックスをクリックします。値を入力するか矢印ボタンを押して、基準レベルを設定します。
3. ダイアログ・ボックスの**Reference**部分で**Position**ボックスをクリックします。値を入力するか矢印ボタンを押して、基準位置を設定します。
4. **OK**をクリックします。

フロントパネル・キー手順

1. **TRACE SETUP**ブロックで**Scale**を押します。
2. **F3 (Ref Level)**を押し、値を入力するか上下の**NAVIGATION**ボタンを押して、基準レベルを設定します。
3. **F3 (Ref Pos)**を押し、値を入力するか上下の**NAVIGATION**ボタンを押して、基準位置を設定します。

周波数レンジ

概要 **概念** **周波数レンジの設定方法**

周波数レンジとは、デバイス測定を実行するために指定する周波数のスパンのことです。

PNAシリーズ・アナライザの周波数レンジ:

周波数分解能

周波数設定の分解能は1 Hzです。

CW周波数

CW時間掃引またはパワー掃引をスティミュラスとする測定は、ある範囲の周波数ではなく単一の周波数で行われます。

注記:周波数値はセキュリティ上の理由で非表示にすることができます。

周波数レンジを設定するには2つの方法があります。

- レンジのスタート周波数とストップ周波数を指定
- レンジの中心周波数とスパンを指定

マウス手順

スタート周波数とストップ周波数で周波数レンジを指定する手順

1. ChannelメニューでStart/Stop...をクリックします。
2. Startボックスをクリックします。値を入力するか矢印ボタンを使って、掃引レンジのスタート周波数を選択します。
3. Stopボックスをクリックします。値を入力するか矢印ボタンを使って、掃引レンジのストップ周波数を選択します。
4. OKをクリックします。

中心周波数とスパンを使って周波数レンジを設定する手順

1. Channelメニューで、Center/Span...をクリックします。
2. Centerボックスをクリックします。値を入力するか矢印ボタンを使って、掃引レンジの中心周波数を選択します。
3. Spanボックスをクリックします。値を入力するか矢印ボタンを使って、掃引レンジの周波数スパンを選択します。
4. OKをクリックします。

フロントパネル・キー手順

スタート周波数とストップ周波数で周波数レンジを指定する手順

1. **CHANNEL SETUP**ブロックで、**Start/Center**または**Stop/Span**を押します。
2. **F1 (Start)**を押します。
3. 値を入力するか上下の矢印を押して、掃引レンジのスタート周波数を選択します。
4. **F2 (Stop)**を押します。
5. 値を入力するか上下の矢印を押して、掃引レンジのストップ周波数を選択します。
6. **Enter**を押します。

中心周波数とスパンを使って周波数レンジを設定する手順

1. **CHANNEL SETUP**ブロックで、**Start/Center**または**Stop/Span**を押します。
2. **F3 (Center)**を押します。
3. 値を入力するか上下の矢印を押して、掃引レンジの中心周波数を選択します。
4. **F4 (Span)**を押します。
5. 値を入力するか上下の矢印を押して、掃引レンジのスパン周波数を選択します。
6. **Enter**を押します。

測定パラメータ

概要　　概念　　パラメータの設定方法

本器では、下記のパラメータを使ってデバイスの電気的特性を測定できます。

- Sパラメータ(一定の比)
- 任意比(独自の比測定を選択)
- 絶対パワー

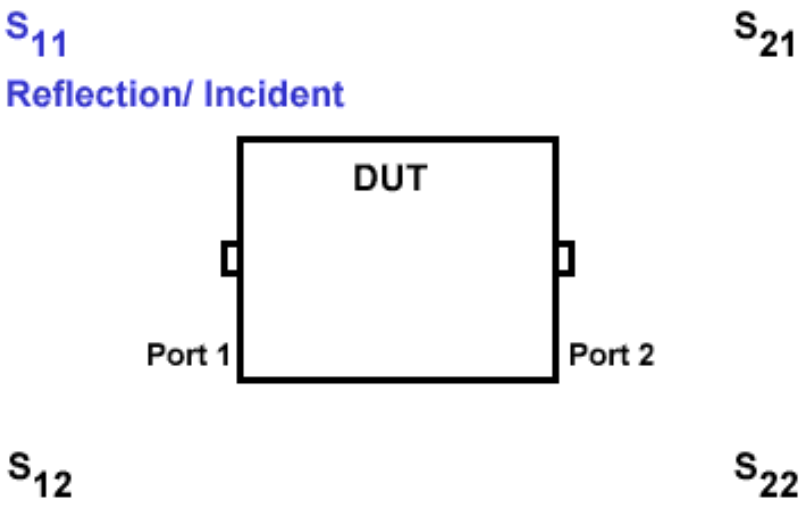
Sパラメータ

Sパラメータ(散乱パラメータ)は、デバイスによって信号がどのように変化するかを評価するために用いられます。被試験デバイスの反射および伝送特性が記述されます。Sパラメータの記法では、下記の番号付け方式によって2つの複素量の比を表します。

$S_{\text{out in}}$

ここで:

図の上でマウスを動かすと、Sパラメータのシグナル・フローを見ることができます。

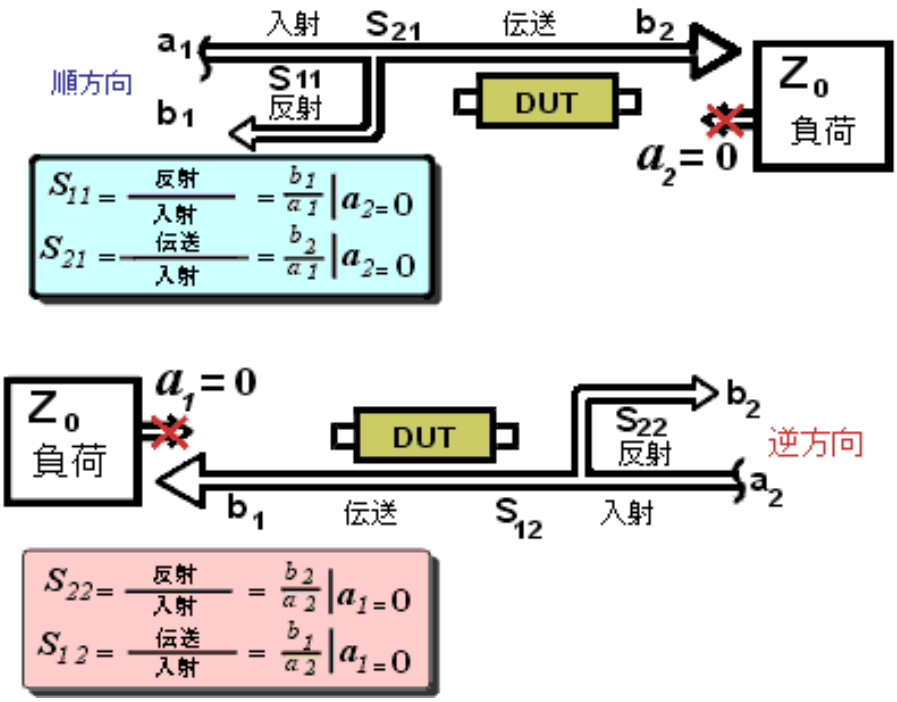


本器には2つのテスト・ポートがあり、1ポートまたは2ポートのデバイスを測定できます。Sパラメータ測定では入力テスト・ポートを定義できます。選択したポートに基づいて、測定の方向(順方向または逆方向)が自動的に切り換わります。これにより、1種類の接続で2ポート・デバイスの4個のSパラメータすべてを測定できます。

2ポート・デバイスのSパラメータは、S_{11}、S_{21}、S_{12}、S_{22}.の4個です。下の**アニメーション図**でSパラメータをマウスでポイントすると、シグナル・フローとSパラメータの関係を見ることができます。

2ポート・デバイスのSパラメータに関するさらに詳しい説明を下の図に示します。ここで:

- a = デバイスに入る信号
- b = デバイスを通って伝送される信号



注記: Sパラメータはすべて複素数の線形比です。

注記: 測定確度は、校正標準の品質と、測定接続の技法に依存します。

注記: Sパラメータ測定の確度は、負荷ポート(信号が入力されない方のポート)を終端する方法にも依存します。完全な負荷以外のものを使用すると、a_1またはa_2が0でなくなります(これはSパラメータの定義に反します)。正確なSパラメータ補正には、信号源および負荷整合を補正する2ポート誤差補正が非常に重要です。詳細については、「校正」を参照してください。

Sパラメータを使った一般的測定

Sパラメータは下記の一般的な測定項目と同等です。

Sパラメータを使った一般的な測定には下記のものがあります。

Sパラメータのフロー・グラフ

下のフロー・グラフは、被試験2ポート・デバイスまたは回路のシグナル・フローを図で表したものです。このフロー・グラフは本器のフロントパネルにも記載されています。

任意比

任意比を使うと、入力信号と基準信号の比を独自に選択できます。A、B、R1、R2の各レシーバが使用できます。任意比測定の一般的な用途には下記のものがあります。

- カスタム測定でA、B、R1、R2レシーバへの直接経路を使用し、独自の比測定を選択
- 本器のSパラメータ測定で実現されるよりも大きいダイナミック・レンジが必要な測定

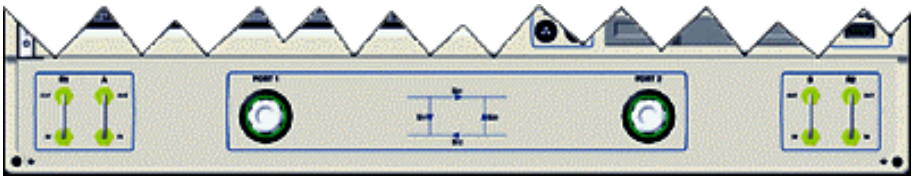
A、B、R1、R2の各レシーバに対しては、本器のフロントパネルから接続が可能です。

- フロントパネルの**A (in)**コネクタと**B (in)**コネクタは、AレシーバとBレシーバへの直接接続に使用します。これには下記の利点があります。
  - テスト・ポート結合器が信号経路から外れるため、感度が向上します。
  - AまたはBレシーバの経路から約15 dBの損失がなくなります。
- フロントパネルの**A (out)**コネクタと**B (out)**コネクタは、テスト・ポート1とテスト・ポート2の結合器の結合アームに直接アクセスするために使います。

- フロントパネルの**R1**コネクタと**R2**コネクタは、増幅器と結合器(ユーザが用意)を使った外部コンポーネントのハイパワー・テストなどの特別な機器設定のために使います。

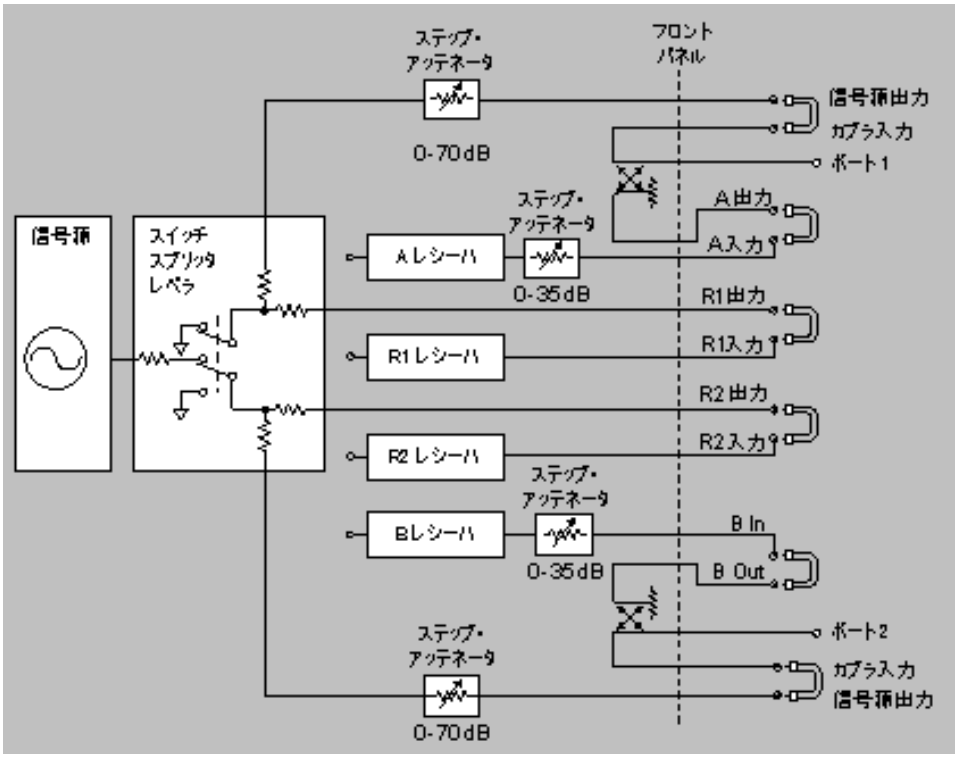
下の図は、本器内部の信号経路を示します。

下の図は、本器フロントパネル上の上記コネクタの位置を示します。



オプション015

構成可能テスト・セット・オプションが装備されている場合、信号源出力と結合器入力との間の信号経路にフロントパネル・アクセス・ループを追加できるようになります。また、両ポートのレシーバ経路に35 dBアッテネータ(5 dBステップ)が追加されます。これにより、低レベル信号の測定の際に感度を上げることができ、またコンポーネントや他の周辺機器を追加することでさまざまな測定アプリケーションに対応できます。詳細については、ハイパワー・コンポーネント測定のチュートリアルを参照してください。



絶対パワー

絶対パワー・パラメータは、A、B、R1、R2レシーバに入力されるデバイスの絶対パワーを調べるために使います。

注記: 位相測定や群遅延測定、およびアベレージングをオンにした測定では、シングル入力は使用できません。

ボタンをクリックすると手順が表示されます。

Sパラメータ測定

マウス手順

1. **Trace**メニューで**Measure**をクリックします。
2. 目的のSパラメータをクリックします。

フロントパネル・キー手順

1. **TRACE SETUP**ブロックで**Measure**を押します。
2. **F1**を押すとS_{11}が選択されます。
3. **F2**を押すとS_{21}が選択されます。
4. **F3**を押すとS_{12}が選択されます。
5. **F4**を押すとS_{22}が選択されます。
6. **Enter**を押します。

任意比測定

マウス手順

1. **Trace**メニューで**Measure**をクリックします。
2. **Measure...**をクリックします。
3. **Trace Type**ボックスで**Arbitrary Ratio**を選択します。
4. **Source Port**ボックスで信号源ポートを選択します。
5. **Input**ボックスで入力を選択します。
6. **Output**ボックスで出力を選択します。
7. **OK**をクリックします。

フロントパネル・キー手順

1. **COMMAND**ブロックで**Menu/Dialog**を押します。
2. **TRACE SETUP**ブロックで**Measure**を押します。
3. 下矢印を押して**Trace Type**ボックスで**Arbitrary Ratio**を選択します。
4. 右タブを押して**Source Port**ボックスに移動します。上下の矢印を押して信号源ポートを選択します。
5. 右タブを押して**Input**ボックスに移動します。上下の矢印を押して入力を選択します。
6. 右タブを押して**Output**ボックスに移動します。上下の矢印を押して出力を選択します。
7. **COMMAND**ブロックで**OK**を押します。

絶対パワー測定

マウス手順

1. **Trace**メニューで**Measure**をクリックします。
2. **Measure...**をクリックします。
3. **Trace Type**ボックスで**Unratioed Power**を選択します。
4. **Source Port**ボックスで信号源ポートを選択します。
5. **Input**ボックスで入力を選択します。
6. **OK**をクリックします。

フロントパネル・キー手順

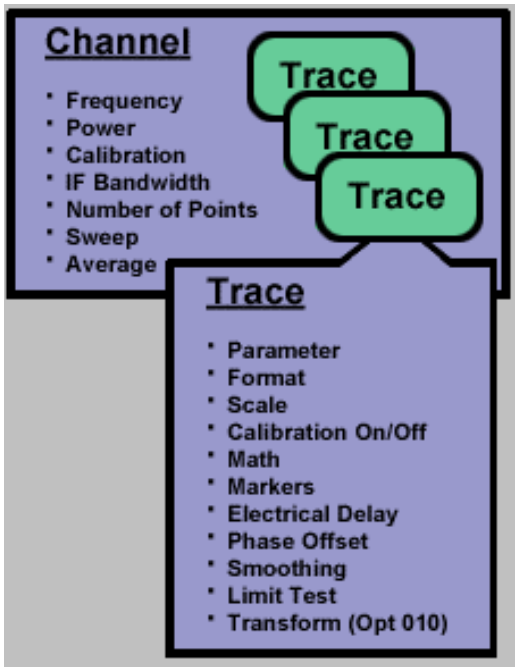
1. **COMMAND**ブロックで**Menu/Dialog**を押します。
2. **TRACE SETUP**ブロックで**Measure**を押します。
3. 下矢印を押して**Trace Type**ボックスで**Unratioed Power**を選択します。
4. 右タブを押して**Source Port**ボックスに移動します。上下の矢印を押して信号源ポートを選択します。
5. 右タブを押して**Input**ボックスに移動します。上下の矢印を押して入力を選択します。
6. **COMMAND**ブロックで**OK**を押します。

複数のトレースとチャネルの表示

複数の測定を構成するにはさまざまな方法があります。適切な構成ができたら、ウィンドウのサイズや位置を変更して、用途に合わせて本器の画面を見やすくカスタマイズできます。

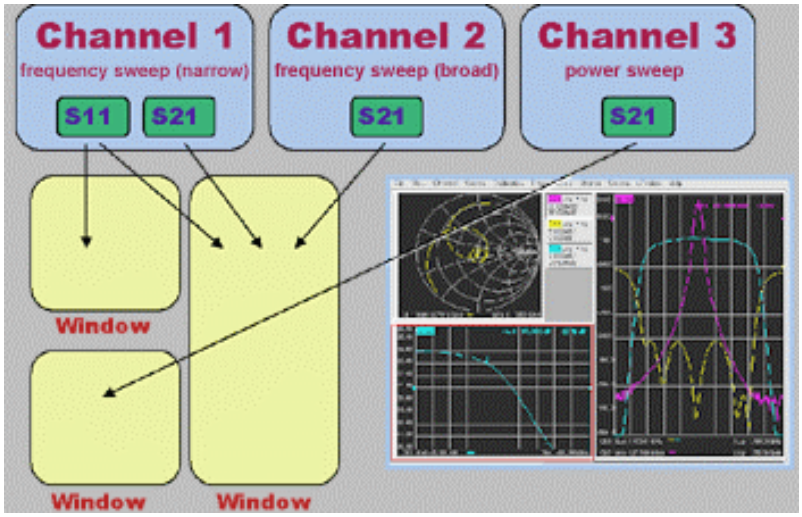
複数のチャネル、トレース、ウィンドウ

- **チャネル**は、データ収集に関連する属性の組合わせをそれぞれ独立に持っています。これらの属性は、そのチャネルに割り当てられたすべてのトレースに適用されます。本器は4つの独立したチャネルをサポートします。
- **トレース**は、収集データに対して実行される演算機能に関する属性の固有の組合わせを持っています。本器は1画面あたり最大16のトレースをサポートします。
- **ウィンドウ**はトレースを表示するために用いられます。本器は画面上に1、2、3、4のいずれかの数のウィンドウを表示できます。



以下のことが可能です。

- **複数のトレースとチャネル**を作成して、被試験デバイスの特性を測定し、表示できます。
- **複数のウィンドウ**を作成して、測定データを見やすく表示できます。



測定の限界

- 1画面あたり4つのウィンドウ
- 1ウィンドウあたり4つのデータ・トレースと4つのメモリ・トレース
- 1画面あたり16のトレース
- 4つの独立した測定チャネル

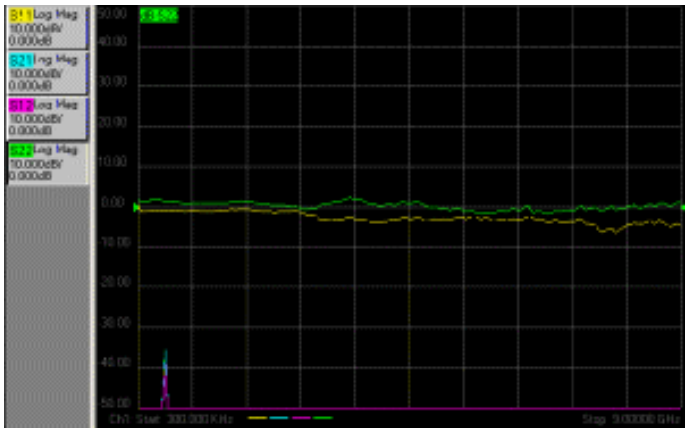
構成済み測定条件設定

この機能は、複数のトレース、チャネル、ウィンドウを使った一般的な測定条件設定を簡単に作成するためのものです。

- 構成済み測定条件設定は4種類あります。
- これらの設定はそれぞれ**新しいトレースを作成**します。
- 既存のトレースとその設定は、この機能を使用する前に保存しない限り失われます。

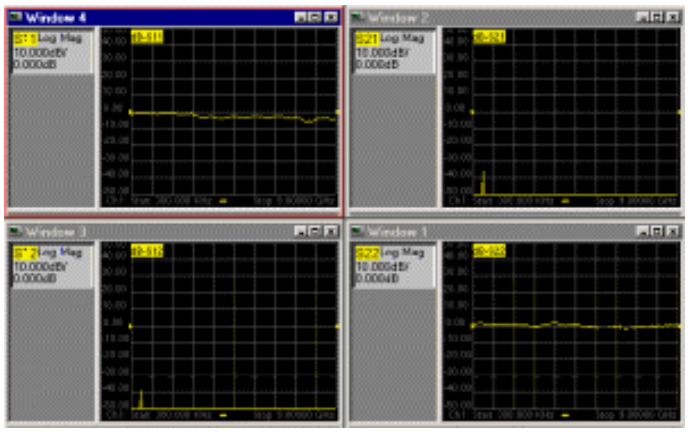
条件設定A

- 4つのトレースS11、S21、S12、S22
- 1つのウィンドウ、対数振幅フォーマット
- 1つのチャネル: ウィンドウ1にS11、S21、S12、S22



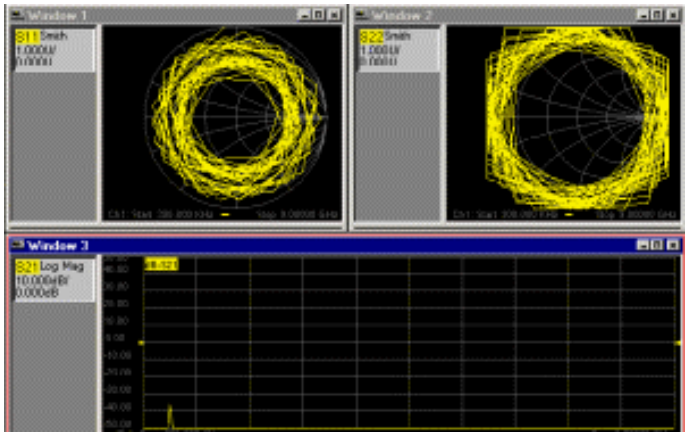
条件設定B

- 4つのトレース
- 4つのウィンドウ、対数振幅フォーマット
- 1つのチャネル: ウィンドウ1にS11; ウィンドウ2にS21; ウィンドウ3にS12; ウィンドウ4にS22



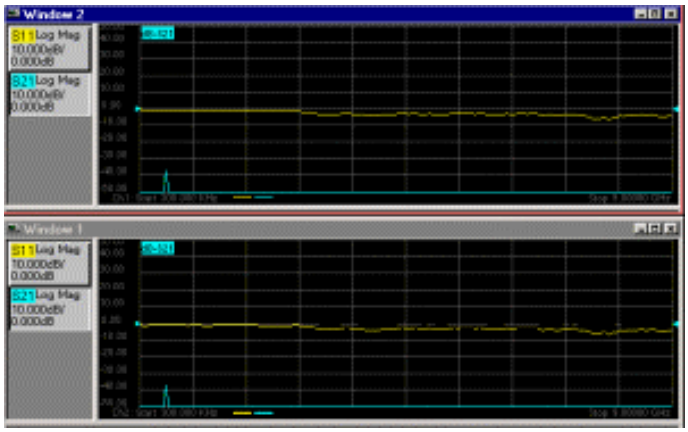
条件設定C

- 3つのトレース
- 3つのウィンドウ、スミス・チャート・フォーマット2つと対数振幅フォーマット1つ
- 1つのチャネル: ウィンドウ1にS11、スミス・チャート・フォーマット; ウィンドウ2にS22、スミス・チャート・フォーマット; ウィンドウ3にS21、対数振幅フォーマット



条件設定D

- 4つのトレース
- 2つのウィンドウ、どちらも対数振幅フォーマット
- 2つのチャネル: ウィンドウ1にチャネル1のS11とS21;
  ウィンドウ2にチャネル2のS11とS21



表示の配列

本器には以下の機能があります。

- マウスを使ってウィンドウを**移動**または**サイズ変更**
- **Window**メニューにあるタイル表示、カスケード表示、最小化、最大化の各機能を使ってウィンドウを**配列**
- **ウィンドウ配列機能**を使って、4つの異なる構成にウィンドウを**配列**
- 各ウィンドウの右上隅にあるMicrosoft Windowsの最大化、復元、最小化機能を使って、ウィンドウを**サイズ変更**または**非表示化**
- 複数ウィンドウ構成の1つのウィンドウを以下の方法で**最大化**
- ウィンドウの隅にある**最大化**ボタン
- **UTILITY**ブロックにある**Maximize/Restore**フロントパネル・キー(**DISPLAY**ブロックの**Window**キーで選択)

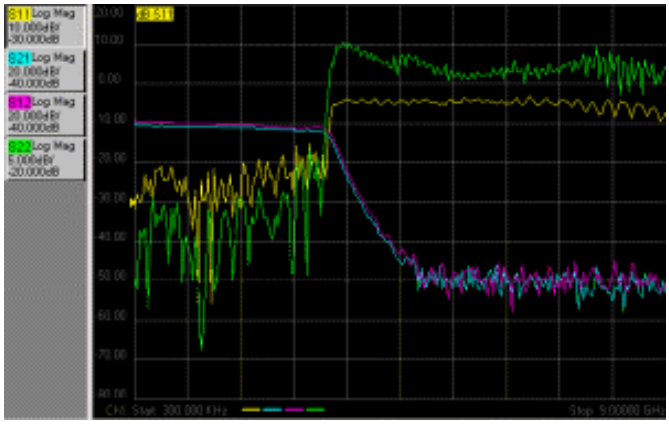
ウィンドウ配列機能

ウィンドウ配列機能

- 4つのウィンドウ構成が可能です。
- オーバレイ
- 2分割
- 3分割
- 4分割
- 既存のトレース、チャネル、フォーマットが用いられます。
- ウィンドウ内のトレースとチャネルはソート・アルゴリズムに基づいて表示されます。

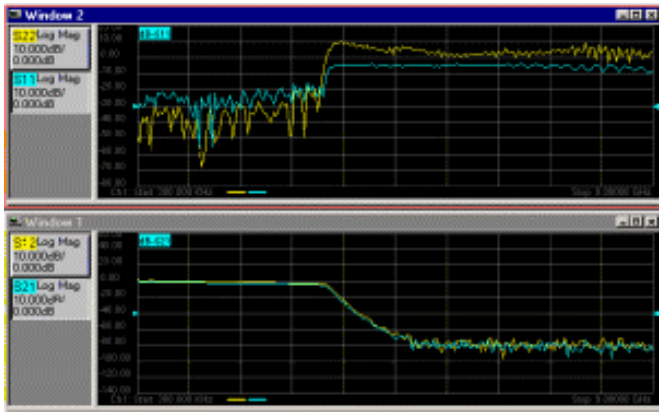
オーバレイ配列

- この構成では、既存のトレースすべてが1つのウィンドウに重なり合って表示されます。
- 既存のトレースが4つより多い場合、この選択は使用できなくなります。



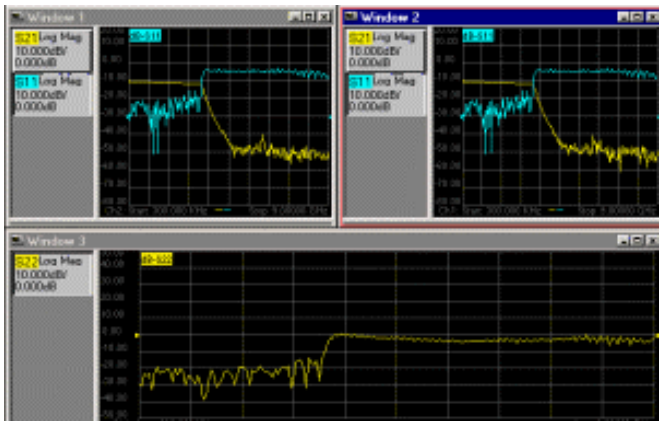
2分割配列

- この構成では、既存のすべてのトレースが、垂直に積み重ねられた2つのウィンドウに配置されます。
- 既存のトレースが8つより多い場合は、この選択は使用できなくなります。



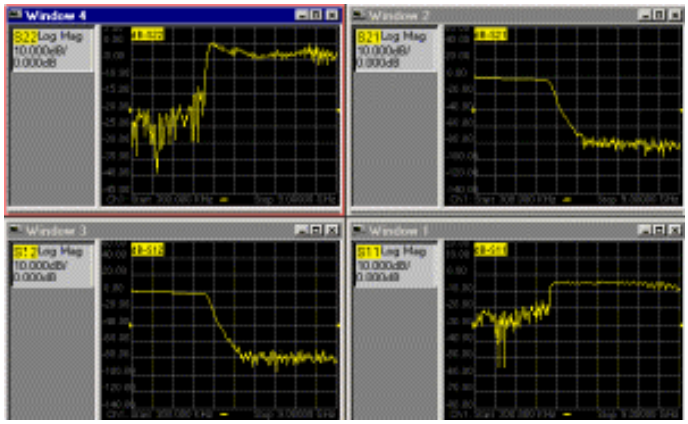
3分割配列

- この構成では、既存のトレースすべてが3つのウィンドウ(上に2つ、下に3つ)に配置されます。
- 既存のトレースが12より多い場合、この選択は使用できなくなります。



4分割配列

- この構成では、既存のトレースすべてが4つのウィンドウ(上下左右に1つずつ)に配置されます。



ソート・アルゴリズム

ウィンドウ配列機能のソート・アルゴリズムは以下のように設計されています。

- トレースをその性質に基づいて各ウィンドウに分配します。
- 共通の性質を持つトレースをまとめます。

ソート・アルゴリズムで用いられるトレースの性質を重要さの順に示します。

1. フォーマット: 円形(極座標、スミス・チャート)と直交座標(対数振幅、リニア振幅、群遅延など)
2. チャネル番号
3. 伝送/反射

注記: ウィンドウ1つあたりトレース4つまでという本器の制限は、このアルゴリズムより優先します。

構成済み測定条件設定

マウス手順

1. **Window**メニューで**Meas Setups**をクリックします。
2. 目的の測定条件設定をクリックします。

フロントパネル・キー手順

1. **DISPLAY**ブロックで**Measure Setups**を押します。
2. **F1 (Setup A)**を押すと1つのウィンドウに4つのトレースが表示されます。
3. **F2 (Setup B)**を押すと4つのウィンドウに4つのトレースが表示されます。
4. **F3 (Setup C)**を押すと3つのウィンドウに4つのトレースが表示されます。
5. **F4 (Setup D)**を押すと2つのウィンドウに4つのトレースが表示されます。

ウィンドウ配列構成の選択

マウス手順

1. **Window**メニューで**Arrange**をクリックします。
2. 目的のウィンドウ配列をクリックします。

フロントパネル・キー手順

1. **DISPLAY**ブロックで**Arrange Windows**を押します。
2. **F1 (Overlay)**を押すと1つのウィンドウにすべてのトレースが表示されます。
3. **F2 (Stack 2)**を押すと2つのウィンドウにすべてのトレースが表示されます。
4. **F3 (Setup C)**を押すと3つのウィンドウにすべてのトレースが表示されます。
5. **F4 (Setup D)**を押すと4つのウィンドウにすべてのトレースが表示されます。

複数ウィンドウ構成の1つのウィンドウを最大化

マウス手順

1. 最大化したいウィンドウの**最大化**ボタンを押します。
2. もう一度**最大化**ボタンを押すと、ウィンドウが元のサイズに戻ります。

フロントパネル・キー手順

1. **DISPLAY**ブロックで**Window**を押して、最大化したいウィンドウを選択します。
2. **UTILITY**ブロックで**Maximize/Restore**キーを押します。
3. もう一度**Maximize/Restore**キーを押すと、ウィンドウが元のサイズに戻ります。

掃引

概要 **概念** **リニア、パワー、CW掃引の設定方法** **セグメント掃引の設定方法**

掃引とは、指定されたスティミュラス値のシーケンスを使って測定データ・ポイントを連続的に収集することです。

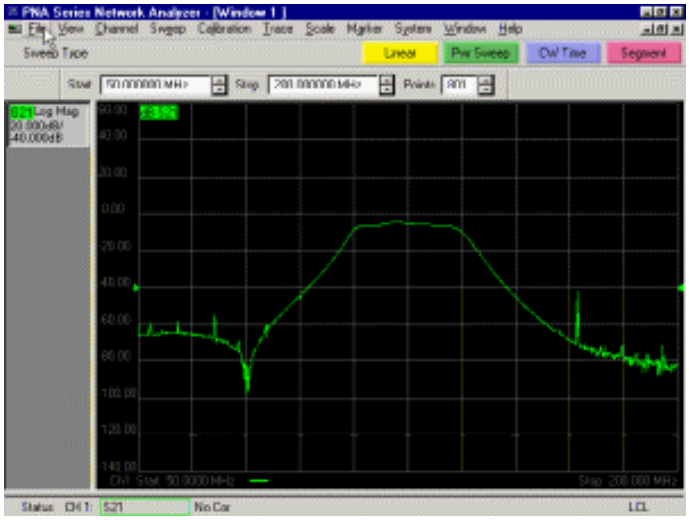
掃引のタイプ

掃引には4つのタイプがあります。

リニア周波数

これは**デフォルト**の掃引タイプです。周波数レンジ内の周波数がリニアに連続して掃引されます。

下の図は、掃引レンジ50.000 MHz～200.000 MHzでのフィルタのリニア周波数掃引を示します。



パワー

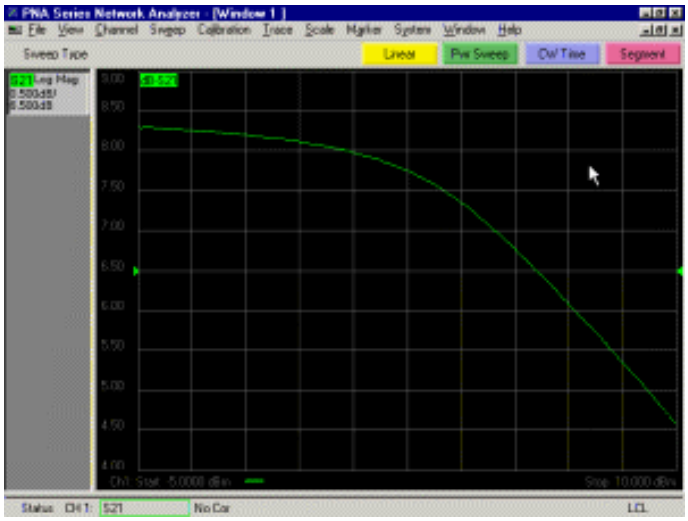
パワー掃引タイプは、パワーに敏感な回路の測定に用いられます。例としては、利得圧縮やAGC(自動利得制御)のスロープが挙げられます。

- 1つの周波数でパワーがスタート値からストップ値まで掃引されます。
- パワー掃引はステップ掃引です。

  信号源の振幅は、パワー掃引レンジ内の各データ・ポイントで離散的なパワー・ステップを取りながら増加します。

  ステップのサイズは、データ・ポイントの数とパワー・レンジによって決まります。
- 測定に必要なパワー掃引レンジが得られるように、信号源パワー・レンジと手動アッテネータ位置を選択します。 パワー・レベルを参照してください。

下の図は、パワー・レンジ-5.0 dBm～+10 dBmでの増幅器のパワー掃引を示します。

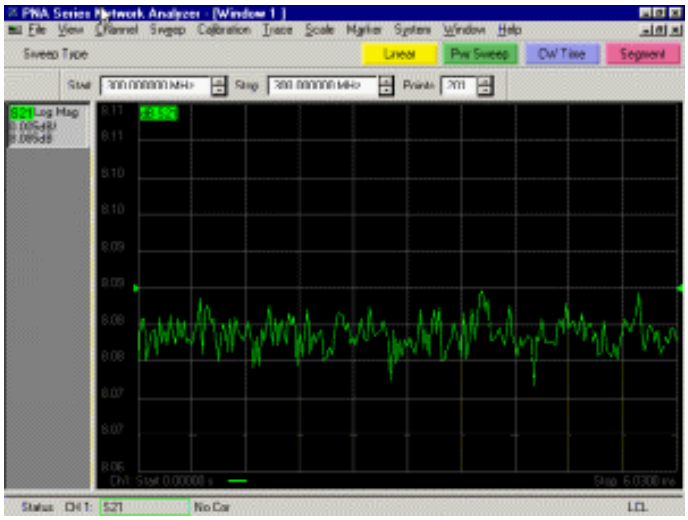


CW時間

CW時間掃引タイプは、オシロスコープの機能に似ています。

- 本器は1つの周波数に設定されます。
- 応答データは時間に対して表示されます。
- データは精密で一様な時間間隔で連続的にサンプリングされます。時間間隔は、掃引時間とポイント数-1で決まります。

下の図は、時間レンジ0～6 msでの増幅器のCW時間掃引を示します。



セグメント掃引

セグメント掃引タイプを使うと、任意の周波数ポイントのリストを掃引することができます。合計最大1601個のデータ・ポイントからなる周波数サブ掃引(「セグメント」と呼ぶ)を指定できます。セグメント掃引は、1つのチャネルのすべてのトレースを対象とします。各セグメントに対して独立した周波数レンジとポイント数を定義できます。また、各セグメントに下記のものを指定できます。

- 独立のIF帯域幅(測定の最適化参照)
- 独立のパワー・レベル
- 独立の掃引時間

掃引レンジ全体またはすべてのセグメントに対して測定校正を実行すると、各セグメントに対して校正済み測定が可能になります。

本器は下記のように動作します。

- 定義されたすべてのセグメントを周波数が低いものから順に並べ替えます。
- 各ポイントを測定します。
- 収集した全てのデータを複合した**1つのトレース**を表示します。

セグメント掃引タイプに関しては、下記のことを考慮する必要があります。

- セグメントの周波数レンジは、他のセグメントの周波数レンジと重ならないようにする必要があります。
- 機械的ステップ・アッテネータの過度の摩耗を避けるため、すべてのセグメントのパワー・レベルが同じアッテネータ・レンジ内にある必要があります。パワー・レベルを参照してください。

リニア周波数掃引

マウス手順

1. **Sweep**メニューで、**Sweep Type**をクリックします。
2. **Sweep Type**ボックスで**Linear Frequency**をクリックします。
3. ダイアログ・ボックスの**Sweep Properties**エリアで**Start**ボックスをクリックします。値を入力するか矢印ボタンを押して、掃引レンジのスタート周波数を選択します。
4. ダイアログ・ボックスの**Sweep Properties**エリアで**Stop**ボックスをクリックします。値を入力するか矢印ボタンを押して、掃引レンジのストップ周波数を選択します。
5. ダイアログ・ボックスの**Sweep Properties**エリアで**Points**ボックスをクリックします。値を入力するか矢印ボタンを押して、掃引レンジのデータ・ポイント数を選択します。
6. **OK**をクリックします。

フロントパネル・キー手順

1. **CHANNEL SETUP**ブロックで**Sweep Type**を押します。
2. **F1 (Linear)**を押します。
3. **CHANNEL SETUP**ブロックで**Sweep Setup**を押します。
4. **F2 (Points)**を押し、値を入力するか上下の矢印を押して、データ・ポイント数を設定します。
5. **Enter**を押します。

パワー掃引

マウス手順

1. **Sweep**メニューで**Sweep Type**をクリックします。
2. **Sweep Type**ボックスで**Power Sweep**をクリックします。
3. ダイアログ・ボックスの**Sweep Properties**エリアで**Start**ボックスをクリックします。値を入力するか矢印ボタンを押して、掃引レンジのスタート・パワー・レベルを選択します。
4. ダイアログ・ボックスの**Sweep Properties**エリアで**Stop**ボックスをクリックします。値を入力するか矢印ボタンを押して、掃引レンジのストップ・パワー・レベルを選択します。
5. ダイアログ・ボックスの**Sweep Properties**エリアで**CW Freq**ボックスをクリックします。値を入力するか矢印ボタンを押して、パワー掃引のCW周波数を選択します。
6. **OK**をクリックします。
7. **Sweep**メニューで**Number of Points**をクリックします。値を入力するか矢印ボタンを押して、掃引のパワー・レンジのデータ・ポイント数を選択します。
8. **OK**をクリックします。

フロントパネル・キー手順

1. **CHANNEL SETUP**ブロックで**Sweep Type**を押します。
2. **F2 (Pwr Sweep)**を押します。
3. **CHANNEL SETUP**ブロックで**Power**を押します。
4. **F1 (Start Power)**を押し、値を入力するか上下の矢印を押して、掃引のスタート・パワー・レベルを設定します。
5. **Enter**を押します。
6. **F2 (Stop Power)**を押し、値を入力するか上下の矢印を押して、掃引のストップ・パワー・レベルを設定します。
7. **Enter**を押します。
8. **CHANNEL SETUP**ブロックで**Start/Center**または**Stop/Span**を押します。
9. **F3 (CW Freq)**を押し、値を入力するか上下の矢印を押して、掃引のCW周波数を入力します。
10. **Enter**を押します。

CW時間掃引

マウス手順

1. **Sweep**メニューで、**Sweep Type**をクリックします。
2. **Sweep Type**ボックスで**CW Time**をクリックします。
3. ダイアログ・ボックスの**CW Freq**エリアで**Start**ボックスをクリックします。値を入力するか矢印ボタンを押して、掃引のCW周波数を選択します。
4. ダイアログ・ボックスの**Sweep Properties**エリアで**Sweep Time**ボックスをクリックします。値を入力するか矢印ボタンを押して、掃引時間を選択します。
5. ダイアログ・ボックスの**Sweep Properties**エリアで**Points**ボックスをクリックします。値を入力するか矢印ボタンを押して、パワー掃引のデータ・ポイント数を選択します。
6. **OK**をクリックします。

フロントパネル・キー手順

1. **CHANNEL SETUP**ブロックで**Sweep Type**を押します。
2. **F3 (CW Time)**を押します。
3. **CHANNEL SETUP**ブロックで**Start/Center**を押します。
4. **F1 (CW Freq)**を押し、値を入力するか上下の矢印を押して、掃引のCW周波数を入力します。
5. **Enter**を押します。
6. **CHANNEL SETUP**ブロックで**Sweep Setup**を押します。
7. **F1 (Time)**を押し、値を入力するか上下の矢印を押して、掃引時間を設定します。
8. **Enter**を押します。
9. **F2 (Points)**を押し、値を入力するか上下の矢印を押して、掃引のデータ・ポイント数を設定します。
10. **Enter**を押します。

セグメント掃引の手順

マウス手順

セグメント掃引の選択

1. **Sweep**メニューで、**Sweep Type**をクリックします。
2. **Sweep Type**ボックスで**Segment Sweep**をクリックします。
3. 掃引の各セグメントに独立したパワー・レベルを設定する場合は、ダイアログ・ボックスの**Sweep Properties**エリアにある**Independent Power Levels**チェック・ボックスをクリックします。
4. 掃引の各セグメントに独立したIF帯域幅を設定したい場合は、ダイアログ・ボックスの**Sweep Properties**エリアにある**Independent IF Bandwidth**チェック・ボックスをクリックします。
5. 掃引の各セグメントに独立した待ち時間を設定したい場合は、ダイアログ・ボックスの**Sweep Properties**エリアにある**Independent Dwell Time**チェック・ボックスをクリックします。
6. ダイアログ・ボックスの**Sweep Properties**エリアにある**Show Table**ボックスをクリックしてセグメント・テーブルを表示させます。
7. **OK**をクリックします。

セグメントを定義する手順:

1. セグメントを追加するには、**Sweep**メニューの**Segment Table**サブメニューで**Insert Segment** をクリックします。
2. セグメントを削除するには、削除するセグメント番号をクリックして選択します。**Sweep**メニューの**Segment Table**サブメニューで**Delete Segment** をクリックします。
3. セグメント・テーブルの最初のセグメントの**TESTA**ボックスをクリックします。矢印ボタンを押してセグメントをON状態に切り換えます。
4. セグメント・テーブルの最初のセグ**STRTA**メントのボックスをクリックします。このセグメントのスタート周波数の値を入力します。
5. セグメント・テーブルの最初のセグメントの**STOP**ボックスをクリックします。このセグメントのストップ周波数の値を入力します。
6. セグメント・テーブルの最初のセグメントの**POINTS**ボックスをクリックします。このセグメントのデータ・ポイント数の値を入力します。
7. この手順のステップ4で独立したパワー・レベルを選択した場合、セグメント・テーブルの最初のセグメントの**POWER**ボックスをクリックします。このセグメントのパワー・レベルの値を入力します。
8. この手順のステップ5で独立したIF帯域幅を選択した場合、セグメント・テーブルの最初のセグメントの**IFBW**ボックスをクリックします。このセグメントのIF帯域幅の値を入力します。
9. この手順のステップ6で独立した待ち時間を選択した場合、セグメント・テーブルの最初のセグメントの**DWL TIME**ボックスをクリックします。このセグメントの待ち時間の値を入力します。
10. **OK**をクリックします。

フロントパネル・キー手順

セグメント掃引の選択

1. **CHANNEL SETUP**ブロックで**Sweep Type**を押します。
2. **F4 (Segment)**を押します。
3. セグメント掃引の特性を定義または変更するには、**COMMAND**ブロックで**Menu/Dialog**を押し、**CHANNEL SETUP**ブロックで**Sweep Type**を押します。**Sweep Type**ダイアログ・ボックスが表示されます。
4. 右タブを押してダイアログ・ボックスの**Sweep Type**エリアに移動します。下矢印を押して**Segment Sweep**を選択します。
5. 掃引の各セグメントに独立したパワー・レベルを設定する場合は、右タブを押してダイアログ・ボックスの**Sweep Type**エリアにある**Independent Power Levels**ボックスに移動し、Clickを押します。
6. 掃引の各セグメントに独立したIF帯域幅を設定する場合は、下矢印を押してダイアログ・ボックスの**Sweep Type**エリアにある**Independent IF Bandwidth**ボックスに移動し、Clickを押します。
7. 掃引の各セグメントに独立した待ち時間を設定する場合は、下矢印を押してダイアログ・ボックスの**Sweep Type**エリアにある**Independent Dwell Time**ボックスに移動し、Clickを押します。
8. 右タブを押してダイアログ・ボックスの**Sweep Properties**エリアにある**Show Table**ボックスに移動し、セグメント・テーブルを表示させます。

セグメントを定義する手順:

1. **Click**を押します。**STATE**オプションとして**ON**と**OFF**が表示されます。上下の矢印を押して目的のステートに移動します。
2. 右タブを押してセグメント・テーブル**の****START**エントリに進みます。値を入力するか上下の矢印を押して、この掃引セグメントのスタート周波数を設定します。
3. 右タブを押してセグメント・テーブルの**STOP**エントリに進みます。値を入力するか上下の矢印を押して、この掃引セグメントのストップ周波数を設定します。
4. 右タブを押してセグメント・テーブルの**POINTS**エントリに進みます。値を入力するか上下の矢印を押して、この掃引セグメントのデータ・ポイント数を設定します。
5. **Sweep Type**ダイアログ・ボックスで**Independent IF Bandwidth**ボックスを選択した場合、右タブを押してセグメント・テーブルのIFBWエントリに移動します。値を入力するか上下の矢印を押して、この掃引セグメントのIF帯域幅を設定します。
6. **Sweep Type**ダイアログ・ボックスで**Independent Power Levels**ボックスを選択した場合、右タブを押してセグメント・テーブルの**POWER**エントリに移動します。値を入力するか上下の矢印を押して、この掃引セグメントのパワー・レベルを設定します。
7. **Sweep Type**ダイアログ・ボックスで**Independent Dwell Time**ボックスを選択した場合、右タブを押してセグメント・テーブルのTIME エントリに移動します。値を入力するか上下の矢印を押して、この掃引セグメントの待ち時間を設定します。

8. セグメントを追加するには、右タブを押して**Sweep**メニューに移動します。下矢印を押して**Segment Table**に移動します。右タブを押して**Segment Table**メニューを表示させます。下矢印を押して**Insert Segment**エントリに移動します。Clickまたは**COMMAND**ブロックのOKボタンを押して、**Insert Segment**を選択します。
9. セグメントを削除するには、削除するセグメントをセグメント・テーブルで選択します。右タブを押してSweepメニューに移動します。下矢印を押して**Segment Table**に移動します。右タブを押して**Segment Table**メニューを表示させます。下矢印を押して**Delete Segment**エントリに移動します。Clickまたは**COMMAND**ブロックのOKボタンを押して**Delete Segment**を選択します。
10. セグメント・テーブルの各エントリに対してステップ10～16を繰り返します。
11. **COMMAND**ブロックでOKボタンをクリックします。

トリガ

概要 **概念** **トリガの設定方法**

トリガとは、本器が測定掃引を実行するきっかけとなる信号です。また、トリガ設定によって、測定掃引がどのように開始され、本器がいつ掃引を停止して**ホールド**状態に戻るかが決まります。

本器では、トリガ機能をきわめて柔軟に設定することができます。

トリガ･モデル

トリガ・モデルのアニメーションでは、トリガ信号をボールとして表しています。目標は、ボールをチャネルに当てることです。このモデルには3つのチャネルがあります。チャネルにボールが当たると、そのチャネルの測定が開始されます。トリガ・モデルのアニメーションをご覧ください。

- 上の箱(トリガ・ソース)はボールの入った箱です。有効なトリガ・イベント(本器が掃引中でない場合に限って生成されます)が発生すると、ボールが箱から出てチャネルに向かいます。ボールがどこから来るかは、**トリガ・ソース**の設定で決まります。
- **内部**のボックス
- **外部**のボックス
- または、**手動**で1個ずつシステムに投入
- ボールが真ん中のボックスに入った場合、ボールがシステム内でどのように動くかは**トリガ・スコープ**設定で決まります。
- 設定が**グローバル**の場合、1個のボールが各チャネルに1個ずつの合計3個になります。3個のボールは誘導路によってそれぞれのチャネルに1個ずつ送られます。
- 設定が**チャネル**の場合、1個のボールがそのまま次の有効なチャネルに送られます。
- チャネルがボールを受け入れるかどうか、何個のボールを受け入れるかは、そのチャネルの**チャネル・トリガ・ステート**設定によって決まります。

トリガ･ソース

本器で選択可能なトリガ信号ソースには、内部、外部、手動の3つがあります。設定が選択されると、**存在するすべてのチャネルに対して**そのソースが用いられます。

- 有効なトリガ信号が生成されるのは、本器が掃引中でない場合に限ります。

内部トリガ(デフォルト)

- 本器が制御します。
- 本器が測定を終了すると、ただちにトリガ信号が発生します。

手動トリガ

手動トリガ信号は下記のどれかによって生成されます。

- Triggerメニューまたはダイアログ・ボックスの**Trigger!**をクリック
- リモート・トリガ・コマンドを送信
- フロントパネルの**Trigger**キーを押す

測定実行中に手動トリガ信号を送ろうとした場合、**トリガ要求は無視されます**。トリガ信号が認識されるためには、測定が終了する(本器が掃引していない状態になる)まで待つ必要があります。

外部トリガ

- リアパネルの補助入出力コネクタ(ピン19)から供給されます。
- TTL信号

- レベル・センシティブ: アクティブ・ハイまたはアクティブ・ロー
- パルス幅は1 μs以上、掃引時間以下でなければなりません(掃引時間より長いパルスは複数のトリガを発生させる可能性があります)。
- 外部トリガ・レベルをトリガ・レベルに維持することにより、連続掃引モードを実現できます。このモードをオーバライドするには、外部トリガ・メニューからシングル・モードを選択します。シングル・モードを選択すると、トリガ信号が有効ステートに最初に遷移したときだけ本器は応答し、それ以後はトリガがシングルにリセットされるまでホールド・モードに入ります。

外部トリガ・レディ

- トリガ・レディ・ラインも、リアパネルの補助入出力コネクタからアクセスできます(ピン18)。
- アクティブ・ローTTL信号
- トリガ・レディ・ラインがローのときに有効な外部トリガ信号を受信した場合、測定実行中はトリガ・レディ・ラインがハイになります。例えば、本器が交互掃引モードで2ポート校正を使用している場合、データを更新するには4回の掃引が必要です。トリガ・レディ・ラインは、4回の掃引がすべて終了し、データが更新されるまでハイ(ノット・レディ)になります。

トリガ・スコープ

トリガ・スコープは、どの測定チャネルがトリガ信号を受け取るかを指定します。トリガ・スコープ設定には**グローバル**と**チャネル**の2つがあります。トリガ・モデルのアニメーションで、トリガ・スコープによるトリガ信号フローの違いを確認してください。

グローバル(デフォルト)

- トリガ・ソースからのトリガ信号を動作中のすべてのチャネルに送ります。
- **1つのトリガ信号**で、**動作中のすべてのチャネル(ホールド・モードのチャネルを除く)**の測定データを収集できます。

チャネル

- トリガ・ソースからのトリガ信号をアクティブ・チャネルだけに送信します。
- どれかのチャネルでトリガ・ステートが**ポイント**に設定されている場合、自動的に選択されます。

アクティブ・チャネルの測定サイクルが終了すると、次のチャネルが自動的に選択されます。チャネル・トリガ・ステートがホールドのチャネル以外のチャネルが順番に選択されます。セットアップされているチャネルまたはホールド状態でないチャネルが1つしかない場合、アクティブ・チャネルは変更されません。

チャネル・トリガ・ステート

チャネル・トリガ・ステートを選択することにより、チャネルが受け入れるトリガ信号の数を指定できます。選択できるステートは、連続、グループ、シングル、ホールドの4つです。

- チャネルが**トリガ信号を受け入れるのは**、チャネルが**ホールド・モード**でないときだけです。

トリガ信号が受け入れられると、**そのチャネルのすべての測定が開始されます(Pointステートの場合を除く)**。例えば、チャネル3でS_{21}測定とS_{22}測定を行っている場合、1つのトリガ信号で両方の測定が開始されます。トリガ・モデルのアニメーションを見て、チャネル・トリガ・ステートによりトリガ信号のフローがどのように変化するかを確認してください。

指定された数のトリガ信号を受け取ったチャネルでは下記のことが起こります。

- アーミングされないホールド・ステートに入ります。
- トリガ信号は無視されます。

連続(デフォルト)

チャネルは**無制限の数のトリガ信号を受け入れます**。

グループ

チャネルは**指定された数のトリガ信号だけを受け入れます**。

シングル

チャネルは**1個のトリガ信号を受け入れます**。

ホールド

トリガ・イベントは無視されます。トリガが発生しても、ホールド・モードのチャネルには影響しません。このモードのチャネルは**新しいデータを受け入れません**(スムージングや時間領域などのポストプロセッシング機能を使えば、画面が更新される場合はあります)。

ポイント

チャネルは1ポイントずつ掃引されます。ポイント・ステートのチャネルがトリガ信号を受信すると、現在の掃引の次のデータ・ポイントが収集されます。ポイント・ステートのチャネルはそのチャネルの測定がすべて終了するまでトリガ信号を受け入れます。

Pointステートのチャネルでは、SingleステートとGroupステートの動作も異なります。

- **シングル**: そのチャネルのすべての測定が1回ずつ実行されるまで、チャネルはトリガ信号を受け入れます。例えば、101個のデータ・ポイントからなる1つのトレースを持つチャネルは、101個のトリガ信号を受け入れます。
- **グループ**: そのチャネルのすべての測定が指定した回数ずつ実行されるまで、チャネルはトリガ信号を受け入れます。例えば、101個のデータ・ポイントからなる1つのトレースを持つチャネルは、グループ設定が3の場合、303個のトリガ信号を受け入れます。

本器は1回の掃引でAとBの両方の入力を測定します。このため、

- S_{11}とS_{21}が同じ掃引で測定されます。
- S_{12}とS_{22}が同じ掃引で(反対側のポートから)測定されます。

すなわち、**ポイント・ステート**では、各ポイントでのS_{11}とS_{21}(またはS_{12}とS_{22})の2つの測定が同時に実行されます。

注記: 掃引設定はチャネルに依存します。チャネルが交互掃引に設定されている場合、1回の掃引で測定される入力は1つだけです。交互掃引に設定されているチャネルでは、1つのポイントで一度に実行される測定は1つだけです。

注記: ポイント・ステートはトリガ・スコープ設定がグローバルの場合は使用できません。どれかのチャネルがポイント・ステートに設定されている場合、トリガ・スコープはチャネルに設定されます。

トリガ設定とリスタート

リスタートは存在するすべてのチャネルを対象とします。**RESTART**コマンドを受信した場合、本器は掃引を停止し、チャネル・トリガ・ステートは下記のように変化します。

- **RESTART**受信時に**ポイント・ステート**にあったチャネルは、**ポイント・ステート**でなくなります。
- チャネルが**Holdステート**にあった場合、トリガ・カウントが1にリセットされます(チャネルはシングル・トリガ信号を受け入れます)。
- チャネルが**グループ・ステート**にあった場合、RESTART受信時と同じトリガ・カウントがそのまま用いられます。
- チャネルが**連続ステート**にあった場合、チャネルはそのまま無制限の数のトリガ信号を受け入れる状態になります。

その後、本器は掃引を再開します。トリガ・ソースが**手動**に設定されている場合、1個のトリガ信号が送信されます。既存のチャネルはそれぞれのチャネル・トリガ・ステートに従って応答します。画面上のトレースはそれぞれ最低1回は更新されます。

ボタンをクリックすると下記の手順が表示されます。

単純なトリガ設定

マウス手順

1. **Sweep**メニューで**Trigger**をクリックします。
2. **Continuous(連続)**、**Single(シングル)**、Hold(ホールド)、Restart(リスタート)の中から、目的の設定をクリックします。

フロントパネル・キー手順

1. **CHANNEL SETUP**ブロックで**Trigger**を押します。
2. **連続**に設定するには **F1**を押します。
3. **シングル**に設定するには**F2**を押します。
4. **ホールド**に設定するには**F3**を押します。
5. **リスタート**に設定するには**F4**を押します。

トリガの設定

マウス手順

1. **Sweep**メニューで**Trigger**をクリックします。
2. **Trigger...**をクリックします。
3. ダイアログ・ボックスの**Trigger Source**エリアで、**Internal(内部)**、**External(外部)**、**Manual(手動)**の中からトリガ・ソースを選択します。
4. ダイアログ・ボックスの**Trigger Scope**エリアで、**Global(グローバル)**または**Channel(チャネル)**の中からトリガ・スコープを選択します。
5. ダイアログ・ボックスの**Trigger State**エリアで、対象のチャネルを選択します。
6. ダイアログ・ボックスの**Trigger State**エリアで、**Continuous(連続)**、**Enable Groups(グループ)**、**Single(シングル)**、**Point Sweep(ポイント)**、**Hold(ホールド)**の中からトリガ・ステートを選択します。**Enable Groups**の場合、グループ数も入力します。
7. **OK**をクリックします。

フロントパネル・キー手順

1. **COMMAND**ブロックで**Menu/Dialog**を押します。**CHANNEL SETUP**ブロックで**Trigger**を押します。
2. **NAVIGATION**ブロックで下矢印を押して、**Internal(内部)**、**External(外部)**、**Manual(手動)**の中から目的の設定を選択します。
3. 右タブを押してダイアログ・ボックスの**Trigger Scope**エリアに移動します。
4. 下矢印を押して、**Global(グローバル)**または**Channel(チャネル)**の中からトリガ・スコープを選択します。
5. 右タブを押してダイアログ・ボックスの**Trigger State**エリアに移動します。
6. 対象のチャネルを選択します。
7. 下矢印を押して、**Continuous(連続)**、**Single(シングル)**、**Hold(ホールド)**、**Point Sweep(ポイント)**、**Enable Groups(グループ)**のどれかのトリガ・ステートに移動します。**Click**を押して選択します。**Enable Groups**を選択した場合、グループ数を入力します(**Point Sweep**が選択できるのは、**Manual**および**Channel**が設定されている場合だけです)。
8. **COMMAND**ブロックで**OK**を押します。

パワー･レベル

概要　　概念　　パワーの設定方法

パワー・レベルは、テスト・ポートにおけるアナライザの信号源信号のパワーです。

下記の項目ではパワー・レベルの詳細について説明しています。 ボタンをクリックすると下記の項目について知ることができます。

パワー･レンジ

注記: PNAモデル番号の特定レンジについては、周波数 レンジを参照してください。

ポート間のパワー結合

デフォルトでは、PORT 1とPORT 2のパワー・レベルは結合されています。 各ポートの出力パワー設定は同じです。測定アプリケーションによっては、 PORT 1とPORT 2で異なるパワー・レベルが必要となる場合があります。本器は、 ポート・パワーを非結合にすることができるので、 各ポートにパワーを個別に設定できます。

例えば、高利得増幅器の利得とリバース・アイソレーションを測定したい場合、 各ポートのパワーを個別に設定する必要があります。増幅器の入力ポートに必要なパワーが、 出力ポートに必要なパワーよりもはるかに低いからです。 詳細については、ポートを結合しない場合の減衰 を参照してください。

注意! 入力パワー・レベルが次に示す最大値を超えると、 アナライザのレシーバを損傷するおそれがあります: テスト・ポートへの 入力: +30 dBm (1.0 W) レシーバへの 直接入力: +15 dBm (31.6 mW)。

信号源減衰

アナライザ信号源は、プログラマブル・ステップ・アッテネータを使って、 フルの95 dBパワー・レンジをカバーします。アッテネータは、アナライザ基準経路内の入射パワーのレベルを変えずに、 被試験デバイスへのテスト信号のパワー・レベルを調整します。 これにより、以下が実現できます。

- 信号源信号の確度と性能の 向上
- より正確な信号源整合

アッテネータには8つの異なる位置があるので、 8つの異なるパワー・レンジのパワー・レベルが使用可能です。

- 各レンジは、25 dBの総仕様スパンを持ちます。
- 各パワー・レンジの最適レンジは、レンジ中央の10 dB帯域です。 このレンジで、信号源レベリング・システムの最高の 確度と性能が得られます。アッテネータ・レンジ の図を参照してください。
- アッテネータ設定は、手動でも 自動でも選択できます。

注記: フルの誤差補正確度が得られるのは、 校正を実行したときのパワー・レベルを使用した場合のみです。ただし 測定校正を実行したときと同じアッテネータ・レンジ内で パワーを変更したときには、比測定の場合、フルに近い確度で測定を実行できます (比測定でない場合は、確度が低下します)。

自動減衰

アッテネータ・コントロールがAuto (デフォルト)の場合、

- 測定器の全動作レンジ内で任意の パワー・レベルを選択できます。
- 全アッテネータ・レンジのうち最適な レンジが使用されるように、アッテネータの位置と信号源パワーが自動的に選択されます。

詳細については、減衰値および遷移ポイントを参照してください。

手動減衰

以下に示すような測定アプリケーションでは、信号源パワーとアッテネータを 手動で設定する必要があります。

- 反射増幅器の テスト (発振器または条件により不安定な増幅器)

  テストには、広い周波数レンジにわたって信号源との非常に良好なインピーダンス整合(20dBリターン・ロス)が必要となります。 広い周波数レンジでテストを実行するときには、アッテネータの方が減衰されていないアナライザの信号源よりも、 優れたインピーダンス整合を提供します。 詳細については、ハイパワー・ コンポーネント測定のチュートリアルを参照してください。

  - 最適な信号源整合を得るには、Auto Offを選択し、減衰レベルを10または20 dB(あるいは それ以上)に設定します。

- パワー掃引

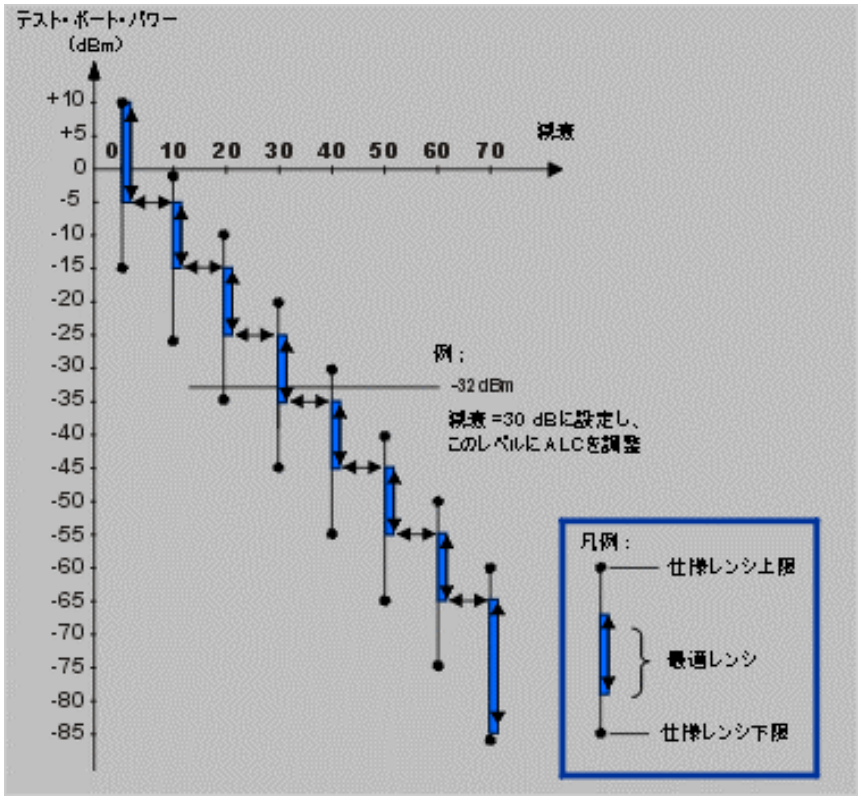
  連続スイッチングによる早期の磨耗を防ぐため、 アッテネータではパワー掃引中の設定の切り替えはできないようになっています。 Auto設定の場合、希望のパワー掃引レンジをフルにカバーできない 可能性があります。

  - 測定に必要なフルの希望パワー掃引レンジを得るには、Auto Offおよび信号源パワー・レンジとアッテネータ位置の 組合わせを選択します。

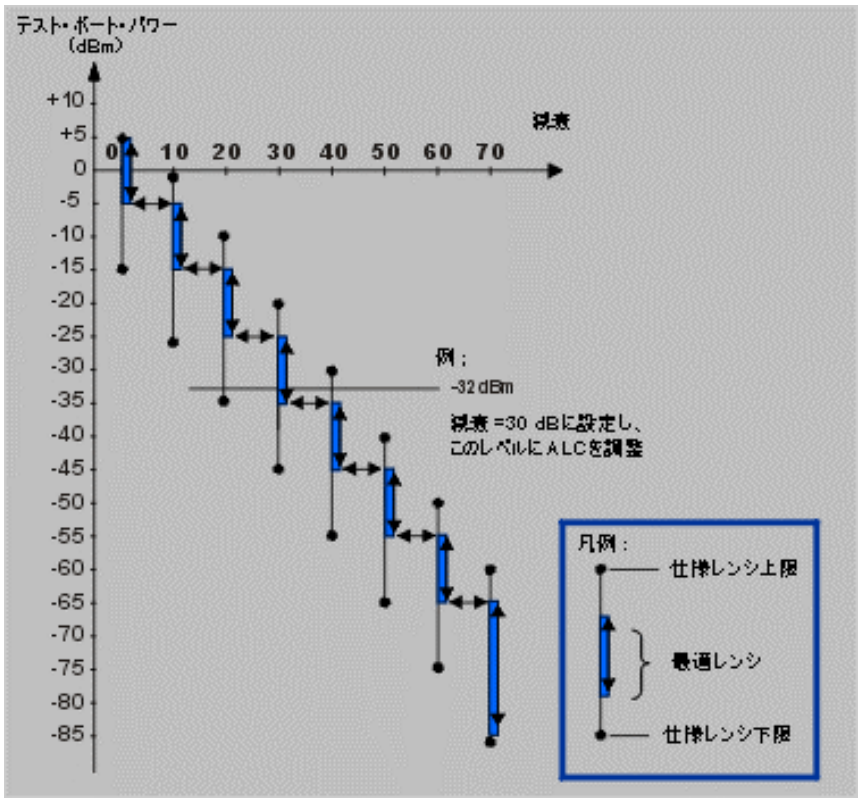
Manualアッテネータ・モードの場合、 以下を実行する必要があります。

- 信号源パワーを選択する。
- 減衰設定を選択する。
- 最適レンジで組合わせを選択する。

減衰値、300 kHz - 6 GHz



減衰値、6 GHz - 9 GHz



ポートを結合しない場合の減衰

テスト・ポート・パワーを結合しないときには、ポート1とポート2との間で ステップ・アッテネータを切り替えることができます。メカニカル・ステップ・アッテネータの早期の磨耗を防ぐため、 アナライザでは下記のことが行われます。

- 2つのパワー・レンジ間での連続的なスイッチングを防止しています。
- 各チャネルを1回測定した後、自動的にホールド・モード になります。

アベレージングをオンにすると、各チャネルに対して指定回数の 掃引を終了するまで、ホールド・モードに 入りません。

パワー・スロープ

パワー・スロープによって、周波数を高くしたときのケーブルとテスト・フィクスチャの パワー損失を補正できます。

- パワー・ スロープ機能をイネーブルにすると、掃引周波数が高くなるにつれてポートの出力パワーが 増加(または減少)します。
- パワー・スロープの 単位はdB/GHzです。
- パワー・スロープとして設定可能な値は0.5、 1、1.5、または2(正または負)だけです。

レシーバの減衰(オプション015を使用)

注意! 入力パワー・レベルが次に示す最大値を超えると、アナライザのレシーバを 損傷するおそれがあります: テスト・ポートへの 入力: +30 dBm (1.0 W) レシーバへの 直接入力: +15 dBm (31.6 mW)。

アナライザのレシーバへの損傷を防ぐには、 被試験デバイスからの出力信号を減衰する必要があります。 オプション015は、 両方のレシーバに内部アッテネータを提供します。

- レンジ: 0 ～ 35 dB
- 分解能: 5 dB

ボタンをクリックすると手順が表示されます。

パワー・レベルの設定

注意! 入力パワー・レベルが次に示す最大値を超えると、アナライザのレシーバを 損傷するおそれがあります: テスト・ポートへの 入力: +30 dBm (1.0 W) レシーバへの 直接入力: +15 dBm (31.6 mW)

マウス手順

1. Channel メニューで、Powerをクリックします。
2. Test Port Powerボックスで、値を入力するか、矢印ボタンを押して、 Port 1に対して希望のパワー・レベルを選択します。ポート・パワーは結合されているので(デフォルト)、 これによってPort 2のパワーも設定されます。
3. 各ポートに対してパワー・レベルを個別に設定したい場合は、 Port Power Coupled ボックスをクリックして、 ポートを非結合にします。Port ボックスで、希望のポートを選択します。この手順のステップ2を繰り返します。
4. OKをクリックします。

フロントパネル・キー手順

1. CHANNEL SETUPブロックで、Powerを押します。
2. F1 (Level)を押し、 値を入力するか、上矢印または下矢印を押して希望のパワー・レベルを 設定します。
3. Enterを押します。

被試験デバイスへの パワーのオフ

マウス手順

1. Channel メニューで、Powerをクリックします。
2. Power Onボックスをクリックし、パワーをOffステートに切り替えます。
3. OKをクリックします。

フロントパネル・キー手順

1. CHANNEL SETUPブロックで、Powerを押します。
2. F2 (ON/off)を押して、パワーをOffステートに切り替えます。

減衰の手動設定

マウス手順

1. Channel メニューで、Powerをクリックします。
2. ダイアログ・ボックスのAttenuator Control部分で、Auto ボックスをクリックし、アッテネータの自動制御をオフにし、手動制御をオンにします。
3. Attenuator Controlボックスをクリックし、値を入力するか、矢印ボタンをクリックして 希望の減衰を選択します。
4. OKをクリックします。

フロントパネル・キー手順

1. CHANNEL SETUPブロックで、Powerを押します。
2. F3 (Atten-Auto)を押し、 減衰制御を手動ステートに切り替えます。
3. 値を入力するか、上矢印または下矢印を押して、 希望の減衰量を選択します。
4. Enterを押します。

パワー・スロープの設定

マウス手順

1. Channel メニューで、Powerをクリックします。
2. ダイアログ・ボックスのPower Slope部分で、Slope チェック・ボックスをクリックします。
3. Slope (dB/GHz)ボックスをクリックし、値を入力するか、矢印ボタンをクリックして希望のスロープ値を選択します。
4. OKをクリックします。

フロントパネル・キー手順

1. CHANNEL SETUPブロックで、Powerを押します。
2. F4 (Slope)を押し、 値を入力するか、上矢印または下矢印を押して希望のパワー・スロープ値を 設定します。
3. Enterを押します。

アナライザ･レシーバの減衰レベルの設定(オプション015使用時のみ)

注意! 入力パワー・レベルが次に示す最大値を超えると、アナライザのレシーバを 損傷するおそれがあります: テスト・ポートへの 入力: +30 dBm (1.0 W) レシーバへの 直接入力: +15 dBm (31.6 mW)。

マウス手順

1. Channel メニューで、Powerをクリックします。
2. ダイアログ・ボックスのReceiver Attenuation部分で、Receiver AまたはReceiver Bのボックスをクリックします。値を入力するか、矢印ボタンを押して 希望のレシーバ減衰を選択します。
3. Enterを押します。

フロントパネル・キー手順

1. COMMAND ブロックで、Menu/Dialogを押します。
2. CHANNEL SETUPブロックで、Powerを押します。
3. 右タブを押し、Powerダイアログ・ボックスのReceiver Attenuation部分でReceiver AまたはReceiver Bを 選択します。 値を入力するか、上矢印または下矢印を押して希望の減衰量を選択します。
4. Enterを押します。
5. COMMAND ブロックで、OKを押します。

ステップ2. 測定の最適化

- 測定確度の改善
- アクセサリの影響の低減
- 低損失2ポート・デバイスの反射確度の改善
- ダイナミック・レンジの改善
- 電気長の長いデバイスの測定確度の改善
- 位相測定確度の改善
- トレース・ノイズの低減
- レシーバ・クロストークの低減
- データ・ポイント数の改善
- 測定安定度の改善
- 測定スループットの改善
- 掃引実行の高速化
- 複数の測定ステートの切り替え
- データ転送の高速化

測定確度の改善

ボタンをクリックすると、確度向上のための測定最適化の方法を見ることができます。

- ダイナミック・レンジの拡大
- データ・ポイント数の増加
- 位相測定確度の改善
- 電気長の長いデバイスの測定の改善
- 低損失2ポート・デバイスに対する反射確度の改善
- 測定安定度の改善
- トレース・ノイズの低減
- レシーバ・クロストークの低減
- アクセサリの影響の低減

アクセサリの影響

概要　　概念　　アクセサリの影響を減らす方法

デバイス測定の結果が、測定構成中のアクセサリによって影響を受ける場合があります。アクセサリの影響を減らすために、本器の2つの機能のどちらかを利用できます。

アクセサリの影響を減らすには、測定構成に合わせて最適の方法を選択します。

パワー・スロープによるケーブル損失の補正

長いケーブルなどのアクセサリが測定構成に存在するために周波数に応じたパワー損失が発生する場合、パワー・スロープ機能を使用します。この機能は、指定した率(dB/GHz)でアナライザの信号源出力を増加させるものです。

ゲーティングによる応答の選択的除去

ゲーティングとは時間領域の機能(オプション010)で、応答を数学的に除去するものです。反射応答と伝送応答のどちらにもゲーティングを設定できますが、結果はそれぞれ異なります。

- **反射応答にゲーティングを使用する**と、必要な応答(フィルタのリターン・ロスなど)が不要な応答(アダプタの反射やコネクタの不整合など)から分離されます。
- **伝送応答にゲーティングを適用する**と、電気長が長いマルチパス・デバイスの特定のパスが分離されます。

時間領域モードでゲーティングを適用した後、変換をオフにしてゲートがアクティブな状態でのデバイスの周波数応答を調べることができます。時間フィルタ・ゲートは下記の形状と動作のものが使用できます。

- **バンドパス形状**　通過領域外の応答を除去します。
- **ノッチ形状**　阻止領域内の応答を除去します。

詳細については、ゲート・フィルタを参照してください。

ボタンをクリックすると、アクセサリの影響を減らすための手順が表示されます。

ケーブル損失の補正

マウス手順

1. **Channel**メニューで**Power**をクリックします。
2. スロープ機能がまだオンになっていない場合、**Slope**チェック・ボックスをクリックします。
3. **dB/GHz**ボックスに、周波数掃引の際に信号源出力を増加させる率を入力します。**OK**をクリックします。

フロントパネル・キー手順

1. **CHANNEL SETUP**ブロックで**Power**を押します。
2. アクティブ・エントリ・ツールバーで**F4 (Slope)**を押します。
3. **ENTRY**ブロックに、周波数掃引の際に信号源出力を増加させる率を入力します。**Enter**を押します。

不要応答の除去(オプション010時間領域のみ)

マウス手順

1. **Trace**メニューで**Transform**をクリックします。
2. **Transform**チェック・ボックスをクリックして変換モードをオンにします。
3. **Transform Mode**の下で目的のモードをクリックします。
   - Low Pass Impulse(ローパス・インパルス)
   - Low Pass Step(ローパス・ステップ)
   - Band Pass(バンドパス)
4. ローパスを選択した場合、**Set Freq. Low Pass**をクリックします。
5. **Category**ボックスで下矢印をクリックして**Gating**を選択します。
6. **Gate Type**ボックスで下矢印をクリックしてゲート動作を選択します。
   - Bandpass(バンドパス)
   - Notch(ノッチ)
7. ゲート境界を設定するため、**Start**と**Stop**または、**Center** と**Span**を設定します。
8. **Gating**チェック・ボックスをクリックして、ゲーティング機能をオンにします。**OK**をクリックします。
9. **Trace**メニューで**Transform**をクリックします。
10. **Category**ボックスで下矢印をクリックして**Transform**を選択します。**Transform**チェック・ボックスをクリックし、**OK**をクリックします。

フロントパネル・キー手順

1. **COMMAND**ブロックで**Menu/Dialog**を押します。
2. **NAVIGATION** ブロックで右タブと下矢印を押して**Trace**に移動し、**Transform**まで移動します。**Click**を押します。
3. 右タブを押して**Transform**チェック・ボックスに移動し、**Click**を押します。
4. 右タブと下矢印を押して**Transform Mode**に移動し、目的の選択肢に移動します。
   - Low Pass Impulse(ローパス・インパルス)
   - Low Pass Step(ローパス・ステップ)
   - Band Pass(バンドパス)
5. ローパスを選択した場合、**Set Freq. Low Pass**に移動して**Click**を押します。
6. **Category**ボックスに移動し、下の**Gating**を選択します。

7. **Gate Type**ボックスに移動し、下でゲート動作を選択します。
   - Bandpass(バンドパス)
   - Notch(ノッチ)
8. ゲート境界を設定するため、**Start**と**Stop**または、**Center** と**Span**を設定します。
9. **Gating**チェック・ボックスに移動して**Click**を押し、**OK**を押します。
10. **Category**ボックスに移動し、下の**Transform**を選択します。
11. **COMMAND**ブロックで**Menu/Dialog**を押します。
12. **Trace**メニューに移動し、下の**Transform**に移動します。**Click**を押します。
13. **Transform**チェック・ボックスに移動し、**Click**を押して**OK**を押します。

低損失2ポート・デバイスの反射確度

概要　　概念　　反射確度の測定方法

1ポート校正を使った反射測定を正確に実行するには、測定対象でないポートを終端する必要があります。

1ポート校正を使って反射応答を正確に測定するには、反対側の(測定対象でない)ポートを終端する必要があります。これは特に、フィルタの通過帯域やケーブルなど、低損失の双方向デバイスの場合に重要です。1ポート校正では、方向性、信号源整合、周波数応答は補正されますが、負荷整合は補正されません。したがって、負荷整合誤差はできる限り小さくする必要があります。

注記: 2ポート校正では、負荷整合を含む12個の誤差項がすべて補正されます。

測定対象でないポートを終端するには下記の方法を使います。

- 高品質の終端負荷(校正キットに含まれるものなど)をデバイスの測定対象でないポートに接続します。この方法では、フルSOLT 2ポート校正に近い測定確度が得られます。
- デバイスの測定対象でないポートを本器に直接接続し、デバイス出力と本器との間に10 dBの精密アッテネータを挿入します。これにより、本器の実効負荷整合がアッテネータの値の約2倍(20 dB)改善されます。

ヒント:

- リバース・アイソレーションが大きい増幅器などのデバイスの測定では、負荷整合を考慮する必要はありません。

下のグラフは、被試験デバイスの出力を精密10 dBアッテネータで終端**した**場合と**しない**場合の測定の不確かさを示します。

凡例

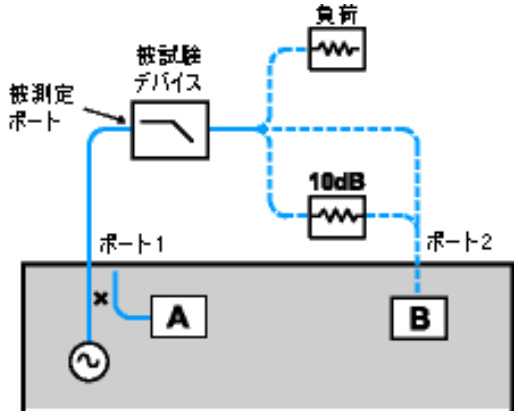
下の計算は、高品質10 dBアッテネータを付加することによる本器の負荷整合の改善を示します。

注記: 対応するリニア値を括弧の中に示します。

計算

低損失2ポート・デバイスに対して正確な反射測定を実行するには、下記のどれかを実行します。

- 1ポート校正を実行した後で、高品質の負荷をデバイス出力に接続します。
- 1ポート校正を実行した後で、デバイス出力と本器との間に10 dBアッテネータを挿入します。
- 2ポート測定校正を実行した後で、デバイス出力を直接本器に接続します。

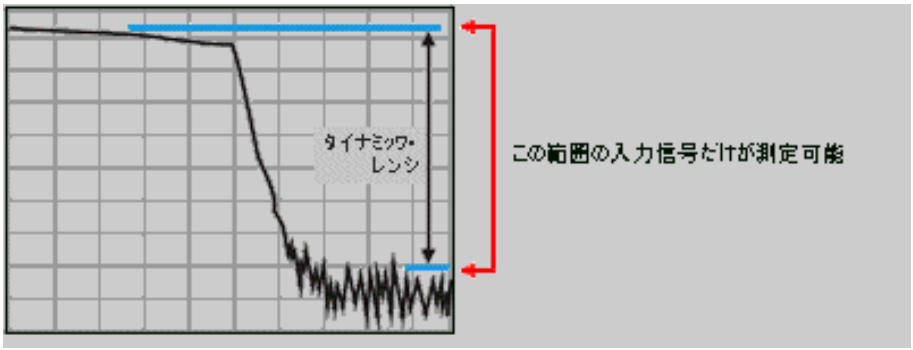


ダイナミック･レンジ

概要　　概念　　ダイナミック・レンジを広げる方法

ダイナミック・レンジとは、本器のレシーバの最大入力パワーと、測定可能な最小パワー(ノイズ・フロア)との差を指します。測定が有効であるためには、入力信号がこの範囲内になければなりません。

フィルタの通過帯域や除去比のように､信号振幅の変化が非常に大きいものを測定する場合、ダイナミック・レンジを広げることが重要になります。下にサンプル測定のダイナミック・レンジを示します。



測定の不確かさを減らすには､本器のダイナミック・レンジがデバイスの応答よりも域幅を狭広くなければなりません。例えば、信号がノイズ・フロアより10 dB以上大きければ、確度が向上します。ダイナミック・レンジを広げるには下記の方法が役立ちます。

注記: 本器の設定の中には相互に影響するものがあるため、ダイナミック・レンジを広げると測定時間が増加する場合があります。

デバイスの入力パワーを増やす

- 被試験デバイスに入力されるパワーを増やすと、デバイスの出力パワーを本器がより正確に検出して測定できるようになります。
- 本器のレシーバへの入力が大きすぎると、圧縮歪みが生じ、場合によってはレシーバを損傷するおそれがあります。

注意! レシーバ入力の損傷レベルは+15 dBmです。

ヒント: 本器にオプション015(構成可能テスト・セット)が装備されている場合、外部ブースタ増幅器を使って被試験デバイスへの入力パワーを増やすことにより､さらにダイナミック・レンジを広げることができます。

レシーバのノイズ･フロアを下げる

下記の方法を使って、ノイズ・フロアを下げ、本器のダイナミック・レンジを広げることができます。

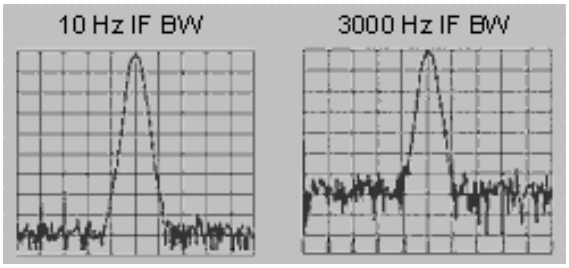
- IF帯域幅を狭める
- 掃引アベレージングを追加または変更する

ヒント: 本器のノイズ・フロアに近い信号を測定する場合、本器のレシーバの間のクロストークを減らす必要があります(レシーバ・クロストークを参照)。

IF帯域幅を狭める

- IFレシーバ帯域幅を狭めることにより、測定に対するランダム・ノイズの影響が減少します。IF帯域幅を10分の1にするごとに、ノイズ・フロアが10 dB下がります。ただし、IF帯域幅を狭めると掃引時間が長くなります。
- ネットワーク・アナライザは、受信信号をRF/マイクロ波周波数から41.67 kHzという低い中間周波数(IF)に変換します。IFバンドパス・フィルタの帯域幅はIF帯域幅と呼ばれ、40 kHzから最低1 Hzまで調整可能です。
- IF帯域幅はチャネルごと、またはセグメント掃引の各セグメントごとに独立に設定できます。

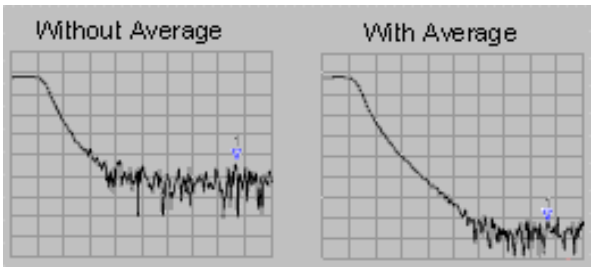
IF帯域幅を狭めることによる影響



掃引アベレージングの追加または変更

- 掃引アベレージングとは、測定に対するランダム・ノイズの影響を減らすプロセスのことです。
- 指定したアベレージング係数によって重み付けされた一連の掃引の平均値に基づいて各データ・ポイントが計算されます。
- 全掃引回数がアベレージング係数に等しくなるまで、新しい掃引がトレース平均に加えられていきます。
- アベレージングの回数(アベレージング係数)が大きいほど、ノイズ低減効果が大きく、ダイナミック・レンジが広がります。

掃引アベレージングの効果



ヒント:

- 測定アベレージング機能は、各チャネルに独立に指定できます。
- 全測定時間はアベレージング回数に応じて増加します。
- 周波数ノイズ一般を最小化したい場合、測定アベレージングを使ってもIF帯域幅を使っても同じノイズ低減効果が得られます。
- 非常に小さいノイズを最小化したい場合、測定アベレージングを使うほうがシステム帯域幅を狭めるよりも効果的です。

ボタンをクリックすると、ダイナミック・レンジを広げるための手順が表示されます。

デバイスの入力パワーを増やす

注意!被試験デバイスに利得がある場合、最大信号源出力パワーはレシーバを損傷するため使用できません。レシーバ入力の損傷レベルは+15 dBmです。

マウス手順

1. **Channel**メニューで**Power**をクリックします。
2. **Power On**チェック・ボックスを選択します。
3. **Port Selection**の下で、信号源出力として使用するポートを選択します。
4. **Test Port Power**ボックスに値を入力するか、矢印ボタンを押して必要な値を選択します。
5. **OK**をクリックします。

フロントパネル・キー手順

1. **CHANNEL SETUP**ブロックで**Power**を押します。
2. **NAVIGATION**ブロックで**F1 (Level)**を押します。
3. **ENTRY**ブロックにパワーの値を入力して、**Enter**を押します。

レシーバのノイズ・フロアを下げる

レシーバのノイズ・フロアを下げるには、IF帯域幅を狭めます。

マウス手順

IF帯域幅を狭める

1. **Sweep**メニューで**IF Bandwidth**をクリックします。
2. **IF Bandwidth**ボックスに値を入力するか、矢印ボタンを押して、狭める帯域幅の値を選択します。
3. **OK**をクリックします。

掃引アベレージングを追加または変更する

1. **Channel**メニューで**Average**をクリックします。
2. **Average ON**チェック・ボックスを選択して、アベレージング機能をオンにします。
3. **Average Factor**ボックスに値を入力するか、矢印ボタンを押して、測定の際のアベレージング回数を増やします。
4. **OK**をクリックします。

注記: Channel、**Restart Average**をクリックすると、アベレージング対象となる一連の掃引が新たに開始されます。

フロントパネル・キー手順

IF帯域幅を狭める

1. **CHANNEL SETUP**ブロックで**Sweep Setup**を押します。
2. **NAVIGATION**ブロックで**F3 (Bandwidth)**を押します。
3. **ENTRY**ブロックにIF帯域幅の値を入力し、**Enter**を押します。

掃引アベレージングを追加または変更する

1. **CHANNEL SETUP**ブロックで**Average**を押します。
2. **NAVIGATION**ブロックで**F1 (Avg#)**を押します。
3. **ENTRY**ブロックに測定のアベレージング回数を入力し、**Enter**を押します。
4. **F2 (Avg on/OFF)**を押してアベレージング機能をオンにします。

注記: F4 (Restart)を押すと、アベレージング対象の一連の掃引が新たに開始されます。

電気長の長いデバイスの測定

概要　　概念　　電気長の長い
デバイスの測定方法

被試験デバイスからの信号の周波数は、その時点でのアナライザの周波数と必ずしも正確に一致しません。これは不正確な測定結果の原因となります。

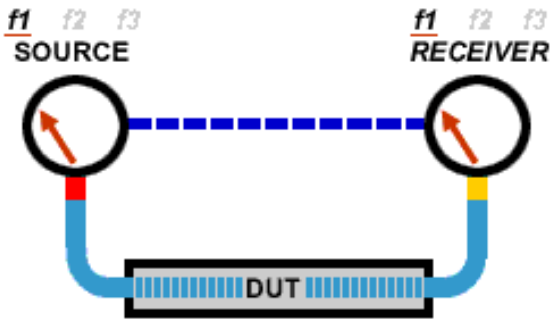
この問題を解決して測定確度を上げるには、3つの方法があります。

不正確な測定結果が生じる理由と、その状況に対処する方法について以下に説明します。

デバイス遅延によって不正確な結果が生じる理由

下の図はこの状況の例を示します。

- ネットワーク・アナライザでは、信号源とレシーバが相互に位相ロックされており、ある周波数スパンを同時に掃引します。
- 被試験デバイス(DUT)を通るシグナル・フローが、周波数ごとに異なる色で示されています。
- 図からわかるように、スティミュラス周波数がDUTを通過し、レシーバに前に届く本器は新しい周波数に同調します。これにより不正確な測定結果が生じます。



.

本器が長いケーブルを測定する場合、ケーブルの端での信号周波数が本器の信号源周波数に対して遅れます。この周波数シフトが本器のIF検出帯域幅(通常は数kHz)に対して無視できないくらい大きければ、IFフィルタのロールオフにより測定結果に誤差が生じます。

測定確度を改善する方法

電気長が長いデバイスを測定する場合、下記の方法で時間遅延を補正できます。

掃引速度を下げる

下記の方法を使うことにより、掃引速度を下げることができます。

- 掃引時間を長くする
- IF帯域幅を狭める
- ポイント数を増やす

ステップ掃引を使用

- アナログ掃引からステップ掃引に切り換えることにより、掃引速度を下げることができます。この動作モードでは、本器の信号源が各測定ポイントにステップ状に移行します。
- 各ステップ(測定ポイント)に本器が留まる時間(待ち時間)は設定可能です。

Rチャネルに電気長を追加

掃引速度を下げる代わりに、ケーブルまたはフィクスチャの電気長を補正することができます。

a. 本器のフロントパネルにあるRチャネル・ジャンパを取り外します。
b. ジャンパの代わりに、被試験デバイスとほぼ同じ長さのケーブルを接続します。
c. 本器を高速掃引に設定します。

この方法は、基準経路とテスト経路の遅延を等しくすることにより、本器の比例法伝送測定に周波数シフト誤差が生じないようにするものです。

注記: この方法は、ケーブルまたはフィクスチャから遅延が生じる場合に有効です。デバイスの遅延が大きい場合、この方法が使用できるのは未校正測定だけです。

ボタンをクリックすると、電気長の長いデバイスを補正する手順が表示されます。

掃引速度を下げる

マウス手順

掃引時間を長くする

1. **Sweep**メニューで**Sweep Time**をクリックします。
2. **Sweep Time**ボックスに値を入力するか、矢印ボタンを押して、掃引時間の値を選択します。
3. **OK**をクリックします。

IF帯域幅を狭める

1. **Sweep**メニューで**IF Bandwidth**をクリックします。
2. **IF Bandwidth**ボックスに値を入力するか、矢印ボタンを押して、帯域幅を狭める値を選択します。
3. **OK**をクリックします。

ポイント数を増やす

1. **Sweep**メニューで**Number of Points**をクリックします。
2. リスト中の値を選択するか、**Custom**を選択して任意の値を入力します。
3. **OK**をクリックします。

フロントパネル・キー手順

掃引時間を長くする

1. **Sweep Setup**、**F1 (Time)**を押します。
2. **ENTRY**ブロックから掃引速度の値を入力し、**Enter**を押します。

IF帯域幅を狭める

1. **Sweep Setup**、**F3 (Bandwidth)**を押します。
2. **ENTRY**ブロックからIF帯域幅の値を入力し、**Enter**を押します。

ポイント数を増やす

1. **CHANNEL SETUP**ブロックで**Sweep Setup**を押します。
2. **F1 (Points)**を押し、上下の矢印を使って値に移動し、**Click**を押して表示された値を選択します。または、**ENTRY**ブロックから値を入力して**Enter**を押します。

ステップ掃引を使用

マウス手順

1. **Sweep**メニューで**Sweep Setup**をクリックします。
2. **Stepped Sweep**チェック・ボックスをクリックします。
3. **Dwell Time**ボックスに値を入力するか、矢印ボタンを使って、各測定ポイントに本器が留まる時間を入力します。
4. **OK**をクリックします。

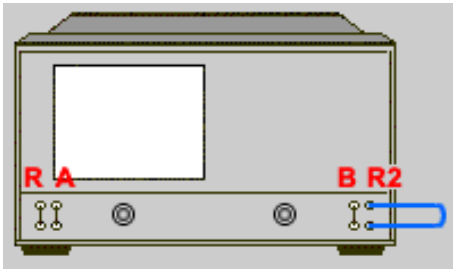
フロントパネル・キー手順

1. **COMMAND**ブロックで**Menu/Dialog**を押します。**CHANNEL SETUP**ブロックで**Sweep Setup**を押します。
2. **NAVIGATION**ブロックで右タブを押して**Stepped Sweep**チェック・ボックスに移動し、**Click**を押してステップ掃引モードを選択します。
3. 右タブを押して**Dwell Time**ボックスに移動し、上下の矢印を使って、各測定ポイントに本器が留まる時間を入力します。または、**ENTRY**ブロックから値を入力して**Enter**を押します。
4. **COMMAND**ブロックで**OK**を押します。

Rチャネルに電気長を追加

1. 被試験デバイスと同じ電気長のケーブルを用意します。
2. フロントパネルを通じて、Rチャネルにケーブルを接続します。
   - S_{11}およびS_{21}測定の場合は、R1チャネルにケーブルを追加します。
   - S_{22}およびS_{12}測定の場合は、R2チャネルにケーブルを追加します。

S_{22}およびS_{12}測定の構成



位相測定確度

概要　　概念　　位相確度の改善方法

位相測定の確度を改善するために、本器のさまざまな機能が利用できます。
正確な位相測定を実行するには、下記の機能が役立ちます。

電気遅延

- 電気遅延機能を使うと、デバイスによるリニア位相シフトを補正することができます。これにより、デバイスのリニア位相からの偏移だけを観察することができます。
- 電気遅延は、可変長の無損失伝送ラインを数学的にシミュレートする機能です。
- 電気遅延は測定トレースごとに独立に設定できます。

ポート延長

- ポート延長機能は、校正の後で測定基準面を移動するものです。これにより、校正の再実行を避けられる場合があります。ポート延長が役立つ場合として、下記の2例が挙げられます。
  - 校正を実行した後で、ある長さのケーブルを測定構成に追加しなければならなくなった場合。ポート延長を使うことで、特定のポートにケーブル長が追加されたことを本器に教えることができます。
  - デバイスがテスト・フィクスチャ内部にあるために、デバイスに対して直接校正を実行できない場合。ポート延長を使えば、フィクスチャによる時間遅延(位相シフト)を補正することができます。
- ポート延長は、特定のポートに対応するすべての測定に自動的に電気長を適用します。
- フロントパネルにあるレシーバ直接アクセス・コネクタを使用している場合は、Aレシーバ とBレシーバに電気長を追加します。

注記: ポート延長では、ケーブル、アダプタ、フィクスチャによる損失や不整合は補正されません。電気長の追加分だけが補正されます。

位相オフセット

位相オフセットは、最大360°までの指定した量だけ位相測定を調整するものです。この機能は下記のように使います。

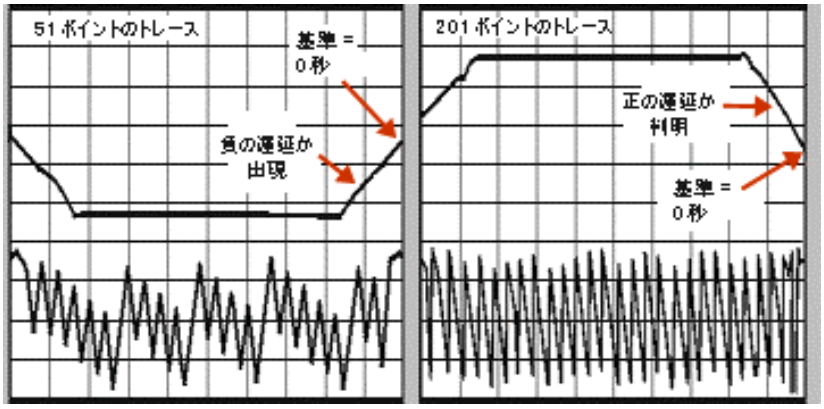
- **位相測定の表示を改善するため。**これは振幅測定で基準レベルを変えるのと似ています。位相応答を中央に配置したり、画面に合わせて整列したりできます。
- **測定で見込まれる位相シフトをエミュレートするため。**例えば、後でケーブルを追加することがわかっている場合、そのケーブルの長さによる位相シフトを位相オフセットで追加することにより、実際のデバイス測定結果をシミュレートすることができます。

周波数ポイントの間隔

- 本器は、離散的な周波数ポイントでデータをサンプリングした後、ポイントをつなげて画面上のトレースを作成します。
- デバイスによる位相シフトが隣接する周波数ポイントの間で180°を超える場合、位相スロープが逆転したように表示されることがあります。これは、データがアンダーサンプリングされ、エリアジングが発生したことが理由です。

群遅延測定の際に位相スロープが逆転すると、群遅延の符号が変わります。例えば、下の図はSAWバンドパス・フィルタの測定結果です。

- 1つめの測定には51個のポイントがあり、負の群遅延が示されていますが、これは物理的に不可能です。応答が0秒の基準ラインより下にあるからです。
- 2つ目の測定ではポイントが201個に増えており、正の群遅延が示されています。すなわち、応答は0秒の基準ラインより上にあります。



ヒント:

- 測定にエリアジングが発生しているかどうかをチェックするには、周波数ポイントの間隔を狭めてみて、表示されるデータが変化するかどうかを観察します。エリアジングが発生している場合、修正するにはポイント数を増やすか、周波数スパンを狭めます。

ボタンをクリックすると、位相測定の確度を改善する手順が表示されます。

リニア位相シフトを補正

マウス手順

1. **Scale** メニューで**Electrical Delay**をクリックします。
2. **Electrical Delay**ボックスに値を入力するか矢印ボタンを押して、必要な値を選択します。
3. **Velocity Factor**ボックスに値を入力するか、矢印ボタンを押して0～1.0の値を選択します。これは、測定対象デバイスの媒体の速度因子を示します。

- 1.0は真空中の光速に対応します。
- ポリエチレン誘電体中の代表的な光速は0.66にあたります。
- テフロン誘電体中の代表的な光速は0.70にあたります。

4. **OK**をクリックします。

フロントパネル・キー手順

1. **COMMAND**ブロックで、**Menu/Dialog**を押します。
2. **NAVIGATION**ブロックで右タブと下矢印を押して**Scale**メニューに移動し、**Electrical Delay**まで移動します。**Click**を押します。
2. **Electrical Delay**ボックスに**ENTRY**ブロックから値を入力して**Enter**を押すか、上下の矢印を押して必要な値を選択します。**Click**を押します。
3. 右タブを押して**Velocity Factor**に移動します。
4. **Velocity Factor**ボックスに**ENTRY**ブロックから値を入力するか、上下の矢印を押して必要な値を選択し、**Click**を押します。0～1.0の値を選択します。これは測定対象のデバイスの媒体の速度因子を示します。

- 1.0は真空中の光速に対応します。
- ポリエチレン誘電体中の代表的な光速は0.66にあたります。
- テフロン誘電体中の代表的な光速は0.70にあたります。

基準面を電気的に移動

マウス手順

1. **Calibration**メニューで**Port Extensions.**をクリックします。
2. **Port Extensions**チェック・ボックスをクリックします。
3. 基準面を延長するポートに対して延長値を指定します。
4. **Velocity Factor**ボックスで0～1.0の値を選択します。これは校正後に追加したケーブルまたはデバイスの媒体の速度因子を示します。**OK**をクリックします。

- 1.0は真空中の光速に対応します。
- ポリエチレン誘電体中の代表的な光速は0.66にあたります。
- テフロン誘電体中の代表的な光速は0.70にあたります。

ヒント:

十分な遅延の値を入力したかどうかは次のようにしてわかります。

A. デバイスの代わりにショートを接続します。

B. 位相応答がフラットになるまでポート延長を調整します。

注記: ほとんどのショート校正標準は遅延が0でないため、この方法で遅延を調整するとショート遅延の2倍に等しい遅延誤差が生じます。ショート校正標準のオフセット遅延は、標準定義を調べればわかります。

フロントパネル・キー手順

1. **COMMAND**ブロックで、**Menu/Dialog**を押します。
2. **NAVIGATION**ブロックで右タブと下矢印を押して**Calibration**メニューに移動し、**Port Extensions**まで移動します。**Click**を押します。
3. **Click**を押して**Port Extensions**チェック・ボックスを選択します。
4. 右タブを押して、基準面を延長する入力またはポートのボックスに移動します。値を入力して**Enter**を押します。
5. 右タブを押して**Velocity Factor**ボックスに移動します。
6. **Velocity Factor**ボックスで上下の矢印を使って0～1.0の値を選択します。これは校正後に追加したケーブルまたはデバイスの媒体の速度因子を示します。**OK**を押します。

- 1.0は真空中の光速に対応します。
- ポリエチレン誘電体中の代表的な光速は0.66にあたります。
- テフロン誘電体中の代表的な光速は0.70にあたります。

ヒント:

十分な遅延の値を入力したかどうかは次のようにしてわかります。

A. デバイスの代わりにショートを接続します。

B. 位相応答がフラットになるまでポート延長を調整します。

注記: ほとんどのショート校正標準は遅延が0でないため、この方法で遅延を調整するとショート遅延の2倍に等しい遅延誤差が生じます。ショート校正標準のオフセット遅延は、標準定義を調べればわかります。

位相測定結果にオフセットを付加

マウス手順

1. **Scale** メニューで**Phase Offset**をクリックします。
2. **Phase Offset**ボックスに値を入力するか、矢印ボタンを押して必要な値を選択します。
3. **OK**をクリックします。

フロントパネル・キー手順

1. **COMMAND**ブロックで**Menu/Dialog**を押します。
2. **NAVIGATION**ブロックで右タブと下矢印を押して**Scale**メニューに移動し、**Phase Offset**まで移動します。**Click**を押します。
3. **Phase Offset**ボックスに**ENTRY**ブロックから値を入力して**Enter**を押すか、上下の矢印を押して必要な値を選択します。**Enter**、**OK**を押します。

エリアジングのチェック

マウス手順

1. **Sweep**メニューで**Number of Points**をクリックします。
2. リスト中の値を選択するか、**Custom**を選択してキーボードまたは**ENTRY**ブロックから現在よりも小さい値を入力します。
3. **OK**をクリックします。

   表示されたデータが変化した場合、ポイント数を増やすか、周波数スパンを狭めます。

フロントパネル・キー手順

1. **CHANNEL SETUP**ブロックで**Sweep Setup**を押します。
2. **F2 (Points)**を押します。
3. **NAVIGATION**ブロックで上下の矢印を押して値を小さくし、**Click**を押して表示された値を選択します。

   表示されたデータが変化した場合、ポイント数を増やすか、周波数スパンを狭めます。

トレース・ノイズ

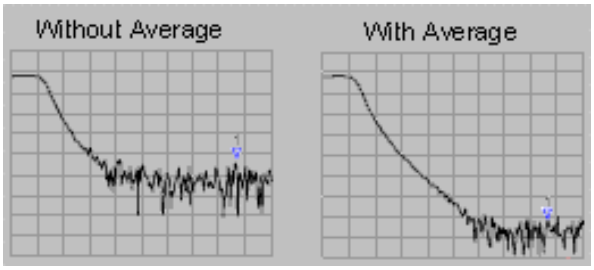
概要　　概念　　トレース・ノイズを減らす方法

トレース・ノイズを減らすために、本器のいくつかの機能が使用できます。
データ・トレースに対するノイズの影響を減らすには、下記の機能が役立ちます。

掃引アベレージング

- 掃引アベレージングとは、測定に対するランダム・ノイズの影響を減らすプロセスのことです。
- 指定したアベレージング係数によって重み付けされた一連の掃引の平均値に基づいて各データ・ポイントが計算されます。
- 全掃引回数がアベレージング係数に等しくなるまで、新しい掃引がトレース平均に加えられていきます。
- アベレージングの回数(アベレージング係数)が大きいほど、ノイズ低減効果が大きく、ダイナミック・レンジが広がります。
- ノイズの低減に関しては、掃引アベレージングとIF帯域幅を狭めることとは同じ効果があります。

掃引アベレージングの効果



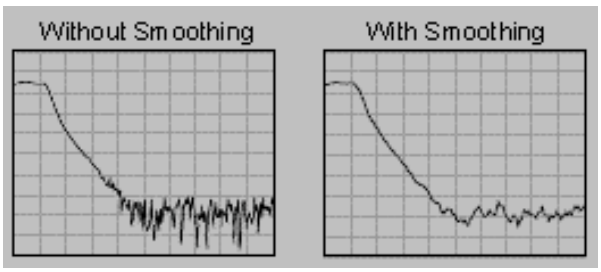
トレース・スムージング

スムージングを適用することにより、広帯域測定データの比較的小さいp-pノイズ値を小さくすることができます。スムージングでは、アクティブ・チャネルのデータが、表示されたトレースの一部分(スムージング・アパーチャ)において平均されます。

ヒント:

- 明確な結果を得るために、表示ポイント数を十分多くしておきます。
- 共振の大きいデバイスや、トレースの変動が大きいデバイスに対してスムージングを使用すると、測定誤差が生じるおそれがあります。
- スムージング機能は各トレースに対して独立に設定できます。

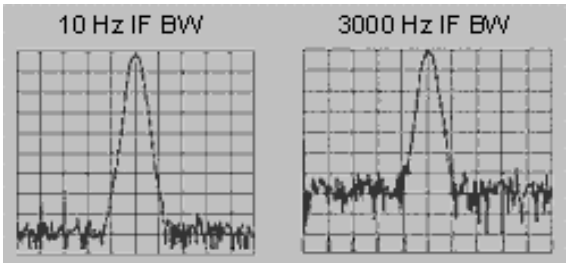
トレースに対するスムージングの影響



IF帯域幅

- IFレシーバ帯域幅を狭めることにより、測定に対するランダム・ノイズの影響が減少します。IF帯域幅を10分の1にするごとに、ノイズ・フロアが10 dB下がります。ただし、IF帯域幅を狭めると掃引時間が長くなります。
- ネットワーク・アナライザは、受信信号をRF/マイクロ波周波数から41.67 kHzという低い中間周波数(IF)に変換します。IFバンドパス・フィルタの帯域幅はIF帯域幅と呼ばれ、40 kHzから最低1 Hzまで調整可能です。
- IF帯域幅はチャネルごと、またはセグメント掃引の各セグメントごとに独立に設定できます。

IF帯域幅を狭めることによる影響



ボタンをクリックすると、トレース・ノイズを減らすための手順が表示されます。

掃引アベレージングの適用または変更

マウス手順

1. **Channel**メニューで**Average**をクリックします。
2. アベレージング機能がまだオンになっていない場合、**Average On**チェック・ボックスを選択します。
3. **Average Factor**ボックスで、測定のアベレージング回数の値を増やします。
4. **OK**をクリックします。

注記: Channel、**Restart Average**をクリックすると、アベレージング対象となる一連の掃引が新たに開始されます。

フロントパネル・キー手順

1. **CHANNEL SETUP**で、**Average**を押します。
2. アベレージング機能がまだオンになっていない場合、**F2 (Avg ON/OFF)**を押してアベレージング機能をオンにします。
3. **F1 (Avg#)**を押します。
4. **ENTRY**ブロックに測定のアベレージング回数を入力し、**Enter**を押します。

注記: F4 (Restart)を押すと、アベレージング対象の一連の掃引が新たに開始されます。

トレース・スムージングの適用

マウス手順

1. **Trace**メニューで**Smoothing**をクリックします。
2. スムージング機能がまだオンになっていない場合、**Smoothing ON**チェック・ボックスを選択します。
3. スムージング・アパーチャを下記のどちらかの方法で指定します。
   - **Percent of Span**ボックスに、スムージング・パーセンテージの値を入力します(最大値は25%)。
   - **Points**ボックスに、スムージング範囲のポイント数を入力します(最大値は測定掃引の全ポイント数の25%)。
4. **OK**をクリックします。

フロントパネル・キー手順

1. **COMMAND**ブロックで**Menu/Dialog**を押します。
2. **NAVIGATION**ブロックで右タブと下矢印を押して**Trace**メニューに移動し、**Smoothing**まで移動して**Click**を押します。
3. スムージング機能がまだオンになっていない場合、**Click**を押して**Smoothing ON**チェック・ボックスを選択します。
4. 右タブを押して、下記のどちらかの方法でスムージング・アパーチャの値を選択します。
   - **Percent of Span**ボックスに、スムージング・パーセンテージの値を入力します(最大値は25%)。
   - **Points**ボックスに、スムージング範囲のポイント数を入力します(最大値は測定掃引の全ポイント数の25%)。
5. **COMMAND**ブロックで**OK**を押します。

IF帯域幅を狭める

マウス手順

1. **Sweep**メニューで**IF Bandwidth**をクリックします。
2. **IF Bandwidth**ボックスで、矢印ボタンを押すか値を入力して、狭める帯域幅の値を選択します。
3. **OK**をクリックします。

フロントパネル・キー手順

1. **CHANNEL**ブロックで**Sweep Setup**を押します。
2. **NAVIGAITON**ブロックで**F3 (Bandwidth)**を押します。
3. **ENTRY**ブロックにIF帯域幅の値を入力し、**Enter**を押します。

測定スループットを上げる

レシーバ･クロストーク

概要 **概念** **クロストークを減らす方法**

クロストークとは、アナライザの信号経路の間のエネルギー漏れのことです。これは高損失の伝送測定で問題になります。しかし、掃引および測定校正の選択によって、クロストークの影響を減らすことができます。

クロストークの下記の性質を考慮します。

- アナライザの信号経路間のエネルギー漏れは、伝送測定の誤差に寄与します。
- クロストークは下記の測定に影響します。
- 損失の大きいデバイス(開位置のスイッチなど)
- ダイナミック・レンジの大きいデバイス(フィルタなど)
- レシーバのクロストークを減らすには、下記の方法があります。

交互掃引に設定

この掃引では、下記の理由でレシーバのクロストークが減少します。

- 交互掃引では1回の掃1つのレシーバ引でだけが測定されます。
- 一方のレシーバが測定されている間、もう一方のレシーバはオフになります。

ヒント:

- 反射(レシーバA)と伝送(レシーバB)が別々の掃引で測定されます。このため、反射測定と伝送測定の両方を行う場合、交互掃引では掃引サイクル時間が2倍になります。
- 伝送だけまたは反射だけの測定では、交互掃引を使うことにより、掃引サイクル時間を変えずにクロストークの影響を最小化することができます。
- クロストークが見えるようにするには、ノイズ・フロアをかなり低くする必要があります。このためには、アベレージング機能を使うか、IF帯域幅を狭めます。

測定校正を実行

伝送測定の場合、レスポンス/アイソレーション測定校正を行うことにより、下記の理由でクロストークを減らすことができます。

- 測定校正中に漏れ信号が測定され、差し引かれます。
- 校正によりアイソレーションが改善され、ノイズ・フロアのみに制限されるようになります。
- 校正プロセスでクロストークを減らす方法については、系統誤差 / アイソレーションを参照してください。

ヒント:

- アイソレーションは一般にノイズ・フロアより小さいため、アイソレーション校正を行う際には、掃引アベレージングを使ったりIF帯域幅を狭めたりといったノイズ低減手法を使用する必要があります。

ボタンをクリックすると、レシーバ･クロストークを減らすための手順を見ることができます。

交互掃引に設定

マウス手順

1. **Sweep**メニューで**Sweep Setup**をクリックします。
2. 交互掃引がオンになっていない場合、**Alternate Sweeps**チェックボックスを選択します。
3. **OK**をクリックします。

注記: 交互掃引モードは、各測定チャネルに対して独立に設定できます。複数の測定チャネルが使用されている場合、すべてのチャネルに対してこの手順を繰り返します。

フロントパネル・キー手順

1. **COMMAND**ブロックで**Menu/Dialog**を押します。
2. **CHANNEL SETUP**ブロックで**Sweep Setup**を押します。
3. 交互掃引がオンになっていない場合、**NAVIGATION**ブロックで右タブを押して、**Alternate Sweeps**ボックスに移動し、Clickを押します。
4. **COMMAND**ブロックで**OK**を押します。

注記: 交互掃引モードは、各測定チャネルに対して独立に設定できます。複数の測定チャネルが使用されている場合、すべてのチャネルに対してこの手順を繰り返します。

レスポンス/アイソレーション測定校正を実行

マウス手順

注記: この手順は伝送測定専用です。

1. **Calibration**メニューで**Calibration Wizard**をクリックします。
2. **Select Calibration Type**の下で**Cal Type**をクリックします。
3. **THRU Resp and Isol**、**OK**をクリックします。
4. プロンプトに従って、校正の残りの手順を実行します。すなわち、校正標準タイプとコネクタ・タイプを選択し、プロンプトに従って標準を測定します。詳細については、測定校正の選択と実行を参照してください。

フロントパネル・キー手順

注記: この手順は伝送測定専用です。

1. **CHANNEL SETUP**ブロックで、**Cal**を押し、**F1 (Cal Wizard)**を押します。
2. **NAVIGATION**ブロックでClickを押し、上下の矢印を押して**THRU Resp and Isol**に移動し、**Click**を押します。
3. **COMMAND**ブロックで**OK**を押します。

NAVIGATIONキーを使って、プロンプトに従って校正の残りの手順を実行します。すなわち、校正標準タイプとコネクタ・タイプを選択し、プロンプトに従って標準を測定します。詳細については、測定校正の選択と実行を参照してください。

ポイント数

概要 **概念** **ポイント数を増やす方法**

データ・ポイントまたは「ポイント」とは、1つのスティミュラス値に対する測定値を表すデータのサンプルです。本器が1回の掃引で測定するポイントの数は指定可能です。

- ほとんどのデータ処理動作はポイント単位で行われます。「掃引」とは、一連のデータ・ポイント測定を、スティミュラス値の列に対して連続して行うものです。
- データ・ポイントの数を多くすると、トレース分解能が上がり、デバイスの応答をよりよく定義できます。
- デフォルト値は201ポイント/掃引です。

本器の掃引時間はポイント数に比例して変化しますが、全体としての測定サイクル時間はポイント数に比例しません。ポイント数やその他の設定による掃引時間の変化の詳細については、技術仕様の理解を参照してください。

ヒント:

- トレース分解能を最高にするには、データ・ポイント数を最大にします。
- スループットを上げるには、確度が許容できる範囲でデータ・ポイントの数を減らします。
- 最適なポイント数を見つけるには、ポイント数をそれ以上増やしても測定結果に大きな違いがなくなる値を探します。
- 正確な測定校正を実現するには、測定に使用するのと同じポイント数で校正を実行します。

ボタンを押すと、データ・ポイントを増やして測定確度を上げるための手順が表示されます。

マウス手順

1. **Sweep**メニューで**Number of Points**をクリックします。
2. リスト中の値を選択するか、C**ustom**を選択して任意の値を入力します。
3. **OK**をクリックします。

フロントパネル・キー手順

1. **CHANNEL SETUP**ブロックで**Sweep Setup**を押します。
2. **NAVIGATION**ブロックで**F2 (Points)**を押し、上下の矢印で値を選択して**Click**を押します。
3. あるいは、**ENTR**Yブロックで任意の値を入力して**Enter**を押します。

測定の最適化

1つの測定には、相互に依存するさまざまな設定があります。これらの設定は測定アプリケーションの目標(スループットの高さまたはデータの正確さ)に合わせて変更することができます。設定を注意深く調整することにより、最適な測定結果が得られる設定を見つけることができます。

- 測定スループットの改善
- 測定確度の改善

測定の安定度

概要 **概念** **測定の安定度を上げる方法**

測定が不安定になる理由はいくつか考えられます。再現性のある測定結果を得るために、安定した測定環境を実現するための方法を以下に紹介します。

測定の確度に悪影響を与える要因として以下のものが挙げられます。ボタンをクリックすると、これらについての説明を見ることができます。

周波数ドリフト

本器の周波数確度は、内部10 MHz周波数発振器に基づいています。安定度とエージングの仕様については、技術仕様の理解を参照してください。

さらに高い周波数確度と安定度を要求する測定アプリケーションの場合、内部周波数標準の代わりに、ユーザが用意した高安定度外部周波数信号源をリアパネルの10 MHz基準入力コネクタに接続することができます。

温度ドリフト

温度による膨張または収縮により、下記のコンポーネントの電気特性が変化します。

- 本器内部のデバイス
- 校正キット標準
- 被試験デバイス
- ケーブル
- アダプタ

測定に対する温度ドリフトの影響を減らすには、下記の方法を使います。

- 温度制御された環境を使用する
- 校正キット・デバイスの温度を安定させる
- 校正手順中には必要なとき以外校正キット・デバイスに触れないようにする
- 周囲温度が測定校正温度から±1℃の範囲に入るようにする

不正確な測定校正

測定校正が不正確な場合、被試験デバイスの真の応答を測定することはできません。正確な校正を保証するには、下記の方法を実行してみてください。

- 被試験デバイスを接続するポイント、すなわち基準面で校正を実行します。
- 測定校正の実行後に他のアクセサリ(ケーブル、アダプタ、アッテネータなど)をテスト・セットアップに追加した場合、ポート延長機能を使って、付加された電気長と遅延を補正します。
- 校正プロセスで使用する定義と一致した校正標準を使用します。

詳細については、正確な測定校正を参照してください。

デバイス接続

再現性のある測定結果を得るには、よい接続が重要です。よい接続のためには下記の方法を実行します。

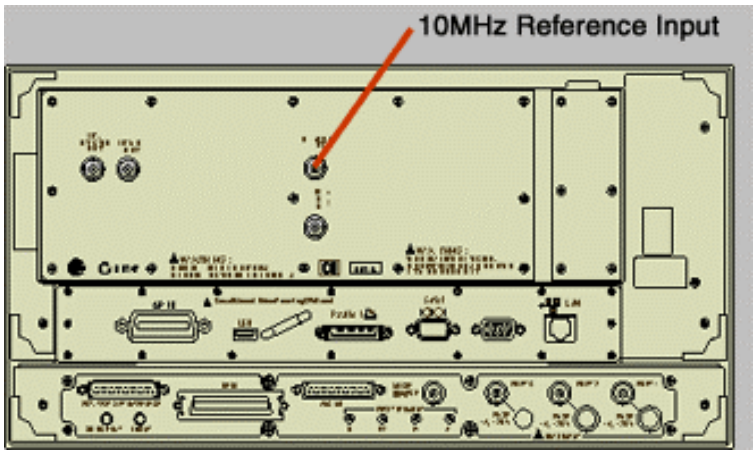
- 測定セットアップに含まれるすべてのコンポーネントのコネクタを検査し、清掃します。
- 正しい接続方法を使用します。
- 測定中にケーブルを動かさないようにします。

ボタンをクリックすると、測定の安定度を上げるための手順が表示されます。

外部周波数基準の使用

1. 外部周波数基準信号を図のようにリアパネルのコネクタに入力します。

- 入力周波数: 10 MHz±10 ppm
- 入力レベル: -15～+20 dBm
- 入力インピーダンス: 200Ω



室温の制御

1. 測定校正またはデバイス測定を実行する30分前に本器の電源をオンにします。
2. デバイス測定は温度制御された環境で実行します。仕様および特性はすべて、(特に注記しない限り)25℃±5℃の範囲で有効です。
3. 周囲温度が測定校正温度から±1℃の範囲内になるようにします。

校正標準温度の安定化

1. 測定校正を実行する1時間前に校正キットのケースを開け、標準を保護フォームから取り出しておきます。
2. 測定校正実行中には、必要なとき以外校正キット・デバイスに触れないようにします。

よい接続の実行

1. **拡大鏡を使ってすべてのコネクタを検査します。下記のような損傷の徴候がないかどうか調べます。**

- メッキの剥げ
- 中心導体の曲がり、折れ、位置ずれ
- 深い掻き傷、くぼみ、肩の丸まり
- 汚れや金属粒子

2. **すべてのコネクタを、イソプロピル・アルコールとリントフリー綿棒で清掃します。**

- アルコールが蒸発するまで待ちます。
- 圧搾空気を軽く当ててコネクタを乾かします。

3. **デバイスを正しく接続します。**

- 静電気防止マットとリスト・バンドを使って、作業者の体とすべてのデバイスをアースします。
- コネクタの位置を合わせてまっすぐにはめ込みます。
- コネクタのナットだけを回します。
- 接続の最後にトルク・レンチを使います。

測定スループットの改善

ボタンをクリックすると、測定スループットを上げるための最適化方法が表示されます。

- **掃引速度の最大化**
- **測定ステートの切替えの高速化**
- **データ転送速度の最大化**

掃引速度

概要　　概念　　掃引速度を最高にする方法

最高の測定スループットを得るために考慮すべきことの1つは、測定掃引を高速化することです。

測定アプリケーションの掃引速度を最高にするには、下記の項目を調整します。

掃引設定

以下の各設定を説明に従って変更してみてください。

- **周波数スパン** デバイスに必要な周波数だけを測定するようにします。
- **セグメント掃引** セグメントを使って、必要な場所の測定データだけに注目します。
- **ステップ掃引オフ** 可能ならリニア掃引モードを使って掃引時間を最小化します。
- **自動掃引時間** このデフォルトを使うと、現在の設定で最高速の掃引が行われます。
- **ポイント数** 測定に必要な最小限のポイント数を使用します。

*機器ステート: 初期設定状態、35 kHz帯域幅、CF=1 GHz、スパン=100 MHz、補正オフ。表示オンの場合は21 msを加算。サイクル・タイムには、掃引時間と再トレース時間が含まれます。

ポイント数やその他の設定が掃引サイクル・タイムに及ぼす影響については、技術仕様の理解を参照してください。

ノイズ低減設定

下記の設定の組合わせを使うと、掃引時間を短縮しながら許容できる測定結果を得ることができます。

- **IF帯域幅** 許容できるトレース・ノイズとダイナミック・レンジが得られる範囲で、できるだけ広いIF帯域幅を使用します。
- **アベレージング** アベレージング係数を小さくするか、アベレージングをオフにします。

測定校正選択

- 必要な確度レベルが実現できる最も高速なタイプの校正を選択します。
  - 未補正測定と、レスポンス校正または1ポート校正を使って行われた測定では、掃引速度はほとんど変わりません。
  - フル2ポート校正を使用すると、1個のSパラメータしか表示されない場合でも、誤差補正のために4個のSパラメータすべてをアップデートするため、正逆両方向の掃引が必要になります。

詳細については、測定校正を参照してください。

不要な機能

本器はすべてのアクティブ機能に対して情報を更新しなければなりません。掃引速度をさらに上げるには、測定アプリケーションに不要なアナライザ機能をすべてオフにします。

- 使用していないトレース
- 使用していないマーカ
- スムージング
- リミット・テスト
- トレース演算
- 表示

本器の掃引速度はいくつかの測定設定に依存します。最高の掃引速度と必要な測定結果が得られる設定が見つかるまで、いろいろな設定を試してみてください。

掃引設定の変更

マウス手順

測定スパンの変更

1. 掃引スティミュラス設定にアクセスします。
   - **Channel**メニューで、周波数掃引の**Start/Stop**または**Center/Span**をクリックします。
   - **Channel**メニューで、パワー掃引の**Power**をクリックします。
2. **Start (Center)**ボックスに値を入力するか矢印ボタンを押して、スパンを小さくする値を選択します。
3. **Stop (Span)**ボックスに値を入力するか矢印ボタンを押して、スパンを小さくする値を選択します。
4. **OK**をクリックします。
5. 掃引タイプを変更する方法についてはセグメント掃引を参照してください。

ステップ掃引モードをオフにする

1. **Sweep**メニューで**Sweep Setup**をクリックします。
2. **Stepped Sweep**ボックスをクリックしてクリアし、**OK**をクリックします。

自動掃引時間に変更

1. **Sweep**メニューで**Sweep Time**をクリックします。
2. 値がそれ以上減らなくなるまで下矢印ボタンをクリックし、**OK**をクリックします。

ポイント数の変更

1. **Sweep**メニューで**Number of Points**をクリックします。
2. リストされた値の1つを選択するか、**Customize**を選択して任意の値を指定します。

3. **OK**をクリックします。

フロントパネル・キー手順

周波数スパンの変更

1. 掃引スティミュラス設定にアクセスします。
   - **CHANNEL SETUP**ブロックで、周波数掃引の**Start/Center**を押し、**F1 (Start)**または**F3 (Center)**を押します。
   - **CHANNEL SETUP**ブロックで、パワー掃引の**Power**を押し、**F1 (Start)**を押します。
2. アクティブ・エントリ・ボックスで、**ENTRY**ブロックから値を入力し、**Enter**を押します。
3. **F2 (Stop)**または**F4 (Span)**を押し、**ENTRY**ブロックから値を入力して**Enter**を押します。
4. 掃引タイプを変更する方法についてはセグメント掃引を参照してください。

ステップ掃引モードをオフにする

1. **COMMAND**ブロックで**Menu/Dialog**を押します。
2. **CHANNEL SETUP**ブロックで**Sweep Setup**を押します。
3. **Stepped Sweep**ボックスがチェックされている場合、右タブを押して**Stepped Sweep**ボックスに移動し、**Click**を押してチェック・ボックスをクリアします。
4. **COMMAND**ブロックで**OK**を押します。

自動掃引時間に変更

1. **CHANNEL SETUP**ブロックで**Sweep Setup**を押します。
2. **F1 (Time)**を押し、**ENTRY**ブロックで0、**Enter**を押します。

ポイント数の変更

1. **CHANNEL SETUP**ブロックで**Sweep Setup**を押します。
2. **F2 (Points)**を押します。
3. **NAVIGATION**ブロックで上下の矢印を押して値のリストを順番にたどり、目的の値が表示されたら**Click**を押して選択します。または、**ENTRY**ブロックから値を入力し、**Enter**を押します。

ノイズ低減設定の変更

マウス手順

IF帯域幅の変更

1. **Sweep**メニューで**IF Bandwidth**をクリックします。
2. **IF Bandwidth**ボックスに値を入力するか矢印ボタンを押して、帯域幅を広げる値を選択します。
3. **OK**をクリックします。

アベレージング係数の変更

4. **Channel**メニューで**Average**をクリックします。
5. **Average Factor**ボックスで、測定アベレージング回数を小さい値に変更します。
   - または、**Average On**ボックスをクリックしてチェック・ボックスをクリアし、アベレージング機能をオフにします。
6. **OK**をクリックします。

注記: アベレージング対象の一連の掃引を新たに開始するには、**Channel**、**Restart Average**をクリックします。

フロントパネル・キー手順

IF帯域幅の変更

1. **Sweep Setup**、**F3 (IF Bandwidth)**を押します。
2. IF帯域幅値を入力してIF帯域幅を広げます。
3. **ENTRY**ブロックで**Enter**を押します。

アベレージング係数の変更

4. **CHANNEL SETUP**ブロックで**Average**を押します。
5. **F1 (Avg#)**を押し、アベレージング回数を減らす値を入力して**Enter**を押します。
   - または、**F2 (Avg ON/OFF)**を押してアベレージング機能をオフにします。

注記: アベレージング対象の一連の掃引を新たに開始するには、**F4 (Restart)**を押します。

不要な機能をオフにする

マウス手順

不要なトレースの削除

1. トレース・ステータス・バーをクリックして、不要なトレースをアクティブにします。
2. **Trace**メニューで**Delete Trace**をクリックします。

不要なマーカをオフにする

1. **Marker**メニューで**Select Marker**をクリックし、リスト上の不要なマーカをクリックします。
   - すべてのマーカをオフにするには、**All Off**をクリックします。
2. マーカ・ツールバーが表示されていない場合、**View**メニューで**Toolbars**をクリックし、**Marker**をポイントしてクリックします。
3. **On**チェック・ボックスをクリックしてクリアします。

スムージングをオフにする

1. **Trace**メニューで**Smoothing**をクリックします。
2. **Smoothing On**ボックスをクリックしてチェック・ボックスをクリアし、スムージン

グ機能をオフにします。**OK**をクリックします。

リミット・テストをオフにする

1. **Trace**メニューで**Limit Test**をクリックします。
2. **Limit Test ON**ボックスをクリックしてチェック・ボックスをクリアし、リミット機能をオフにします。**OK**をクリックします。

演算機能をオフにする

- **Trace**メニューで**Data Math**リストをポイントし、**Data**をクリックして現在の測定トレースだけを表示させます。

フロントパネル・キー手順

不要なトレースの削除

1. 複数のウィンドウが画面に表示されている場合、このステップを実行します。
   - **DISPLAY**ブロックで**Window**、**F2 (Next Window)**を、不要なトレースが表示されているウィンドウが選択されるまで押します。
2. **DISPLAY**ブロックで**Trace**、**F2 (Next Trace)**を、不要なトレースがアクティブになるまで押します。
3. **F4 (Delete)**を押します。

不要なマーカをオフにする

1. **TRACE SETUP**ブロックで**Marker**を押します。
2. **F1**または**F2**または**F3**(**Marker 1**、**Marker 2**、**Marker 3**)を押して、オフにするマーカを選択します。その他のマーカを表示するには、もう一度**Marker**を押します。
3. **F4 (Off)**を押します。
4. すべてのマーカをオフにするには、下記のステップを実行します。
   a. **COMMAND**ブロックで**Menu/Dialog**を押します。
   b. **NAVIGATION**ブロックで右タブと下矢印を押して**Marker**メニューに移動し、**All Off**まで移動します。**Click**を押します。

スムージングをオフにする

1. **COMMAND**ブロックで**Menu/Dialog**を押します。
2. **NAVIGATION**ブロックで右タブと下矢印を押して**Trace**メニューに移動し、**Smoothing**まで移動します。**Click**、 **Click**を押します。
3. **COMMAND**ブロックで**OK**を押します。

リミット・テストをオフにする

- **TRACE SETUP**ブロックで**Limit Table**、**F1 (Test ON/OFF)**を押します。

演算機能をオフにする

- **TRACE SETUP**ブロックで**Math/Memory**、**F3 (Data Only)**を押して、現在の測定トレースだけを表示させます。

高速な校正の選択

マウス手順

1. **Calibration**メニューで**Calibration Wizard**、**Cal Type**をクリックします。
2. **OPEN Response**、**SHORT Response**、**THRU Response**、**1-Port**のどれかを選択し、**OK**をクリックします。
3. **Measure Stds...**をクリックし、ウィザードのプロンプトに従います。

詳細については、校正の選択と実行を参照してください。

フロントパネル・キー手順

1. **CHANNEL SETUP**ブロックで**Cal**、**F1 (Cal Wizard)**を押します。
2. **NAVIGATION**ブロックで**矢印**と**Click**を押して、**OPEN Response**、**SHORT Response**、**THRU Response**、**1-Port**のどれかを選択します。
3. **COMMAND**ブロックで**OK**を押します。
4. **NAVIGATION**ブロックで右タブを押して**Measure Stds...**を選択し、**Click**を押します。
5. 校正ウィザードのプロンプトに従います。

詳細については、校正の選択と実行を参照してください。

複数測定

概要　　概念　　複数測定の
高速な実行方法

デバイスの特性評価に複数の測定が必要な場合、スループットを改善するためにさまざまな方法が使用できます。

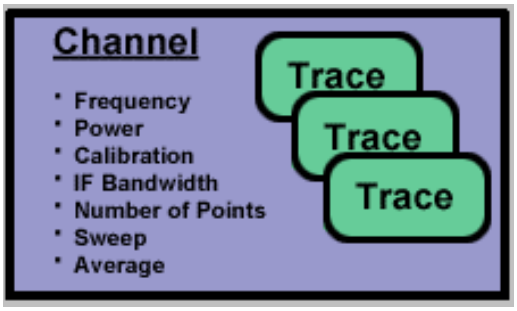
これらの方法を試してみて、測定アプリケーションのニーズに最も合うものを見つけてください。

スループットを上げるための測定条件設定

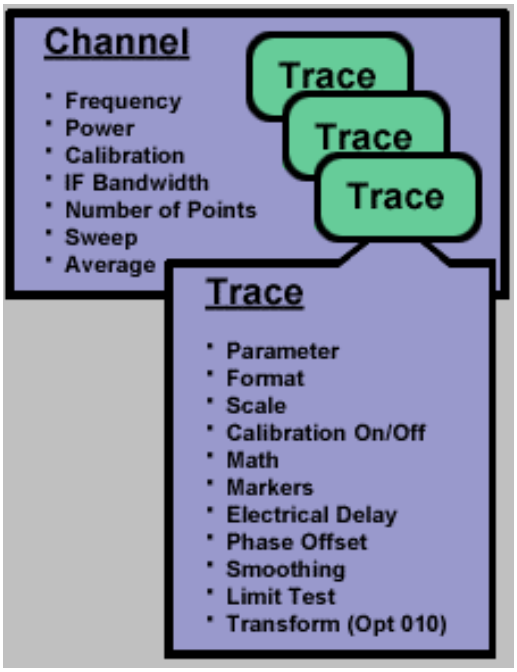
複数の測定が必要なデバイスに対するスループットを最適化するには、本器の動作について理解しておくのが役立ちます。この知識があると、アプリケーションに最適な測定シナリオを準備することができます。下記の機能ブロックを使って、任意の組合わせで測定を配列することができます。

- 1～4の独立した測定チャネル
- ウィンドウ1つあたり1～4のトレース
- 1画面あたり1～4のウィンドウ
- 1画面あたり1～16の測定トレース
- 1画面あたり1～16のトレース

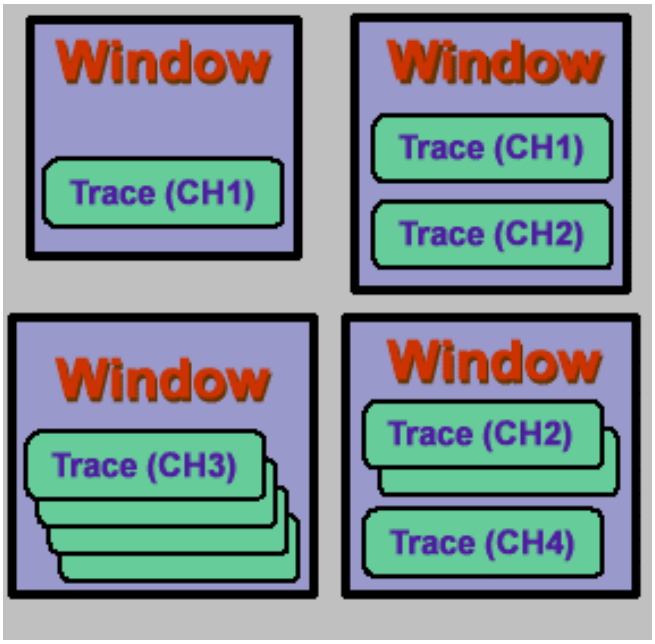
チャネルは、データ収集に関連する属性の組合わせをそれぞれ独立に持っています。



トレースは、収集データに対して実行される演算機能に関する属性の固有の組合わせを持っています。本器は1画面あたり最大16のトレースをサポートします。



ウィンドウはトレースを表示するために用いられます。

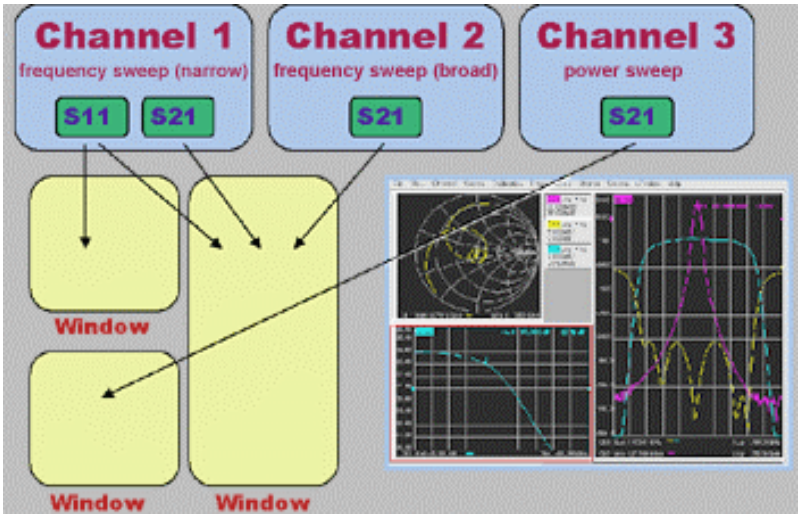


測定セットの作成

すべてのデバイス測定が1つの機器ステートに含まれるように測定を配列すれば、それを保存することにより後でリコール機能1つで一連の測定を呼び出すことができます。詳細については複数のトレースとチャネルを参照してください。

測定セットを作成する場合、以下の機能があることを考慮します。

- 複数のチャネルのトレースを1つのウィンドウに表示
- 複数のチャネルのトレースを複数のウィンドウに表示



セグメント掃引の使用

セグメント掃引は、被試験デバイスの特性評価の際に下記の設定を変更する必要がある場合に有効です。

- 周波数レンジ
- パワー・レベル
- IF帯域幅
- ポイント数

セグメント掃引を使うと、独立した属性を持つ複数の周波数レンジのセットを定義できます。これにより、変化する特性を持つデバイスを1回の測定掃引で測定できます。詳細についてはセグメント掃引を参照してください。

選択的に測定をトリガ

測定トリガを使って、下記のように測定を実行できます。

- データが高速に変化する測定だけを連続的にアップデート
- データがあまり変化しない測定は不定期にアップデート

例えば、4つの測定チャネルが下記のようにセットアップされているとします。

- 2つのチャネルはフィルタのチューニングに使用するデータを測定
- 2つのチャネルはフィルタの帯域外応答のデータを測定

常時モニタしたいのは、フィルタのチューニングに使用するデータだけです。すべてのチャネルを連続的にアップデートすると、本器の応答が遅くなり、フィルタを効率的にチューニングできなくなるおそれがあります。

注記: 頻度の低い測定は手動でトリガするか、リモート・インタフェース・コマンドでトリガします。

測定の切替えを自動化

デバイスの特性評価に必要ないくつかの測定の間にわずかな違いがある場合、測定設定を自動的に切り換えたほうが高速になる場合があります。

測定の呼出しの高速化

- 測定をリコールする最も効率的な方法は、測定のセット(機器ステート)の形でリコールすることです。
- 複数の測定を含む機器ステートをリコールする時間は、1つの測定を含む機器ステートをリコールする時間に比べてわずかに長いだけです。
- リコール機能はそれぞれに時間がかかります。測定をセットにまとめ、そのセットをリコールすれば、リコールにかかる時間を節約できます。

ボタンをクリックすると、複数の測定が必要なデバイスに対するスループットを改善する手順が表示されます。

測定セットの作成

マウス手順

1. **Window**メニューで**Meas Setups**をクリックし、下記のセットアップのどれかをクリックします。
   - **Setup A**: 1つのチャネルの4つのSパラメータすべてを、対数振幅フォーマットの1つのウィンドウに表示します。
   - **Setup B**: 1つのチャネルの4つのSパラメータすべてを、対数振幅フォーマットのウィンドウ1つずつに表示します。
   - **Setup C**: 1つのチャネルの3つのSパラメータ(S_{11}、S_{22}、S_{21})を1つずつのウィンドウに別々に表示します。S_{11}とS_{22}はスミス・チャート・フォーマット、S_{21}は対数振幅フォーマットになります。
   - **Setup D**: 2つのチャネルのそれぞれのS21およびS11パラメータを、チャネルごとに別々の対数振幅フォーマットのウィンドウに表示します。
2. 測定ニーズに合わせて設定を変更します。

詳細については、測定ステートの選択を参照してください。

フロントパネル・キー手順

1. **DISPLAY**ブロックで**Measure Setups**を押し、アクティブ入力選択のどれかのキー(**F1**、**F2**、**F3**、**F4**)を押して測定のセットを設定します。
   - **Setup A**: 1チャネル、4個のSパラメータ、対数振幅、1つのウィンドウ
   - **Setup B**: 1チャネル、4個のSパラメータ、対数振幅、4つのウィンドウ
   - **Setup C**: 1チャネル、3個のSパラメータ、スミス・チャートと対数振幅、3つのウィンドウ
   - **Setup D**: 2チャネル、2個のSパラメータ(4つのトレース)、対数振幅、2つのウィンドウ
2. 測定ニーズに合わせて設定を変更します。

詳細については、測定ステートの選択を参照してください。

選択的に測定をトリガ

この手順は、下記のように動作する2つの異なる測定を準備する方法を示します。

- 1つの測定は連続的にデータをアップデート
- もう1つの測定は不定期にデータをアップデート

マウス手順

1. **Windows**メニューで**Meas Setups**、**Setup D**をクリックします。
2. 測定ニーズに合わせて設定を変更します。チャネル1が連続的にアップデートする測定、チャネル2が不定期にアップデートする測定とします。

連続アップデートの測定トリガの設定

3. **Sweep**メニューで**Trigger**、**Trigger...**をクリックします。
4. **Trigger Source**の下で**Internal**をクリックします。
5. **Channel Trigger State**の下で**Channel 1**を選択し、**Continuous**をクリックします。

不定期アップデートの測定トリガの設定

6. **Channel Trigger State**の下で**Channel 2**を選択し、**Single**、**OK**をクリックします。
   - 本器に複数回の掃引をトリガさせたい場合、**Enable Groups**チェック・ボックスをクリックし、掃引回数を入力します。
7. **System** メニューで**Keys, Trigger**をクリックします。

測定のアップデート

8. 下のウィンドウをクリックしてチャネル2をアクティブ・チャネルにします。
9. アクティブ・エントリ・ツールバーで、設定するトリガのタイプをクリックします。
   - 1回のトリガで1回の掃引をさせたい場合は、**Single**をクリックします。
   - 1回のトリガで複数回の掃引をさせたい場合は、**Groups**をクリックします。

注記: 測定のトリガを開始するには、その測定のトレースがアクティブでなければなりません。

フロントパネル・キー手順

1. **DISPLAY**ブロックで**Measure Setups**、**F4 (Setup D)**を押します。
2. 測定ニーズに合わせて設定を変更します。チャネル1が連続的にアップデートする測定、チャネル2が不定期にアップデートする測定とします。

連続アップデートの測定トリガの設定

3. **CHANNEL SETUP**ブロックで**Channel**、**F1 (Ch 1)**を押します。
4. **Trigger**、**F1 (Continuous)**を押します。

不定期アップデートの測定トリガの設定

5. **CHANNEL SETUP**ブロックで**Channel**、**F2 (Ch 2)**を押します。
6. **Trigger**、**F2(Single)**を押します。
7. 複数回の掃引を本器にさせたい場合は、次のステップを実行します。
   a. **COMMAND**ブロックで**Menu/Dialog**を押します。
   b. **CHANNEL SETUP**ブロックで**Trigger**を押します。
   c. **NAVIGATION**ブロックで右タブを押して**Enable Groups**チェック・ボックスに移動し、Clickを押します。
   d. 右タブを押して**Enable Groups**ボックスに移動し、掃引回数を入力します。
   e. **COMMAND**ブロックで**OK**を押します。
   f. 測定セットをトリガするには、**Trigger**、**F1 (Groups)**を押します。

測定のアップデート

8. **CHANNEL SETUP**ブロックで**Channel**、**F2 (Ch 2)**、**Trigger**を押します。
9. アクティブ・エントリ・ツールバーで、設定するトリガのタイプをクリックします。
   - 1回のトリガで1回の掃引をさせたい場合は、**F2 (Single)**をクリックします。
   - 1回のトリガで複数回の掃引をさせたい場合は、**F1 (Groups)**をクリックします。

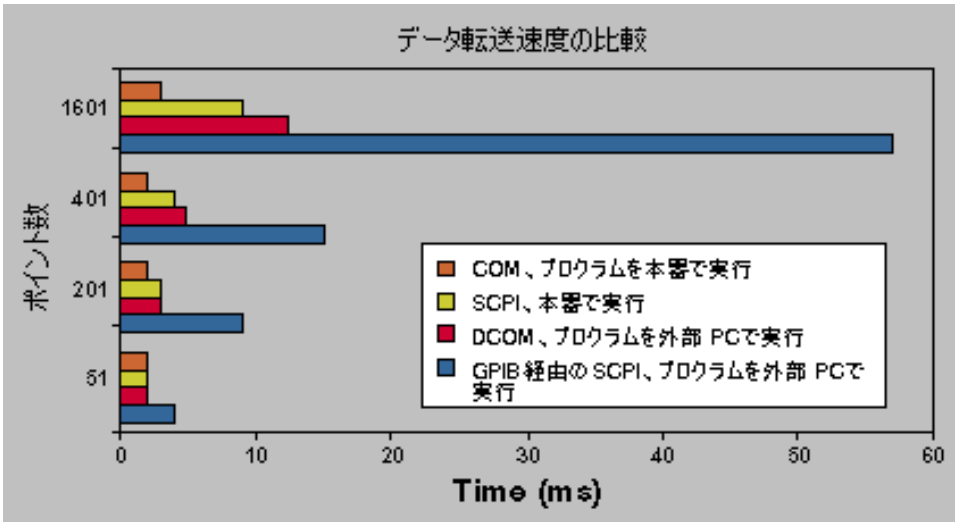
データ転送速度

概要　　概念　　データ転送速度を上げる方法

測定スループットを最大化するには、データ転送を高速化することが役立ちます。

データ転送速度を改善するには、下記の方法を試してみてください。

- **シングル掃引モードを使用**して、データ転送を開始する前に測定が終了するようにします。
- **必要最小限のデータを転送**します。例えば、リニアに配置された多数のポイントを持つフル・トレースの代わりに、セグメント掃引による少ないポイント数のトレースを使います。または、トレース転送を使わずにマーカを使います。
- SCPIプログラムを使って自動化アプリケーションを実現する場合、**REALデータ・フォーマット**を選択することにより高速な転送が可能です。
- SCPIプログラムによる自動化アプリケーションの場合、**LAN経由のSCPIを使用**します。
- 自動化アプリケーションで転送速度を最大にするには、**COMプログラムを使用**します。データ転送時間を参照してください。



ボタンをクリックすると手順が表示されます。

シングル掃引モードの使用

ボタンをクリックすると手順を見ることができます。

マウス手順

1. **Sweep**メニューで、**Trigger**、**Trigger...**をクリックします。
2. **Trigger Source**の下で、**Manual**をクリックします。
3. **Channel Trigger State**の下で、転送する測定結果のチャネルを選択し、**Continuous**、**OK**をクリックします。

測定のアップデート

4. **System**メニューで**Keys, Trigger**をクリックします。
5. アクティブ・エントリ・ツールバーで、**Single**をクリックします。

フロントパネル・キー手順

1. **CHANNEL SETUP**ブロックで**Channel**を押し、転送する測定結果のチャネルに対応するアクティブ・エントリ・キー(**F1**、**F2**、**F3**、**F4**)を押します。
2. **Trigger**、**F2(Single)**を押します。

チャネル2の測定のアップデート

8. **CHANNEL SETUP**ブロックで、**Trigger**、**F4(Restart)**を押します。

ステップ3. 測定校正の実行

- 測定校正の概要
- 測定誤差
- 校正標準
- 測定校正の選択と実行
- 正確な校正
- 校正の有効性

測定校正の概要

概要　　概念　　校正の実行方法

測定校正とは、既知の校正標準を測定することにより系統誤差項を評価し、以降の測定からこれらの誤差の影響を除去するプロセスのことです。校正は測定誤差を減らすことにより本器の確度を向上させます。

ボタンをクリックすると下記事項についての説明が表示されます。

測定校正とは何か

- 校正では、誤差モデルと呼ばれる式によって、系統誤差のいくつかが除去されます。高品質の標準(ショート、オープン、負荷、スルーなど)を測定することにより、誤差モデルの誤差項を解くことができます。系統誤差の詳細については、測定誤差を参照してください。
- 校正にはいくつかの方法があり、実行する測定の種類と、必要な確度のレベルによって選択が可能です。
- 校正済み測定の確度は、校正キットの標準の品質と、校正キット定義ファイルに記録された標準モデル(定義)の正確さに依存します。校正キット定義ファイルは本器内部に記憶されています。正確な測定を行うには、実際に使用する校正キットに一致する校正キット定義を使用する必要があります。
- カスタム校正標準を作成する場合(フィクスチャ内測定など)、校正標準の特性を評価し、その定義をユーザ変更校正キット・ファイルに記録します。校正キット・ファイルの変更方法については、校正標準を参照してください。

校正の選択と手順の詳細については、測定校正の選択と実行を参照してください。

なぜ校正が必要なのか

- どのような形の誤差補正も必要としない完璧なハードウェアを作成することは不可能です。大部分のデバイスに対して誤差補正が不要なくらいに正確なハードウェアを作成することでさえ、莫大な費用がかかります。
- ネットワーク解析の確度は、ネットワーク・アナライザ外部の要因から大きな影響を受けます。接続ケーブルやアダプタなど、測定セットアップの構成部品によって、振幅と位相に変動が生じ、被試験デバイスの実際の応答が覆い隠される場合もあります。
- 最善の方法は、性能と費用のバランスを取りながら、実用的にできるだけ正確なハードウェアを作成することです。この場合、測定確度を向上させるために校正が非常に役立ちます。

校正を行った方がよい場合

- 実現可能な最高の確度が必要な場合
- コネクタ・タイプやインピーダンスの違いにアダプタで対応する場合
- 被試験デバイスと本器のテスト・ポートとをケーブルで接続する場合
- 測定の周波数スパンが広いか、デバイスの電気長が長い場合
- 被試験デバイスの入力または出力にアッテネータなどのデバイスを接続する場合

テスト・セットアップが上記の条件のどれかに当てはまる場合、以下のシステム特性が影響を受ける可能性があります。

- デバイス入力での振幅
- 周波数応答確度
- 方向性
- クロストーク(アイソレーション)
- 信号源整合
- 負荷整合

被試験デバイスに対して測定校正を実行する手順の概要を以下に示します。

1. 本器と被試験デバイスを測定構成に従って準備します。
2. 測定を最適化するために適切なアナライザ設定を選択します。
3. 被試験デバイスを取り外し、校正ウィザードで校正タイプと校正キットを選択します。
4. 校正ウィザードのプロンプトに従って、選択した校正タイプに必要な校正標準を接続して測定します。
   - 校正標準の測定結果に基づいて、本器が誤差項を計算します。
   - 校正ウィザードの手順が終了すると、誤差項が校正セットとして本器のメモリに自動的に記憶されます(記憶された校正セットを呼び出すことができます)。
5. 被試験デバイスを再接続し、データを解析します。誤差補正が適用されると、誤差項の影響がデバイスの測定結果から除去されます。

測定誤差

概要　　ドリフト誤差とランダム誤差　　系統誤差　　誤差項のモニタ

どれほど注意して測定を行っても、あらゆる測定には何らかの不正確さが存在します。ただし、測定誤差にどれほど注意すべきかは、必要な測定確度によって異なります。例を挙げます。

- デバイスが動作するかどうかだけが問題で、大まかな測定で十分な場合は、測定誤差について気にする必要は通常ありません。
- 制限値ぎりぎりの増幅器が利得仕様を満たすかどうかを決定したい場合、測定確度に十分注意する(すなわち測定誤差を減らす)必要があります。

測定誤差の原因を理解することは、確度改善の方法を知るために役立ちます。ボタンをクリックすると、測定誤差の2つの基本的原因と、誤差項をモニタする方法について知ることができます。

ドリフト誤差

- ドリフト誤差は、測定器またはテスト・システムの性能が校正実行後に変化することに起因します。
- ドリフト誤差の主要な原因は、テスト・セット内部の接続ケーブルの熱膨張特性と、マイクロ波周波数コンバータの変換安定度です。ドリフト誤差は再度校正(s)を行うことによって除去できます。
- 校正がどれくらいの時間有効であるかは、テスト環境でのテスト・システムのドリフト速度に依存します。

ドリフトを最小化するには、環境温度を一定にすることが役立ちます。詳細については、測定安定度を参照してください。

ランダム誤差

ランダム誤差は予測不可能であり、誤差補正によって除去することができません。ただし、測定確度に対するランダム誤差の影響を最小化する方法はあります。

ランダム誤差の主な原因は下記の3つです。

機器ノイズ誤差

ノイズとは、本器のコンポーネントから生成される不要な電気的擾乱のことです。下記のような種類があります。

- レシーバの広帯域ノイズ・フロアによる低レベル・ノイズ
- テスト・セット内部のLO信号源のノイズ・フロアおよび位相ノイズによる高レベル・ノイズまたはトレース・データのジッタ

ノイズ誤差を減らすには、下記のような方法があります。

- 被測定デバイスの信号源出力を増やす。
- IF帯域幅を狭める。
- 複数の測定掃引のアベレージングを使用する。

スイッチ再現性誤差

- 本器では、信号源アッテネータ設定を切り換えるために、機械的RFスイッチが用いられています。
- 機械的RFスイッチをオンにする際に、接点が前回オンにしたときとは異なる閉じ方をすることがあります。本器内部でこのようなことが起きると、測定確度に悪影響を与えます。
- スイッチ再現性誤差の影響を低減するには、重要な測定の途中でアッテネータを切り換えないようにします。

コネクタ再現性誤差

コネクタに摩耗が起きると、電気的性能が変化します。コネクタ再現性誤差を低減するには、コネクタの取り扱いに関する注意事項を守るようにします。デバイス接続を参照してください。

系統誤差は、アナライザとテスト・セットアップの不完全性から生じます。

- これらの誤差は再現性があり(したがって予測可能であり)、時間によって変化しないと考えられます。
- 系統誤差は校正プロセスによって評価され、測定の際に数学的に除去されます。
- しかし、系統誤差を完全に除去することはできません。校正プロセスの限界のために、必ずいくらかの残留誤差が存在します。残留(測定校正後の)系統誤差は以下の原因から生じます。
  - 校正標準の不完全性
  - コネクタ・インタフェース
  - 接続ケーブル
  - 測定機器

反射測定では以下の3つの系統誤差が生じます。

伝送測定では下記の3つの系統誤差が生じます。

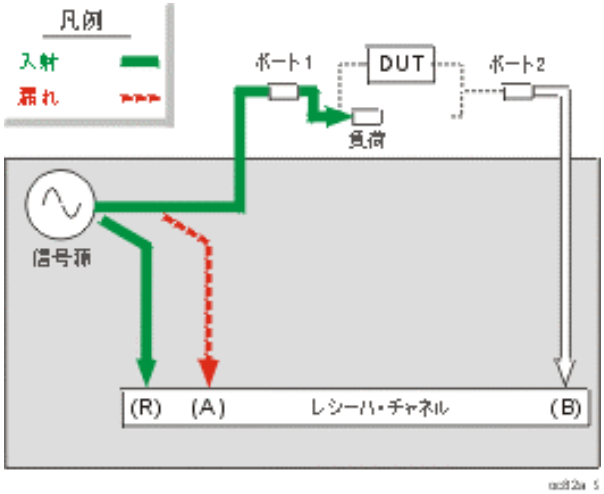
注記: 以下の6つの系統誤差に関する図は、順方向測定向けに構成されたハードウェアを示します。逆方向測定の場合、本器内部のスイッチングにより、ポート2が信号源、ポート1がレシーバになります。チャネル'A'が伝送レシーバ、チャネル'B'が反射レシーバ、チャネル'R2'が基準レシーバになります。これら6つの系統誤差に2方向を掛けて合計12の系統誤差が2ポート・デバイスには生じます。

方向性誤差

ネットワーク・アナライザはどれも、方向性結合器またはブリッジを使って反射測定を実行します。

理想的な結合器では、DUTからの反射信号だけがチャネル'A'に現れます。実際には、入射信号のごく一部が結合器のフォワード・パスから漏れてチャネル'A'に現れます。

この測定誤差を方向性誤差と呼びます。アナライザはこの誤差を測定して低減することができます。



方向性誤差の測定と低減の仕組み

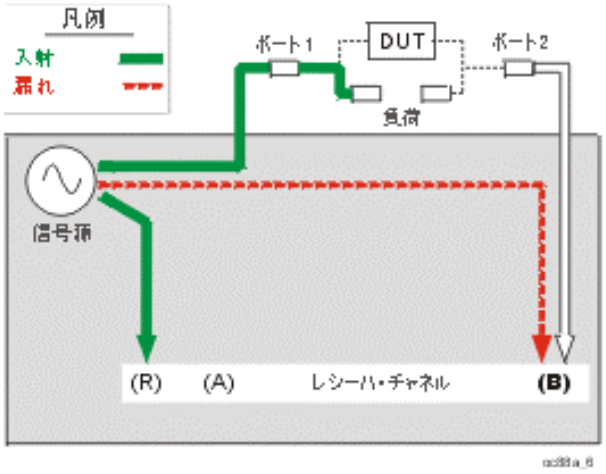
1. 校正中に、負荷標準がポート1に接続されます。負荷からの反射はないと仮定します。
2. 以降の反射測定から方向性誤差が減算されます。

アイソレーション誤差

理想的には、チャネル'B'では DUTから伝送された信号だけが測定されます。

実際には、ごくわずかな信号が本器内部のさまざまな経路からチャネル'B'に漏れてきます。

信号漏れ、別の呼び方ではクロストークは、アイソレーション誤差であり、本器が測定して低減することができます。

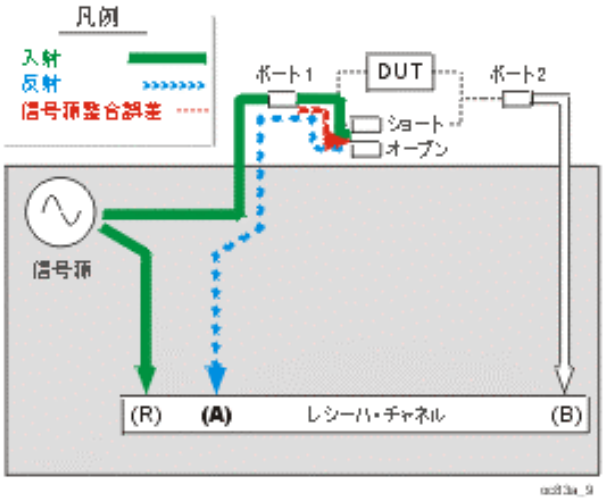


アイソレーション誤差の測定と低減の仕組み

1. 校正中に、負荷標準がポート1とポート2に接続されます。
2. チャネル'B'で測定される信号は、本器内部のさまざまな経路からの漏れです。
3. 以降の伝送測定からアイソレーション誤差が減算されます。

信号源整合誤差

理想的な反射測定では、DUTから反射された信号がすべてチャネル'A'で測定されます。
実際には、DUTから反射された信号の一部がポート1で再び反射され、チャネル'A'で測定されません。
この測定誤差は信号源整合誤差と呼ばれ、本器が測定して低減することができます。

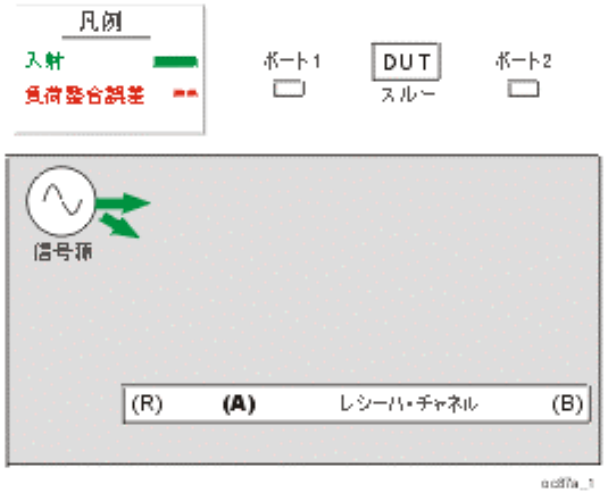


信号源整合誤差の測定と低減の仕組み

1. 校正中に、ショート標準がポート1に接続されます。ショートからの既知の反射がチャネル'A'で測定され、本器に記憶されます。
2. オープン標準がポート1に接続されます。オープンからの既知の反射がチャネル'A'で測定され、本器に記憶されます。
3. 本器は記憶された2つの信号を比較します。2つの信号の差は信号源整合誤差に起因するものです。
4. 以降の反射および伝送測定から信号源整合誤差が減算されます。

負荷整合誤差

理想的な伝送測定では、入射信号はDUTから伝送されてチャネル'B'で測定されます。
実際には、一部の信号がポート2で反射され、チャネル'B'で測定されません。
この測定誤差は負荷整合誤差と呼ばれ、本器が測定して低減することができます。



負荷整合誤差の測定と低減の仕組み

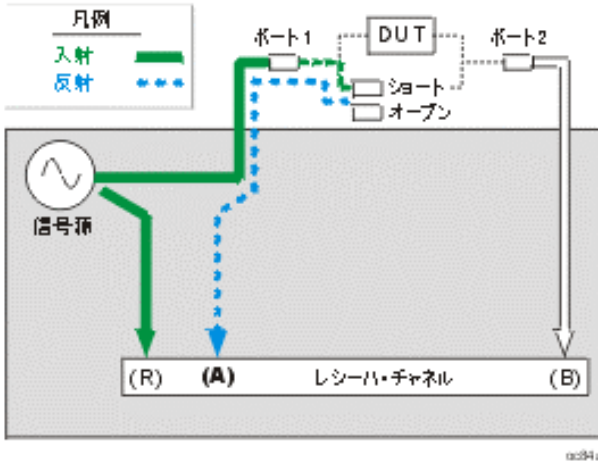
1. ポート1とポート2のテスト・コネクタを接合することにより、完全なゼロレングス・スルー接続が形成されます(これが不可能な場合、スルーライン・アダプタを挿入します)。これにより、ポート2に既知の大きさの入射信号が入ります。
2. チャネル'A'で測定される信号は、ポート2からの反射信号です。
3. 以降の伝送および反射測定から得られた 負荷整合誤差が減算されます。

周波数応答反射トラッキング誤差

反射測定は、チャネル'A'(反射チャネル)の信号とチャネル'R1'(基準チャネル)の信号とを比較することによって行われます。これを比測定(A/R1)と呼びます。
理想的な反射測定では、チャネル'A'とチャネル'R1'のレシーバの周波数応答は同一です。
実際にはそうではなく、周波数応答反射トラッキング誤差が生じます。これは、振幅と位相が周波数の関数として変化するすべてのテスト変動のベクトル和です。この変動には以下のような要因があります。

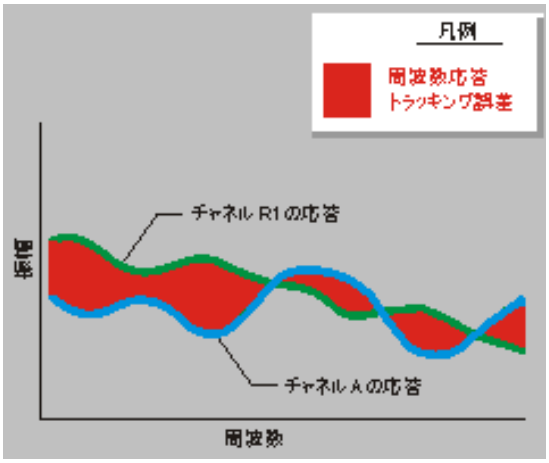
- 信号分離デバイス
- テスト・ケーブル
- アダプタ
- 基準信号パスとテスト信号パスの間のバラツキ

周波数応答反射トラッキング誤差は、本器が測定して低減することができます。



周波数応答反射トラッキング誤差の測定と低減の仕組み

1. 校正中に、ショート標準がポート1に接続されます。ショートからの既知の反射がチャネル'A'で測定され、本器に記憶されます。
2. オープン標準がポート1に接続されます。オープンからの既知の反射がチャネル'A'で測定され、本器に記憶されます。
3. 記憶された2つのチャネル'A'信号が平均されます。
4. チャネル'A'の平均応答が、'R1'で測定された基準信号と比較されます。
5. この結果は、チャネル'A'とチャネル'R1'のレシーバの周波数応答の差です(下の図を参照)。この周波数応答反射トラッキング誤差が、以降のDUT測定から減算されます。



注記: 反射応答校正では、1種類の校正標準(オープンまたはショート)だけが測定されるので、誤差補正に対するその部分だけが用いられます。

周波数応答伝送トラッキング誤差

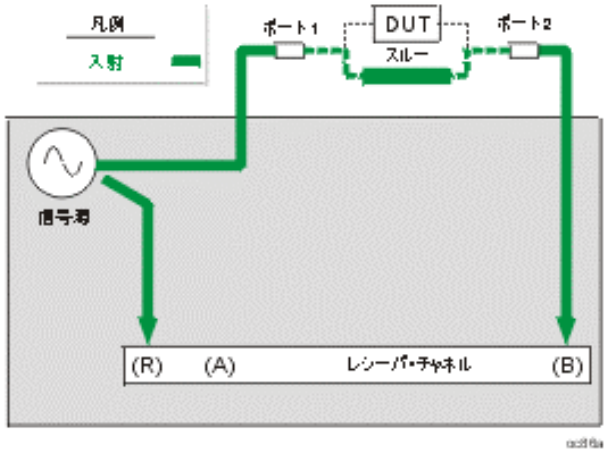
伝送測定は、チャネル'B'(伝送チャネル)の信号とチャネル'R1'(基準チャネル)の信号とを比較することによって行われます。これを比測定(B/R1)と呼びます。

理想的な伝送測定では、チャネル'B'とチャネル'R1'のレシーバの周波数応答は同一です。

実際にはそうではなく、周波数応答伝送トラッキング誤差が生じます。これは、振幅と位相が周波数の関数として変化するすべてのテスト変動のベクトル和です。この変動には以下のような要因があります。

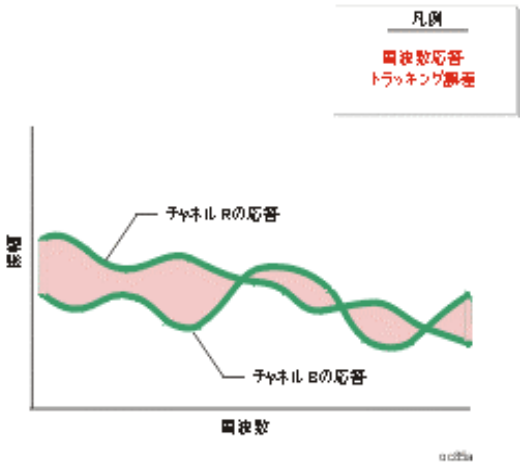
- 信号分離デバイス
- テスト・ケーブル
- アダプタ
- 基準信号パスとテスト信号パスの間のバラツキ

周波数応答伝送トラッキング誤差は、本器が測定して低減することができます。



周波数応答伝送トラッキング誤差の測定と低減の仕組み

1. 校正中に、ポート1とポート2のテスト・コネクタを接合することにより、完全なゼロレングス・スルー接続が形成されます(これが不可能な場合、スルーライン・アダプタを挿入します)。これにより、ポート2に既知の大きさの入射信号が入ります。
2. チャネル'B'とチャネル'R1'で測定が行われ、結果が記憶されます。
3. チャネル'R1'とチャネル'B'の信号が比較されます。
4. 結果はチャネル'B'とチャネル'R1'のレシーバの周波数応答の差です(下の図を参照)。この周波数応答伝送トラッキング誤差が以降のデバイス測定から減算されます。



.

系統誤差の測定結果は誤差項として保存されます。この誤差項をモニタすることにより、本器の状態と測定の正確さを知ることができます。

誤差項を印刷または保存しておくことにより、現在の誤差項と、同じ機器と校正キットを使って以前に生成された結果とを定期的に比較することができます。以前に生成された結果が得られない場合、サービス・ガイドの第4章「誤差項」に記載されている代表値または代表的な誤差項測定結果を参照してください。

- 安定したシステムでは、約6ヶ月以上の間再現性のある誤差項が生成されるはずです。
- 同一の周波数レンジ、パワー、レシーバ設定に対する誤差項が突然変化した場合、システム・コンポーネントのトラブルシューティングを行う必要があります。誤差項のトラブルシューティング方法については、サービス・ガイドの第4章「誤差項」を参照してください。
- 長期にわたるわずかな誤差項の変化は、ドリフトまたはコネクタとケーブルの摩耗に伴うものであるのが普通です。コネクタの清掃やゲージング、あるいはケーブルの検査で問題が解決することもよくあります。ボタンをクリックすると、誤差項を測定する手順が表示されます。

以下の手順は、誤差項の読取りと記憶の方法を示します。

マウス手順

1. フルSOLT 2ポート校正を実行します。詳細については、測定校正の選択と実行を参照してください。
2. **System**メニューで**Service**、**Utilities**をポイントします。**Evaluate Error Corr Data**をクリックします。
3. ダイアログ・ボックスで、ポート1またはポート2の誤差項の1つを選択します。

- Applyをクリックします。誤差項トレースとダイアログ・ボックスは表示されたままなので、別の選択を行うことができます。
- OKをクリックします。ダイアログ・ボックスがクローズし、誤差項トレースは表示されたままになります。

4. **メニューでAutoscale**をクリックして、測定結果の表示が見やすくなるようにスケールを調整します。
5. マーカを使って誤差項トレースの詳細を調べます。
6. 微妙な障害やわずかな性能低下が疑われる場合、誤差項の大きさを、同じ測定器と校正キットで以前に生成された値と比べてみます。この比較により、問題を精密に調べることができます。以前に生成された値がない場合、あるいはさらに詳細について知りたい場合は、サービス・ガイドの第4章「誤差項」を参照してください。

フロントパネル・キー手順

1. フルSOLT 2ポート校正を実行します。詳細については、測定校正の選択と実行を参照してください。
2. **COMMAND**ブロックで**Menu/Dialog**を押します。
3. **NAVIGATION**ブロックで右タブを押して、**System**メニューに移動します。
4. 下矢印を押して**Service**に移動します。**Click**を押します。
5. 下矢印を押して**Utilities**に移動します。**Click**を押します。
6. **Evaluate Error Corr Data**がデフォルトで選択されています。**Click**を押します。
7. 右タブを押して、ダイアログ・ボックスの**Port 1 Error Terms**または**Port 2 Error Terms**部分を選択します。上下の矢印キーを押して、ポート1またはポート2の誤差項の1つを選択します。
   - 右タブを押してApplyを選択します。Clickを押します。誤差項トレースとダイアログ・ボックスは表示されたままなので、別の選択を行うことができます。
   - 右タブを押してOKを選択します。Clickを押します。ダイアログ・ボックスがクローズし、誤差項は表示されたままになります。
8. **COMMAND**ブロックで**Menu/Dialog**を押します。
9. 右タブを押して**Scale**メニューを選択します。
10. Autoscaleがデフォルトで選択されています。**Click**を押して、測定結果の表示が見やすくなるようにスケールを調整します。
11. マーカを使って誤差項トレースの詳細を調べます。
12. 微妙な障害やわずかな性能低下が疑われる場合、誤差項の大きさを、同じ測定器と校正キットで以前に生成された値と比べてみます。この比較により、問題を精密に調べることができます。以前に生成された値がない場合、あるいはさらに詳細について知りたい場合は、サービス・ガイドの第4章「誤差項」を参照してください。

誤差項の自動読取り

誤差項の読取りと記憶のためのSCPIおよびCOMコマンドについては、Data / Error Termsを参照してください。

校正標準

概要　　概念　　標準の変更方法

測定校正の際には、本器は標準を測定し、その結果を標準の「理想モデル」と数学的に比較します。その差が誤差項に分離され、後の誤差補正の際にデバイス測定結果から除去されます。

あらゆる校正標準に対しては、電気遅延、インピーダンス、損失などが精密に決定され、定義として付属しています。本器はこれらの定義を記憶しており、それを使って誤差補正項を計算します。

校正キット

校正キットには、校正プロセスで本器が測定する標準(デバイス)が含まれています。さまざまなコネクタ・タイプ向けの校正キットがAgilent Technologiesから提供されています。キットには通常、以下のものが含まれます。

- ショート1個
- オープン1個
- インピーダンス整合負荷1個

定義

- 校正「標準」(番号1～8で示す)とは、系統誤差の測定に用いられる固有の既定義物理デバイスです。例えば85033E 3.5 mm校正キットの場合、標準1はショートです。
- 標準「タイプ」とは、その標準と組み合わせて使用するモデルの形式または構造を定義する以下の4つの基本タイプの1つです。
  - ショート
  - オープン
  - 負荷
  - スルー
- 標準1のタイプはショートです(3.5 mm校正キットの場合)。
- 標準「特性」とは、選択したモデルで用いられる標準の数値的、物理的特性です。

ポートとのインタフェースにアダプタを必要とするキットの場合、アダプタは位相整合されているのが普通です。校正キットに含まれない標準を使用する場合は、その特性をユーザ定義キットに記述します。これについては次の「校正キットの変更」で説明します。

本器で使用できる標準校正キット(本器に定義が記憶されているものを含む)のリストは、使用可能アクセサリの項にあります。

校正キット･ファイルの変更

ユーザ定義校正キット・ファイルを作成する必要があるのは、以下のような場合です。

- 使用するポートのコネクタ・タイプがCal Kitメニューにリストされているが、校正標準(ショートなど)の特性が本器に記憶されている校正キット定義と異なる場合。
- 使用するポートのコネクタ・タイプと一致するものが、本器に記憶されている校正キット定義に存在しない場合。
- ノンインサータブル・デバイスの校正を行う場合。正確な測定校正を参照してください。

既定義の校正キット・ファイルを変更し、ユーザ定義校正キットとして保存することもできます。元の既定義の校正キット・ファイルは変更されずに残ります。

注記: ここでは校正キット変更については詳詳しく説明致しません。詳細については、Webサイトwww.agilent.com/find/assistでProduct Note 8510-5A, "Specifying Calibration Standards for the HP 8510 Network Analyzer"(カタログ番号5956-4352)を検索の上ご覧ください。

校正標準とその定義を作成、変更する手順は以下のとおりです。

a. 標準デバイスの選択
b. Modify Calibration Kitメニューの使用

標準デバイス

最初に、必要な確度を満たす測定校正方法を選択します(使用できる測定校正方法の説明については、測定校正の選択と実行を参照してください)。選択した方法によって、校正に必要な標準デバイスが決まります。

同軸校正標準としては、以下の3種類が一般的です。

- 0電気長ショート
- シールド付きオープン
- 整合負荷終端(現時点では本器はスライディング・ロードやオフセット・ロードをサポートしません)

注記: 「スルー標準」は通常は物理的な標準ではなく、テスト・ポート同士を直接接続することにより形成される長さ0のスルーです。スルー標準の詳細については、「正確な校正」を参照してください。

測定校正に使用する校正標準は、DUTと同じ媒体(同軸、導波管、マイクロストリップなど)中になければなりません。ニーズを満たすには下記の方法があります。

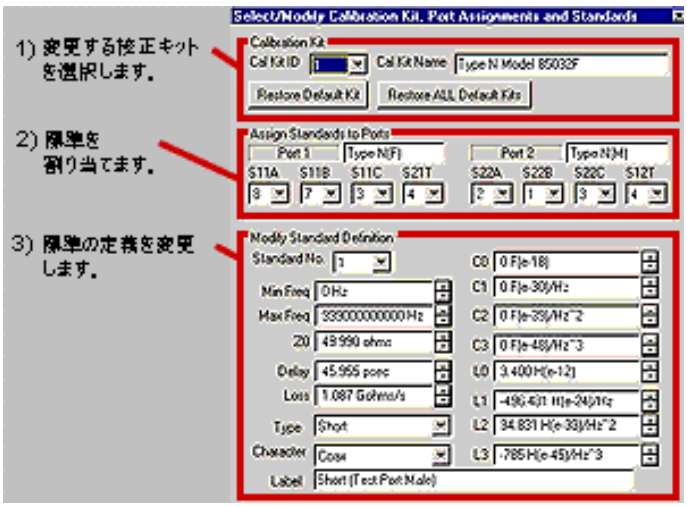
- 標準を購入する
- 標準を作成する
- 既存の標準を変更する

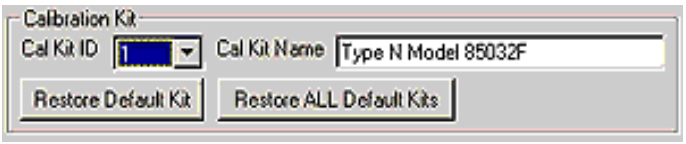
どの方法を使用する場合も、校正標準は以下の基準に基づいて選択します。

- **応答が精密に定義され**、再現性があって温度に対して安定していること。
- それぞれのインピーダンスが**明確に異なっている**こと。例えば:
  - 長さ0のショートとシールド付きオープンは、180°の位相分離を示します。
  - 整合精密負荷は、ショートとオープンに対して40～50 dBの振幅分離を示します。
- 必要な周波数レンジの全域にわたる**広い周波数範囲**を持つこと。

Modify Calibration Kitメニュー

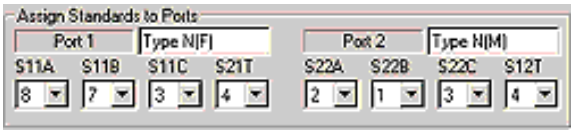
各機能について知るには下の図を使用してください。





Calibration Kitブロックには下記のものが含まれます。

- **Cal Kit ID** 10個の校正キットIDから1つを選択できます(下の表を参照)。
  - ID番号1～6は、既定義の校正キット定義です。将来の測定でこれらのキットを使用する可能性がある場合、これらのキットの定義は変更しないでください。
  - ID番号7～10はユーザ定義校正キットです。既定義の校正キットを変更してユーザ・キットとして保存することができます。元の既定義の校正キットは変更されずに残ります。
- **Cal Kit Name** 校正キット名(編集可能)。
- **Restore Default Kit** 現在選択されている校正キットを既定義の値に戻します。
- **Restore ALL Default Kits** すべての校正キットを既定義の値に戻します。



Assign Standards to Portsブロックには下記のものが含まれます。

- **Port 1** ポート1のコネクタ・タイプ(編集可能)。
- **Port 2** ポート2のコネクタ・タイプ(編集可能)。
- 標準を割り当てるための8個のクラス(各ポートに4個ずつ):

注記: 上の例では、N型標準がさまざまなクラスに割り当てられています。

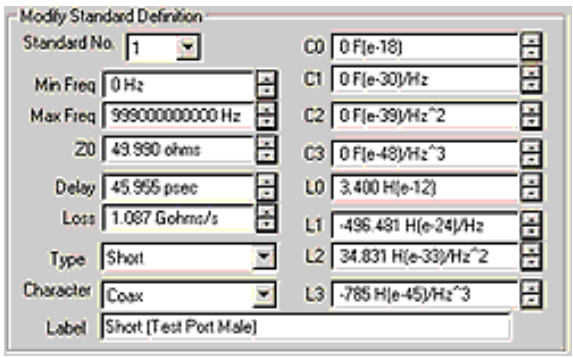
校正クラスは、誤差項計算における位置と標準とを対応付ける役割を果たします。A、B、Cの各クラスは下記の計算に用いられます。

- 方向性(S11C、S22C)
- 信号源整合(S11A、S11B、S22A、S22B)
- 反射トラッキング・エラー(S11A、S11B、S22A、S22B)

"T"クラスは下記の計算に用いられます。

- 伝送トラッキング・エラー(S21T、S12T)
- 負荷整合(S21T、S12T)

8つの校正クラスのそれぞれに校正標準を割り当てる必要があります。校正キットのデフォルト番号は以下のとおりです。



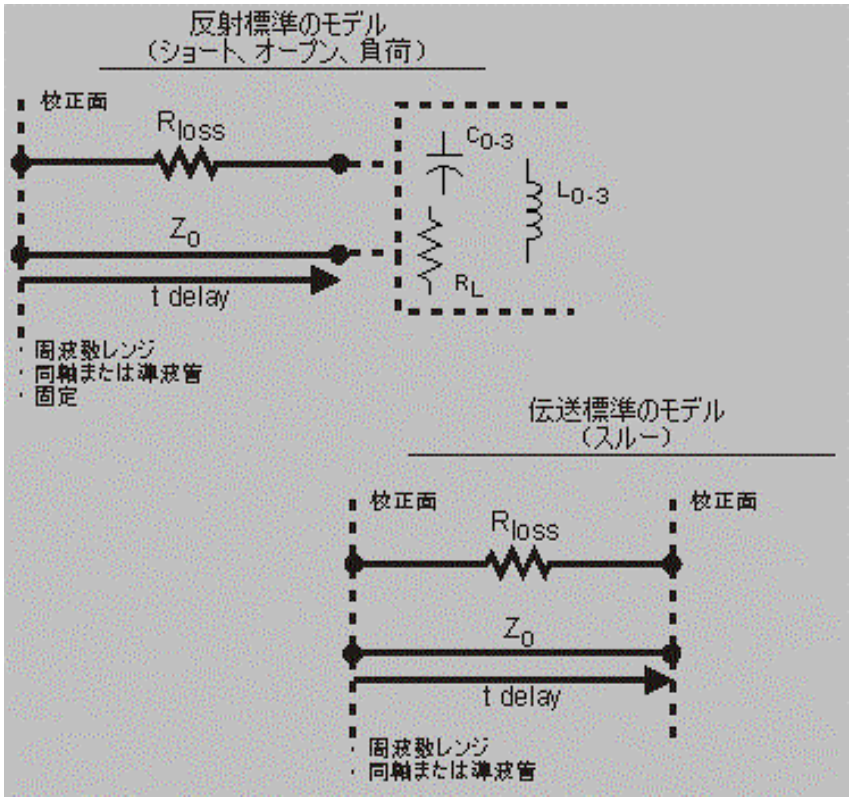
Modify Standard Definitionブロックには下記のものが含まれます(定義の完成のために必要)。

- **標準番号**(1～8)
- **最小周波数**
- **最大周波数**
- $\mathbf{Z_o}$(インピーダンス)
- **遅延**
- **損失**
- **タイプ**(オープン、ショート、負荷、スルー)
- **性質**(同軸、導波管)
- **ラベル**
- キャパシタンス値(**C0**、**C1**、**C2**、**C3**) - オープンの記述にのみ使用。
- インダクタンス項(**L0**、**L1**、**L2**、**L3**) - ショートの記述にのみ使用。

校正キットの定義は、選択した標準の電気的特性です。これらの電気的特性を求めるには2つの方法があります。

- 各標準の物理寸法と材料から導出
- 校正済みのネットワーク・アナライザで測定

特性



- Z_0
  - 定義する標準と実際の測定面との間のオフセットのインピーダンス。
  - 通常はシステムの特性インピーダンスに設定。
- 遅延
  - 定義する標準と実際の測定面との間の一様な長さの伝送ラインと等価。
  - オープン、ショート、負荷については、測定面から標準までの1方向の通過時間(秒単位)を遅延として入力します。
  - スルーについては、測定面の間の1方向の通過時間を遅延として入力します。
  - 遅延を測定または計算するには、標準の精密な物理長を速度係数で割ります。

注意: 校正中に2つのポートをスルー・アダプタによって接続する場合、これらのポートのそれぞれに使用する校正キットを、スルー・アダプタに対する同一の遅延値で変更する必要があります。

- 損失
  - 同軸ケーブルの1方向の長さにおける表皮効果によるエネルギー損失を指定します。
  - 損失の値は、1 GHzでのΩ/s単位で入力します。
  - 多くのアプリケーションでは、損失値を0に設定しても悪化は無視できる程度です。
  - 標準の損失を計算するには、秒単位の遅延と1 GHzでのdB単位の損失を測定し、下記の計算を行います。

$$\text{Loss}\left(\frac{\Omega}{s}\right) = \frac{\text{loss (dB)} \times Z_0(\Omega)}{4.3429(\text{dB}) \times \text{delay(s)}}$$

- C0、C1、C2、C3
  - 高い周波数では、オープンの反射特性が不完全になるのが普通です。これは、フリンジング・キャパシタンスのために、周波数によって変化する位相シフトが生じるからです。この現象は防ぐことができませんが、本器の計算にはオープンのキャパシタンス・モデルが組み込まれています。このモデルは周波数の関数としての3次の多項式であり、多項式の係数はユーザ定義可能です。
  - キャパシタンス・モデルの式は下記の通りです。
  - $C = (C0) + (C1 \times F) + (C2 \times F^2) + (C3 \times F^3)$
  - (Fは測定周波数)
  - 式の各項は、オープンの指定時に下記のように定義されます。
    - C0項は3次多項式の定数項であり、Fで表されます。
    - C1項はF/Hzで表されます。
    - C2項はF/Hz^2で表されます。
    - C3項はF/Hz^3で表されます。
- L0、L1、L2、L3
  - 高い周波数では、ショートの反射特性が不完全になるのが普通です。これは、残留インダクタンスのために、周波数によって変化する位相シフトが生じるからです。この現象は防ぐことができませんが、本器の計算にはショートのインダクタンス・モデルが組み込まれています。このモデルは周波数の関数としての3次の多項式であり、多項式の係数はユーザ定義可能です。
  - インダクタンス・モデルの式は以下のとおりです。
  - $L = (L0) + (L1 \times F) + (L2 \times F^2) + (L3 \times F^3)$
  - (Fは測定周波数)
  - 式の各項は、ショートの指定時に下記のように定義されます。
    - L0項は3次多項式の定数項であり、Hで表されます。
    - L1項はH/Hzで表されます。
    - L2項はH/Hz^2で表されます。
    - L3項はH/Hz^3で表されます。

標準の定義ができたら、ダイアログ・ボックスのそれぞれのフィールドに入力します。

ほとんどの校正キットでは、ショート、オープン、負荷のオス型とメス型で同じ定義が用いられます。

既存のほとんどの校正キットでは、「スルー」標準の定義(通常は標準番号4)は「長さ0のスルー」の定義です。遅延と損失が0であることに注意してください。これは「スルー」標準が存在しないことを意味します。校正の際には、2つのテスト・ポートを直接接合します。詳細については、正確な測定校正を参照してください。

注記: 測定結果の確度は、校正標準がその定義にどの程度一致するかに直接左右されます。標準が損傷したり摩耗したりした場合、測定確度は低下します。

電子校正(ECal)

- ECalは完全に電子的な校正ソリューションです。
  - 1ポート(反射)校正とフル2ポート校正を高速かつ容易に実現できます。
  - オペレータの誤操作の余地も少なくなります。
  - 各標準(校正モジュール内部に存在)は、PINダイオードやFETスイッチによって切り換えられるため、摩耗することがありません。
- 校正モジュールの特性評価には、TRL校正済みのネットワーク・アナライザが用いられます。
- ECalの確度は、正確なTRL校正に比べれば劣ります。

詳細については、Agilent計測お客様窓口にお問い合わせください。

ボタンをクリックすると、校正キット・ファイルの変更手順が表示されます。

マウス手順

1. **Calibration**メニューで**Advanced Modify Cal Kit**をクリックします。
2. ダイアログ・ボックスの**Calibration Kit**部分にある**Cal Kit ID**ボックスをクリックします。値を入力するか、矢印ボタンを使って、目的の校正キットのID番号を選択します。
   - 現在選択されている校正キットを既定義の値に戻すには、Restore Default Kitをクリックします。
   - すべての校正キットを既定義の値に戻すには、Restore ALL Default Kitsをクリックします。
3. **Cal Kit Name**ボックスで、既定義の校正キット名を変更して新しい名前にします。

ポートへの標準の割当て

1. ダイアログ・ボックスの**Assign Standards to Ports**部分で、**Port 1**ボックスをクリックします。このコネクタ・タイプを編集します。
2. **Port 2**ボックスで、このコネクタ・タイプを編集します。
3. 8個の校正クラス・ボックス(S11Aなど)に、新しい標準番号を入力するか、ボックスの矢印ボタンをクリックして標準番号を選択します。

標準定義の変更

1. ダイアログ・ボックスの**Modify Standard Definition**部分で、**Standard No**ボックスをクリックします。新しい標準番号を入力するか、矢印ボタンをクリックして標準番号を選択します。
2. 前のステップで選択した標準番号に対して、標準のすべての定義(最小周波数、最大周波数、Z0、遅延、損失など)を入力します。定義の値を入力するか、ボックスの矢印ボタンをクリックして値を選択します。
3. 各標準番号に対してステップ1と2を繰り返します。
4. **OK**をクリックします。

フロントパネル・キー手順

1. **COMMAND**ブロックで、**Menu/Dialog**を押します。
2. **NAVIGATION**ブロックで、右タブを押して**Calibration**メニューに移動します。
- **下矢印を押して、**Advanced Modify Cal**に移動します。**
- Click**を押します**。
3. ダイアログ・ボックスのCalibration Kit部分にある**Cal Kit ID**ボックスが自動的に選択されています。値を入力するか、上または下の矢印を押して校正キットID番号を選択します。
4. 右タブを押して、**Cal Kit Name**ボックスに移動します。既定義の校正キット名を変更するには、新しい校正キット名を入力します。
5. 右タブを押して**Restore Default Kit**に移動します。現在選択されている校正キットを既定義の値に戻すには、**Click**を押します。
6. 右タブを押して**Restore ALL Default Kits**に移動します。すべての校正キットを既定義の値に戻すには、**Click**を押します。

ポートへの標準の割当て

1. **NAVIGATION**ブロックで右タブを押して、ダイアログ・ボックスの**Assign Standards to Ports**部分にある**Port 1**ボックスを選択します。コネクタ・タイプを編集します。
2. 右タブを押して**Port 2**ボックスに進みます。コネクタ・タイプを編集します。
3. 右タブを押して、8個の校正クラス・ボックス(S11Aなど)のそれぞれに移動します。標準番号を入力するか、上または下の矢印を押して、各ボックスの標準番号を選択します。

標準定義の変更

1. 右タブを押して、ダイアログ・ボックスの**Modify Standard Definition**部分にある**Standard No.**ボックスを選択します。標準番号を入力するか、上または下の矢印を押して標準番号を選択します。
2. 前のステップで選択した標準番号に対して、標準の定義(最小周波数、最大周波数、Z0、遅延、損失など)を入力します。定義の値を入力するか、上または下の矢印を押して各ボックスの値を選択します。
3. 各標準番号に対してステップ1と2を繰り返します。
4. **COMMAND**ブロックで**OK**を押します。

測定校正の選択と実行

概要　　校正の選択　　ウィザードと機能　　校正の実行方法

どの測定校正を選択して実行すればよいかは、下記の要素に依存します。

- 実行する測定
- 測定アプリケーションに要求される確度のレベル

ボタンをクリックすると下記事項についての説明が表示されます。

測定校正方法

一般に、校正を正確にしようとすると、下記の結果が生じます。

- 校正の実行時間が長くなります。
- 測定時間が長くなります。

したがって、測定校正方法を選択する際には、デバイス測定でどの程度の不確かさが許容されるかを知っておくことが重要です。

Sパラメータと測定の向き(順方向、逆方向、両方)に応じて、下記の測定校正方法が選択できます。

校正方法: なし

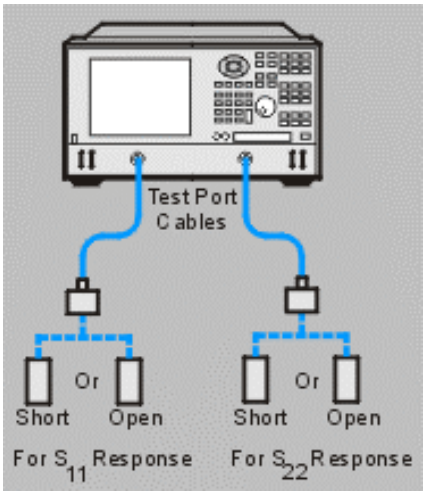
- すべてのSパラメータ測定に使用できます。
- 校正標準測定はありません。
- 最も高速だが確度は劣ります。
- あまり確度を必要としない測定だけに使用します。

オープンまたはショート・レスポンス校正(反射)

- **1**ポートに対する**反射**測定だけに使用します。
- オープンまたはショート標準を測定します。
- 非常に高速で、ある程度の確度を実現する校正です。

注記: 2ポート・デバイスに対して反射測定を行う場合、1ポート反射校正を実行した後で出力ポートを終端する必要があります。終端しないと、出力ポートからの反射信号のために測定の確度が低下します。挿入損失が小さいデバイスでは特に注意が必要です。

オープンまたはショート・レスポンス校正の構成

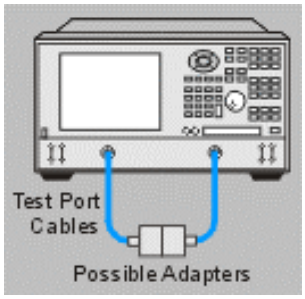


スルー・レスポンス校正(伝送)

- **1**方向の**伝送**測定だけに使用します。
- スルー標準を測定します。
- 非常に高速で、ある程度の確度を実現する校正です。

注記: 本器の定義済み校正キットのスルー定義は、0長のスルーを前提としています。アダプタをスルーとして使用する場合、最高の確度を得るにはスルーの特性評価を行う必要があります。詳細については、正確な測定校正を参照してください。

スルー・レスポンス校正の構成

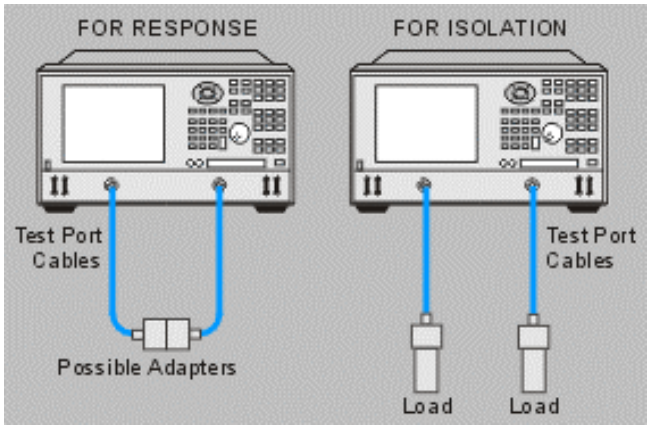


スルー・レスポンス/アイソレーション校正(伝送)

- **1**方向の**伝送**測定だけに使用します。
- vのアイソレーション部分は、挿入損失が大きいデバイスの測定向けです。
- スルーおよび負荷標準を測定します。
- 非常に高速で、ある程度の確度を実現する校正です。

注記: 本器の定義済み校正キットのスルー定義は、0長のスルーを前提としています。アダプタをスルーとして使用する場合、最高の確度を得るにはスルーの特性評価を行う必要があります。詳細については、正確な測定校正を参照してください。

スルー・レスポンス/アイソレーション校正の構成

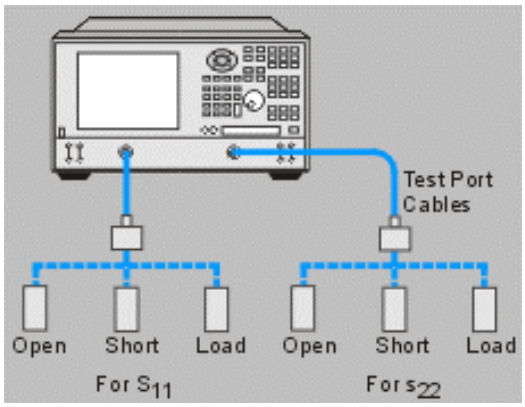


1ポート反射校正

- **1**ポートに対する**反射**測定だけに使用します。
- ショート、オープン、負荷標準を測定します。
- 1ポート測定で高い確度を実現します。

注記: 2ポート・デバイスに対する反射測定を実行する場合、1ポート反射校正を実行した後で、出力ポートを終端します。そうしないと、出力ポートから反射された信号のために測定確度が低下します。特に、挿入損失が小さいデバイスの場合にこれが問題となります。

1ポート(反射)校正の構成

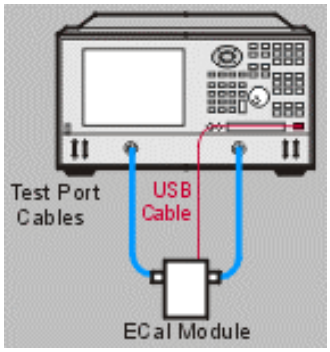


ECal 1ポート反射校正

- **1**ポートに対する反射測定だけに使用します。
- ソリッドステート・モジュール(USBコネクタに接続)を測定します。
- 1ポート測定で高い確度を実現します。
- テスト・セットアップの摩耗を最小限に抑えられます。
- オペレータ・エラーの余地が少なくなります。
- 校正時間が短縮されます。

注記: 2ポート・デバイスに対する反射測定を実行する場合、1ポート反射校正を実行した後で、出力ポートを終端します。そうしないと、出力ポートから反射された信号のために測定確度が低下します。特に、挿入損失が小さいデバイスの場合にこれが問題となります。

ECal校正の構成

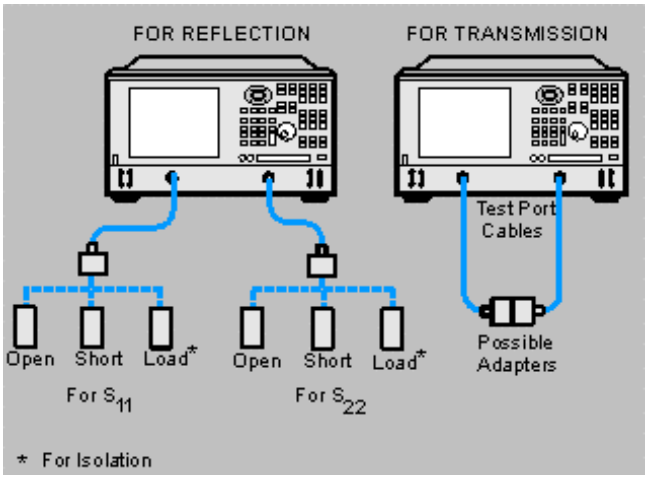


フル2ポートSOLT校正(ショート、オープン、負荷、スルー)

- すべてのSパラメータ測定に使用できます。
- ショート、オープン、負荷、スルー標準を測定します。
- 順方向で6個、逆方向で6個、合計12個の誤差が除去されるため、「12項誤差補正」と呼ばれます。
- きわめて高い測定確度を実現します。

注記: アイソレーション校正は、挿入損失が大きいデバイスを測定する際に役立ちます。挿入損失が大きいデバイスを測定する場合以外は、アイソレーション校正を実行しないでください。アイソレーション校正では本器のノイズ・フロアに近い低レベル信号の測定が必要なので、誤差が加わるおそれがあります。

フルSOLT 2ポート校正の構成



フルTRL 2ポート校正(スルー、反射、ライン)

- すべてのSパラメータ測定に使用できます。
- スルー、反射(ショートまたはオープン)、ライン(伝送)標準を測定します。
- 順方向で6個、逆方向で6個、合計12個の誤差が除去されるため、「12項誤差補正」と呼ばれます。
- 全校正中で最高の確度を実現します。

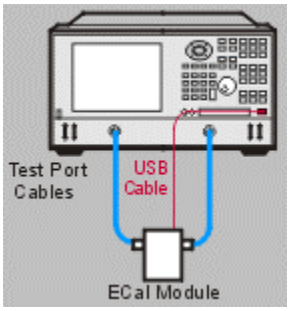
注記: アイソレーション校正は、挿入損失が大きいデバイスを測定する際に役立ちます。挿入損失が大きいデバイスを測定する場合以外は、アイソレーション校正を実行しないでください。本器のノイズ・フロアに近い低レベル信号の測定が必要なので、誤差が加わるおそれがあります。

本器のメモリに定義されているTRL校正キットは、3.5 mm TRLモデル85052Cだけです。3.5 mm以外のコネクタ・タイプに対してTRL校正を実行する場合、TRL標準と定義の変更や作成が必要です。標準の変更の詳細については、校正標準を参照してください。

ECalフル2ポート校正

- すべてのSパラメータ測定に使用できます。
- ソリッドステート・モジュール(USBコネクタに接続)を測定します。
- 順方向で6個、逆方向で6個、合計12個の誤差が除去されるため、「12項誤差補正」と呼ばれます。
- きわめて高い測定確度を実現します。
- テスト・セットアップの摩耗を最小限に抑えられます。
- オペレータ・エラーの余地が少なくなります。
- 校正時間の大幅な短縮が可能です。

ECal校正の構成



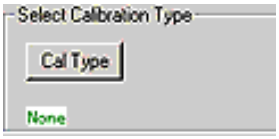
測定校正選択のまとめ

ボタンをクリックすると下記の事項について知ることができます。

校正ウィザード

校正ウィザードは、一連のダイアログ・ボックスを使って校正プロセスを容易に実行するためのものです。

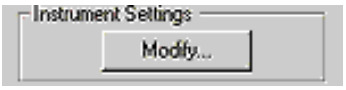
注記: 本器の校正キット・ラベルは、校正標準のコネクタではなくテスト・ポート・コネクタのオス/メスを表します。



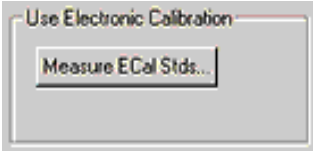
Select Calibration Typeブロックには下記のものがあります。

- 現在選択されている校正タイプ。この例では、"None"(なし)が選択されています。
- **Cal Type**ダイアログ・ボックス。これには下記の校正タイプの選択肢があります。
  - オープン・レスポンス
  - ショート・レスポンス
  - スルー・レスポンス
  - スルー・レスポンス/アイソレーション
  - 1ポート(反射)
  - フルSOLT 2ポート
  - フルTRL 2ポート
  - なし

Cal Typeには、**Omit Isolation (2-Port types)**の選択もあります。これは、2ポート校正のアイソレーション部分を省略するためのものです。

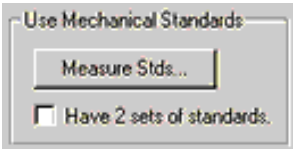


このInstrument Settingsブロックには**Modify**ダイアログ・ボックスがあります。これは本器のスタート/ストップ周波数設定を変更するためのものです。



このUse Electronic calibrationブロックには**Measure ECal Stds**ダイアログ・ボックスがあり、

ECalプロセスをガイドする図と(指定した場合)プロンプトが表示されます。



このUse Mechanical Standardsブロックには下記のものがあります。

- 2つの標準セットの設定。これは、2ポート校正で2つの標準セット(2個のオープンなど)をポート1と2で同時に測定するためのものです。
- **Measure Standards**ダイアログ・ボックス。下記の機能があります。
  - 図とプロンプト(指定した場合)を表示して、機械標準を使った校正プロセスをガイドします。
  - **Show Prompts**設定。校正ウィザードのこの部分でプロンプトを表示するかしないかを切り換えます。
  - **Change Port**設定。既定義の校正キットのポート割当てを変更して、カスタム・スルー/アダプタを特性評価することを可能にします(必要な場合)。この設定ではまた、変更をユーザ定義校正キットとして保存することもできます。スルー標準の詳細については、正確な校正を参照してください。校正キットの変更の詳細については、校正標準を参照してください。
  - **Select Calibration Kit**設定。下記の校正キットの選択が可能です。

注記: キット1～6に対しては、標準校正キット定義が本器に記憶されています。

補正オン/オフ

本器の補正機能のオン/オフを切り換えることができます。

- ONにした場合、表示されたパラメータに対する最新の校正データを使った誤差補正がオンになります。画面の最下部に、Cという注釈表示と、現在の校正タイプ(例えば2-P SOLT)が表示されます。校正に関する表示注釈の詳細については、測定校正の有効性を参照してください。
- OFFにした場合、誤差補正がオフになります。画面最下部にNo Corという注釈表示が表示されます。

校正プロパティ

校正プロパティ表示機能を使うと、現在の校正の下記の設定を表示することができます。

校正データ記憶

- 校正ウィザードの手順の途中で、選択した校正タイプに必要な校正標準(またはモジュール)の測定が終わると、校正セットと機器ステートが*.cstファイルに自動的に保存されます。このファイルは本器のリコール機能を使ってリコールすることができます。
- 校正セットのみ(機器ステートを含まない)を手動で*.calファイルに保存することもできます。このファイルは本器のリコール機能でリコールすることができます。校正データを手動で保存する方法については、ファイルの保存を参照してください。

現在の校正データの変更

- 本器を初期設定すると、校正タイプ「なし」がリコールされます。
- 本器の電源をオンにすると、電源をオフにしたときに使用されていた機器ステートとそれに対応する校正がリコールされます。電源オン/オフ・スイッチを参照してください。

測定校正のリコール

注記: 校正は各チャネルに固有です。例えば、チャネル1で実行した校正を他のチャネルに対してリコールすることはできません。動作モデルを参照してください。

測定校正情報は、校正が実行されたときの機器ステートに結びついています。保存した校正セットを複数の機器ステートに対して使用する場合、測定パラメータ、周波数レンジ、ポイント数が適合する必要があります。

- ステートと校正セット(*.cstファイル)をリコールすると、選択した機器ステートとそれに対応する校正がディスクからリコールされます。
- 校正セット(*.calファイル)をリコールすると、選択した校正がディスクからリコールされますが、対応する機器ステートはリコールされません。
  - 現在のアナライザ設定が校正実行時の設定と正確に一致する場合、校正の確度は非常に高くなります。
  - 補間をオンにしておくと、現在のアナライザ設定が校正実行時の設定と正確に一致しない場合、校正確度が低下します。例外として、現在のアナライザ周波数レンジが校正実行時よりも広い場合、校正はオフになります。
  - 補間をオフにしておくと、アナライザ設定が校正実行時の設定と正確に一致しない場合、校正がオフになります。
- ステート・セット(*.staファイル)をリコールすると、選択した機器ステートがメモリからリコールされますが、対応する校正セットはリコールされません。現在の校正がそのまま用いられます。
  - リコールされたアナライザ設定が校正実行時の設定と正確に一致する場合、校正の確度は非常に高くなります。
  - 補間をオンにしておくと、リコールしたアナライザ設定が校正実行時の設定と正確に一致しない場合、校正確度が低下します。例外として、リコールしたアナライザ周波数レンジが校正実行時よりも広い場合、校正はオフになります。
  - 補間をオフにしておくと、リコールしたアナライザ設定が校正時の設定と正確に一致しない場合、校正がオフになります。

注記: アナライザ設定を変更した場合の校正の有効性の詳細については、測定校正の有効性を参照してください。

ボタンをクリックすると下記の手順が表示されます。

校正ウィザードの使用

マウス手順

1. **Calibration**メニューで**Calibration Wizard**をクリックします。

校正タイプの選択

1. **Cal Type**をクリックします。
2. 目的の校正タイプを選択します。
3. 2ポート校正の場合、**Omit Isolation** (2-Port types)チェック・ボックスがデフォルトで選択されています。2ポート校正プロセスにアイソレーション部分を含める場合、チェック・ボックスをクリックします。
4. **OK**をクリックします。

周波数設定を変更する場合

1. **Modify**をクリックします。
2. 値を入力するか、矢印ボタンを押してスタート/ストップ周波数を選択します。
3. **OK**をクリックします。

電子校正を選択する場合

1. まだ行っていない場合、ステップ"a"～"d"を実行して、ECalモジュールの接続とウォームアップを行います。

- ECalモジュールを本器のUSBコネクタに接続します(ECalモジュールをパラレル・ポート・コネクタに接続しないでください)。
- 電源投入時には、モジュールはWAITステートになります。赤いLED(WAITライト)だけが点灯し、モジュールがウォームアップ・サイクルを開始したことを示します(緑のLEDが一瞬点灯する場合もあります)。
- 約14分後に赤と緑のLEDが両方点灯します。これにより、モジュールが内部動作温度の95%に近づいていることがわかります。
- 約1分後、緑のLED(READYライト)だけが点灯し、赤のLEDは消灯します。これにより、モジュールが動作温度に達し、使用可能になったことがわかります。

2. **Measure ECal**をクリックします。

- ECalプロセスをガイドするプロンプトを表示させたい場合、**Show Prompts**チェック・ボックスをクリックします。

注記: ECalモジュールのウォームアップ時間は、周囲温度によって異なる可能性があります。

注記: ウォームアップ・サイクル中にモジュールを一時的に取り外した場合、最初の電源投入時よりも短い時間でWAITステートからREADYステートに変わります。ただし、赤と緑のLEDが両方点灯した状態(95%のウォームアップ時間)でモジュールを使用した場合、校正品質が低下するおそれあります。工場指定の校正品質を得るには、モジュールがREADY状態(緑のLEDだけが点灯)になるまで待ってから使用してください。

3. **Module**をクリックします。図に示すように接続を行います。
4. **OK**をクリックします。Cal Informationダイアログ・ボックスが画面に表示され、保存された校正セットと機器ステートのファイル名とパスが表示されます。

注記: 選択した校正タイプに必要なモジュールの測定プロセスが終了すると、校正セットと機器ステートが*.cstファイルに自動的に保存されます。このファイルは本器のリコール機能を使ってリコールすることができます。

5. **OK**をクリックします。

機械標準を選択する場合

1. 校正プロセスで2セットの標準を使用する場合、ダイアログ・ボックスのUse Mechanical Standards部分で**Have 2 sets of standards**チェック・ボックスをクリックします。
2. **Measure Stds**をクリックします。

- 校正プロセスをガイドするプロンプトを表示させたい場合、**Show Prompts**チェック・ボックスをクリックします。

3. **Select Cal Kit**をクリックします。
4. 校正キットのタイプを選択します。
5. **OK**をクリックします。

アナライザのポートに割り当てられた校正キットを変更する場合

1. **Change Port**をクリックします。
2. **Cal Kit ID**ボックスに値を入力するか、矢印ボタンをクリックして、校正キットのID番号を選択します(ID番号1～6は既定義の校正キット定義、ID番号7～10はユーザ定義校正キット定義です)。
3. **Cal Kit Name**ボックスで校正キット名を編集します。
4. **Port 1 Model**ボックスで矢印ボタンをクリックして、ポート1に対するAgilent校正キット・モデルを選択します。ポート1の測定基準面のコネクタのオス/メスを選択します。
5. **Port 2 Model**ボックスで矢印ボタンをクリックして、ポート2に対するAgilent校正キット・モデルを選択します。ポート2の測定基準面のコネクタのオス/メスを選択します。
6. **THRU**または**Adapter**ボックスで矢印ボタンを押して、スルーまたはアダプタを選択します。
7. 前のステップでカスタム・スルー/アダプタを選択した場合:
   a. **Specify Custom Thru/Adapter**をクリックします。
   b. **Label** ボックスでラベルを編集します。
   c. **Z0**ボックスに数値を入力するか、矢印をボタンをクリックして目的の値を選択します。
   d. **Delay**ボックスに値を入力するか、矢印ボタンをクリックして目的の値を選択します。
   e. **Loss**ボックスに数値を入力するか、矢印をボタンをクリックして目的の値を選択します。
   f. **OK**をクリックします。
8. **OK**をクリックします。Cal Informationダイアログ・ボックスが表示され、選択した校正キットIDに新しい設定が保存されたことを示します。**OK**をクリックします。

校正標準の測定

1. 校正標準(**OPEN**、**SHORT**、**LOAD**など)を表すボタンをクリックします。図の通りに接続を行います。
2. **OK**、**OK**をクリックします。Cal Informationダイアログ・ボックスが画面に表示され、保存された校正セットと機器ステートのファイル名とパスが表示されます。

注記: 選択した校正タイプに必要な校正標準の測定プロセスが終了すると、校正セットと機器ステートが*.cstファイルに自動的に保存されます。このファイルは本器のリコール機能を使ってリコールすることができます。

3. **OK**をクリックします。

フロントパネル・キー手順

1. **CHANNEL SETUP**ブロックで**Cal**を押します。
2. **F1 (Cal Wizard)**を押します。

校正タイプの選択

1. **NAVIGATION**ブロックで**Click**を押してCal Typeを選択します。
2. 上下の矢印を押して校正タイプを選択します。
3. 2ポート校正を選択した場合、右タブを押してOmit Isolation (2-Port types)チェック・ボックスを選択します。デフォルトではこのボックスはチェックされています。2ポート校正プロセスのアイソレーション部分を実行する場合、**Click**を押します。
4. **COMMAND**ブロックで**OK**を押します。

周波数設定を変更する場合

1. **NAVIGATION**ブロックで右タブを押して**Modify**を選択します。
2. **Click**を押します。
3. 右タブを押して**Start**および**Stop**ボックスを選択します。値を入力するか上下の矢印を押して、スタート/ストップ周波数を選択します。
4. **COMMAND**ブロックで**OK**を押します。

電子校正を選択する場合

1. まだ行っていない場合、ステップ"a"～"d"を実行して、ECalモジュールの接続とウォームアップを行います。

ECalモジュールを本器のUSBコネクタに接続します。

電源投入時には、モジュールはWAITステートになります。赤いLED(WAITライト)だけが点灯し、モジュールがウォームアップ・サイクルを開始したことを示します。

25℃の場合、約14分後に赤と緑のLEDが両方点灯します。これにより、モジュールが内部動作温度の95%に近づいていることがわかります。

約1分後、緑のLED(READYライト)だけが点灯し、赤のLEDは消灯します。これにより、モジュールが動作温度に達し、READY状態になったことがわかります。

2. **NAVIGATON**ブロックで右タブまたは左タブを押して、**Measure ECal Stds**を選択します。**Click**を押します。
3. 右タブを押して**Show Prompts**チェック・ボックスに移動します。ECalプロセスをガイドするプロンプトを表示させたい場合、**Click**を押します。
4. 左タブを押して**Module**を選択します。**Click**ボタンを押し、図の通りに接続を行います。
5. **COMMAND**ブロックで**OK**を押します。Cal Informationダイアログ・ボックスが画面に表示され、保存された校正セットと機器ステートのファイル名とパスを表示します。

注記: 選択した校正タイプに必要なモジュールの測定プロセスが終了すると、校正セットと機器ステートが*.cstファイルに自動的に保存されます。このファイルは本器のリコール機能を使ってリコールすることができます。

6. **OK**を押します。

機械標準を選択する場合

1. **NAVIGATON**ブロックで右タブを押して、ダイアログ・ボックスの**Use Mechanical Standards**部分にある**Have 2 sets of standards**チェック・ボックスに移動します。校正プロセスで2セットの標準を使用する場合、**Click**を押します。
2. 左タブを押して**Measure Stds**に移動します。**Click**を押します。

- 左タブを押して**Show Prompts**チェック・ボックスに移動します。校正プロセスをガイドするプロンプトを表示させたい場合、**Click**を押します。

3. 右タブを押して**Select Cal Kit**に移動します。**Click**を押します。
4. 上下の矢印を使って校正キットのタイプを選択します。
5. **COMMAND**ブロックで**OK**を押します。

アナライザのポートに割り当てられた校正キットを変更する場合

1. **NAVIGATION**ブロックで右タブを押して**Change Port**に移動します。**Click**を押します。
2. **Cal Kit ID**ボックスに値を入力するか、上下の矢印を使って校正キットのID番号を選択します(ID番号1～6は既定義の校正キット定義、ID番号7～10はユーザ定義校正キット定義です)。
3. 右タブを押して**Cal Kit Name**に移動します。校正キット名を編集します。
4. 右タブを押して**Port 1 Model**ボックスに移動します。上下の矢印を使ってポート1に対する校正キット・モデルを選択します。
5. 右タブを押し、上下の矢印を使ってポート1の測定基準面のコネクタのオス/メスを選択します。
6. 右タブを押して**Port 2**ボックスに移動します。上下の矢印を使ってポート2に対する校正キット・モデルを選択します。
7. 右タブを押し、上下の矢印を使ってポート2の測定基準面のコネクタのオス/メスを選択します。
8. 右タブを押して**THRU**または**Adapter**ボックスに移動します。上下の矢印を押してスルーまたはアダプタを選択します。
9. 前のステップでカスタム・スルー/アダプタを選択した場合:
   a. **右タブを押して**Specify Custom Thru/Adapter**に移動します。**Click**を押します。**
   b. **右タブまたは左タブを押して**Label**ボックスに移動します。ラベルを編集します。**
   c. **右タブを押して**Z0**ボックスに移動します。数値を入力するか、上下の矢印を押して目的の値を選択します。**
   d. **右タブを押して**Delay**ボックスに移動します。数値を入力するか、上下の矢印を押して目的の値を選択します。**
   e. **右タブを押して**Loss**ボックスに移動します。数値を入力するか、上下の矢印を押して目的の値を選択します。**
   f. COMMAND**ブロックで**OK**を押します。**
10. **OK**を何回か押すと、**Cal Information**ダイアログ・ボックスが表示され、選択した校正キットIDに新しい設定が保存されたことを示します。**OK**を押します。

校正標準の測定

1. **NAVIGATION**ブロックで右タブまたは左タブを押して、各校正標準(**オープン**、**ショート**、**負荷**など)のボタンに移動します。**Click**を押します。図の通りに接続を行います。
2. **COMMAND**ブロックで**OK**を押します。Cal Informationダイアログ・ボックスが画面に表示され、保存された校正セットと機器ステートのファイル名とパスが表示されます。

注記: 選択した校正タイプに必要な校正標準の測定プロセスが終了すると、校正セットと機器ステートが*.cstファイルに自動的に保存されます。このファイルは本器のリコール機能を使ってリコールすることができます。

3. **OK**を押します。

補正のオン/オフ

マウス手順

1. **Calibration**メニューで**Correction**に移動します。
2. デフォルト設定は補正オンです。
   - 補正を**オン**にしたい場合、選択をクリックして**Correction ON/off**となるようにします。
   - 補正を**オフ**にしたい場合、選択をクリックして**Correction on/OFF**となるようにします。

フロントパネル・キー手順

1. **CHANNEL SETUP**ブロックで**Cal**を押します。
2. 補正機能のステータスは、アクティブ・エントリ・ツールバーにある緑のボックスに表示されます。ステータスを変更する場合(**オフ**から**オン**にするなど)、**F2 (Corr on/off)**を押します。

校正プロパティの表示

マウス手順

1. **Calibration**メニューで**Properties**をクリックします。
2. **OK**をクリックします。

フロントパネル・キー手順

1. **CHANNEL SETUP**ブロックで**Cal**を押します。
2. **F3 (Properties)**を押します。
3. **COMMAND**ブロックで**Cancel**を押します。

正確な測定校正

概要　　概念　　手順

実行しようとしている測定校正が正確かどうかを確認するには、いくつかの方法があります。ボタンをクリックすると下記の事項について知ることができます。

測定基準面

- ほとんどの測定条件設定では、本器のフロントパネル・コネクタに被試験デバイスを直接接続することができません。多くの場合、本器に接続されたテスト・フィクスチャまたはケーブルに被試験デバイスを接続します。
- 最高の測定確度を実現するには、デバイスを接続する点で校正を行います。これを測定基準面と呼びます(図を参照)。これらの点で校正を行うことにより、テスト・セットアップ(本器のポートと基準面との間に用いられるケーブル、テスト・フィクスチャ、アダプタなど)による誤差が測定され、校正プロセスで除去されます。

ポート延長

- 校正後にポート延長機能を使うことにより、下記のような原因による測定基準面の延長に伴う位相シフトを補正することができます。
  - ケーブル
  - アダプタ
  - フィクスチャ
- ポート延長は、テスト・フィクスチャの位相シフトを補正するための推奨される方法です。
  - 例えば、テスト・フィクスチャのデバイス・コネクタが校正標準と異なっているとします。この場合、フィクスチャ内部の電気長を校正することはできません。ポート延長を使えば、フィクスチャ内部の遅延を、それに伴う位相シフトを除去することによって補正できます。ただし、フィクスチャによる不整合と損失はポート延長では補正できません。
- ポート延長に対して入力する遅延の値は、伝送測定と反射測定の両方に適切に適用されます。例を挙げます:
  - S11を測定する際には、ポート1の延長遅延が2回(往復のため)適用されます。
  - S21を測定する際には、ポート1の延長遅延とポート2の延長遅延が1回ずつ適用されます。
- ポート延長は、そのポートに関連するすべての測定に適用されます。

正しくない校正標準を使用することによる影響

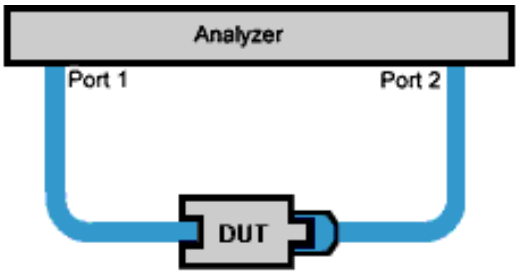
- 校正の際には、被試験デバイスのコネクタとタイプおよびオス・メスの別が一致するコネクタを持つ標準が含まれる校正キットを使用するのが普通です。
- しかし、被試験デバイスと同じコネクタ・タイプが校正キットに存在しない場合もあります。例えば、デバイスのコネクタが3.5 mmで、N型の校正キットを使用する場合などです。この場合、N型の標準をアダプタを使って3.5 mmのテスト・ポートに接続します。

- アダプタは校正の一部であって、テスト・セットアップには含まれないので、これにより反射測定に大きな誤差が生じます。
- 校正プロセスで指定されているのと別の校正標準を使用した場合にも、誤差が生じるおそれがあります。誤差の大きさは、使用する標準が指定された標準とどれだけ異なっているかに依存します。

校正の確度をチェックする方法については、測定校正の正当性を参照してください。.

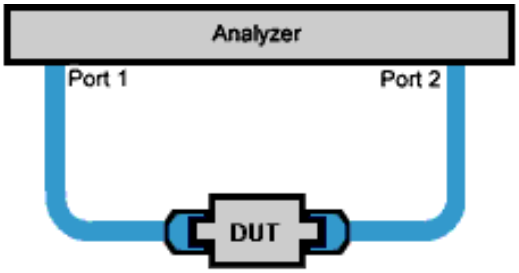
ノンインサータブル・デバイスの校正

下に示すのは、**インサータブル**・デバイスの例です。このデバイスには2個のN型コネクタがあり、一方のポートがオス型、もう一方のポートがメス型です。



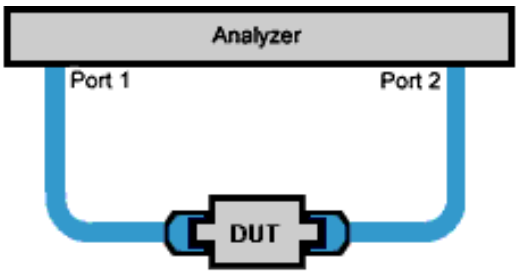
- インサータブル・デバイスに対して校正を行う場合、測定基準面のテスト・ポート同士が直接に接合されます。これをゼロレングス・スルーといいます。その意味は、校正プロセスでアナライザが使用する定義(電気的記述)が、長さ0、遅延なし、損失なし、キャパシタンスなし、インダクタンスなしだということです。
- 校正キットにはスルー標準が含まれない場合もあります。これは、インサータブル・デバイスを使用するという前提に基づいています。

デバイスが**ノンインサータブル**の場合、下の図のように測定基準面のテスト・ポート同士が直接接合できません。このため、アダプタを使ってスルー接続を行う必要があります。このアダプタは測定に含まれないため、測定確度の低下につながります。



ノンインサータブル・デバイスを使用する場合、校正でスルー接続を行うには下記のどれかの方法を使う必要があります。これらの方法は、確度の低い順に並んでいます(最初のものが最も不正確)。

特性未評価スルー・アダプタ



これはノンインサータブル・デバイスの校正方法のうちで最も簡単ですが、確度は最も低くなります。「特性が未知」とは、スルー・アダプタの電気遅延と損失が本器に記録されていないことを指します。

スルー・パスの校正にアダプタが用いられますが、その損失と遅延は未知なので、以降の測定値から除去されません。

特性評価済みスルー・アダプタ

- 特性評価済みスルー・アダプタとは、電気長と損失が本器の校正キット定義に記述されているものをいいます。
- オス-オスまたはメス-メス・アダプタの電池遅延と損失は、アダプタのメーカから提供される場合もあります。提供されていない場合、電気遅延と損失を測定することもできます。ただし、スルー・アダプタ(それ自体がノンインサータブル・デバイス)の測定には、正確な校正が必要です。

高品質スルー・アダプタの特性評価

1. 等価アダプタ交換法(このあとの説明を参照)を使って校正を行います。
2. スルー・アダプタを特性評価します。
3. 校正ウィザードを使って、スルー・アダプタの校正キット定義を変更します(この方法だと、等価アダプタ交換校正は1回しか行う必要がありません)。
4. 特性評価済みのスルーを使って校正を行います。

注記: 特性評価済みスルー・アダプタは、スルー・パス校正だけに使用し、デバイスのテストには使用しません。このため、特性評価済みスルー・アダプタの寿命を延ばすことができます(後で説明するように、等価アダプタ交換法では特性評価済みアダプタの1つがすべてのデバイスのテストに用いられます)。

等価アダプタ交換法

- 等価アダプタ交換校正法は、コネクタ・タイプとオス・メスの別が等しいコネクタを持つデバイス(例えば両側にメス型SMAコネクタを持つもの)に対して非常に有効な方法です。
- この方法では、性能が等しく、オス・メスの別が異なるコネクタを持つ2個の精密整合アダプタを使用します。
  - 例えば、両端にメス型SMAコネクタを持つデバイスを、APC-7 mmテスト・ケーブルを使って測定する場合、7 mm-オス型3.5 mmアダプタと、7 mm-メス型3.5 mmアダプタを使用します。

- アダプタの特性を一致させるため、Agilentの校正キットには下記の特性が等しいアダプタが含まれています。
  - 特性インピーダンス
  - 挿入損失
  - 電気遅延
  - 整合

等価アダプタ交換校正法のプロセス概要

1. スルー接続に必要なアダプタを使って、2ポート校正の伝送部分を実行します。
2. このアダプタを外し、代わりに2番目のアダプタを使って、校正の反射部分を両方のテスト・ポートに対して実行します。
3. アダプタを交換することにより一方のテスト・ポートのオス・メスが変化するため、校正手順の終了後にテスト・デバイスを(2番目のアダプタをつけたまま)挿入して測定することができます。
4. 校正後に残る誤差は、2個のアダプタの差に等しくなります。

等価アダプタ交換校正法の手順を参照してください。

2ポート校正のアイソレーション部分

- 2ポート校正を実行する際には、校正のアイソレーション部分を省略することが可能です。
- 校正のアイソレーション部分では、クロストークすなわちデバイスが存在しないときのテスト・ポート間の信号漏れを補正します。
- アイソレーション校正は、挿入損失が大きいデバイスをテストする場合に実行します。例を挙げます。
  - ある種のフィルタの阻止帯域
  - 開位置のスイッチ
- アイソレーション校正では、誤差モデルにノイズが追加されます。これは、測定が本器のノイズ・フロアに非常に近いところで行われるからです。測定確度を改善するには以下のようにします。
  - アイソレーション校正は必要なときだけ実行します。
  - ノイズを減らすため掃引アベレージを使用します。
- アイソレーション校正の最善の方法は、測定対象のデバイスを本器の各テスト・ポートに接続し、他の2個のデバイス・ポートを終端することです。これにより、アナライザに接続されるインピーダンスは以降のデバイス測定のときと同一になります。
- 上記の方法が実際的でない場合(テスト・フィクスチャ内や、デバイスが1個しかない場合など)、次善の方法として、本器の信号源ポートに終端したデバイス、負荷ポートに終端を接続します。
- 使用できるデバイスがない場合、あるいはデバイスがチューニングされる(そのためにポート整合が変化する)場合、本器の両方のポートに終端を接続します。

ボタンをクリックすると下記の手順が表示されます。

ポート延長機能の使用

マウス手順

1. 測定基準面で校正を実行します。
2. Calibrationメニューで、**Port Extensions**をクリックします。
3. **Port Extensions**チェックボックスをクリックしてポート延長機能をオンにします。
4. **Input**または**Port**ボックスをクリックします。値を入力するか、矢印ボタンをクリックして、入力またはポートの測定基準面とデバイスとの間の遅延を選択します。十分な遅延の値を入力したかどうかは次のようにしてわかります。
   a. 被試験デバイスの代わりにショートを接続します。
   b. ポート延長を調整して、位相応答がフラットになるようにします(なお、ほとんどのショート標準は遅延が0でないため、この方法で遅延を調整するとショート遅延の2倍に等しい遅延誤差が生じます)。
5. Velocity Factorボックスをクリックします。値を入力するか、矢印ボタンをクリックして、校正後に挿入されたデバイス内の伝送媒体(ケーブルやテスト・フィクスチャなど)の速度因子を選択します。速度因子のデフォルト値は1.0で、自由空間中の光速(2.997925×10^8 m/s)に等しくなります。
6. **OK**をクリックします。

フロントパネル・キー手順

1. **測定基準面で校正を実行します。**
2. **COMMAND** ブロックで**Menu/Dialog**を押します。
3. **NAVIGATION**ブロックで、右タブまたは下矢印を押して**Calibration**メニューに移り、**Port Extensions**まで移動します。**Click**を押します。
4. ポート延長機能をオンにするため、**Click**を押します。
5. 右タブを押して**Input**または**Port** ボックスに移動します。値を入力するか、上または下の矢印を押して、入力またはポートの測定基準面とデバイスとの間の遅延を選択します。十分な遅延の値を入力したかどうかは次のようにしてわかります。
   a. 被試験デバイスの代わりにショートを接続します。
   b. ポート延長を調整して、位相応答がフラットになるようにします(なお、ほとんどのショート標準は遅延が0でないため、この方法で遅延を調整するとショート遅延の2倍に等しい遅延誤差が生じます)。
6. 右タブを押して、**Velocity Factor**ボックスに移動します。校正後に挿入されたデバイス内の伝送媒体(ケーブルやテスト・フィクスチャなど)の速度因子を入力します。速度因子のデフォルト値は1.0で、自由空間中の光速(2.997925×10^8 m/s)に等しくなります。
7. **OK**を押します。

スルー･アダプタの特性評価

1. フルSOLT 2ポート校正を、等価アダプタ交換法で実行します。この方法を使用するのは、被試験デバイス(スルー・アダプタ)がノンインサータブル・デバイスだからです。
2. 周波数レンジ設定のほぼ中央でスルー・アダプタの遅延を測定します。測定値を記録します。
3. 周波数レンジ設定のほぼ中央で、スルー・アダプタの挿入損失を測定します。測定値を記録します。
4. これらの値を使って、スルー・アダプタの校正キット定義を変更します。このためには、校正ウィザードでChange Portを選択します。測定校正の選択と実行を参照してください。
5. 特性評価済みスルー・アダプタを含むように変更された校正キットを使って、校正を実行します。

等価アダプタ交換校正法の実行

1. 校正ウィザードで、フルSOLT 2ポート校正を選択します。校正ウィザードの詳細については、「測定校正の選択と実行」を参照してください。
2. 下の図のように、ポート1とポート2の間にアダプタAを接続します。

3. 校正ウィザードのMeasure Mechanical Standardsダイアログ・ボックスでThruを選択します。
4. ポート1とポート2の間のアダプタAを取り外します。
5. アダプタB(被試験デバイスとオス･メスが異なるコネクタを持つもの)を適切なポート(この例ではポート1)に接続します。このアダプタは測定セットアップの一部となります。下の図を参照してください。

6. 校正ウィザードのMeasure Mechanical Standardsダイアログ・ボックスで、Open, Short, and Loadを選択します。
   a. プロンプトが表示されたら、ポート1に対して選択された標準をアダプタBに接続します。
   b. プロンプトが表示されたら、ポート2に対して選択された標準を、ポート2に接続されたケーブルの端に接続します。
7. 下の図に示すように、アダプタBを接続した状態でデバイスを測定します。

注記: 校正に用いられるアダプタBは、測定セットアップの一部です。すなわち、このアダプタに直接テスト・デバイスを接続するため、精密整合アダプタのうち1個が摩耗します。別の方法として、等価アダプタ交換法でスルー・アダプタを特性評価することもできます。

測定校正の有効性

概要　　概念　　校正の検証方法

場合によっては、測定校正の有効性を確認しなければならないことがあります。このトピックでは、校正データが有効で、測定結果にとって必要なレベルの確度を実現しているかどうかを検証する方法について説明します。

ボタンをクリックすると、下記の項目について知ることができます。

補正なし(No Cor)

本器の周波数スパンを広げた場合、現在の校正は無効になり、誤差補正がオフになります(補間機能をオンにしていても)。このように補正がオフになった場合、本器のステータス・バーに"**No Cor**"と表示されます。

誤差補正能力の低下の可能性(C*)

補間

補間機能はデフォルトでオンになっています。以下の測定設定のどれかを変更すると、現在の校正の品質が低下するおそれがあります。この場合、本器のステータス・バーに"**C***"と表示されます。

- 周波数レンジを狭める
- ポイント数を減らす

補間機能がオンの場合、新しい設定に対する誤差補正が元の誤差項から計算されます。測定確度がどれくらい低下するかは、測定ポイント間で振幅と位相シフトがどれくらい変化するかに依存します。変化が小さいほど、補間された誤差補正の性能は高くなります。ただし、補間された誤差補正を使う場合、システム性能は仕様外となります。

注記: ステータス・バーに"**C***"が表示された場合、校正の有効性に疑問があります。これが問題になるかどうかを確認するには、元の校正(ステータス・バーに"**C**"と表示された状態)での測定トレースと、現在の測定トレース(ステータス・バーに"**C***"と表示された状態)の測定トレースとを比較します。2つのトレースの確度の差と、測定アプリケーションで要求される確度に基づいて、校正がニーズを満たすかどうかを判定できます。「校正確度の検証」を参照してください。

補間機能がオフになっている場合、上記の設定を変更すると、現在の校正が無効になり、誤差補正はオフになります。このように補正がオフになった場合、本器のステータス・バーに"**No Cor**"と表示されます。

誤差補正に疑問が生じる場合

以下の測定設定のどれかを変更すると、対応する誤差補正アルゴリズムが本器に存在しないため、現在の校正の品質が低下するおそれがあります。この場合、校正に疑問があることを示すためにステータス・バーに"**C***"と表示されます。

- 掃引時間
- IF帯域幅
- ポートのパワー/減衰
- ステップ掃引のオン/オフ

注記: 最高品質の誤差補正が得られるのは、測定校正が実行されたパワー・レベルに限ります。ただし、測定校正を実行したのと同じアッテネータ・レンジでパワーを変化させる場合、Sパラメータ測定の確度はわずかしか低下しません。別のアッテネータ・レンジを選択した場合、誤差補正の品質はさらに低下します。

校正確度の検証

校正の確度を検証するには、以下の2つの方法のどちらかを使います。

1. 校正を実行した後、スルー接続を行って、測定トレースを本器の内部メモリまたはフロッピー・ディスクに保存します。このトレースを後でリコールして、現在の測定トレース(前と同じケーブルとスルー接続を使用したもの)と比較します。さらに測定確度を改善するには、以下の演算機能を実行して結果の違いを見ます。

- 伝送測定ではデータ/メモリ
- 反射測定ではデータ-メモリ

注記: 校正が許容範囲内(すなわち有効)であるかどうかを簡単に確認するには、リミット・ラインをリコールする方法があります。

2. **クイック・チェック**
   - 実行した校正の有効性に疑問があり、念のために校正の再実行を検討している場合、その前にクイック・チェックのプロセスを実行してみてください。クイック・チェックは短時間で実行できるので、校正が有効であった場合は大幅に時間を節約できます。
   - 校正の品質を検証するには、校正のすぐ後でクイック・チェックを実行するのが最善です。
   - これらのチェックで校正実行後に繰返し問題が発見される場合、ベリフィケーション・キットを使った完全な検証を実行してください(サービス・ガイド参照)。

注記: ショートとオープンの校正標準には多少の電気遅延があります。本器はこの遅延を補正するために、電気遅延の分だけ校正結果にオフセットを付加します。したがって、校正後に標準を測定すると、標準は長さオフセット0のショートやオープンには見えません。これは本器が正常に動作している証拠です。

ボタンをクリックすると下記の手順が表示されます。

注記: 下記の手順で、ポートとは測定基準面のポートを指します。測定基準面については、「正確な測定校正」を参照してください。

クイック・チェック

反射測定のチェック

注記: この手順は、ポート1に対するS_{11}測定に関するものです。ポート2に対するS_{22}測定の場合も、全体としての手順は同じです。

注記: 元の校正に使用したのと同じ校正標準は使用しないでください。同じ標準を測定しても、元の校正で実行された補正の再現性が検証されるだけです。

1. ポート1に校正標準を接続しない状態で、S_{11}の大きさが0 dB付近(おおむね±1 dB以内)であることを確認します。
2. 負荷校正標準をポート1に接続します。S_{11}の大きさは、負荷に応じて-30 dB～-40 dBより小さい値になるはずです。
3. オープンまたはショート校正標準をポート1に接続します。S_{11}の大きさは0 dB付近(0.数dB以内)になるはずです。

伝送測定のチェック

注記: この手順はS_{21}測定に関するものです。ポート2に対するS_{12}測定の場合も、全体としての手順は同じです。

1. ポート1とポート2をスルー接続します。S_{21}の大きさは0 dB付近(0.数dB以内)になるはずです。
2. S_{21} アイソレーションを検証するには、ポート1とポート2にそれぞれ1個ずつの負荷を接続します。S_{21}の大きさは、IF帯域幅に応じて、-60 dBより小さい値になるはずです。

補間のオン/オフ

マウス手順

1. **Calibration**メニューで**Interpolation**に移動します。
2. デフォルト設定は補間オンです。
3. 補間を**オン**にしたい場合、項目をクリックして**Interpolation ON/off**と表示させます。
4. 補間を**オフ**にしたい場合、項目をクリックして**Interpolation on/OFF**と表示させます。

フロントパネル・キー手順

1. **COMMAND**ブロックで**Menu/Dialog**を押します。
2. **NAVIGATION**ブロックで、右タブを押して**Calibration**メニューに移動します。
3. 下矢印を押して**Interpolation**に移動します。
4. デフォルト設定は補間オンです。
5. 補間を**オン**にしたい場合、**Click**を押して**Interpolation ON/off**と表示させます。
6. 補間を**オフ**にしたい場合、**Click**を押して**Interpolation on/OFF**と表示させます。

ステップ4. データの解析

- マーカを使ったデータの検索
- リミットを使ったデバイスのテスト
- 演算機能を用いたデータ処理

マーカ

概要　　マーカ・タイプ　　マーカ機能　　アクティブ化と変更の方法　　検索方法

マーカは、測定データの数値の読取り、特定の値の検索、スティミュラス設定の変更などに使います。このトピックでは、マーカのすべての側面について説明します。

1つの測定トレースには最大10個のマーカを配置できます。

マーカ・タイプ

マーカには、ノーマルと固定の2つのタイプがあります。

ノーマル

ノーマル・マーカは、固定されたスティミュラス位置(直交座標フォーマットではX軸)を持ち、**データ振幅(Y軸)の変化に応答**します。マーカのスティミュラス値を変更すると、マーカがX軸上で左右に移動します。このマーカ・タイプをいずれかのマーカ検索タイプと組み合わせることで、目的のデータを探すことができます。

固定

固定マーカは、トレース上で最初に配置された位置にデータ位置(直交座標フォーマットではY軸)が固定され、**トレース・データの振幅に応じて移動しません**。マーカのスティミュラス値を変更すると、マーカがX軸上で左右に移動しますが、Y軸上の位置は初期値のままで変化しません。このマーカ・タイプを使うと、被試験デバイスに変更を加える前と後の状態を簡単に比較できます。例えば、フィルタのチューニング前と後のテスト結果の違いを固定マーカによって記録することができます。

基準マーカ

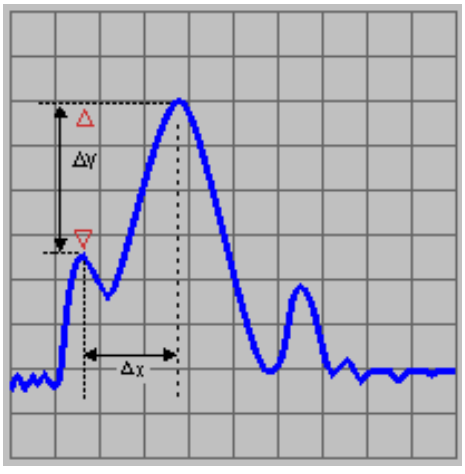
基準位置に相対的にデータを表示することができます。

- まず、最初の位置に**基準マーカ**を配置します。
- 次に、2番目の位置に別のマーカ(**デルタ・マーカ**)を配置します。

デルタ・マーカが表示するデータとスティミュラスの情報は、基準マーカのデータとスティミュラスを基準とした値です。

基準マーカは、**Marker**メニューおよびダイアログ・ボックスまたは、**マーカ・ツールバー**から配置できます。また、SCPIまたはCOMコマンドも使用できます。

注記: 基準マーカがオンになっていない状態でデルタ・マーカを作成すると、専用の基準マーカが自動的にオンになります。この場合でも、基準マーカを目的の位置に配置するのは手動で行う必要があります。



マーカによる検索

マーカを使って、特定の基準に基づいて測定データを検索できます。

検索タイプ

マーカ検索には7つのタイプがあります。

注記: ピークまたはターゲット検索タイプで一致する有効なデータがない場合、マーカは現在位置から動きません。

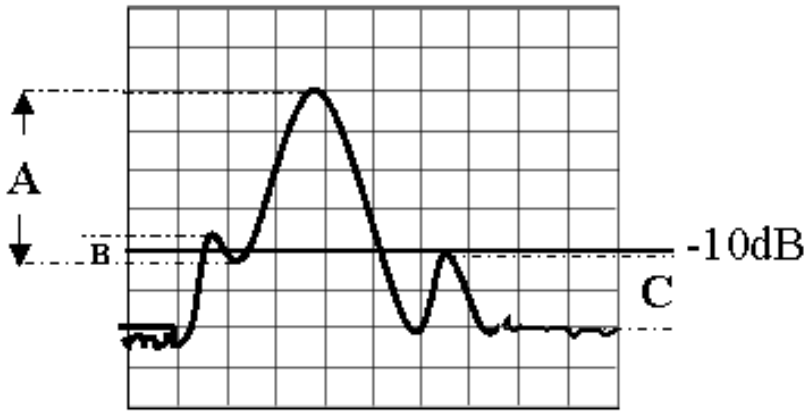
「ピーク」とは何か

本器が何を「ピーク」と見なすかを定義するために、下記の2つのピーク基準設定が使用できます。

- **しきい値** - 最小振幅(dB)。ピークが有効と見なされるには、しきい値レベルより**上**でなければなりません。両側の谷はしきい値レベルより下でもかまいません。
- **エクスカーション** - ピークと両側の谷の間の垂直距離(dB)。ピークと見なされるためには、ピークから両側にエクスカーション値だけデータ値が低下する必要があります。

例:

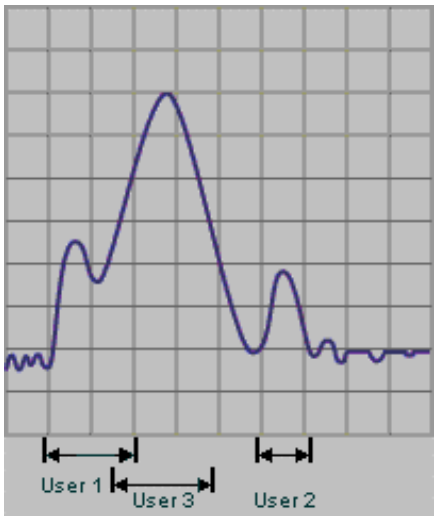
しきい値設定: **-10dB**
エクスカーション設定: **1dB**
スケール = 1 dB / Div
ピークA = 有効ピーク(しきい値およびエクスカーション設定よりも上)
ピークB = 無効ピーク(エクスカーション設定より下)
ピークC = 無効ピーク(しきい値設定より下)



検索領域

検索領域設定は、X軸(直交座標フォーマットの場合)に沿って検索されるスティミュラス値を特定のスパンに制限する場合に使います。この**ユーザ・スパン**のスタート/ストップ・スティミュラス値を設定します。スタート値がストップ値より大きい場合、マーカは移動しません。下の図は検索領域の例を示します。

- 新規マーカのデフォルト領域は「フル・スパン」です。
- 各チャネルに対して9個のユーザ定義領域があります。
- ユーザ定義領域は互いに重なっていてもかまいません。
- 定義された1つの領域を複数のマーカが使うこともできます。



検索の実行

マーカ検索を開始するには、**Execute**ボタンを押す必要があります。**Execute**ボタンはすべての検索タイプで使用する必要があります。

また、**Tracking**ボックスをチェックすると、1回の掃引が終わるたびに**Execute**ボタンを押すのと同じことになり、検索が連続的にアップデートされます。

注記: 検索は、**Tracking**ボックスがチェックされた後の最初の掃引から始まり、現在入力されている検索タイプおよび領域情報に基づいて行われます。**Tracking**がオンの場合、検索基準が適切かどうかを、データの使用前に必ずチェックしてください。

注記: **Tracking**が選択されているマーカに対しては、スティミュラス設定を手動で変更することはできません。

ボタンをクリックすると情報が表示されます。

マーカ機能 - 機器設定の変更

マーカを使って下記の機器設定を変更できます。

アクティブ・マーカは画面上に□という記号で表示されます。その他のマーカはすべて非アクティブであり、画面上では□で表示されます。

離散マーカ

- **離散マーカ**がオフの場合、マーカは指定されたスティミュラス値を持つ**補間された**データ・ポイントを返します。
- **離散マーカ**がオンの場合、マーカは指定されたスティミュラス値に最も近い**実際のデータ・ポイント**を返します。

注記: **Discrete Marker**がオンの場合、スティミュラス値は指定された値と異なる可能性があ

ります。

マーカ表示

マーカ情報の表示のしかたを変更できます。

マーカ･データのフォーマット

マーカ読み値データのデフォルトのフォーマットは、トレースのフォーマット(トレース・デフォルト)と同じです。マーカ読み値データのフォーマットは、現在の表示フォーマットと独立に、12のフォーマットの中から選択できます。これにより、必要な測定ウィンドウの数を減らして、測定セットアップを単純化できます。フォーマット・オプションは下記の通りです。

下の図はさまざまなデータ・フォーマットのマーカを示します。

マーカ･テーブル

アクティブ・トレースの**マーカ・データ一覧**を表の形で表示できます。マーカ・データは、上の図に示すように、各マーカに対して指定されたフォーマットで表示されます。

ボタンをクリックすると手順が表示されます。

マーカをオンにする

マウス手順

1. ウィンドウの**トレース・ステータス**表示領域で目的のトレースをクリックして選択します(トレース・ステータス表示の詳細については、アナライザ画面のカスタマイズを参照してください)。
2. **Marker**メニューで**Marker**をクリックします。
3. ダイアログ・ボックスの**Marker**部分でマーカ番号を選択します。
4. **On**ボックスをクリックして、選択したマーカをオンにします。
5. ダイアログ・ボックスの**Stimulus**エリアに、目的のマーカ位置を入力します。
6. **Advanced**ボタンをクリックします(ステップ6～9はオプション)。
7. **Marker Type**を選択します。: **Normal**(ノーマル)(デフォルト)または**Fixed**(固定)
8. **Discrete marker**ボックスでマーカの動作を選択します。:
   補間 (デフォルト、ボックスのチェックなし)または**離散**
9. **OK**をクリックします。

注記: マーカをオンにするには、**マーカ・ツールバー**を使う方法もあります。この場合、上の手順のステップ1、3、4、5を実行します(マーカ・ツールバーの詳細については、アナライザ画面のカスタマイズを参照してください)。

フロントパネル・キー手順

1. **DISPLAY**ブロックで**Trace**を押して目的のトレースを選択します。目的のトレースが選択されるまで**F2 (Next Trace)**を押します。
2. **TRACE SETUP**ブロックで**Marker**を押します。
3. **F1 (Marker 1)**または**F2 (Marker 2)**または**F3 (Marker 3)**を押して値を入力するか、上下の矢印を押して目的のマーカ位置を設定します。
4. これ以外のマーカを追加するには、**TRACE SETUP**ブロックでもう一度**Marker**を押します。
5. **F1 (Marker 4)**または**F2 (Marker 5)**または**F3 (Marker 6)**を押して値を入力するか、上下の矢印を押して目的のマーカ位置を設定します。
6. これ以外のマーカを追加するには、**TRACE SETUP**ブロックでもう一度**Marker**を押します。
7. **F1 (Marker 7)**または**F2 (Marker 8)**または**F3 (Marker 8)**を押して値を入力するか、上下の矢印を押して目的のマーカ位置を設定します。

マーカ・タイプ(ノーマルまたは固定)およびマーカ動作(補完または離散)を設定する手順:

1. **COMMAND**ブロックで**Menu/Dialog**ボタンを押します。**TRACE SETUP**ブロックで**Marker**を押します。
2. Markerダイアログ・ボックスで右タブを押して**Advanced**を選択します。Clickを押します。
3. 右タブを押して**Discrete marker**ボックスを選択します。Clickを押して**離散**マーカ動作を選択します(デフォルトは**補間**)。
4. 右タブを押して**Marker Type**ボックスを選択します。下矢印を押して**Fixed**(固定)を選択します(デフォルトは**Normal**(ノーマル))。
5. **COMMAND**ブロックで**OK**を押します。

マーカによるアナライザ設定の変更

マウス手順

1. ウィンドウの**トレース・ステータス**表示領域で目的のトレースをクリックして選択します(トレース・ステータス表示の詳細については、アナライザ画面のカスタマイズを参照してください)。

2. **Marker**メニューで**Select Marker**をクリックします。目的のマーカ番号をクリックします。
3. **Marker**メニューで**Marker Function**をクリックします。
4. 変更したいアナライザ機能をクリックします。

- **Start(スタート)**
- **Stop(ストップ)**
- **Center(中心)**
- **Ref Level(基準レベル)**
- **Delay(遅延)**

5. **OK**をクリックします。

注記: アナライザ設定は**マーカ・ツールバー**からも変更できます(**スタート**、**ストップ**、**中心**、**スパン**のみ)(マーカ・ツールバーの詳細については、アナライザ画面のカスタマイズを参照してください)。

フロントパネル・キー手順

1. **DISPLAY**ブロックで**Trace**を押して目的のトレースを選択します。目的のトレースが選択されるまで**F2 (Next Trace)**を押します。
2. **TRACE SETUP**ブロックで**Marker**を押します。
3. **F1 (Marker 1)**または**F2 (Marker 2)**または**F3 (Marker 3)**を押して、目的のマーカを選択します。
4. これ以外のマーカを選択するには、**TRACE SETUP**ブロックでもう一度**Marker**を押します。**F1 (Marker 4)**または**F2 (Marker 5)**または**F3 (Marker 6)**を押して、目的のマーカを選択します。
5. これ以外のマーカを選択するには、**TRACE SETUP**ブロックでもう一度**Marker**を押します。**F1 (Marker 7)**または**F2 (Marker 8)**または**F3 (Marker 9)**を押して、目的のマーカを選択します。
6. **TRACE SETUP**ブロックで**Marker Function**ボタンを押します。
7. 変更したい機能を選択します。

- **F1 (Center)(中心)**
- **F2 (Ref Level)(基準レベル)**
- **F3 (Delay)(遅延)**

他のマーカ機能(**スタート**または**ストップ**)を設定する手順:

1. **COMMAND**ブロックで**Menu/Dialog**ボタンを押します。**TRACE SETUP**ブロックで**Marker Function**を押します。
2. 右タブを押して目的のマーカ機能を選択します。
3. **COMMAND**ブロックで**OK**を押します。

マーカ・データの表示

マウス手順

1. **Marker**メニューで**Marker Table**をクリックします。
2. **Show Table**をクリックします。

注記: マーカ・テーブルとマーカ読み値は、**View**メニューからオンにすることもできます(アナライザ画面のカスタマイズ参照)。

フロントパネル・キー手順

1. **TRACE SETUP**ブロックで**Marker Table**ボタンを押します。
2. **F4 (Hide)**を押すとマーカ・テーブルが非表示になります。

マーカ・データ・フォーマットの変更

マウス手順

1. ウィンドウの**トレース・ステータス**表示領域で目的のトレースをクリックして選択します(トレース・ステータス表示の詳細については、アナライザ画面のカスタマイズを参照してください)。
2. **Marker**メニューで**Marker**をクリックし、**Advanced**をクリックします。
3. ダイアログ・ボックスの**Marker**部分で、目的のマーカ番号を選択します。
4. ダイアログ・ボックスの**Format**部分で、目的のマーカ・データ・フォーマットを選択します。
5. **OK**をクリックします。
6. **OK**をクリックします。

フロントパネル・キー手順

1. **COMMAND**ブロックで**Menu/Dialog**ボタンを押します。**TRACE SETUP**ブロックで**Marker**を押します。
2. Markerダイアログ・ボックスで右タブを押して**Advanced**ボタンを選択します。Clickボタンを押します。
3. **NAVIGATION**ブロックで下矢印を押して、目的のマーカ番号を選択します。
4. 右タブを押して**Format**ボックスを選択します。下矢印を押して目的のマーカ・フォーマットを選択します。
5. **COMMAND**ブロックで**OK**を押します。
6. **COMMAND**ブロックで**OK**を押します。

ボタンをクリックすると手順が表示されます。

データ検索

マウス手順

1. ウィンドウの**トレース・ステータス**表示領域で目的のトレースをクリックして選択します(トレース・ステータス表示の詳細については、アナライザ画面のカスタマイズを参照してください)。
2. **Marker**メニューで**Marker Search**をクリックします。
3. ダイアログ・ボックスの**Marker**部分で、目的のマーカ番号を選択します。
4. ダイアログ・ボックスの**Search Type**部分で、目的の検索タイプを選択します。
- **Next Peak(次ピーク)**、**Peak Left(左ピーク)**、**Peak Right(右ピーク)**の場合: **Threshold**(しきい値)レベルと**Excursion**(エクスカーション)レベルを入力します。
- **Target**(ターゲット)の場合、**Target**(ターゲット)値を入力します。
- **Bandwidth**(帯域幅)の場合、レベルを入力します。
5. ダイアログ・ボックスの**Search Domain**部分で、目的の検索領域を選択します。**Full Span**(フル・スパン、デフォルト)以外の検索領域を選択した場合、ダイアログ・ボックスの**User Span**エリアが表示されます。ユーザ・スパンの**Start**(スタート)値と**Stop**(ストップ)値を入力します。
6. 検索を開始する手順:
- **手動** - **Execute**ボタンをクリックします。
- **自動** - **Tracking**ボックスをクリックします(マーカでマークされたデータが変更されたときに検索が実行されます)。
7. **OK**をクリックします。

フロントパネル・キー手順

1. **DISPLAY**ブロックで**Trace**を押して目的のトレースを選択します。目的のトレースが選択されるまで**F2 (Next Trace)**を押します。
2. **F1 (Marker 1)**または**F2 (Marker 2)**または**F3 (Marker 3)**を押して、目的のマーカを選択します。
3. これ以外のマーカを選択するには、**TRACE SETUP**ブロックでもう一度**Marker**を押します。**F1 (Marker 4)**または**F2 (Marker 5)**または**F3 (Marker 6)**を押して、目的のマーカを選択します。
4. これ以外のマーカを選択するには、**TRACE SETUP**ブロックでもう一度**Marker**を押します。**F1 (Marker 7)**または**F2 (Marker 8)**または**F3 (Marker 9)**を押して、目的のマーカを選択します。
5. **TRACE SETUP**ブロックで**Marker Search**を押します。
6. 目的の検索タイプを選択します。

- **F1 (Max)(最大値)**
- **F2 (Min)(最小値)**
- **F3 (Left Peak)(左ピーク)**
- **F4 (Right Peak)(右ピーク)**

その他の**マーカ検索**パラメータを設定する手順:

1. **COMMAND**ブロックで**Menu/Dialog**ボタンを押します。**TRACE SETUP**ブロックで**Marker Search**を押します。
2. 下矢印を押して、目的のマーカ番号を選択します。
3. Markerダイアログ・ボックスで右タブを押して**Search Type**エリアを選択します。下矢印を押して目的の検索タイプを選択します。右タブを押して、選択した検索タイプのエントリ・ボックスに移動します。

- **Next Peak(次ピーク)**、**Peak Left(左ピーク)**、**Peak Right(右ピーク)**の場合: **Threshold**(しきい値)レベルと**Excursion**(エクスカーション)レベルを入力します。
- **Target**(ターゲット)の場合、**Target**(ターゲット)値を入力します。
- **Bandwidth**(帯域幅)の場合、**Threshold**(しきい値)レベルを入力します。

4. Markerダイアログ・ボックスで右タブを押して、**Search Domain**エリアを選択します。下矢印を押して、目的の検索領域を選択します。
5. **Full Span**(フル・スパン、デフォルト)以外の検索領域を選択した場合、ダイアログ・ボックスの**User Span**エリアが表示されます。右タブを押して**User Span**エリアに移動し、**Start**(スタート)値と**Stop**(ストップ)値を入力します。
6. 検索を開始するには、右タブを押して下記のどちらかに移動し、Clickボタンを押して選択します。

- **手動** - **Execute**ボタンをクリックします。
- **自動** - **Tracking**ボックスをクリックします(マーカでマークされたデータが変更されたときに検索が実行されます)。

7. **COMMAND**ブロックで**OK**を押します。

基準マーカによる相対測定の実行

マウス手順

1. ウィンドウの**トレース・ステータス**表示領域で目的のトレースをクリックして選択します(トレース・ステータス表示の詳細については、アナライザ画面のカスタマイズを参照してください)。
2. **Marker**メニューで**Marker**をクリックします。
3. ダイアログ・ボックスの**Marker**部分で、**Reference**を選択します。
4. **On**ボックスをクリックして基準マーカをオンにします。
5. ダイアログ・ボックスの**Stimulus**エリアに、目的の基準マーカ位置を入力します。
6. ダイアログ・ボックスの**Marker**部分で、基準マーカに対する相対測定に使うマーカの番号を選択します。

7. **On**ボックスをクリックしてマーカをオンにします。
8. **Delta marker**ボックスをクリックして、基準マーカに対する相対測定にこのマーカを使用できるようにします。

基準マーカとデルタ・マーカのマーカ動作(補間または離散)を設定する手順:

1. **Marker**ダイアログ・ボックスで**Advanced**ボタンをクリックします。
2. ダイアログ・ボックスの**Marker**部分で、**Reference**を選択します。
3. **Discrete marker**ボックスでマーカの動作を選択します。
- **補間** (デフォルト、ボックスのチェックなし)
- **離散**
4. ダイアログ・ボックスの**Marker**部分で、デルタ・マーカのマーカ番号を選択します。
5. ステップ3を繰り返します。
6. **OK**をクリックします。
7. **OK**をクリックします。

注記: 相対測定用のマーカをオンにするには、**マーカ・ツールバー**を使う方法もあります(マーカ・ツールバーの詳細については、アナライザ画面のカスタマイズを参照してください)。

フロントパネル・キー手順

1. **DISPLAY**ブロックで**Trace**を押して目的のトレースを選択します。目的のトレースが選択されるまで**F2 (Next Trace)**を押します。
2. **TRACE SETUP**ブロックで**Marker**を押します。
3. **F1 (Marker 1)**または**F2 (Marker 2)**または**F3 (Marker 3)**を押して値を入力するか、上下の矢印を押して目的のマーカ位置を設定します。
4. これ以外のマーカを追加するには、**TRACE SETUP**ブロックでもう一度**Marker**を押します。
5. **F1 (Marker 4)**または**F2 (Marker 5)**または**F3 (Marker 6)**を押して値を入力するか、上下の矢印を押して目的のマーカ位置を設定します。
6. これ以外のマーカを追加するには、**TRACE SETUP**ブロックでもう一度**Marker**を押します。
7. **F1 (Marker 7)**または**F2 (Marker 8)**または**F3 (Marker 8)**を押して値を入力するか、上下の矢印ボタンを押して目的のマーカ位置を設定します。

マーカ動作(補間または離散)を設定する手順:

1. **COMMAND**ブロックで**Menu/Dialog**ボタンを押します。**TRACE SETUP**ブロックで**Marker**を押します。
2. Markerダイアログ・ボックスで右タブを押して**Advanced**ボタンを選択します。Clickボタンを押します。
3. 上矢印を押して、**Reference**(基準)マーカを選択します。
4. 右タブを押して**Discrete marker**ボックスを選択します。Clickを押して**離散**マーカ動作を選択します(デフォルトは**補間**)。
5. 右タブを押してもう一度**Marker**ボックスを選択します。下矢印を押してデルタ・マーカのマーカ番号を選択します。
6. 右タブを押して**Discrete marker**ボックスを選択します。Clickを押して**離散**マーカ動作を選択します(デフォルトは**補間**)。
7. **COMMAND**ブロックで**OK**を押します。

リミットを使用したデバイスのテスト

概要　　概念　　リミット・テストの方法

リミットは、デバイスをテストするために定義する性能パラメータです。 リミット・ラインとリミット・テストによって、 測定したデータをあらかじめ定義した条件と比較することができます。

リミット

リミットは以下のように定義されます。

- セグメントごとに設定します。**各セグメントは、スティミュラス・スパンの一部となります。**
- **最大** または**最小** リミットとして設定するか、あるいは**OFF**に設定します。
- 各リミット・セグメントに対して、**開始 および終了スティミュラス値**を 指定します。
- 各リミット・セグメントに対して、**開始 および終了レスポンス値**を 指定します。
- 最大16の測定に対して独立に設定できます。**各測定のセグメント数は最大100です**。

リミット・テーブル

リミットの定義、編集、アクティブ化、および表示は、リミット・テーブルを使って行います。

	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

リミット・ライン

リミット・ラインは、チャネルに対して指定したリミットをアナライザの画面に 目に見える形として表現したものです。リミット・ラインは下記に 使用できます。

- デバイスをチューニングするときに、オペレータに**目印**を提供する。
- デバイス仕様に適合するための**標準 基準**を提供する。
- データ対仕様の**比較**を表示する。

下記のタイプのリミット・ラインを作成できます。

- フラット
- 斜線
- 点

リミット・ラインは、画面で**表示**または **非表示**にすることができます。リミット・ラインを 表示すると、

- **測定 トレース**が**緑**色で表示されます (測定トレースには、一般的に緑色を使用します)。
- **リミット・ライン** が**橙**色で表示されます。

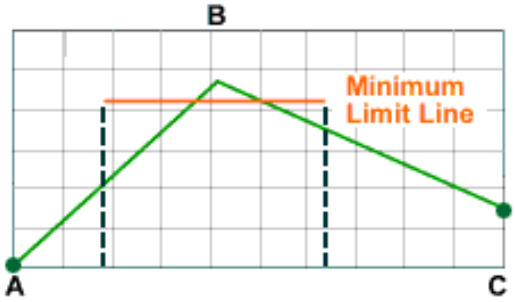
注記:指定したリミットは、 リミット・ラインが表示されていない場合でも有効です。

リミット・テスト

リミット・テストは 測定データを 定義したリミットと比較し、各測定データ・ポイントの**パス またはフェール**情報を提供します。

リミットは、実際の測定データ・ポイントでのみ チェックされます。ポイント密度が不十分

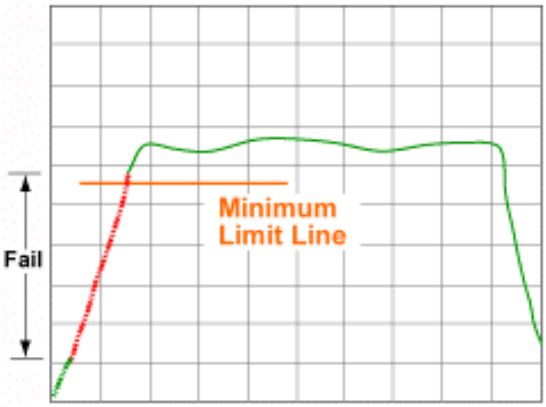
であると、リミット・テストでフェールが表示されない場合でも、 デバイスが仕様の範囲外にある可能性があります。 下の図は、最小ポイントを使った測定を示します。



- リミット・ラインの始まりは、 横周波数軸に沿ったポイントAとポイントBとの**間**にあります。
- リミット・ラインの終わりは、周波数軸に沿ったポイントBとポイントCとの**間**に あります。

リミット・ラインの始まりと終わりとの間には、**1個の測定ポイントしかありません**。 この特殊な例では、リミット・テストの結果が予測に反して、フェールでなく **パス**になります。 これは、テストされた唯一のポイント(ポイント B)がリミット・テストに合格したためです。ポイントAおよびCはフェールになりますが、これらのポイントは表示されたリミットによってテストされません。

以下では、同じ測定を51個のポイントを使って行っています。 測定がリミット・テストに「フェール」になっていることがわかります。



この測定を評価するには、より多くの測定ポイントが必要でした。

上の図に示すように、リミット・テストをオンにすると、

- リミット・テストに **フェール**になった測定トレースの部分は、 **赤色で表示されます**。
- **フェールにならなかった**測定トレースの部分は、**緑色** (変更なし)のままです。

注記: リミット値によって囲まれていない測定トレース部分は、 テストされていませんが、緑色のままです。

リミット・ラインの表示とリミット・テストの使用は、**同時 または別々に**行うことができます。

リミット・ラインとリミット・テストは、 **スミス・チャートまたは極座標フォーマットでは使用できません**。リミット・ラインがオンのときにスミス・チャートまたは極座標フォーマット に変更すると、リミット・ラインとリミット・テストが 自動的にディスエーブルになります。

注記: リミット・テストは、 測定データ・トレースでのみ実行されます。リミット・テストを**メモリ・トレースで実行することはできません**。

ボタンをクリックすると手順が表示されます。

リミット・セグメントの定義

マウス手順

1. **Trace** メニューで、**Limit Test...**、**Show Table**、**OK**をクリックします。
2. リミット・テーブルの**Type** エリアで、Limit Segment 1に対して**MIN** または**MAX**を選択します。
3. Limit Segment 1の**BEGIN STIMULUS** エリアをクリックします。希望の値を入力します。
4. Limit Segment 1の**END STIMULUS** エリアをクリックします。希望の値を入力します。
5. Limit Segment 1の**BEGIN RESPONSE** エリアをクリックします。希望の値を入力します。
6. Limit Segment 1の**END STIMULUS** エリアをクリックします。希望の値を入力します。
7. 希望の各リミット・セグメントに対してステップ2～6を繰り返します。

フロントパネル・キー手順

1. **TRACE SETUP**ブロックで、**Limit Table**を押します。
2. **F3 (Show)**を押して、 リミット・テーブルを表示します。
3. **NAVIGATION** ブロックで、**Click**を押して、 Limit Segment 1のType選択肢を表示します。Downボタンを押して、 **MIN**または**MAX**を選択します。
4. 右タブを押して、Limit Segment 1の**BEGIN STIMULUS**エリアを表示します。**ENTRY** キーを押して、希望の値を入力します。**Enter**を押します。
5. 右タブを押して、Limit Segment 1の**END STIMULUS**エリアを表示します。**ENTRY** キーを押して、希望の値を入力します。**Enter**を押します。
6. 右タブを押して、Limit Segment 1の**BEGIN RESPONSE**エリアを表示します。**ENTRY** キーを押して、希望の値を入力します。**Enter**を押します。
7. 右タブを押して、Limit Segment 1の**END RESPONSE**エリアを表示します。**ENTRY** キーを押して、希望の値を入力します。**Enter**を押します。
8. 左タブと下矢印を押して、次のリミット・セグメントを 表示します。
9. 希望の各リミット・セグメントに対してステップ3～8を繰り返します。

リミット・テーブルの表示/非表示

マウス手順

1. **Trace** メニューで、**Limit Test**をクリックします。
2. **Show Table**または**Hide Table**をクリックします。
3. **OK**をクリックします。

以下に、マウスを使ったリミット・テーブルの表示/非表示の別の方法を 示します。

1. **View** メニューで、**Tables**をクリックします。
2. **Limit Table**をクリックして、その表示ステートを変更します。

フロントパネル・キー手順

1. **TRACE SETUP**ブロックで、**Limit Table**を押します。
2. **F3 (Show)** または**F4 (Hide)**を押します。

リミット・ラインのオン

マウス手順

1. **Trace** メニューで、**Limit Test**をクリックします。
2. **Limit Line ON**をクリックします。
3. **OK**をクリックします。

フロントパネル・キー手順

1. **TRACE SETUP**ブロックで、**Limit Table**を押します。
2. **F2 (Line ON/Off)**を押して、リミット・ラインを表示します。

リミット・テストのオン

マウス手順

1. **Trace** メニューで、**Limit Test**をクリックします。
2. **Limit Test ON**をクリックします。
3. リミット・テストのフェールを、サウンドによって通知してほしい場合は、**Sound ON Fail**をクリックします 。
4. **OK**をクリックします。

フロントパネル・キー手順

1. **TRACE SETUP**ブロックで、**Limit Table**を押します。
2. **F1 (Test ON/Off)**を押して、リミット・テストをオンにします。

演算機能

概要　　概念　　演算機能の実行方法

アクティブ・データ・トレースとメモリ中のトレースとを使って、4種類の演算を実行できます。また、アクティブ・データ・トレースの統計値として、平均値、標準偏差、p-p値を表示することができます。

トレース演算

表示されたデータ・トレースに対して**4種類の演算**を実行できます。

- データ / メモリ
- データ - メモリ
- データ * メモリ
- データ + メモリ

演算機能は、表示向けにフォーマットされる前の**複素データに対して実行**されます。

演算機能を実行するには、被試験デバイスの測定トレース(または相当するもの)をアクティブ・チャネル・メモリにあらかじめ記憶しておく必要があります。Trace Viewオプションを使うことにより、現在のデータ・トレースとともに**メモリ・トレースを表示**できます。

注記: SCPIまたはCOMコマンドを使って、メモリに記憶される「トレース」を定義することもできます。このためには、データ・トレースのデータ・ポイントと同じ数の複素データ・ペアをコマンドで入力します。データ・トレースは、メモリにデータをロードする前に構成されている必要があります。入力された複素データ・ペアは、実際の測定ポイントのスティミュラス値に対応するようにマッピングされます。

データ / メモリ

データ / メモリは、トレース演算の**最も一般的な使用法**です。この機能は、利得や減衰の測定のように、2つのトレースの比を求める場合に使用します。

1. **この機能を使用するには、被試験デバイスの測定トレースをアクティブ・チャネル・メモリにあらかじめ記憶しておく必要があります。**
2. その後、**Data / Memory**を実行することにより、データをメモリの内容で除算します。
3. データがメモリの内容で正規化され、結果が表示されます。
4. データを対数振幅フォーマットで表示すれば、記憶されたトレースからの偏移を測定することができます。

データ - メモリ

データ - メモリは、記憶されたトレースを使って、複素測定データに対するベクトル減算を実行します。例えば、ベクトル誤差の測定結果を記憶しておいて、デバイス測定結果からその誤差を除去することができます。

データ * メモリとデータ + メモリ

データ * メモリとデータ + メモリは、測定データと記憶データに使用するよりも、独自の式やアルゴリズムをデータ・トレースに対して適用するために使うのが普通です。例えば、デバイスの前に直列に接続された理論的な増幅器の利得をシミュレートする場合､次のようにします。

1. **増幅器の周波数応答を記述するアルゴリズムを作成します。**
2. データ・トレースのデータ・ポイントと同じ数の複素データ・ペアを入力します。
3. SCPIまたはCOMコマンドを使って、データ・ペアをメモリにロードします。

4. ロードした複素データ・ペアは、実際の測定ポイントのスティミュラス値に対応するようにマッピングされます。
5. 次に、**データ + メモリ**または**データ * メモリ**機能を使って、周波数応答と測定データとを加算または乗算します。詳細については、COMコマンド、PutComplexメソッドを参照してください。

注記: データ・トレースは、メモリにデータをロードする前に構成されている必要があります。

トレース統計

アクティブ・データ・トレースの統計値を計算して表示できます。下記の統計値があります。

- 平均値
- 標準偏差
- p-p値

統計値は、スティミュラス・スパン全体に対しても、その一部に対しても計算できます。1チャネルあたり9つのユーザ・レンジが指定できます。これらのユーザ・レンジは、そのチャネルに対してマーカ検索で指定する検索領域と同じものです。これらは同じメモリ・レジスタを使用するため、用いられるスティミュラス・スパンは同一です。あるチャネルに対してマーカ検索の検索領域を指定してある場合、対応するユーザ・レンジを選択することで同じスパンを呼び出すことができます。チャネルのユーザ・レンジは互いに重なっていてもかまいません。

トレース統計の便利な使用法の1つとして、通過帯域のリップルのp-p値を求めることができます。この場合、最小値と最大値を個別に求める必要はありません。

トレース統計の計算方法は、データ表示に用いられているフォーマットに基づいて決まります。

- 直交座標データ・フォーマットの場合、表示されているスカラ・データから計算されます。
- 極座標フォーマットとスミス・チャート・フォーマットの場合、対数振幅フォーマットで表示されるデータから計算されます。

ボタンをクリックすると手順が表示されます。

トレース演算の選択

マウス手順

1. **Trace**メニューで**Math/Memory**をクリックします。
2. **Data>Memory**をクリックして、目的のトレースをメモリに記憶します。
3. ダイアログ・ボックスの**Data Math**部分で、実行する演算機能を選択します。
4. ダイアログ・ボックスの**Trace View Options**部分で、トレース・ビューを選択します。
5. **OK**をクリックします。

フロントパネル・キー手順

1. **TRACE SETUP**ブロックで**Math/Memory**を押します。
2. **F1 (Data>>Mem)**を押して、目的のトレースをメモリに記憶します。
3. **F2 (Data/Mem)**を押してデータ/メモリを選択します。
4. **F3 (Data Only)**を押して、データ・トレースのみを表示します。
5. **F4 (DataMem)**を押して、データ・トレースとメモリ・トレースの両方を表示します。

データ/メモリ以外の演算機能を選択するには、下記の手順を実行します。

1. **COMMAND**ブロックで**Menu/Dialog**を押します。
2. **TRACE SETUP**ブロックで**Math/Memory**を押します。
3. Clickを押して、目的のトレースをメモリに記憶します。
4. 右タブを押してダイアログ・ボックスの**Data Math**エリアを選択します。上下の矢印を押して、実行するトレース演算を選択します。
5. 右タブを押して、ダイアログ・ボックスのTrace View Optionsエリアを選択します。上下の矢印を押して、トレース・ビューを選択します。
6. **COMMAND**ブロックで**OK**を押します。

トレース統計の設定

マウス手順

1. **Trace**メニューで**Trace Statistics**をクリックします。
2. **Statistics**ボックスをクリックしてトレース統計をオンにします。
3. 使用するユーザ・レンジを選択します。
4. **Start**ボックスに、ユーザ・レンジを開始するスティミュラス値を入力します。
5. **Stop**ボックスに、ユーザ・レンジを終了するスティミュラス値を入力します。
6. **OK**をクリックします。

フロントパネル・キー手順

1. **COMMAND**ブロックで**Menu/Dialog**を押します。右タブを使って**Trace**メニューを選択します。下矢印を使って**Trace Statistics**を選択します。Clickを押します。
2. Clickを押してトレース統計をオンにします
3. 右タブを押してダイアログ・ボックスのUser Range選択エリアに移動します。下矢印を押して、使用するユーザ・レンジを選択します。
4. 右タブを押して**Start**ボックスに移動します。ユーザ・レンジを開始するスティミュラス値を入力します。

5. 右タブを押して**Stop**ボックスに移動します。ユーザ・レンジを終了するスティミュラス値を入力します。
6. **COMMAND**ブロックで**OK**を押します。

ステップ5. データの出力

- ドライブ・マッピング
- ファイルの保存とリコール
- 表示された測定結果のプリント

ドライブ・マッピング

概要　　概念　　ドライブをマッピングする方法

ドライブ・マッピングを使うと、本器と外部コンピュータとの間でディスク・ドライブを共有できます。

ドライブ・マッピングを使うと、本器と外部コンピュータとの間でディスク・ドライブを共有し、ファイルの保存や呼出しを行うことができます。

- 本器のディスク・ドライブが、PCのディレクトリから別のドライブのように見えます。
- コンピュータのディスク・ドライブが、本器のディレクトリから別のドライブのように見えます。
- これにより、本器のディスク・ドライブとコンピュータのディスク・ドライブが、ファイルを扱う際のソースの選択肢となります。

注記: 以下の手順には、マウスとキーボードが必要です。

注記: 以下の手順では、外部コンピュータがWindows NT 4.0以降で動作する必要があります。

本器から外部コンピュータへのドライブのマッピング

コンピュータ名の確認

1. 以下の手順を実行するには、本器のコンピュータ名を知る必要があります。本器のコンピュータ名を表示、変更するには、下記の2つのステップを実行します。
   a. **System、 Windows Taskbar、Settings、Control Panelをクリックし、System アイコンをクリックします。**
   b. Network Identification**タブを選択します。必要ならコンピュータ名を記録しておきます。コンピュータ名を変更するには、**Properties**をクリックし、新しいコンピュータ名を入力します。**OK**をクリックします。**

新規ユーザの追加

2. **Users and Passwords**をクリックします。
3. **Add**ボタンが淡色表示になっている場合、**"Users must enter a user name and password to use this computer"**のチェックボックスをクリックします。**Add**をクリックします。
4. **User Name**ボックスに、ユーザ名(外部コンピュータで使用するのと同じもの)を入力します。**Full Name**ボックスにフルネームを入力します。**Next**をクリックします。
5. **Password**ボックスにパスワード(外部コンピュータで使用するのと同じもの)を入力します。**Confirm Password**にもう一度パスワードを入力します。**Next**をクリックします。
6. "What level of access do you want to grant this user?"というプロンプトに対して、**Other**を選択し、デフォルトの選択肢**Administrators**を選択します。**Finish**をクリックします。

ドライブの共有名の作成

7. Addressボックスで下矢印をクリックし、**My Computer**をクリックします。外部コンピュータと共有するドライブを右クリックします。**Sharing**をクリックします。
8. **Share this folder**を選択します。**Share name**ボックスに共有名を入力します。**OK**を押します。

共有ドライブへのマッピング

9. 外部コンピュータのデスクトップで、**Windows NT Explorer**をクリックします。**Tools**メニューで**Map Network Drive**をクリックします。
10. **Path**ボックスに「¥¥コンピュータ名¥共有名」と入力します。

11. **Connect**ボックスにログイン名を入力します。**OK**をクリックします。

外部コンピュータから本器へのドライブのマッピング

1. 外部コンピュータのデスクトップで、**Windows NT Explorer**をクリックします。ドライブのリストで、共有するドライブを右クリックします。**Sharing**をクリックします。
2. ダイアログ・ボックスで**Shared As**を選択します。**Share Name**ボックスで、矢印キーを使うか、ドライブの共有名を入力します(任意の名前が使用できます)。**OK**をクリックします。
3. 本器のデスクトップで**Windows Explorer**をクリックします。**Tools**メニューで**Map Network Drive**をクリックします。
4. **Folder**ボックスに「¥¥PCホスト名(またはID番号)¥共有名(ステップ2で選択した名前)」と入力します。**Finish**をクリックします。
5. **Connect as**ボックスにログイン名を入力します。
6. **Password**ボックスに、外部コンピュータで使用するログオン・パスワードを入力します。**OK**をクリックします。

ファイルのセーブ/リコール

概要　　概念　　セーブ/リコール方法

本器では、内部ハード・ディスク、フロッピー・ディスク、外部CD-RWドライブ、PC、サーバなどに、さまざまなファイル・フォーマットでデータを保存し、リコールすることができます。

ボタンをクリックすると下記の事項について知ることができます。

ファイルの保存

セーブ機能を使うと、バイナリ・ファイルのデータは内部ハード・ディスクの自動的に選択されたパスとファイル名に書き込まれます。以前にパスが定義されていない場合、デフォルトのC:¥Program Files¥Agilent¥Network Analyzer¥Documentsが自動的に選択され、ファイル名としては"default.cst"が用いられます。下記のどれかの方法でファイルの名前を変更して保存しておかない限り、次にセーブ機能を使ったときにこのファイルは上書きされます。

- オートセーブ
- 名前を付けて保存
- リネーム(WindowsのExplorerを使用)

オートセーブ

オートセーブ機能を使うと、バイナリ・ファイルのデータは内部ハード・ディスクの自動的に選択されたパスに保存されます。以前にパスが定義されていない場合、デフォルトのC:¥Program Files¥Agilent¥Network Analyzer¥Documentsが自動的に選択されます。さらに、新しい.cstファイル名が自動的に選択されて表示されます。この機能を使うと、前にセーブしたファイルを上書きすることはありません。

名前を付けて保存

この機能を使うと、以下の任意の場所を指定して、バイナリまたはASCIIのファイルにデータを保存できます。

- 内部ハード・ディスク
- フロッピー・ディスク
- 外部CD-RWドライブ、PC、サーバ(マッピングされたネットワーク・ドライブを使用)

バイナリ・ファイル・タイプ

以下のバイナリ・ファイル・タイプは、測定に関連するデータをセーブ・リコールするためだけに用いられます。これらのファイル・タイプはAgilent PNAシリーズ・アナライザ専用です。

- .cst
- .sta
- .cal

.cstファイルには下記のデータが含まれます。

- 本器で設定されているすべての測定(リミット・ラインとマーカを含む)
- 本器で設定されているすべての校正セット
- Windowsの表示配置

.cstファイルをリコールすることにより、機器ステートと校正データ・ファイルの両方を測定のたびにリコールする時間が節約できます。

.staファイルには下記のデータが含まれます。

- 本器で設定されているすべての測定(リミット・ラインとマーカを含む)
- Windowsの表示配置

注記: .staファイルには校正データ・セットは含まれません。

.staファイルをリコールすることにより、機器ステートを測定のたびにリコールする時間が節約できます。

.calファイルには、校正実行時に作成される測定補正データだけが含まれます。.calファイルには機器ステート・データは含まれません。測定補正データは校正実行時の機器ステートに結びついています。したがって、測定確度を最高にするには、校正実行時に用いられた機器ステート・データ(.staファイル)を.calファイルとともにリコールする必要があります。

注記: 校正プロパティ機能を使うことにより、.calファイルが保存されたときの機器設定を見ることができます。詳細については、「測定校正の選択と実行」を参照してください。

測定校正は各チャネルに固有です。例えば、チャネル1で実行した校正を他のチャネルに対してリコールすることはできません。

ASCIIファイル・タイプ

以下のASCIIファイル・タイプは、本器の内部ハード・ディスクまたはフロッピー・ディスクに保存されます。これらのファイルを外部CD-RWドライブ、PC、サーバに出力することにより、表計算やCAEなどのプログラムで利用することができます。

- .prn
- .s1p
- .s2p

注記: ASCIIファイル・タイプでは、1つのアクティブ・チャネルの1つのアクティブ・トレースだけが保存されます。1つのASCIIファイルに複数の測定を保存することはできません。

.prnフォーマットは主にMicrosoft(R) Excelなどの表計算ソフトウェアで用いられます。.prnファイルには、現在のアクティブ・トレースの補正済みスティミュラス/レスポンス・データがカンマ区切りのリストの形で記録されます。下に例を挙げます。

.prnファイルを表計算ソフトウェアにインポートすることの利点は、.s1pファイルや.s2pファイルの実数・虚数ペアでなく、.prnのフォーマット(対数振幅、リニア振幅、SWRなど)が利用できることです。また、Excelなどの一部の表計算ソフトウェアでは、.prnファイルのデータを表のカラムごとに入力することができます。.prnファイルは出力のみで、本器に読み込むことはできません。

.s1pフォーマットと.s2pフォーマットはTouchstoneフォーマットとも呼ばれ、Agilentのマイクロウェーブ・デザイン・システム(MDS)やアドバンスド・デザイン・システム(ADS)などで用いられます(.s1pファイルと.s2pファイルは表計算ソフトウェアにインポートすることもできますが、.prnファイルの方が適しています)。.s1pファイルと.s2pファイルには、現在のアクティブ・トレースの誤差補正済みスティミュラスおよび実数/虚数ペアのデータが、スペース区切り形式のリストで記録されます。データはフォーマット配列を表し、電気遅延とポート延長の効果を含みます。これらのファイルは出力のみで、本器に読み込むことはできません。.s1pファイルと.s2pファイルのデータでは画面キャプチャ・イメージを表示することはできません。

.s1pファイルには1ポート・コンポーネントの測定データが記録されます。このフォーマットは、現在のアクティブ・トレースの1個のSパラメータの補正済みデータを実数・虚数ペアとして返します。下に例を挙げます。

注記: 上と下の例で:

- Hz = 周波数
- S = Sパラメータ
- RI = 実数・虚数ペア
- R 50.0 = 基準インピーダンス

.s2pファイルには2ポート・コンポーネントの測定データが記録されます。このフォーマットは、現在のアクティブ・トレースの4個のSパラメータの補正済みデータを実数・虚数ペアとして返します。下に例を挙げます。

有効な.s2pファイルを作成するには、本器で2ポート誤差補正を使用している必要があります。2ポート誤差補正が使用されていない場合、S_{11}、S_{21}、S_{12}、S_{22}のどれかの項が得られず、測定されないパラメータのフィールドには0が返されます。ただし、1ポート測定と2ポート測定のどちらが行われたかに関わらず、.s1pファイルと.s2pファイルのどちらも作成できます。

ファイルへのプリント

ファイルへのプリント機能を使うと、画面キャプチャ・イメージの.bmp(ビットマップ)ファイルが下記のユーザ指定の場所に作成されます。

- 内部ハード・ディスク
- フロッピー・ディスク
- 外部CD-RWドライブ、PC、サーバ(マッピングされたネットワーク・ドライブを使用)

.bmpファイルは自動的に現在のパスに保存されます。パスが以前に定義されていない場合、デフォルトのC:¥Program Files¥Agilent¥Network Analyzer¥が自動的に選択されます。

.prnファイルと同様、.bmpファイルはMicrosoft Excel、Word、Paintなどのソフトウェア・アプリケーションにインポートして、画面キャプチャ・イメージを表示するために使用できます。

ファイル名の編集

ファイル名の編集機能を使うと、ファイルをコピーして、ナビゲーション・キーから新しい名前を入力することができます。元のファイルとその名前は変更されません。この機能は**名前を付けて保存**または**ファイルへのプリント**機能を使用する際に、Save Asダイアログ・ボックスから利用できます。下の図を参照してください。

注記: ファイル/フォルダ管理(コピー、削除、リネーム、検索など)の詳細については、Windows 2000のヘルプを参照してください。

ファイルのリコール

本器のリコール機能では、バイナリ・ファイルのリコールのみが可能です。.cstファイルまたは.staファイルを画面キャプチャ・イメージとしてリコールすることはできますが、データはトレース・ホールド機能でセーブされている必要があります(リコールされたファイルはトリガされるまで掃引されません)。バイナリ・ファイル・タイプの詳細については、このトピックの名前を付けて保存の部分を参照してください。

外部PCのリコール機能を使う場合、本器で出力したASCIIファイルまたはビットマップ・ファイルだけがリコールできます。画面キャプチャ・イメージをリコールするには、.prnファイルまたは.bmpファイルをMicrosoft Excel、Word、Paintなどのソフトウェア・アプリケーションでリコールします。ASCIIファイル・タイプの詳細については、このトピックの「名前を付けて保存」の部分を参照してください。

ボタンをクリックすると手順を見ることができます。

注記: 以下の手順には、マウスとキーボードが必要です。

ファイルの保存

マウス手順

1. FileメニューでSaveをクリックします。

フロントパネル・キー手順

1. **UTILITY**ブロックで**Save**を押します。
2. アクティブ・エントリ・ツールバーで**F1 (Save)**を押します。

ファイルを保存するパスとファイル名を示すメッセージが画面にしばらく表示されます。

注記: このファイルは、次にセーブ機能を使ったときに上書きされます。これを防ぐには、オートセーブ、名前を付けて保存、リネーム(WindowsのExplorerを使用)のどれかの方法でファイルの名前を変更して保存します。

ファイルのオートセーブ

注記: この手順にはフロントパネルのキーを使います。

フロントパネル・キー手順

1. **UTILITY**ブロックで**Save**を押します。
2. アクティブ・エントリ・ツールバーで**F3 (Auto Save)**を押します。

ファイルを保存するパスを示すメッセージが画面にしばらく表示されます。新しい.cstファイル名が自動的に選択されて表示されます。これは以前に保存されたファイルを上書きするのを避けるためです。

別の名前または場所へのファイルの保存

マウス手順

1. **File**メニューで**Save As**をクリックします。
2. ダイアログ・ボックスの**Save in**ボックスで、矢印ボタンを押してファイルのパスを選択します。
3. **File name**ボックスに新しいファイル名を入力します。
4. **Save as type**ボックスで矢印ボタンをクリックしてファイルのタイプを選択します。
5. **Save**をクリックします。

フロントパネル・キー手順

1. **UTILITY**ブロックで**Save**を押します。
2. アクティブ・エントリ・ツールバーで**F2 (Save As)**を押します。
3. 左タブを押して**Save in**ボックスに移動します。下矢印を押してファイルを保存するパスに移動します。右タブを押してパスを選択します。
4. **File name**ボックスに新しいファイル名を入力します。
5. 右タブを押して**Save as** typeボックスに移動します。下矢印を押して目的のファイル・タイプに移動します。右タブを押してファイル・タイプを選択します。
6. 右タブを押して**Save**ボックスに移動します。**Click**を押します。

ファイルへのプリント

マウス手順

1. **File**メニューで**Print to File**をクリックします。
2. ダイアログ・ボックスの**Save in**ボックスで矢印ボタンを押して、ファイルを保存するパスを選択します。
3. **File name**ボックスに新しいファイル名を入力します。
4. **Save**をクリックします。

フロントパネル・キー手順

1. **COMMAND**ブロックで、**Menu/Dialog**を押します。
2. **NAVIGATION**ブロックで下矢印を押して**Save As**または**Print to File**に移動します。**Click**を押します。
3. **NAVIGATION**ブロックで左タブを押して**Save in**ボックスに移動します。下矢印を押してファイルを保存するパスに移動します。右タブを押します。
4. **File name**ボックスに新しいファイル名を入力します。
5. 右タブを押して**Save**ボックスに移動します。**Click**を押します。

ファイル名の編集

マウス手順

1. **File**メニューで**Save As**または**Print to File**をクリックします。
2. ダイアログ・ボックスの**Save**ボックスで矢印ボタンをクリックして、ファイルを置くパスを選択します。
3. **Filepath**ボックスでファイル名を選択します。
4. **Edit Filename**をクリックします。
5. ダイアログ・ボックスで、下記のどれかの方法で新しいファイル名を入力します。
   - 外部キーボードを使用します。
   - 文字またはボタン(<<、Select、Back、>>)をクリックします。
6. **OK**、**Save**をクリックします。

フロントパネル・キー手順

1. **COMMAND**ブロックで、**Menu/Dialog**を押します。
2. **NAVIGATION**ブロックで下矢印を押して**Print to File**に移動します。**Click**を押します。
3. 左タブを押して**Save in**ボックスに移動します。下矢印ボタンを押してファイルを置くパスに移動します。右タブを押します。
4. 左タブ・ボタンを押して**Filepath**ボックスに移動します。上下の矢印を使ってファイル名に移動します。**Click**を押します。
5. 右タブを押して**Edit Filename**ボックスに移動します。**Click**を押します。
6. ダイアログ・ボックスで、以下のどれかの方法で新しいファイル名を入力します。
   - 外部キーボードを使用します。
   - 上下の矢印ボタンを押して目的の文字に移動し、**Click**を押します。右タブまたは左タブを押して目的のボタン(<<、Select、Back、>>)に移動し、**Click**を押します。
7. 右タブを押して**OK**に移動し、**Click**を押します。
8. 右タブを押して**Save**に移動し、**Click**を押します。

アナライザへのファイルのリコール

マウス手順

1. 外部データ記憶装置にネットワーク・ドライブをマッピングしていない場合には、マッピングします(ドライブ・マッピング参照)。これにより、外部データ記憶装置のハード・ディスク・ドライブが本器のディレクトリとして見えるようになります。
2. **File**メニューで**Recall**をクリックします。
3. ダイアログ・ボックスの**Look in**ボックスで矢印ボタンをクリックして、リコールするファイルのパスを選択します。
4. **Files of type**ボックスで矢印ボタンをクリックして、ファイル・タイプを選択します。
5. **Filepath**ボックスで、リコールするファイルの名前を選択します。
6. **Recall**をクリックします。

フロントパネル・キー手順

1. 外部データ記憶装置にネットワーク・ドライブをマッピングしていない場合には、マッピングします(ドライブ・マッピング参照)。これにより、外部データ記憶装置のハード・ディスク・ドライブが本器のディレクトリとして見えるようになります。
2. **UTILITY**ブロックで**Recall**を押します。
3. アクティブ・エントリ・ツールバーで**F1**、**F2**、**F3**のどれかを押して、表示されたファイル名の1つを選択します(ファイル名はアルファベット順に表示されます)。別のファイルをリコールするには、**F4 (Open)**を押します。
4. 前のステップで**F4 (Open)**を選択した場合:
   a. 左タブを押して**Look in**ボックスに移動します。下矢印を押してリコールするファイルのパスに移動します。右タブを押します。
   b. 右タブを押します。**Files of type**ボックスで下矢印を押して目的のファイル・タイプに移動します。右タブを押します。
   c. 左タブを押して**Filepath**ボックスに移動します。下矢印を押して目的のタイプのバイナリ・ファイルに移動します。右タブを押します。
   d. 右タブを押して**Recall**ボックスに移動します。**Click**を押します。

外部コンピュータへのファイルのリコール

1. 本器にネットワーク・ドライブをマッピングしていない場合には、マッピングします(ドライブ・マッピング参照)。これにより、本器のハード・ディスク・ドライブがPCのディレクトリとして見えるようになります。
2. 外部コンピュータでWindows Explorer(R)の機能を使って、目的のASCIIまたはビットマップ・ファイルを本器からリコールします。

Windows Explorerの使用法については、外部コンピュータにインストールされている Windowsのヘルプ・システムを参照してください。

表示された測定結果のプリント

ネットワーク・プリンタまたはローカル・プリンタを使って、表示された測定結果をプリントすることができます。ボタンをクリックすると手順が表示されます。

- **ネットワーク上のプリンタへの接続**
- **アナライザへのプリンタの接続**
- **プリントアウトの生成**

ネットワーク上のプリンタへの接続

注記: この手順には、マウスとキーボードが必要です。

1. 本器リアパネルのLANコネクタとネットワークのLANジャックとをLANケーブルで接続します。
2. 本器のデスクトップで**Start**、**Settings**、**Printers**をクリックします。
3. **Add Printer**をダブルクリックして**Add Printer Wizard**を起動します。**Next**をクリックします。
4. **Network Printer**、**Next**をクリックします。
5. 下記のどちらかを実行して目的のプリンタに接続します。

- プリンタの名前を次のフォーマットで入力するか、**Next**をクリックしてネットワーク上のプリンタを検索します: ¥¥printservername¥sharename
- 次のフォーマットでプリンタのURLを入力します: http://printservername/Printers/sharename/.printer

注記: 上記の手順でプリンタに接続できない場合、プリンタのマニュアルを見るか、ネットワーク管理者に問い合わせてください。

6. 画面上の指示に従ってネットワーク・プリンタへの接続を完了します。
7. プリンタのアイコンがPrintersフォルダに表示されます。

アナライザへのプリンタの接続

注記: この手順には、マウスとキーボードが必要です。

1. プリンタ・メーカの説明書に従って、本器の適切なポートにプリンタを接続します。
2. ほとんどのプリンタはWindows 2000によって自動的にインストールされます。ただし、一部の古いプリンタの場合、インストールのために追加の情報を指定する必要があります。Printersフォルダにプリンタのアイコンが表示されない場合、お使いのプリンタのタイプに応じて、下記のどれかの手順を実行してください。

- **パラレル・ポート(LPT)互換プリンタのインストール**
- **USB互換プリンタのインストール**
- **シリアル・ポート(COM)互換プリンタのインストール**

パラレル・ポート(LPT)互換プリンタのインストール

1. 本器にプリンタを接続します。本器の付属品に、パラレル・ポートとセントロニクス・ポートとを接続するためのアダプタ(36ピン・オス(1284-C)-25ピン・メス)が含まれています。

注意: Ext. Test Set Interfaceというラベルの25ピン・メス型ポートをプリンタに接続しないでください。信号ラインの電圧レベルの違いにより、プリンタの入出力部分が損傷されるおそれがあります。

2. 本器のデスクトップで**Start**、**Settings**、**Printers**をクリックします。
3. **Add Printer**をダブルクリックして**Add Printer Wizard**を起動します。**Next**をクリックします。
4. **Local Printer**をクリックし、**Automatically detect**と**install my Plug and Play printer**の2つのチェック・ボックスが選択されていることを確認して、**Next**をクリックします。
5. インストールするプリンタに応じて、Printer Wizardが続行されるか、Found New Hardwareメッセージまたは Found New Hardwareウィザードが表示されて、インストールの開始を示します。表示される指示に従って、プリンタのインストールを完了します。

- もう1つの方法として、プリンタに接続した後、本器を起動または再起動すると、Windows 2000が自動的にFound New Hardwareウィザードを表示します。

6. Printersフォルダにプリンタのアイコンが追加されます。

USB互換プリンタのインストール

本器のUSBポートにプリンタを接続します。このタイプのプリンタはWindows 2000によって自動的に検出され、Found New Hardwareウィザードが起動します。

本器を停止したり再起動したりする必要はありません。画面上の指示に従ってプリンタのインストールを完了します。

Printersフォルダにプリンタのアイコンが追加されます。

シリアル・ポート(COM)互換プリンタのインストール

1. **本器のシリアル・コネクタにプリンタを接続します。**
2. 本器のデスクトップで**Start**、**Settings**、**Printers**をクリックします。
3. **Add Printer**をダブルクリックして**Add Printer Wizard**を起動します。**Next**をクリックします。
4. **Local Printer**をクリックし、**Automatically detect**と**install my Plug and Play printer**の2つのチェック・ボックスが選択されていることを確認して、**Next**をクリックします。
5. 指示に従って下記の項目を選択することにより、プリンタのインストールを完了します。

プリンタ・ポート

プリンタのメーカとモデル

プリンタの名前

6. Printersフォルダにプリンタのアイコンが追加されます。

注記: プラグ・アンド・プレイ・プリンタ(USBやパラレル・ポート(LPT)など)を追加してセットアップする場合、管理者特権は不要です。これに対して、非プラグ・アンド・プレイ・プリンタを本器に直接接続する場合、管理者としてログインするか、管理者グループに属している必要があります。本器がネットワークに接続されている場合、ネットワーク・ポリシー設定のためにこの手順が最後まで実行できない可能性もあります。

プリントアウトの生成

マウス手順

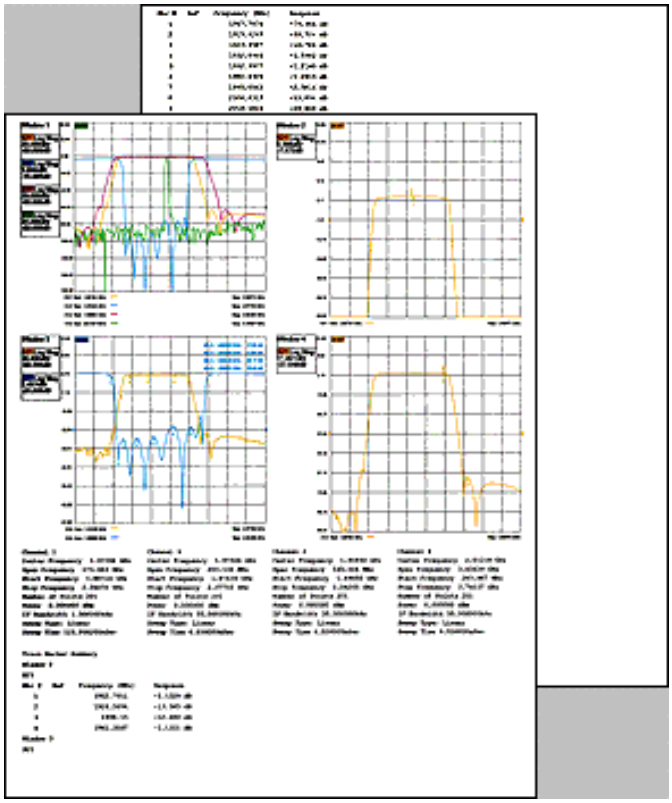
1. **File** メニューで**Print**をクリックします。
2. **Print**ダイアログ・ボックスで**Name**に移動します。矢印ボタンをクリックして目的のプリンタを選択します。**OK**をクリックします。

フロントパネル・キー手順

1. **UTILITY**ボックスで**Print**アイコン・キーを押します。
2. **NAVIGATION**ブロックで上下の矢印を押して、**Name**ボックスで目的のプリンタを選択します。**OK**を押します。

注記: Printダイアログ・ボックスでの選択の詳細については、Windows 2000のヘルプを参照してください。

下の図は、プリントされた画面キャプチャ・イメージを示します。テキスト・フォームのデータが画面キャプチャ・イメージに自動的に付加されることに注意してください。



測定チュートリアル

ボタンをクリックすると、下記の測定について詳しく知ることができます。

- 位相測定リファレンス
- 増幅器パラメータ・リファレンス
- AM-PM変換
- 複素インピーダンス
- リニア位相からの偏移
- 群遅延
- 利得とフラットネス
- 利得圧縮
- リバース・アイソレーション
- ハイパワー・コンポーネント
- 絶対出力パワー
- インピーダンス整合モデル
- 伝送ライン上の反射波
- 時間領域

位相測定リファレンス

本器では、以下の位相測定を使ってデバイスの特性評価を実行できます。
リニア位相からの偏移
群遅延
複素インピーダンス
AM-PM変換
高レベル・コンポーネントのインテグレーションを成功させるには、振幅特性と位相特性の両方を知っておく必要があります。
ボタンをクリックすると、下記の情報が表示されます。

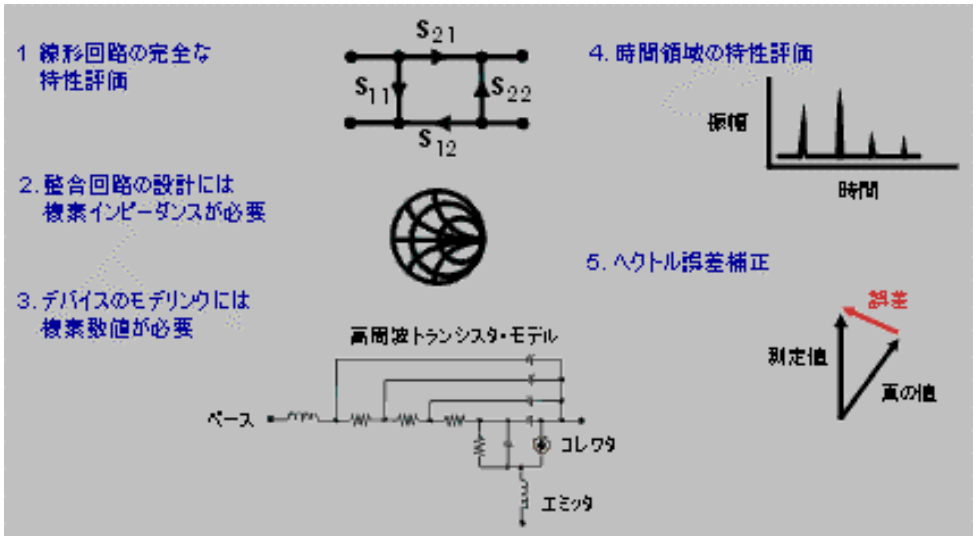
位相測定とは何か
なぜ位相を測定するのか
位相フォーマットの使用
位相測定のタイプ

位相測定とは何か

位相測定は、振幅測定と同様にSパラメータを使って行われます。位相測定は絶対測定ではなく相対(比)測定です。位相測定では、デバイスに入る信号(入射信号)の位相とデバイスの応答信号の位相とを比較します。応答信号は反射信号と伝送信号のどちらかです。正確な校正が実行されていたとすると、2つの信号の位相差(位相シフトとも呼ぶ)は被試験デバイスの電気特性の結果となります。
下の図は、入射信号と伝送信号との位相シフト(時間または度単位)をオシロスコープ画面の形式で示したものです。

なぜ位相を測定するのか

位相の測定は、ネットワーク解析で最も重要な要素の1つです。振幅と位相の両方を測定する理由を下の図に記します。



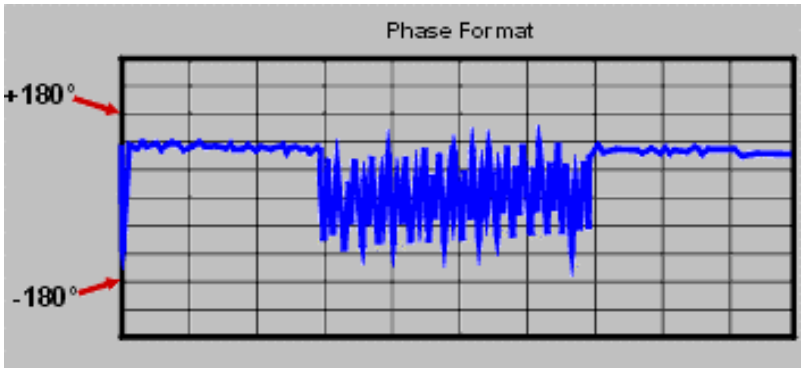
通信システムで信号を通過するためのコンポーネントや回路は、信号歪みをなるべく起こさないようにする必要があります。歪みには下記のものがあります。

- リニア歪み。これは、目的の帯域幅で振幅のフラットネスと周波数に対する位相シフトのリニアリティが維持されないことを指します。
- ノンリニア歪み。AM-PM変換など。

コンポーネントや回路が効率よくエネルギーを伝送または吸収することを確認するためには、その反射性を測定することが重要です。アンテナの複素インピーダンスの測定がよい例です。

位相フォーマットの使用

本器の位相フォーマットは、位相対周波数または位相対パワー測定を表示します。基準信号と試験信号との測定可能な位相差は±180°までです。したがって、位相の値が+180°と-180°の間で変化することにより、下図のようなのこぎり形が描かれます。



のこぎり形は必ずしも+180°と-180°に達するとは限りません。これは、測定が離散的な周波数で行われるため、+180°と-180°のポイントが指定された掃引で測定されない可能性があるからです。

位相測定のタイプ

- **複素インピーダンス データ**は、S11またはS22測定から求められる抵抗、リアクタンス、位相、振幅などの情報です。複素インピーダンス・データはスミス・チャート・フォーマットまたは極座標フォーマットで表示します。
- **AM-PM変換**とは、システムの振幅変動(AM)から生じる不要な位相偏移(PM)の量を測定するものです。AM-PM変換は通常、増幅器に入力されるパワーの1 dBの増加に対する(1 dB利得圧縮ポイントでの)出力位相の変化と定義されます。単位は°/dBです。
- **リニア位相からの偏移**とは、デバイスによる位相歪みを測定するものです。理想的なデバイスの位相シフトは、周波数のリニアな関数になります。この理論的な位相シフトからのずれの大きさのことをリニア位相からの偏移 (または位相リニアリティ)と呼びます。
- **群遅延**とは、デバイスによる位相歪みの別の見方です。群遅延は特定の周波数におけるデバイスの通過時間を測定するものです。本器は測定された位相応答を微分することにより群遅延を計算します。

リニア位相からの偏移と群遅延

本器は、位相シフトのリニアリティをある周波数レンジで測定し、2つの方法で表すことができます。

- 直接、リニア位相からの偏移として
- 導出値、すなわち群遅延として

リニア位相からの偏移測定の利点

- 群遅延よりノイズが少なくなります。
- 位相変調された信号を通すデバイスの特性評価が可能で、秒単位でなく位相単位の結果が得られます。

群遅延測定の利点:

- 位相歪みの指標としてリニア位相からの偏移 よりも解釈が容易です。
- 被試験デバイスのより正確な特性評価が可能です。その理由は、群遅延を求める際に位相リップルの傾きが計算されるからです。位相リップルの傾きは、単位周波数あたりに発生するリップルの数に依存します。p-p位相リップルが等しい2つの位相応答を比較した場合、位相の傾きが大きい応答の方が以下の性質を示します。
  - 群遅延変動が大きい
  - 信号歪みが大きい

増幅器パラメータのリファレンス

- 利得
- 利得フラットネス
- リバース・アイソレーション
- 利得ドリフト対時間
- リニア位相からの偏移
- 群遅延
- リターン・ロス(SWR、ρ)
- 複素インピーダンス
- 利得圧縮
- AM-PM変換

利得

$$\tau = \frac{V_{trans}}{V_{inc}}$$

$$Gain\ (dB) = -20\log_{10}|\tau|$$

$$Gain\ (dB) = P_{out}\ (dBm) - P_{in}\ (dBm)$$

増幅器の出力パワー(Z_0負荷に供給されるもの)と入力パワー(Z_0ソースから供給されるもの)との比。Z_0は特性インピーダンスであり、この例では50Ωです。

信号レベルが低いときには、増幅器の出力パワーは入力パワーに比例します。小信号利得とは、このリニア領域での利得を指します。

入力パワー・レベルが増加し、増幅器の飽和レベルに近づくにつれて、出力パワーは限界に達し、利得が低下します。大信号利得とは、このノンリニア領域での利得を指します。利得圧縮を参照してください。

利得フラットネス

増幅器の周波数レンジにおける利得の変動。小信号利得とフラットネスを参照してください。

リバース・アイソレーション

出力から入力への伝送の尺度。利得測定と似ていますが、信号スティミュラスが増幅器の出力に印加される点が異なります。リバース・アイソレーションを参照してください。

利得ドリフト対時間(温度、バイアス)

他のすべてのパラメータが一定のときの、時間に対する利得の変動の最大値。利得ドリフトは、温度、湿度、バイアス電圧などの他のパラメータの変化に対しても観察されます。

リニア位相からの偏移

リニア位相シフトからの変動の大きさ。理想的には、増幅器による位相シフトは周波数のリニア関数になります。リニア位相からの偏移を参照してください。

群遅延

$$\tau_g\,(\mathrm{sec}) = -\frac{\Delta\theta}{\Delta\omega} = -\frac{1}{360} * \frac{\Delta\theta}{\Delta f}$$

増幅器の通過時間を周波数の関数として測定するもの。位相シフトが完全にリニアなら、周波数に対する変化率は一定であり、群遅延は定数になります。群遅延を参照してください。

リターン・ロス(SWR、ρ)

$$\Gamma = \frac{V_{refl}}{V_{inc}} = \rho\angle\theta$$

$$\text{Reflection coefficient} = \rho$$

$$\text{Return loss (dB)} = -20\log_{10}\rho$$

$$\text{SWR} = \frac{1+\rho}{1-\rho}$$

増幅器の入力または出力における反射不整合を、システムのZ_0特性インピーダンスを基準として測定したもの。

複素インピーダンス

$$Z = \frac{1+\Gamma}{1-\Gamma} * Z_0 = -R + jX$$

複素インピーダンス(1+G)。増幅器から反射されるエネルギーは、増幅器のインピーダンスと直接の関係があります。複素インピーダンスは抵抗成分とリアクタンス成分からなります。複素インピーダンスは、システムの特性インピーダンスと反射係数から導出されます。複素インピーダンスを参照してください。

利得圧縮

増幅器には、利得が入力パワー・レベルに依存しないリニア利得領域があります(小信号利得)。パワーが増加して増幅器の飽和レベルに達すると、利得は減少します。

利得圧縮を求めるには、増幅器の1 dB利得圧縮ポイント(P_{1dB})を測定します。これは、利得が小信号利得に対して1 dB小さくなる出力パワーを指します。これは増幅器のパワー出力能力の一般的な尺度です。利得圧縮を参照してください。

AM-PM変換係数

$$\text{AM/PM} = \frac{\Delta\theta}{\Delta P}$$

入力信号の振幅変化に応じて増幅器の出力信号に生じる位相変化の大きさ。

AM-PM変換係数は、特定のパワー・レベル(通常はP_{1dB}すなわち1 dB利得圧縮ポイント)における°/dB単位で表されます。AM-PM変換を参照してください。

複素インピーダンス

概要　　概念　　測定方法
　　複素インピーダンス

被試験デバイスのS_{11}またはS_{22}測定を実行する際には、直列抵抗/リアクタンスや位相/振幅情報などの複素インピーダンス・データを表示することができます。複素インピーダンス・データを表示するには、スミス・チャート・フォーマットまたは極座標フォーマットを使います。

ボタンをクリックすると下記の事項について知ることができます。:

複素インピーダンスとは何か

複素インピーダンス・データとは、被試験デバイスのS_{11}またはS_{22}測定から得られる情報で、以下のようなものがあります。

- 抵抗
- リアクタンス
- 位相
- 振幅

デバイスから反射されるパワーの大きさは、デバイスと測定システムの両方のインピーダンスに直接関係します。例えば、複素反射係数(Γ)の値が0になるのは、デバイスのインピーダンスとシステムのインピーダンスが正確に一致するときだけです(この場合、ソースから負荷に伝達されるパワーが最大になります)。Γの値は、下記の式によって複素デバイス・インピーダンスと(周波数の関数として)一意に対応します。

$Z_L = [(1 + \Gamma) / (1 - \Gamma)] \times Z_0$

ここで、Z_Lは被試験デバイスのインピーダンス、Z_0は測定システムの特性インピーダンスです。

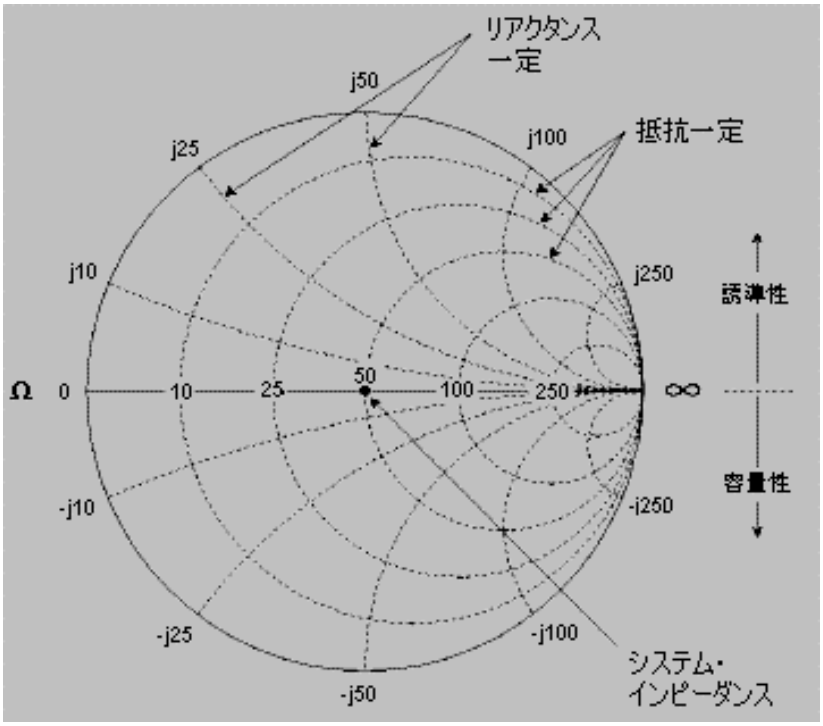
スミス・チャート・フォーマットによるデータの表示

被試験デバイスのS_{11}またはS_{22}測定から複素インピーダンス・データを表示する方法の1つは、本器のスミス・チャート・フォーマットを選択することです。スミス・チャートは複素反射係数(Γ)を被試験デバイスのインピーダンスにマッピングするツールです。

スミス・チャートでは、複素インピーダンス平面が円形のグリッドに変形され、そこから直列抵抗およびリアクタンス(R + jX)を読み取ることができます。本器のマーカ機能では下記の情報が表示されます。

- 抵抗(Ω単位)
- 等価キャパシタンス(F単位)またはインダクタンス(H単位)としてのリアクタンス

以下の説明については、下の図を参照してください。



- 水平軸(実線)はインピーダンスの実数部、すなわち抵抗です。水平軸の中心は常にシステム・インピーダンスを表します。

水平軸と交わる破線の円は、一定の抵抗を表します。水平軸と接する破線の円弧は、一定のリアクタンスを表します。

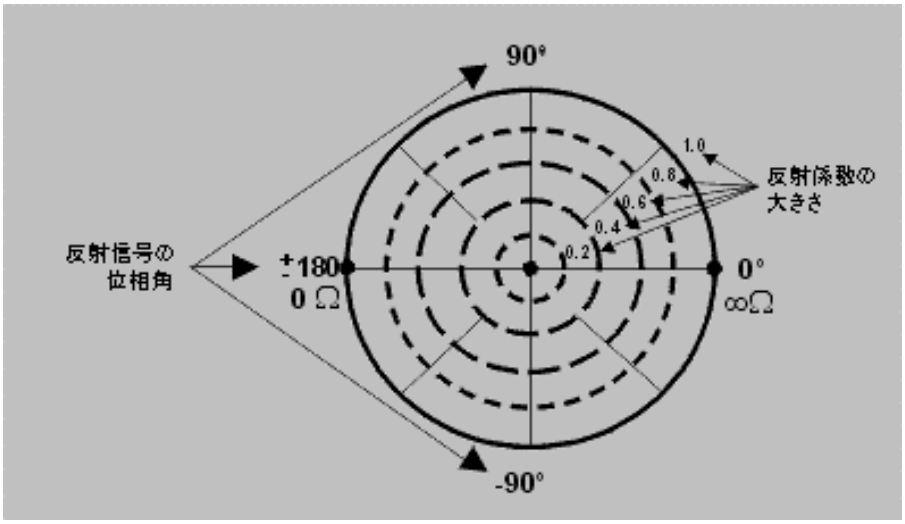
スミス・チャートの上半分はリアクタンス成分が正、すなわちインダクタンス性の領域です。下半分はリアクタンス成分が負、すなわちキャパシタンス性の領域です。

極座標フォーマットによるデータの表示

スミス・チャート・フォーマットの他に、極座標フォーマットを使ってS_{11}またはS_{22}測定から得られた反射係数(Γ)の振幅と位相を表示することもできます。本器のマーカ機能では下記の情報が表示されます。

- リニア振幅(測定単位)または対数振幅(dB)
- 位相(度)

以下の説明については、下の図を参照してください。

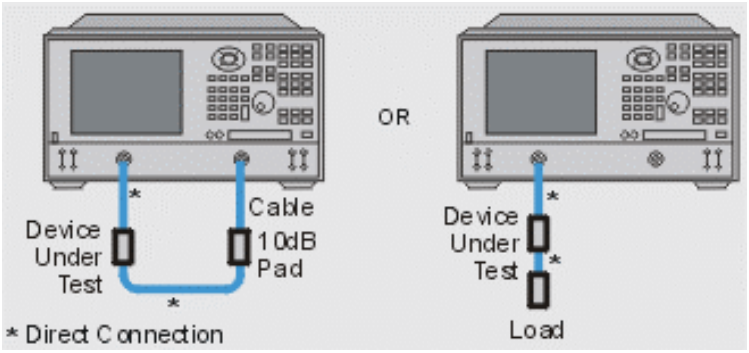


- 破線の円は反射係数を表します。いちばん外側の円が反射係数(Γ)1、すなわち全反射信号に対応します。円の中央は反射係数(Γ)0、すなわち無反射に対応します。
- 放射状の線は、反射信号の位相角を示します。右端の位置は位相角0(反射信号と入射信号が同位相)に対応します。位相角90°、±180°、-90°は、それぞれ極座標表示の上端、左端、下端に対応します。

確度に関する考慮事項

- スミス・チャートについて理解するには、フル・スケール値1.0の場合を見るのが最も簡単です。
- スミス・チャートまたは極座標フォーマットでのマーカの確度を改善するには、離散マーカ・モードをオンにします。
- 反射測定の不確かさは下記の要素に影響されます。
- 方向性
- 反射トラッキング
- 信号源整合
- 負荷整合(2ポート・デバイスの場合)

2ポート校正を使うと、これらの要素の影響を減らすことができます。1ポート校正でも、デバイスの出力が適切に終端されていれば同等の確度が得られます。以下の説明については、下の図を参照してください。



- 本器の2つのポートの間にデバイスを接続する場合、測定確度の改善のためにはデバイスの出力に10 dBのパッドを使用することが推奨されます。2ポート校正を行えば負荷整合が補正されるため、これは必要ありません。
- 2ポート・デバイスを本器の1つのポートだけに接続する場合、高品質の負荷(校正標準など)をデバイスの出力に接続することが推奨されます。

1. 前の図のようにデバイスを接続します。
2. アナライザを初期設定します。
3. S11またはS22測定を準備し、校正し、実行します。
4. インピーダンス・データの表示:
   a. スミス・チャート・フォーマットを選択します。
   b. 測定結果の表示が見やすくなるようにスケールを調整します。
   c. マーカを配置することにより、トレース上の任意の点における複素インピーダンスの抵抗成分とリアクタンス成分を読み取ることができます。
   d. データをプリントするか、ディスクに保存します。

5. アドミタンス・データの表示(必要な場合):
   a. 逆スミス・チャートを表示するマーカ機能を選択します。
   b. 測定結果の表示が見やすくなるようにスケールを調整します。
   c. マーカを配置することにより、トレース上の任意の点における複素インピーダンスのコンダクタンス成分とサセプタンス成分を読み取ることができます。コンダクタンスとサセプタンスはどちらもS(Ωの逆数)単位で測定されます。
   d. データをプリントするか、ディスクに保存します。
6. 反射係数の振幅と位相の表示:
   a. スミス・チャート・フォーマットまたは極座標フォーマットを選択します。
   b. リニア・マーカまたは対数マーカ・フォーマットを選択します。
   c. 測定結果の表示が見やすくなるようにスケールを調整します。
   d. マーカを配置することにより、トレース上の任意の点における反射係数(Γ)の周波数、振幅、位相を読み取ることができます。
   e. データをプリントするか、ディスクに保存します。

群遅延

概要　　概念　　測定方法

群遅延は、位相歪みの度合いを示します。デバイスを通る信号の実際の通過時間を、周波数の関数として表したものです。群遅延を指定する際には、測定に使用するアパーチャを定めることが重要です。

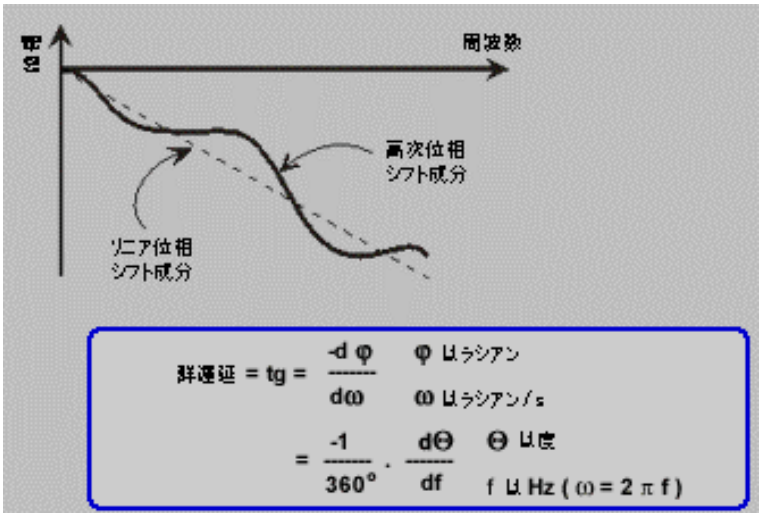
ボタンをクリックすると下記の事項について知ることができます。

群遅延とは何か

群遅延とは下記のものを指します。

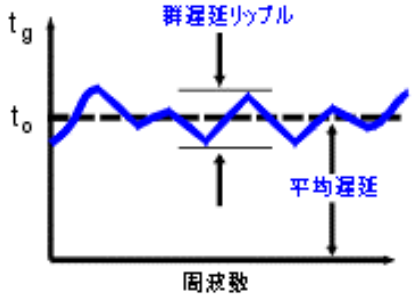
- デバイスの位相歪みの度合いを示します。
- デバイスを通る信号の通過時間を周波数に対して示したものです。
- デバイスの位相特性を周波数で微分したものです。

以下の説明については下の図を参照してください。



デバイスの位相特性には、リニア位相シフト成分と、高次(リニアからの偏移)位相シフト成分とが含まれるのが普通です。

以下の説明については下の図を参照してください。



群遅延測定では:

- リニア位相シフト成分が定数値(平均遅延を表す)に変換されます。
- 高次位相シフト成分が一定の群遅延からの変位(群遅延リップル)に変換されます。
- 群遅延の変位は信号歪みを引き起こします。これは、リニア位相からの偏移が歪みを引き起こすのと同じです。
- 測定トレースは、各周波数成分が被試験デバイスを通過するのにかかる時間を表します。

群遅延の計算方法については、下の式を参照してください。

$$\text{Group Delay} = tg = \frac{-d\varphi}{d\omega} = \frac{-1}{360°} \cdot \frac{d\Theta}{df}$$

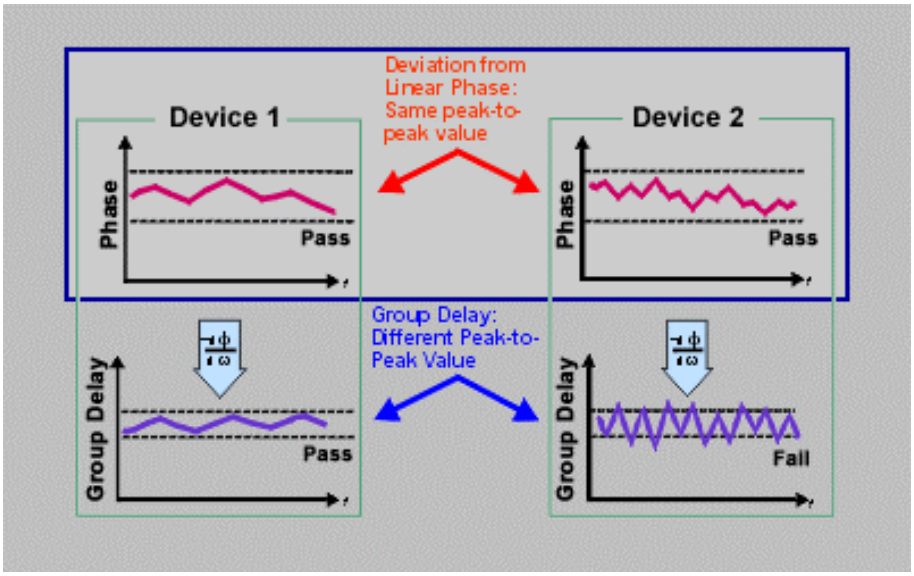
φ in Radians
ω In Radians/Sec
Θ in Degrees
f in Hz ($\omega = 2\pi f$)

- 位相データから位相変化(-d□)が求められます。
- 指定された周波数アパーチャを使って、周波数変化(d□)が求められます。
- 上記2つの値を使って、周波数に対する位相の変化率の近似値が計算されます。
- この近似値は、秒単位の群遅延を表します(指定された周波数アパーチャでの位相変化がリニアであると仮定)。

群遅延とリニア位相からの偏移との違い

多くの場合、群遅延のほうがリニア位相からの偏移よりも位相歪みを正確に表します。

リニア位相からの偏移の結果を、下の図の上の部分に示します。デバイス1とデバイス2は、見かけが異なるにも関わらず同じ値を示します。

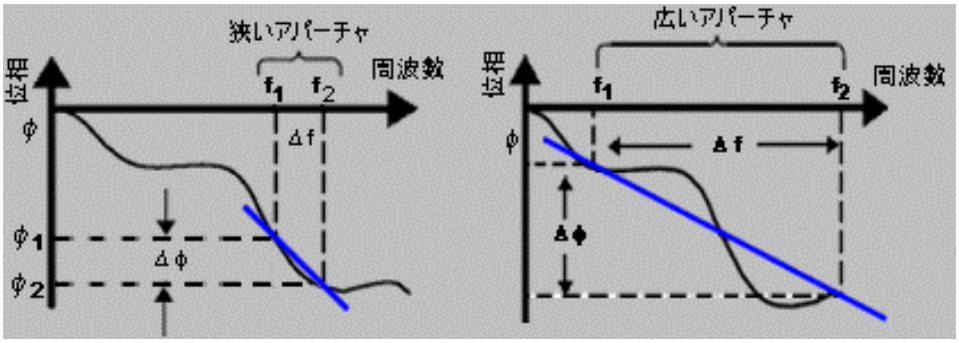


群遅延 の結果を下の部分に示します。
デバイス1とデバイス2の群遅延の値は異なっています。これは、群遅延を求める際には位相リップルの傾きが計算されるからです。位相リップルの傾きは単位周波数あたりに発生するリップルの数に依存します。

アパーチャとは何か

群遅延測定の際には、わずかに異なる2つの周波数での位相が測定され、その結果から位相の傾きが計算されます。2つの位相測定ポイントの周波数間隔(周波数デルタ)のことを、アパーチャと呼びます。アパーチャを変えると群遅延の値が変化する場合があります。アパーチャを増やすと、計算結果のスロープ(デルタ位相)が異なってきます。このため、群遅延データを比較する際には、測定の際に用いられたアパーチャを知ることが重要になります。

以下の説明については下の図を参照してください。



群遅延アパーチャに対する本器のデフォルト設定は、周波数スパンを表示されるポイント数で割ったものです。アパーチャを別の値に設定するには2つの方法があります。

1. 測定ポイントの数または周波数スパンを調整します。
- ポイント数を増やすか周波数スパンを小さくすると、アパーチャが狭まります。
- ポイント数を減らすか周波数スパンを大きくすると、アパーチャが広がります。

注記: アパーチャが広すぎる(隣接する周波数ポイントの間の位相シフトが180°を超える)場合、群遅延データにエラーが生じます。

2. 本器のスムージング機能を使います。
- シングル掃引で、周波数スパンの指定された割合における隣接するデータ・ポイントの移動平均を求めます。
- ポイントの周波数間隔を変更するのと同様の動作になります。
- スムージング・アパーチャでは位相シフトが180°より大きくてもかまわないので、より広いアパーチャを実現できます。

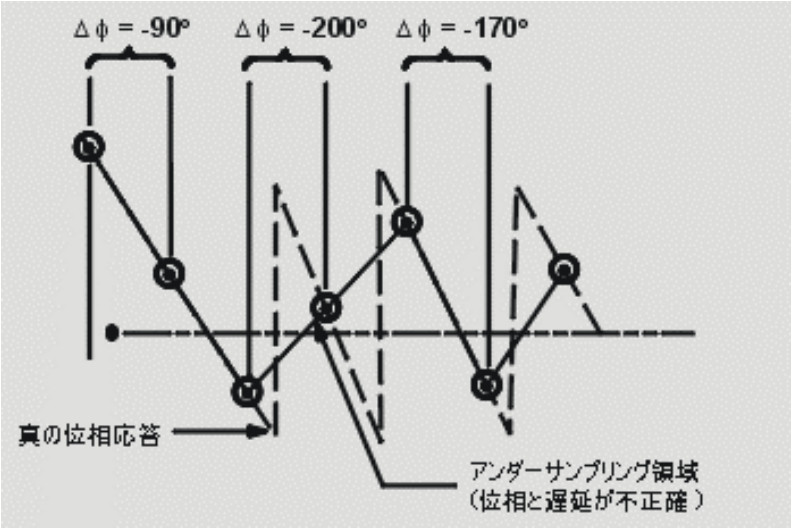
群遅延測定は下記の掃引タイプで実行できます。
- リニア周波数
- リスト周波数掃引セグメント

群遅延アパーチャは周波数間隔とポイント密度によって変化するため､セグメント掃引ではアパーチャが一定になりません｡セグメント掃引で必要なアパーチャを実現するには､追加の周波数ポイントを定義する方法があります。

確度に関する考慮事項

2個の隣接する測定ポイントの位相差は180°よりも小さくなければなりません(下の図を参照)。そうでないと、得られる位相および遅延の情報が正しくなくなります。電気長の長いデバイスを測定する際にアンダーサンプリングが発生するおそれがあります。2個の隣接するポイントの間の位相差が180°より小さくなるようにするには、測定トレースが変化しなくなるまで下記の設定を調整します。
- ポイント数を増やします。
- 周波数スパンを狭めます。

電気遅延を使うことによってもこの影響を補正できます。



群遅延テスト・セットアップにおける支配的な誤差は、周波数応答です。スルー・レスポンス測定校正により、この誤差を大きく減らすことができます。さらに高い確度が必要なら、2ポート測定校正を実行します。

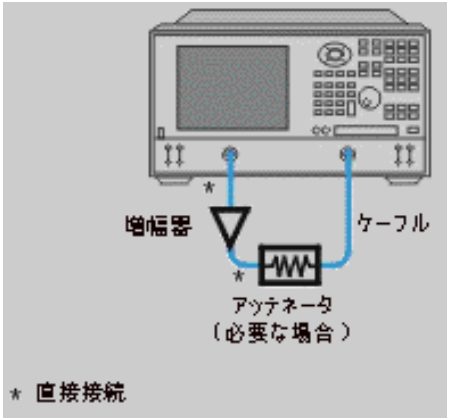
特に増幅器の場合、温度が異なると応答が変わる場合があります。テストは増幅器が目的の動作温度にあるときに行うべきです。

群遅延の測定方法

1. 本器を初期設定します。
2. 被試験デバイスが増幅器の場合、本器の信号源出力の調整が必要な場合があります。

- 本器の信号源出力を、増幅器の出力応答の線形領域(通常は1 dB圧縮ポイントより10 dB下)に設定します。
- 必要なら外部アッテネータを選択して、本器のポート2のレシーバに圧縮や損傷が生じないように増幅器の出力パワーを減衰させます。

3. 下の図のように被試験デバイスを接続します。



4. S_{21}測定を選択します。
5. 被試験デバイスに対する設定を選択します。下記の設定を使用します。

- 測定ポイント数: 最大
- フォーマット: 遅延
- スケール: オートスケール

6. 被試験デバイスを取り除き、測定校正を実行します。
7. 被試験デバイスを再接続します。
8. 測定結果の表示が見やすくなるようにスケールを調整します。
9. 本器のスムージング機能を使ってアパーチャを広げ、必要な細部が見られる範囲でトレースのノイズを減らします。アパーチャを広げる手順は下記の通りです。

- 本器のスムージング機能をオンにします。
- スムージング・アパーチャを変化させます(掃引スパンの25%まで)。

10. マーカを使って、目的の周波数における群遅延(秒単位)を測定します。
11. データをプリントするか、ディスクに保存します。

絶対出力パワー

概要　　概念　　絶対出力パワーの測定方法

絶対出力パワー測定は、絶対パワーを周波数に対して表示します。
ボタンをクリックすると下記の事項について知ることができます。

絶対出力パワーとは何か

絶対出力パワー測定は、本器の入力ポートに存在するパワーを表示します。このパワーは絶対値であり、入射信号や信号源のパワーを基準としたもの(比測定)ではありません。対数振幅フォーマットの場合、グリッドの垂直軸はdBm単位です。これは1 mWを基準としたパワーを表します。

- 0 dBm = 1 mW
- -10 dBm = 100 □W
- +10 dBm = 10 mW

リニア振幅フォーマットの場合、グリッドの垂直軸の値はW単位です。

なぜ絶対出力パワーを測定するのか

絶対パワー測定は、増幅器の出力を相対パワー(比測定)でなく絶対パワーとして定量化する場合に使用します。例えば、利得圧縮測定の際には、絶対出力パワーも測定するのが普通です。これは1 dB圧縮が発生する増幅器の絶対出力パワーを示します。

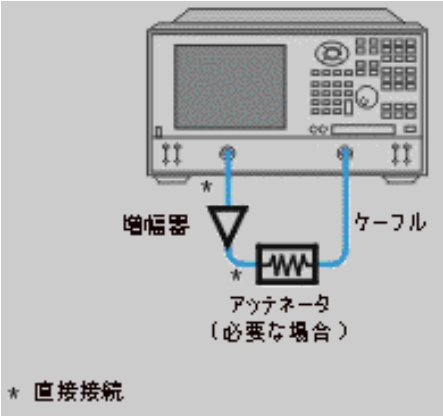
確度に関する考慮事項

- 場合によっては、増幅器の出力を減衰させなければならないことがあります。出力パワーが大きすぎると、下記のような影響が生じるおそれがあります。
- 本器のレシーバが損傷される
- 本器のレシーバの入力圧縮レベルを超過し、不正確な測定結果が生じる
- 増幅器の出力を減衰させるには、次の方法があります。
- アッテネータ
- カプラ
- 増幅器は温度が異なると非常に異なる応答を示す場合があります。テストは増幅器が目的の動作温度にあるときに行うべきです。

絶対出力パワーを測定するには、以下の手順を実行します。

1. アナライザを初期設定します。
2. 絶対パワー測定(レシーバB)を選択します。
3. 本器の信号源出力を0 dBmに設定します。
4. 必要なら外部アッテネータを選択して、本器のポート2でレシーバの圧縮や損傷が生じないように増幅器の出力パワーを減衰させます。

5. 増幅器を下の図のように接続し、DCバイアスを供給します。



6. 被試験増幅器に対するアナライザ設定を選択します。
7. 増幅器を取り外し、測定ポート同士を接続します。データをメモリに記憶します。増幅器の測定の際にアッテネータやケーブルを使用する場合、それらも必ずテスト・セットアップに含めます。
8. 機器ステートをメモリに保存します。
9. 増幅器を再接続します。
10. データ演算機能のData/Memoryを選択します。
11. 測定結果の表示が見やすくなるようにスケールを調整し、マーカを使って目的の周波数での絶対出力パワーを測定します。
12. データをプリントするか、ディスクに保存します。

ハイパワー・コンポーネント測定

概要　概念　ブロック図　標準構成　オプション015　測定方法

ハイパワー・デバイスを測定するために本器を構成することができます。ハイパワー・デバイスとは、下記の性質のどちらかまたは両方を持つもののことです。

- 本器の供給限度よりも大きい入力パワーを必要とするもの。
- 標準のアナライザで測定できるよりも大きい出力パワーを供給するもの。

このため、これらのアナライザ構成は下記のように設計されています。

- デバイスのスティミュラス信号をブースタ増幅器によって増幅します。
- 本器の受信信号に減衰を与えることで、本器のコンポーネントを過大な入力から保護します。

ボタンをクリックすると下記の事項について知ることができます。

ハイパワー測定はなぜ難しいか

これらの構成を使用する際の困難としては下記のものがあります。

- 高レベル信号を安全に受信し、低レベル信号を正確に受信すること。
- 校正と正確な測定。
- 一部の校正が実行できないこと。
- 測定可能なパラメータの数が制限されること。

例えば、一部の構成では逆Sパラメータが測定できません。このため一部の校正が実行できず、測定の確度が制限されます。

ネットワーク・アナライザによるハイパワー測定の詳細について知るには、Agilentのウェブ・ページ(http://¥www.agilent.com/find/pna)でアプリケーション・ノート1287-6、"Using a Network Analyzer to Characterize High-Power Components"(カタログ番号5966-3319E)を検索してください。

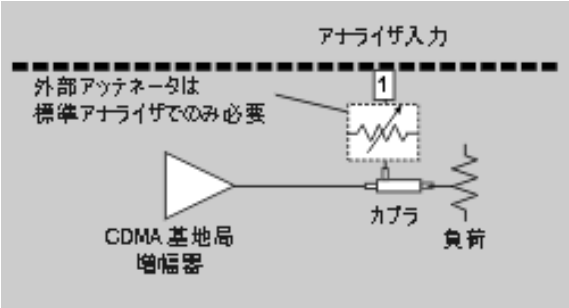
CDMAハイパワー基地局増幅器

CDMAハイパワー基地局増幅器の信号は、無線伝送による損失を受けても移動機で受信が可能なレベルまで増幅する必要があります。CDMA基地局増幅器の代表的な要件として下記の2つが挙げられます。

- 出力パワー: 8～45 W(39～46.5 dBm)
- 利得: 30～40 dB

(これらの要件は、メーカごとのシステム・デザインによって異なる場合があります)

CDMA増幅器の高い出力パワー・レベルを、本器が安全に扱えるレベルまで下げる必要があります。下記のコンポーネントを選択する際には、そのパワー処理能力が十分であるかどうかに注意する必要があります(下の図を参照)。



- 標準のアナライザの場合、被試験デバイスの出力に外部ハイパワー結合器を設置します。この結合器は、本器に入る前に被試験デバイスの出力パワーを下げる役割を果たします。結合器のスルー・アームは、大電力を消費する特殊な負荷で終端されます。結合アームは外部ハイパワー・アッテネータに接続され、それが本器の入力につながれます。
- オプション015(構成可能テスト・セット)を装備したアナライザの場合、上記の外部ハイパワー結合器と負荷だけを使用します。オプション015にはレシーバを保護する内部ステップ・アッテネータ(0～35 dB)が用意されているので、外部ハイパワー・アッテネータは不要です。

注記: ネットワーク・アナライザの安全な動作を保証するため、テスト・ポートの入力パワーは+30 dBm(1 W)を超えないようにしてください。

正確な測定を行うには、測定中の温度条件も考慮する必要があります。例えば、CDMA基地局増幅器をテストする場合、出力結合器に使用する負荷は下記の条件を満たす必要があります。

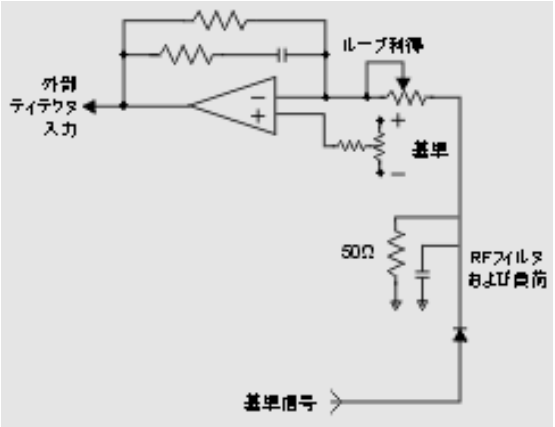
- 被試験増幅器から供給されるパワーを吸収できること。小信号向けに設計された負荷の多くは、最大1 W程度のパワーしか扱うことができません。それ以上のパワーに対しては、より多くのパワーを消費できる特殊な負荷を使用する必要があります。
- インピーダンス特性が温度に対して大きく変化する場合、温度を安定化させておくこと。

外部レベリング

外部レベリングを使ってブースタ増幅器のノンリニア動作を補正することで、測定確度を上げることができます。外部レベリングを使用した測定構成では、ブースタ増幅器の後に結合器を置き、その結合アームからパワー・スプリッタに基準信号を送ります。基準信号はパワー・スプリッタから2つの方向に送られます。

- 本器の内部基準レシーバへ
- 外部レベリング回路を介して本器リアパネルの外部ディテクタ入力へ

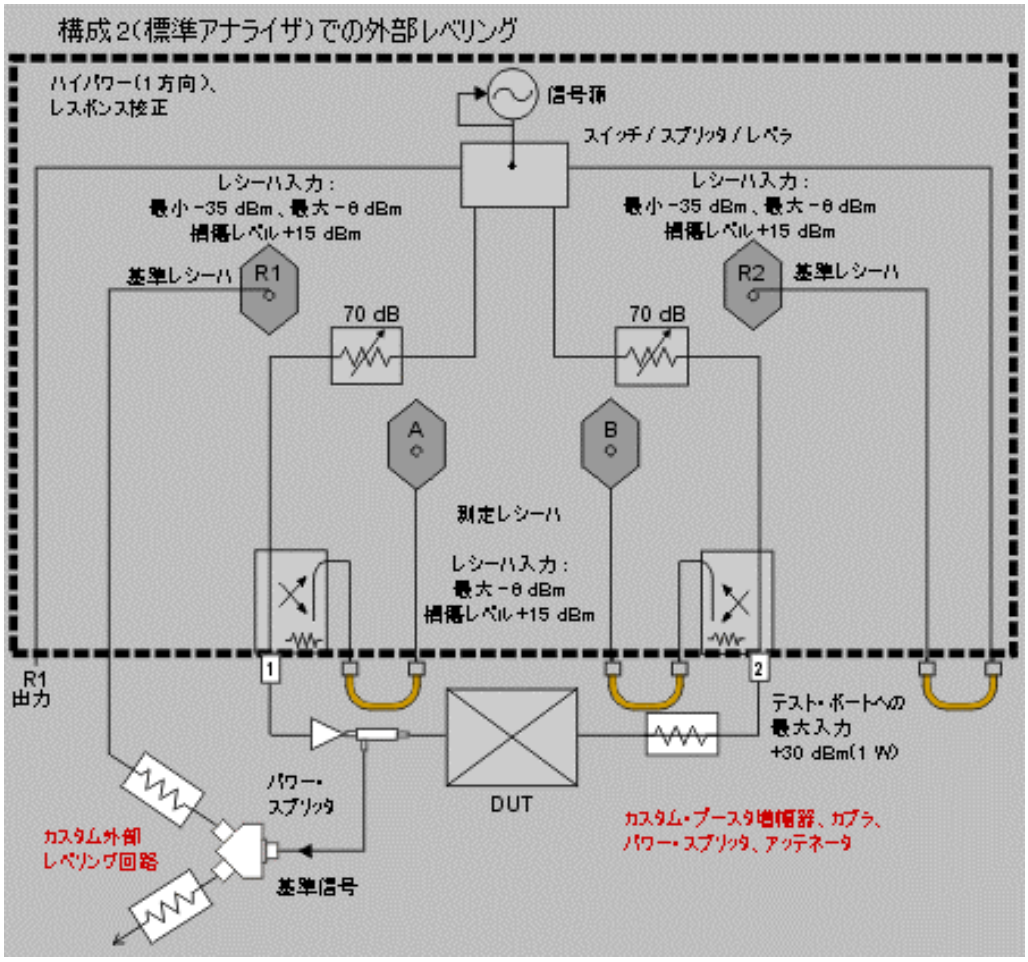
以下の説明に関しては、下の外部レベリング回路(ユーザが用意)の図を参照してください。

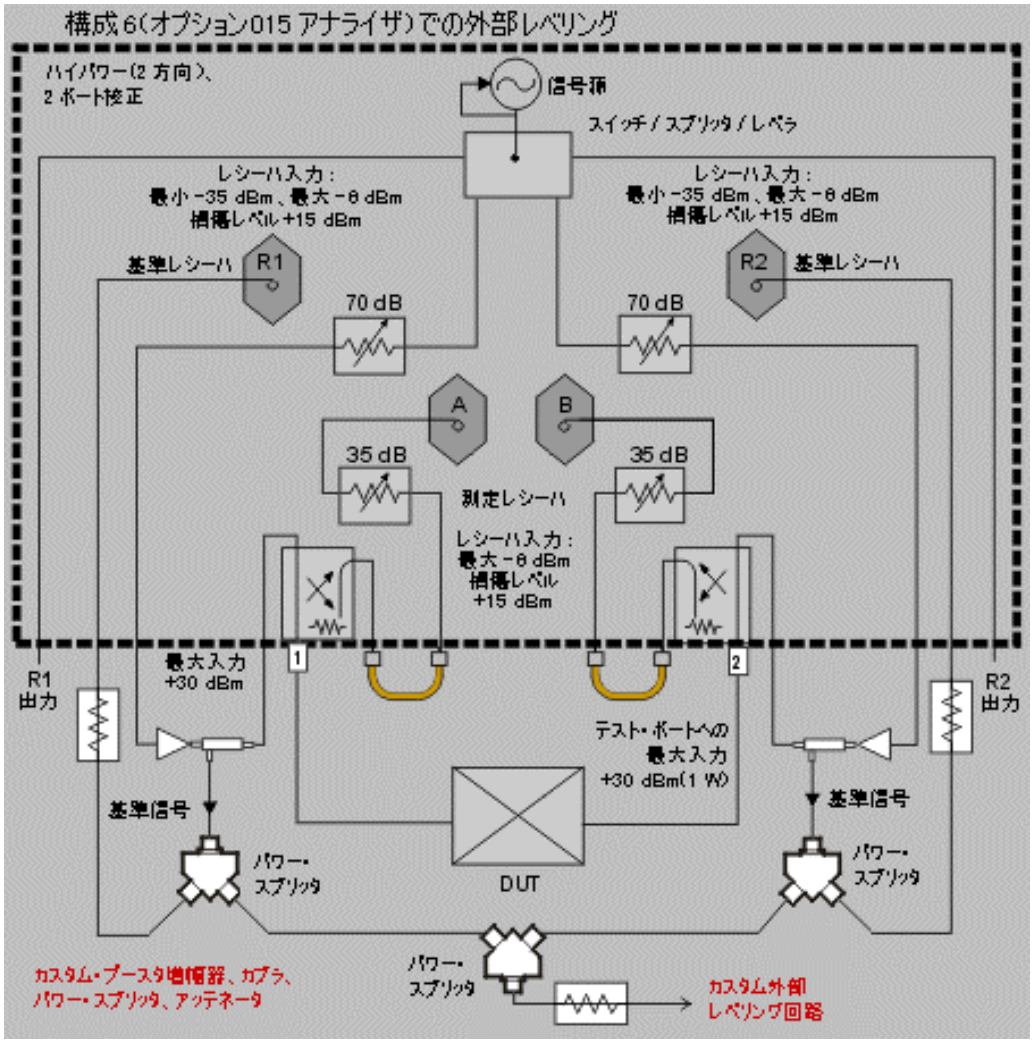


- 外部レベリング回路の主な構成要素は下記の通りです。
- ディテクタ・ダイオード
- オペアンプ
- ダイオードで測定されたパワーの変化がオペアンプ回路で検出されます。この回路は外部ディテクタ入力の電圧をレギュレートし、DUTでの応答がフラットになるようにリアルタイムで信号源出力を調整します。
- 外部レベリングが正しく動作するには、外部ディテクタ入力での電圧が、周波数とパワー・レベルに応じて、-0.002 mVDC～-6 VDCの範囲になければなりません。
- 内部ステップ・アッテネータを使って基準信号パワーをあまり大きく変化させると、外部レベリング機能が有効に働かなくなる場合があります。

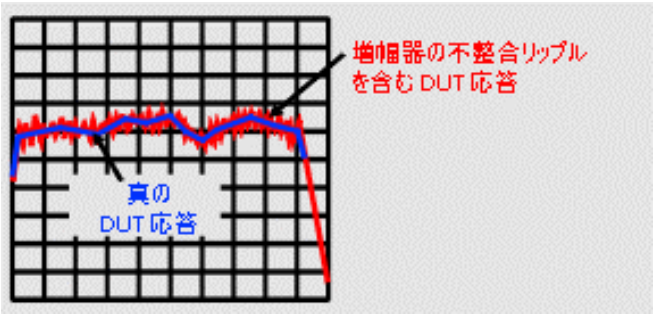
注記: 校正を実行する際には、測定中と同様に信号源レベリング・プロセスを働かせておきます。

下の図は、外部レベリングを使った2つのハイパワー測定校正を示します。1つめのサンプル構成は標準アナライザ用で、順方向でのみハイパワーが使用できます。2つ目のサンプル構成はオプション015(構成可能テスト・セット)を装備したアナライザ用で、順方向と逆方向の両方でハイパワーが使用できます。


構成2(標準アナライザ)での外部レベリング
ハイパワー(1方向)、
レスポンス校正
信号源
スイッチ / スプリッタ / レベラ
レシーバ入力:
最小 -35 dBm、最大 -6 dBm
損傷レベル +15 dBm
基準レシーバ
R1
R2
70 dB
A
B
測定レシーバ
R1
出力
1
2
テスト・ポートへの
最大入力
+30 dBm(1 W)
DUT
パワー・
スプリッタ
カスタム外部
レベリング回路
基準信号
カスタム・ブースタ増幅器、カプラ、
パワー・スプリッタ、アッテネータ



ハイパワー測定構成では、ブースタ増幅器の性能がDUTの測定確度に影響します。下の図に示すように、ブースタ増幅器にリップルがあるとDUT応答にもリップルが生じます。



ブースタ増幅器のリップルには下記のような原因があります。

- 低品質のブースタ増幅器
- 電源電圧のノイズ
- ブースタ増幅器の飽和

測定校正

本器に対して校正を行うと、系統誤差が除去され、測定がより正確になります。このチュートリアルでは、さまざまな構成に対してどのような校正が可能かを示してきました。下記の2つのタイプの校正が紹介されています。

- 2ポート校正: 最も正確。理想的にはDUTがアナライザに接続される点までの、測定セ

ットアップの系統誤差がすべて除去されます。

- スルー・レスポンス校正: 2ポート校正より確度は劣ります。周波数トラッキング誤差だけが除去されます。周波数トラッキング誤差は、チャネルの間の周波数応答の差に起因します。

注記: 校正の際にはすべてのハードウェアを接続しておく必要があります。例えば、ブースタ増幅器、ケーブル、アッテネータ、結合器など、DUT測定の際に使用するものはすべて含めるようにします。

次のトピックでは、最善の校正結果を得るための手法や考慮事項について説明します。

ダイナミック確度

本器のダイナミック確度応答を調べることにより、校正と測定の際のパワー・レベルを最適化することができます。ダイナミック確度とは、校正と測定に使用する信号レベルが異なることに伴う不確かさのことです。レシーバ・パワーが-10～-60 dBmの場合、校正パワー・レベルに関わらず本器はきわめて小さい不確かさを示します。レシーバにおける信号レベルがこの高確度レンジに入るように、校正と測定を実行します。信号レベルが低い場合、ノイズが影響します。レベルが高い場合、レシーバの圧縮が影響します。

校正パワー・レベル

増幅器の測定で、出力パワー・レベルが入力パワー・レベルよりも高い場合、どのパワー・レベルで校正を行えばよいでしょうか。一般的には、ノイズを減らすために、レシーバ圧縮が始まる前の最も高いパワーで校正を行うようにします。

1パワー・レベルと2パワー・レベルの校正

1パワー・レベルの校正

- レスポンス校正では1回の順方向掃引だけが実行されるため、この校正は1つのパワー・レベルでのみ実行できます。
- 2ポート校正も1つのパワー・レベルで実行できます。使用するパワー・レベルは、ダイナミック確度仕様(-10～-60 dBm)から決定します。

2パワー・レベルでの校正

- 2ポート校正では順方向と逆方向の掃引が行われるので、この校正は2つの異なるパワー・レベルで実行できます。
- DUTの測定時に用いられるパワー・レベルにさらに正確に合わせるため、本器のそれぞれのポートを別のパワー・レベルで校正することが可能です。

ハイパワーでの校正標準の使用

ハイパワー・レベルでの校正では、校正標準のパワー処理能力が問題になることがあります。

- 負荷はエネルギーを消費します。したがって、使用するパワー・レベルに対応した負荷を選ぶ必要があります。
- オープン、ショート、スルーの各標準は、エネルギーを消費しないため問題になりません。

注意: 校正中にオープンまたはショートを使用する場合、反射信号がテスト・ポートの最大入力レベル(+30 dBm)を超えないように入射パワー・レベルを設定する必要があります。

AGCループを持つ増幅器のテスト

増幅器の中にはAGC(自動利得制御)ループを持つものがあります。AGCループは、増幅器の出力パワーを一定に保つために、増幅器の入力の変動を考慮して利得を調整するものです。AGCループは、特にハイパワー・レベルで使用する場合、アナライザでの測定に問題を引き起こすおそれがあります。

AGCループを持つ増幅器の場合、本器の信号源出力をオフにすると問題が生じます。1つの周波数帯域だけを使用する場合、掃引と再トレースの間も本器の出力は一定です。しかし、掃引や再トレースの際に周波数帯域が切り替わる場合、本器の信号源出力はオフになります(ブランキング)。このような帯域交差は下記の周波数で発生します。

- 10 MHz
- 750 MHz
- 1500 MHz
- 3000 MHz
- 4500 MHz
- 6500 MHz

本器の信号源出力がオフになると、増幅器のAGCループは出力パワーを一定にするために利得を増やすことによってそれを補償しようとします。ブランキングが終わって掃引が再開されると、本器の出力信号が回復し、増幅器の入力に突然パワーが出現します。増幅器は利得を上げているため、AGCループの応答が十分高速でないと、一時的に大出力が生じ、下記のものを損傷または破壊するおそれがあります。

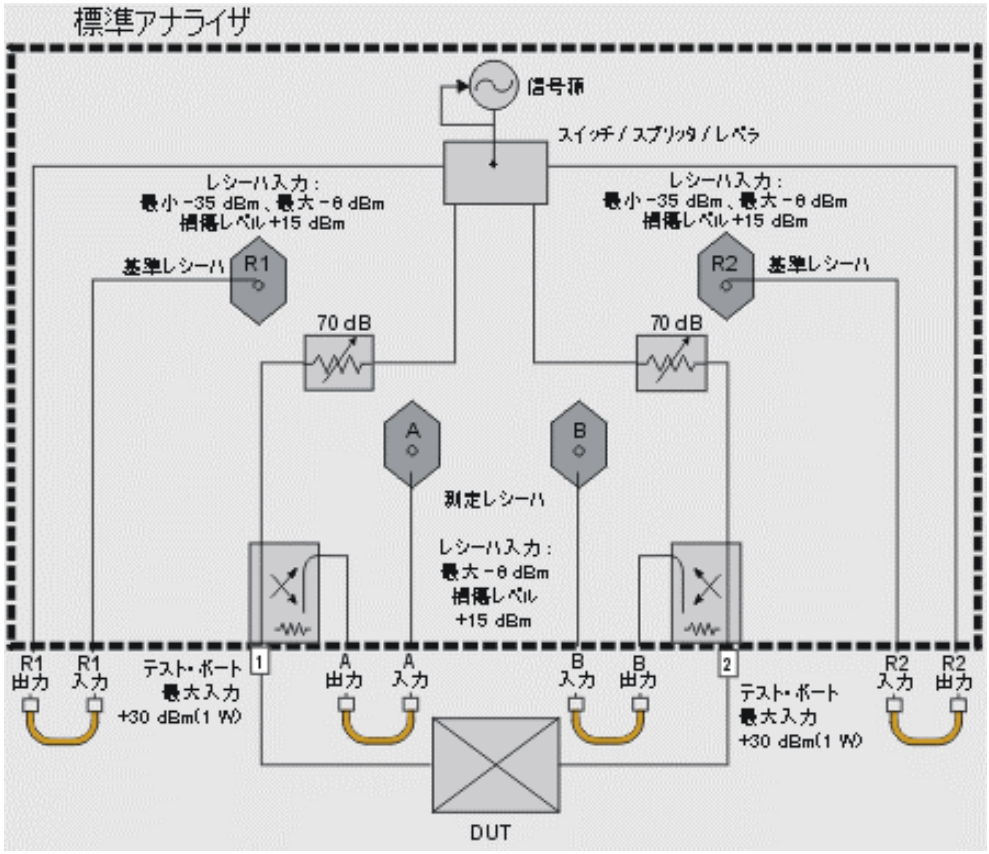
- 増幅器
- 本器のレシーバ

本器が複数の周波数帯域にまたがって掃引または再トレースを行う際には、ブランキングによってこのような損傷や破壊が発生するおそれがあることに注意してください。

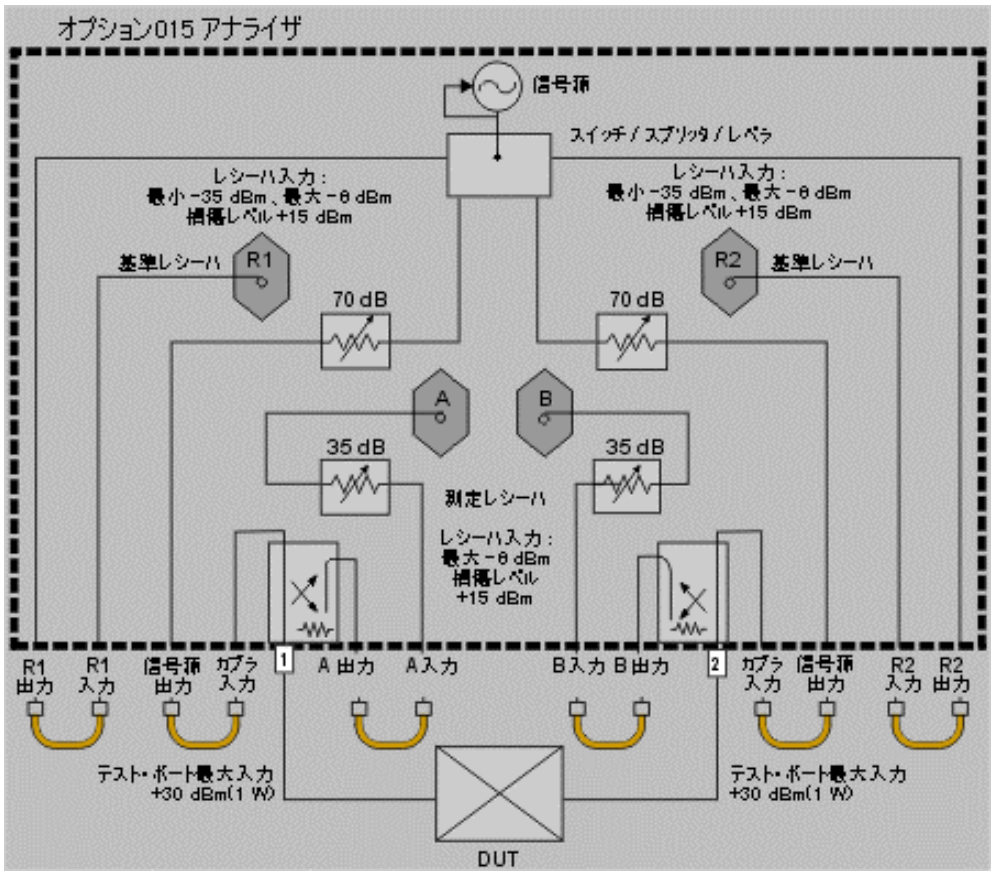
確度に関する考慮事項

- 被試験ハイパワー・デバイスは、温度が異なると非常に異なる応答を示す場合があります。テストは増幅器が目的の温度にあるときに行うべきです。
- 測定の不確かさと柔軟性を改善するには、ブースタ増幅器の前でなく後ろで入射信号を測定します。ブースタ増幅器の後ろで入射信号を測定することにより、信号源不整合の影響が打ち消されます。
- 本器のレシーバを過大な入力から保護するため、外部アッテネータまたは結合器を付加する場合は、以下の手順に従ってください。
- レシーバの入力信号の最大パワーを求めます。
- レシーバで必要なパワー・レベルを差し引きます。
- レシーバ入力パワーが必要なレベルになるように、適切な値のアッテネータまたは結合器(またはその両方)を選択します。
- IF帯域幅を狭めるか、アベレージングを使うと、確度は向上しますが、測定速度は低下します。

標準アナライザのブロック図



オプション015構成可能テスト・セットを装備したアナライザ



オプション015(構成可能テスト・セット)を装備したアナライザには、標準アナライザよりも高いパワー・レベルの測定を可能にする2つの機能が用意されています。

- ポート1とポート2の結合器とレシーバの間に、ステップ・アッテネータ(0～35 dB)が存在します。このステップ・アッテネータは、レシーバにとって最適なパワー・レベルに信号を調整する役割を果たします。信号源減衰を参照してください。
- ポート1と2のRFスティミュラス経路への直接アクセス(SOURCE OUTおよびCOUPLER INチャネル経由)。これにより次のことが可能になります。
- ブースタ増幅器のリバース・アイソレーションが測定確度に悪影響を与えないようにブースタ増幅器を配置。
- 本器の内部コンポーネントを保護するためのハイパワー・アイソレータの追加。

構成選択ガイド

このチュートリアルで紹介する下記の構成は、確度と能力がそれぞれ異なっています。
正しいネットワーク・アナライザ構成を選択するには、下記の要素を考慮する必要があります。

- 被試験デバイス
- 必要な測定
- 必要な確度

測定アプリケーションに最適なアナライザ構成を選択するには、以下のツールが役立ちます。

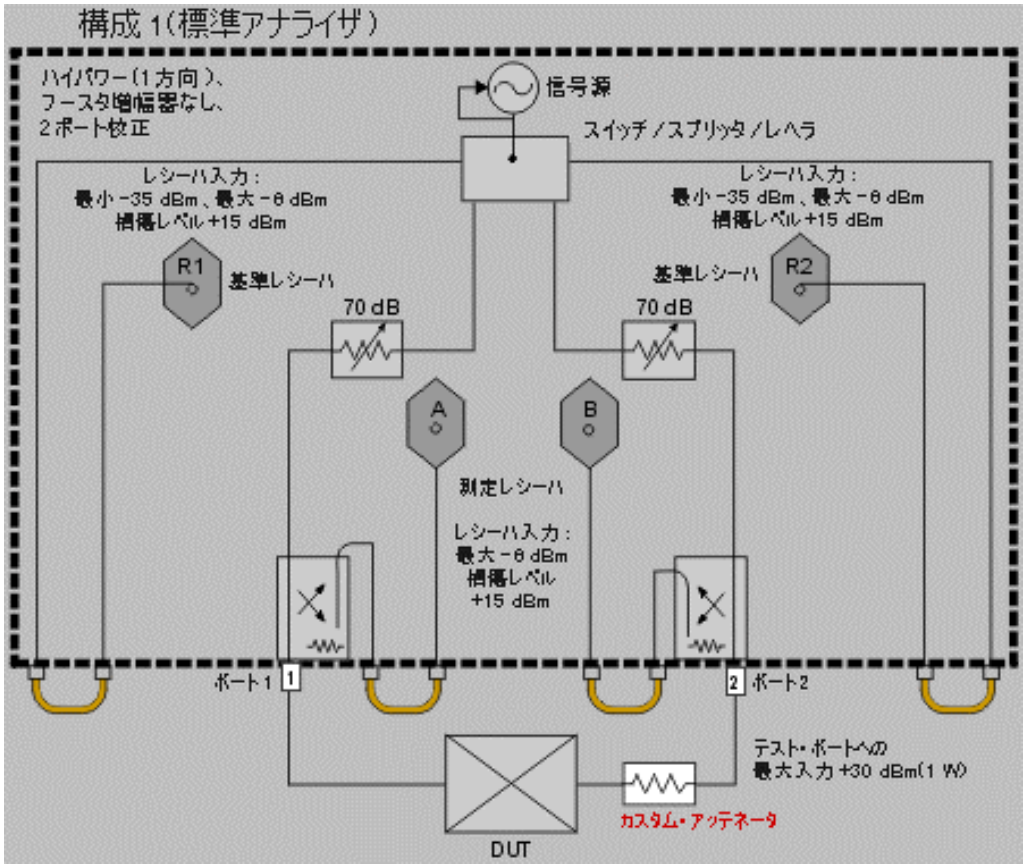
- ハイパワー・テスト構成のブロック図
- 下記に関する情報:
- 機器
- 校正選択
- 各構成の利点と制限
- 標準およびオプション015アナライザの各構成の特徴の表

このチュートリアルで紹介する下記の標準アナライザ用構成は､確度と能力がそれぞれ異なっています。

構成1(標準アナライザ)

この構成の性質:

- 1方向のみのハイパワー測定
- 本器のポート2レシーバに外部減衰を使用
- S_{11}、 S_{12}、S_{21}、 S_{22}
- 2ポート校正
- 外部コンポーネント1個
- ポート結合器でのパワー損失によりダイナミック・レンジが低下
- 下記の特性を持つ被試験デバイスに適合
- 高利得
- 高レベルのドライブ・パワーが不要(本器のパワー・レベル仕様以内)
- 通常はデバイスのリニア領域でテスト



機器

ブロック図に従って下記の接続を行います。

- 本器のポート1とアッテネータまたは結合器との間にDUT
- DUTと本器のポート2との間にアッテネータまたは結合器。DUTは利得が大きいため、大きい出力パワーからレシーバを保護する必要があります。

結合器を使用する場合、結合器のスルー・アームを特性インピーダンス負荷で終端します。結合器の結合アームは、入力信号の一部をポート2に送ります。20 dB結合器の場合、結合アームの信号は入力信号よりも20 dB小さくなります。

校正選択

- 2ポート
- スルー・レスポンス

ポート2の減衰のために、ポート2の未補正方向性が減衰値の2倍だけ低下します。これにより校正の安定度が制限され、S_{22}測定に大きなノイズが発生するおそれがあります。S_{22}測定が必要な場合、次のようにします。

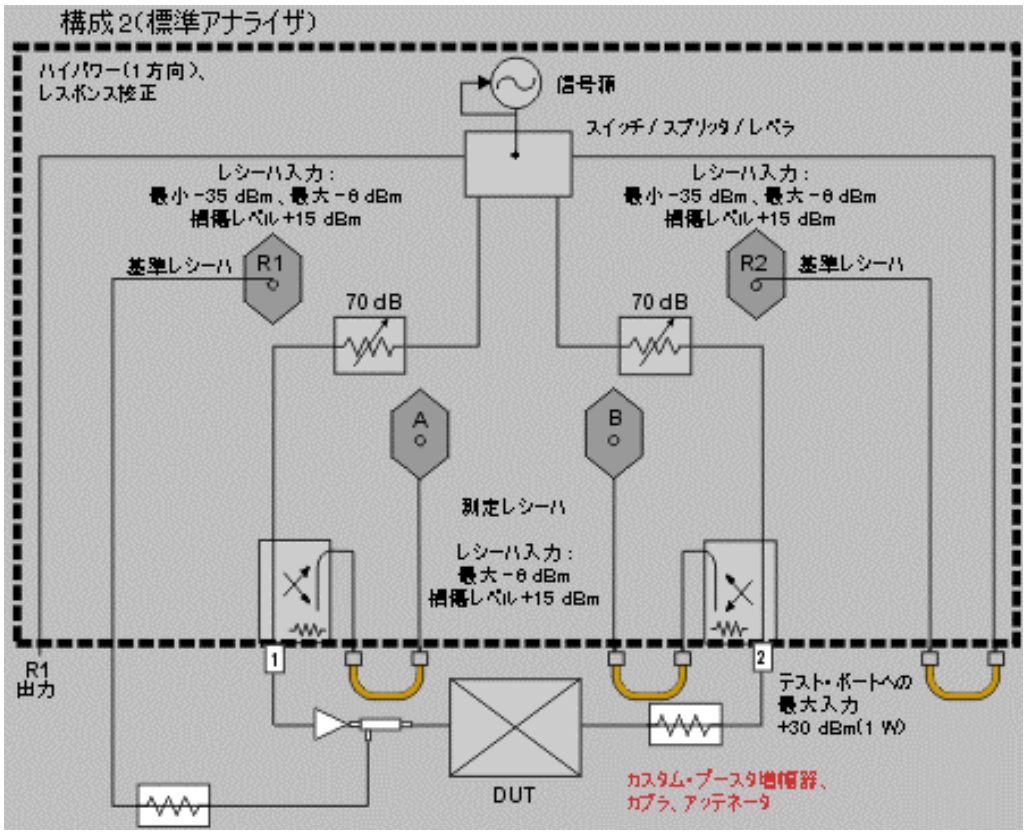
1. より大きいパワー・レベルで逆方向測定の校正を行います。
2. 順方向測定を行う際にはパワー・レベルを下げます。
3. 逆方向測定を行う際にはパワー・レベルを上げます。

構成2(標準アナライザ)

この構成の性質:

- 1方向のみのハイパワー測定
- 本器のポート2レシーバに外部減衰を使用
- S_{21}のみ(S_{11}はブースタ増幅器のために測定不可)
- スルー・レスポンス校正
- ポート2の結合器によるパワー損失のためダイナミック・レンジが低下

- テスト・セットにブースタ増幅器によるパワー損失なし
- 下記の特性を持つ被試験デバイスに適合
- 高利得
- 高レベルのドライブ・パワーのためにブースタ増幅器が必要



機器

ブロック図に従って下記の接続を行います。

- 本器のポート1にブースタ増幅器
- ブースタ増幅器とDUTとの間に結合器
- 結合器の結合アームにアッテネータ
- ブースタ増幅器を通った信号源信号の結合アーム部分をR1 INチャネルへ。この信号は比例法で用いられる基準信号となります。
- 結合器とアッテネータまたは結合器との間にDUT
- DUTの出力と本器のテスト・ポートとの間にアッテネータまたは結合器。DUTは利得が大きいため、大きい出力パワーからレシーバを保護する必要があります。

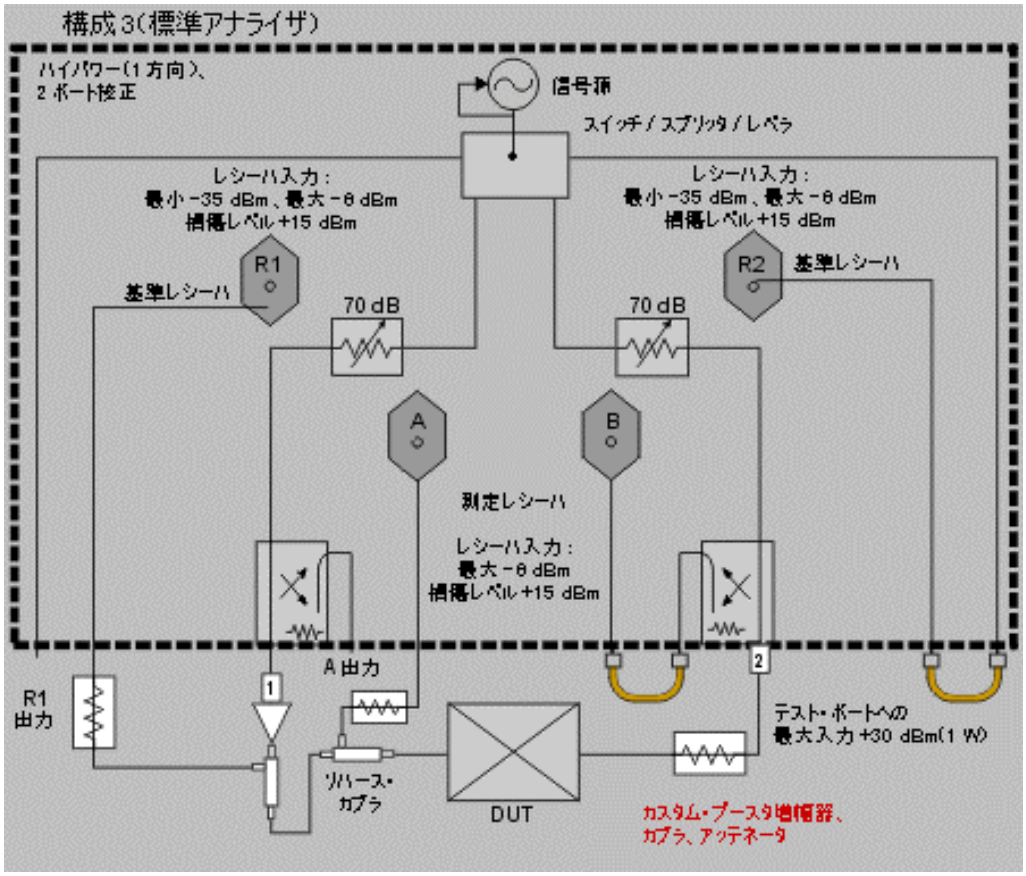
結合器を使用する場合、結合器のスルー・アームを特性インピーダンス負荷で終端します。結合器の結合アームは、入力信号の一部をポート2に送ります。20 dB結合器の場合、結合アームの信号は入力信号よりも20 dB小さくなります。

構成3(標準アナライザ)

この構成の性質:

- 1方向のみのハイパワー測定
- 本器のポート2レシーバに外部減衰を使用
- S_{11}、S_{21}
- 2ポートおよびスルー・レスポンス校正
- 外部コンポーネント6個

- ポート2の結合器によるパワー損失のためダイナミック・レンジが低下
- テスト・セットにブースタ増幅器によるパワー損失なし
- 下記の特性を持つ被試験デバイスに適合
- 高利得
- 高レベルのドライブ・パワーのためにブースタ増幅器が必要



機器

ブロック図に従って下記の接続を行います。

- 本器のポート1にブースタ増幅器
- ブースタ増幅器の出力に結合器
- 結合器の結合アームにアッテネータ
- ブースタ増幅器を通った信号源信号の結合アーム部分をR1 INチャネルへ。この信号は比例法で用いられる基準信号となります。
- 順方向結合器とDUTとの間に逆方向結合器
- 逆方向結合器の結合アームをA INチャネルへ。これは順方向反射測定用です。
- 逆方向結合器とアッテネータまたは結合器との間にDUT
- DUTの出力と本器のテスト・ポートとの間にアッテネータまたは結合器。DUTは利得が大きいため、大きい出力パワーからレシーバを保護する必要があります。

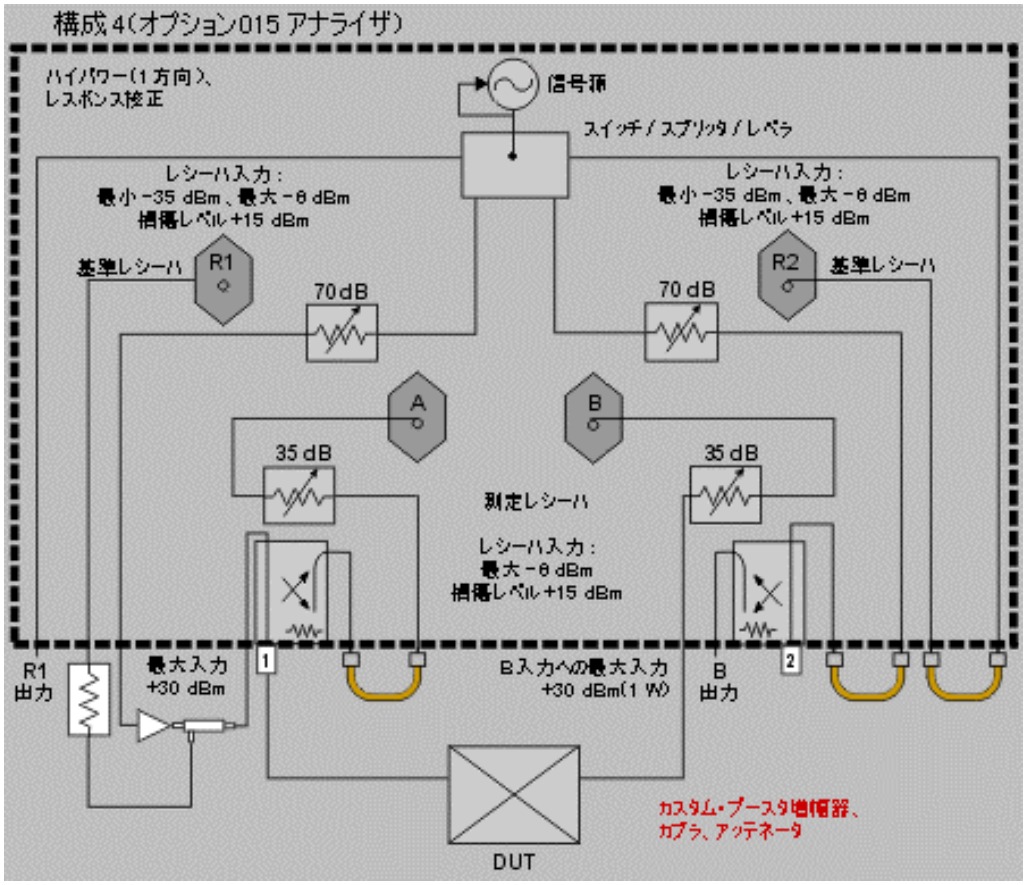
結合器を使用する場合、結合器のスルー・アームを特性インピーダンス負荷で終端します。結合器の結合アームは、入力信号の一部をポート2に送ります。20 dB結合器の場合、結合アームの信号は入力信号よりも20 dB小さくなります。

このチュートリアルで紹介する下記の構成は、確度と能力がそれぞれ異なっています。

構成4(オプション015アナライザ)

この構成の性質:

- 1方向のみのハイパワー測定
- 本器のBレシーバを使用
- S_{11}、S_{21}
- スルー・レスポンス校正
- 外部コンポーネント3個
- B INチャネルの使用により、最大のダイナミック・レンジを実現
- ポート1の結合器によるパワー損失のためダイナミック・レンジが低下
- 下記の特性を持つ被試験デバイスに適合
- 高利得
- 高レベルのドライブ・パワーのためにブースタ増幅器が必要



機器

ブロック図に従って下記の接続を行います。

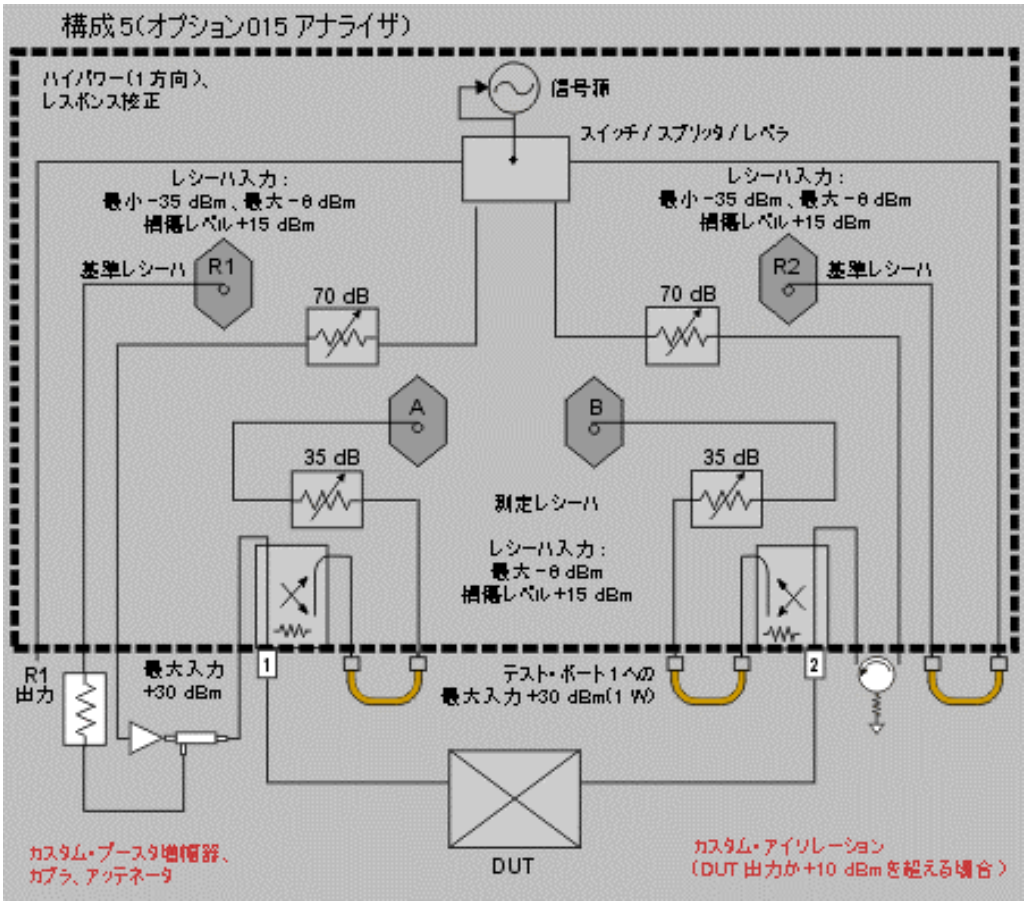
- ポート1のSOURCE OUTチャネルにブースタ増幅器
- ブースタ増幅器とポート1のCOUPLER INチャネルとの間に結合器
- 結合器の結合アームにアッテネータ
- ブースタ増幅器を通った信号源信号の結合アーム部分をR1 INチャネルへ。この信号は比例法で用いられる基準信号となります。
- 本器のポート1とB INチャネルとの間にDUT。B INチャネルを使うことにより、ポート2の結合器によるパワー損失がないので、最大のダイナミック・レンジが得られます。

これらのデバイスは利得が大きいため、大きい出力パワーからレシーバを保護する必要があります。本器の内部ステップ・アッテネータ(0～35 dB)によってBレシーバを保護します。

構成5(オプション015アナライザ)

この構成の性質:

- 1方向のみのハイパワー測定
- 本器の内部コンポーネントを過大なパワーから保護するためにアイソレータを使用
- S_{11}、S_{21}
- スルー・レスポンスおよび2ポート校正
- 外部機器4個
- ポートの結合器によるパワー損失のためにダイナミック・レンジが低下
- アイソレータによりスイッチと信号源を保護
- 下記の特性を持つ被試験デバイスに適合
- 高利得
- 高レベルのドライブ・パワーのためにブースタ増幅器が必要



機器

ブロック図に従って下記の接続を行います。

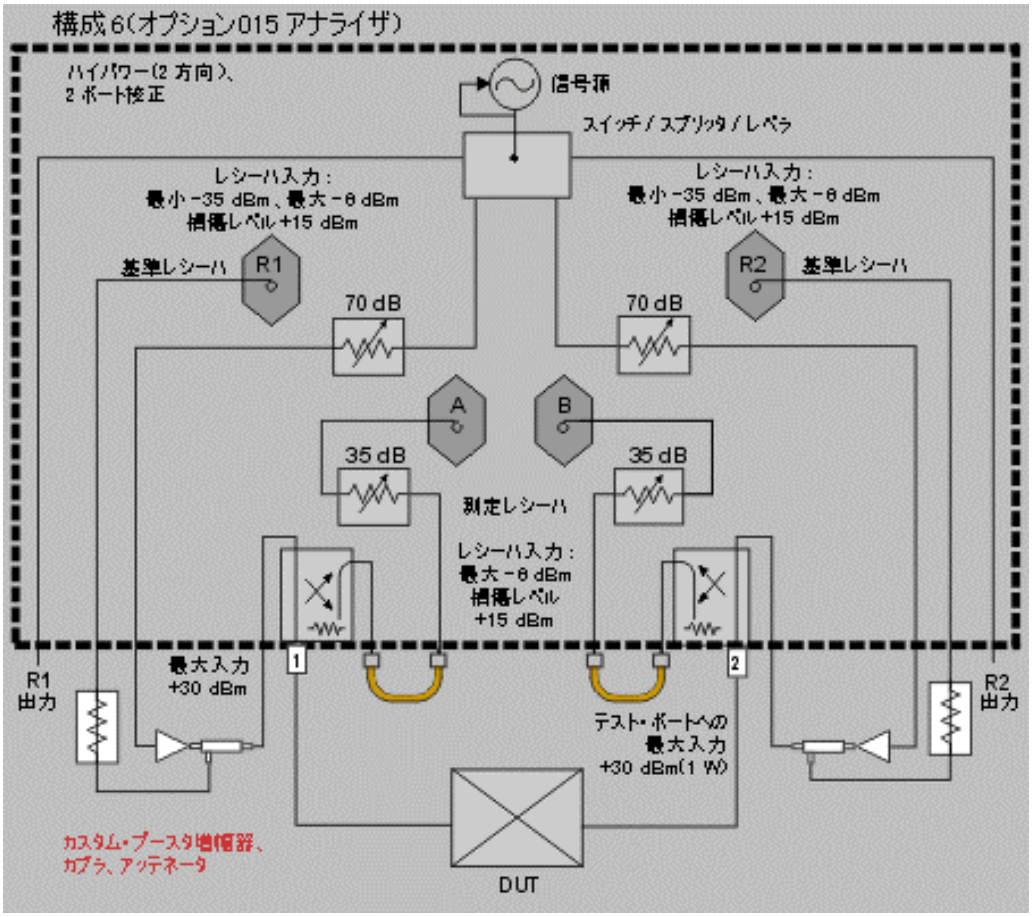
- ポート1のSOURCE OUTチャネルにブースタ増幅器
- ブースタ増幅器とポート1のCOUPLER INチャネルとの間に結合器
- 結合器の結合アームにアッテネータ
- ブースタ増幅器を通った信号源信号の結合アーム部分をR1 INチャネルへ。この信号は比例法で用いられる基準信号となります。
- 本器のポート1とポート2との間にDUT
- ポート2のSOURCE OUTチャネルとCOUPLER INチャネルとの間にアイソレータ

これらのデバイスは利得が大きいため、大きい出力パワーからレシーバを保護する必要があります。本器の内部ステップ・アッテネータ(0～35 dB)によってBレシーバを保護します。

構成6(オプション015アナライザ)

この構成の性質:

- 2方向(順方向、逆方向)のハイパワー測定
- S_{11}、S_{12}、S_{21}、S_{22}
- スルー・レスポンスおよび2ポート校正
- 外部コンポーネント6個
- ポートの結合器によるパワー損失のためにダイナミック・レンジが低下
- 下記の特性を持つ被試験デバイスに適合
- 高利得
- 高レベルのドライブ・パワーのためにブースタ増幅器が必要



機器

ブロック図に従って下記の接続を行います。

- ポート1のSOURCE OUTチャネルにブースタ増幅器
- ブースタ増幅器とポート1のCOUPLER INチャネルとの間に結合器
- 結合器の結合アームにアッテネータ
- ブースタ増幅器を通った信号源信号の結合アーム部分をR1 INチャネルへ。この信号は比例法で用いられる基準信号となります。
- 本器のポート1とポート2との間にDUT
- ポート2のSOURCE OUTチャネルにブースタ増幅器
- ブースタ増幅器とポート2のCOUPLER INチャネルとの間に結合器
- 結合器の結合アームにアッテネータ

- ブースタ増幅器を通った信号源信号の結合アーム部分をR2 INチャネルへ。この信号は比例法で用いられる基準信号となります。

これらのデバイスは利得が大きいため、大きい出力パワーからレシーバを保護する必要があります。本器の内部ステップ・アッテネータ(0～35 dB)によってAおよびBレシーバを保護します。

ハイパワー・デバイスを測定する方法

1. DUTと測定要件に合わせて正しいアナライザ構成を選びます。まだDUTは接続しません。
2. 本器を初期設定します。
3. 被試験デバイスが増幅器の場合、本器の信号源出力の調整が必要な場合があります。
   - 本器の信号源出力を、増幅器の出力応答の線形領域(通常は1 dB圧縮ポイントより10 dB下)に設定します。
   - 必要なら外部アッテネータを選択して、本器のポート2のレシーバに圧縮や損傷が生じないように増幅器の出力パワーを減衰させます。
4. 目的の測定を選択します。
5. DUTを測定します。
6. DUTに対する設定を選択します。
7. DUTを取り除き、測定校正を実行します。DUTの測定に使用するブースタ増幅器、ケーブル、アッテネータ、結合器などは必ず含めます。
8. 機器ステートをメモリに保存します。
9. DUTを再接続します。
10. 測定結果の表示が見やすくなるようにスケールを調整し、マーカを使って測定結果を読み取ります。
11. データをプリントするか、ディスクに保存します。

AM-PM変換

概要　　概念　　AM-PM変換の測定方法

増幅器のAM-PM変換とは、システムに固有の振幅変動(AM)によってもたらされる不要な位相偏移(PM)の量を測定するものです。

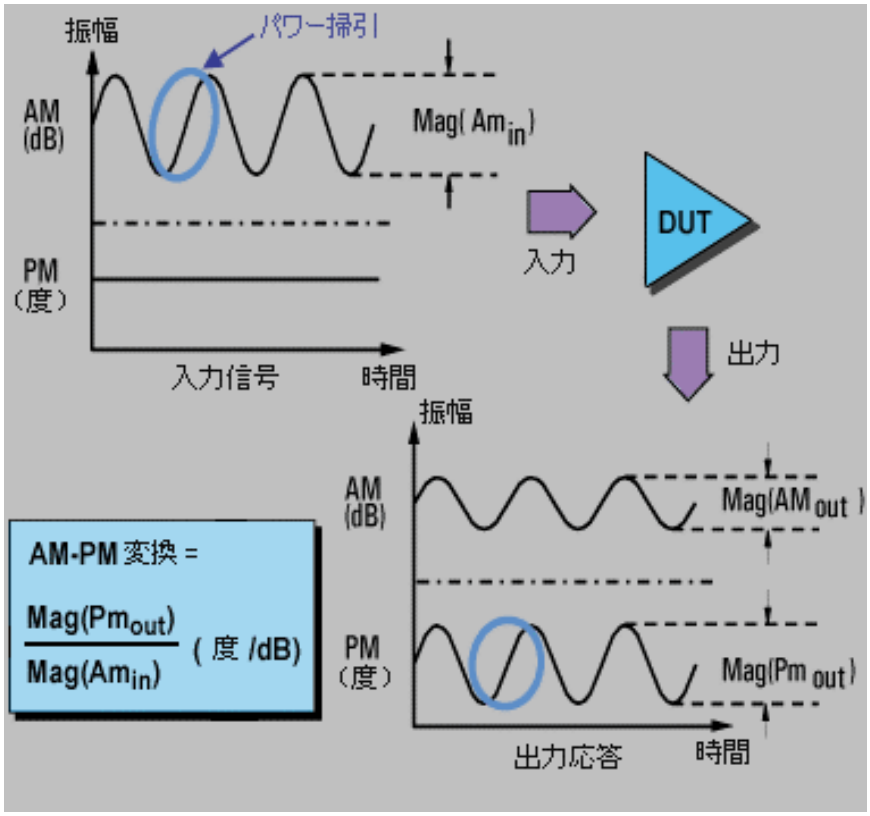
ボタンをクリックすると下記の項目について知ることができます。

AM-PM変換とは何か

AM-PM変換とは、システムの振幅変動(AM)から生じる不要な位相偏移(PM)の量を測定するものです。例えば、通信システムの不要な位相偏移(PM)は以下のような原因で生じます。

- 意図しない振幅変動(AM)
- 電源のリップル
- 熱ドリフト
- マルチパス・フェージング
- 信号振幅の意図的な変調
- QAM
- バースト変調

AM-PM変換は通常、増幅器の入力に対するパワー掃引の1 dBの増加に対する(1 dB利得圧縮ポイントでの)出力位相の変化と定義されます。単位は°/dBです。理想的な増幅器では、位相応答と入力信号のパワー・レベルとは無関係です。



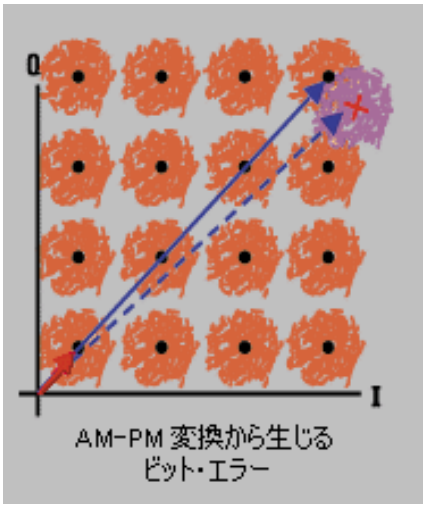
AM-PM変換を測定する理由

AM-PM変換が重要なのは、位相(角度)変調が用いられる以下のようなシステムの場合です。

- FM
- QPSK
- 16QAM

このパラメータが重要なのは、不要な位相偏移(PM)があると、アナログ信号が劣化したり、ディジタル通信システムのビット・エラー・レート(BER)が増加したりするからです。ディジタル通信システムのBERを測定するのは容易ですが、この測定だけではビット・エラーの根本的な原因はわかりません。AM-PM変換はBERの基本的な要因の1つなので、このパラメータを定量化することが通信システムでは重要です。

AM-PM変換からビット・エラーが生じる理由を以下に説明します。下のI/Qダイアグラムを参照してください。



- 望ましいステート変化は、小さい実線のベクトルから大きい実線のベクトルへの変化です。
- AM-PM変換があると、大きいベクトルが実際には点線で示したようになることがあります。これは入力パワー・レベルの変化から生じる位相シフトに原因があります。
- 図に示す64QAM信号(1象限のみを表示)の場合、各ステートの周りの雑音円が実際には重なっており、統計的にはある程度のビット・エラーが発生する可能性があります。

確度に関する考慮事項

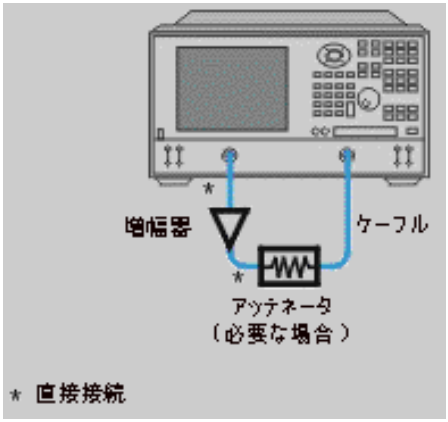
AM-PM変換の測定にこの方法を使う場合、変調周波数はだいたい掃引時間の逆数になります。大部分のネットワーク・アナライザでは、最高速のパワー掃引を使っても変調周波数はかなり低くなります(通常10 Hz以下)。このため、掃引の実行中に多少の温度変化が生じるおそれがあります。特に、非パッケージ・デバイスのように増幅器の熱質量が小さい場合にはこの影響が大きくなります。増幅器のノンリニア動作が温度変化に対してきわめて敏感な場合は、この方法で測定した結果が多少異なる可能性があります(PNAシリーズ・アナライザでは1 ms未満の時間でパワー掃引が可能です)。

- 増幅器は温度が異なると非常に異なる応答を示す場合があります。テストは増幅器が目的の動作温度にあるときに行うべきです。
- 場合によっては、増幅器の出力を減衰させなければならないことがあります。出力パワーが大きすぎると、下記のような影響が生じるおそれがあります。
  - 本器のレシーバが損傷される
  - 本器のレシーバの入力圧縮レベルを超過し、不正確な測定結果が生じる

- 増幅器の出力を減衰させるには、次の方法があります。
  - アッテネータ
  - カップラ
- アッテネータとカップラはテスト・システムの一部なので、校正の際にその周波数応答の影響を考慮する必要があります。これらの影響は正しいエラー訂正手法により低減することができます。

  周波数応答は、AM-PM変換測定セットアップにおける支配的な誤差です。スルー・レスポンス測定校正を行うことにより、この誤差を大幅に減らすことができます。さらに確度を上げるには、2ポート測定校正を実行します。

1. アナライザを初期設定します。
2. パワー掃引モードのS21測定を選択します。
3. パワー掃引のスタート/ストップ・パワー・レベルを入力します。スタート・パワー・レベルは、増幅器のリニア領域になければなりません(通常は1 dB圧縮ポイントより10 dB下)。ストップ・パワーは増幅器応答の圧縮領域になければなりません。
4. 必要なら外部アッテネータを選択して、本器のポート2でレシーバの圧縮や損傷が生じないように増幅器の出力パワーを減衰させます。
5. 増幅器を下の図のように接続し、DCバイアスを供給します。



6. 被試験増幅器に対するアナライザ設定を選択して、選択した周波数で掃引パワー利得圧縮測定を実行します。利得圧縮を参照してください。
7. 増幅器を取り外し、測定校正を実行します。増幅器の測定の際にアッテネータやケーブルを使用する場合、それらも必ず校正セットアップに含めます。
8. 機器ステートをメモリに保存します。
9. 増幅器を再接続します。

10. 基準マーカを使って、増幅器の入力パワーを1 dB利得圧縮ポイントに設定します。マーカをもう1個選択し、基準マーカより1 dB下に応答が来るようにスティミュラス値を調整します。
11. S_{21}測定を対数振幅フォーマットから位相フォーマットに変更します(新たに校正を実行する必要はありません)。
12. マーカの間の位相変化を求めます。この値が、1 dB利得圧縮ポイントでのAM-PM変換係数になります。
13. データをプリントするか、ディスクに保存します。

利得圧縮

概要　　概念　　利得圧縮の測定方法

利得圧縮とは、増幅器の出力に歪みを生じる入力パワーのレベルを測定するものです。増幅器の利得圧縮を求めるためには3つの測定が用いられます。

1) 掃引周波数利得圧縮測定は、1 dBの利得圧縮が初めて発生する周波数ポイントを探します。

2) 掃引パワー利得圧縮測定は、特定の周波数ポイント(測定1で求めたもの)でパワー・ランプを増幅器に入力して、利得が1 dB低下する入力パワーを求めます。

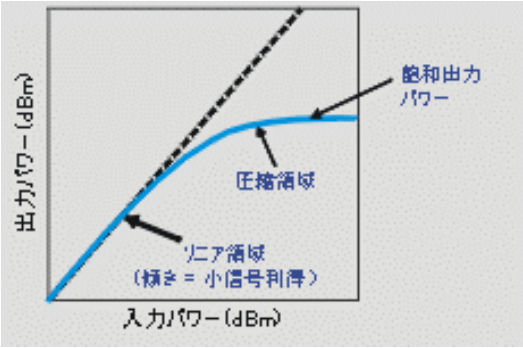
3) 絶対パワー測定は、圧縮点での絶対パワー出力(dBm単位)を求めます。

ボタンをクリックすると下記の事項について知ることができます。

利得圧縮とは何か

利得圧縮とは、増幅器の入力パワーを上げていったときに、あるレベルで増幅器の利得が低下し、出力パワーが直線的に増加しなくなる現象です。

本器にはパワー掃引と周波数掃引の機能が備わっています。パワー掃引は、増幅器の非線形動作を調べるために役立ちます。以下の説明については、下の図(1つの周波数における増幅器の出力パワー対入力パワーのプロット)を参照してください。



- ある特定のパワー・レベルにおける増幅器利得は、曲線の傾きに一致します。
- 増幅器の線形領域では、利得が一定でパワー・レベルに依存しません。この領域での利得を一般に「小信号利得」と呼びます。
- 入力パワーを上げていくと、ある点で増幅器の利得が減少し、圧縮領域に入ります。
- 増幅器圧縮に関する最も一般的な測定は、1 dB圧縮ポイントです。これは、増幅器利得が(小信号利得に対して)1 dB減少する入力パワー(あるいは場合によっては出力パワー)のことです。

なぜ利得圧縮を測定するのか

- 増幅器に正弦波を入力しても、圧縮領域では出力は正弦波になりません。増幅器出力の一部が、入力信号の基本波周波数以外の高調波として現れます。
- さらに入力パワーを上げると、増幅器は飽和し、出力パワーが一定になります。この点では、入力パワーを上げても出力パワーが変化しません。
- 場合によっては(TWT増幅器の場合など)、飽和点からさらに入力を上げると出力パワーが減少することさえあります。この場合、増幅器の利得は負になります。
- 増幅器動作は利得を得ることが目的なので、利得圧縮を生じる入力信号の限界を知っておくことは重要です。

確度に関する考慮事項

本器は増幅器を飽和させるのに十分なパワーを供給する必要があります。本器の信号源が供給できるよりも大きい入力パワー・レベルが必要な場合、プリアンプを使って被試験増幅器の前でパワー・レベルを増幅します(ハイパワー・コンポーネント測定を参照)。プリアンプを使う場合、測定確度を上げるには下記の方法を使います。

- プリアンプの出力に結合器をつなぎ、増幅された入力信号の一部を本器の基準信号に使用します。この構成により、プリアンプの周波数応答とドリフト誤差が(比例法により)測定から除去されます。
- プリアンプ、結合器、アッテネータをテスト・セットアップに含めて、スルー・レスポンス校正を実行します。

必要なら、増幅器の出力パワーを適当に減衰します。出力パワーが大きすぎると下記のような結果が生じるおそれがあります。

- 本器のレシーバが損傷されます。
- 本器のレシーバの入力圧縮レベルを超過します。

増幅器の出力パワーの減衰には、下記のような手段が使用できます。

- アッテネータ
- 結合器

アッテネータと結合器はテスト・システムの一部なので、校正の際にその周波数応答の影響を考慮する必要があります。適切な誤差補正技法により、これらの影響を減らすことができます。

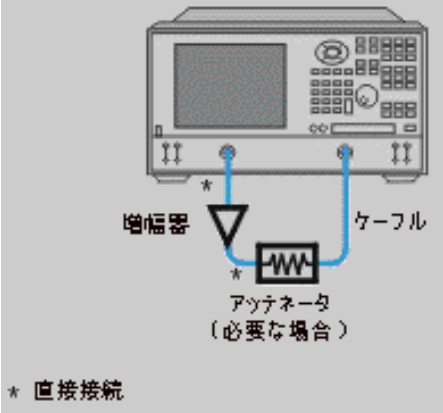
- 利得圧縮測定セットアップにおける支配的な誤差は周波数応答です。スルー・レスポンス測定校正により、この誤差を大きく減らすことができます。
- 増幅器は温度が異なると非常に異なる応答を示す場合があります。テストは増幅器が目的の動作温度にあるときに行うべきです。
- IF帯域幅を狭めるか、測定アベレージングを使うと、確度は向上しますが、測定速度は低下します。

この手順は、増幅器利得圧縮を求めるための3つの測定の実行方法を示します。

掃引周波数利得圧縮測定

掃引周波数利得圧縮測定では、1 dBの利得圧縮が初めて発生する周波数ポイントを探します。

1. 本器を初期設定します。
2. S_{21}測定パラメータを選択します。
3. 本器の信号源出力を、増幅器の出力応答の線形領域(通常は1 dB圧縮ポイントより10 dB下)に設定します。
4. 必要なら外部アッテネータを選択して、本器のポート2のレシーバに圧縮や損傷が生じないように増幅器の出力パワーを減衰させます。
5. 下の図のように増幅器を接続し、DCバイアスを供給します。
6. 被試験増幅器に対するアナライザ設定を選択します。ノイズの影響を減らすには、IF帯域幅を小さくします。



7. 増幅器を取り除き、スルー・レスポンス校正を実行します。増幅器の測定に使用するアッテネータやケーブルは必ず校正セットアップに含めます。
8. 機器ステートをメモリに保存します。
9. 増幅器を再接続します。
10. スパンの中央付近にマーカを配置します。
11. 本器のスケールを1 dB/divに設定します。
12. トレースをメモリに記憶し、Data/Memを表示します。
13. 信号源出力を少しずつ上げていき、トレースのどこかの周波数で1 dBの利得減少が最初に見られたところで止めます。
14. 1 dBの利得減少が最初に発生した周波数をマーカを使って読み取ります。このあとの測定で使用するためにこの周波数を記録しておきます。
15. データをプリントするか、ディスクに保存します。

掃引パワー利得圧縮測定

掃引パワー利得圧縮測定は、特定の周波数(前の測定のステップ13で求めたもの)でパワー・ランプを増幅器に入力して、利得が1 dB低下する入力パワーを求めます。

1. 前の掃引周波数利得圧縮測定がまだ済んでいなければ、ここで実行します。
2. パワー掃引モードのS_{21}測定を準備します。下記の設定を使用します。
   - CW周波数を、前の掃引周波数利得圧縮測定のステップ13で記録した周波数に設定します。
   - 掃引のスタート/ストップ・パワー・レベルを入力します。スタート・パワーは増幅器応答の線形領域(通常は1 dB圧縮ポイントより10 dB下)内の値にします。ストップ・パワーは増幅器応答の圧縮領域内の値にします。

3. スケールを1 dB/divに調整します。
4. マーカ(基準マーカを含む)を使って、利得が1 dB減少する入力パワーを読み取ります。
5. データをプリントするか、ディスクに保存します。

絶対出力パワー測定

1. 絶対パワー測定は、圧縮時の増幅器の絶対出力パワー(dBm単位)を測定します。
   - 絶対パワー測定を選択します。前の図のテスト・セットアップを使用する場合は、B入力を選択します。下記の設定を使用します。
   - 前の掃引パワー利得圧縮測定で使用したCW周波数をそのまま使用します。
2. 利得が1 dB減少する入力パワー・レベル(前の測定のステップ4で求めたもの)にマーカを配置します。
3. 測定結果の表示が見やすくなるようにスケールを調整します。
4. マーカ値を読み取って、利得が1 dB減少する増幅器の絶対出力パワー(dBm単位)を求めます。
5. データをプリントするか、ディスクに保存します。

注記: 絶対パワーには測定校正が適用されません。したがって、アナライザ外部に減衰が存在する場合、手動で補正する必要があります。

リニア位相からの偏移

概要　　概念　　測定方法

リニア位相からの偏移とは、位相歪みを測定するものです。本器の電気遅延機能を使って、位相シフトのリニア部分を測定結果から除去します。これにより、位相シフトのノンリニア部分(リニア位相からの偏移)の高分解能表示が得られます。

ボタンをクリックすると下記の事項について知ることができます。

リニア位相シフトとは何か

位相シフトが発生する理由は、入射信号の波長が高くなるに従って、デバイスの電気長を占める波長が短くなるからです。リニア位相シフトは、デバイスの位相応答が周波数に直線的に比例する場合に発生します。この理想的なリニア位相シフトを本器に表示すると、位相対周波数の測定トレースは直線になります。直線の傾きは、デバイスの電気長に比例します。リニア位相シフトは(フラットな振幅応答とともに)、信号を歪みなく伝送するための必要条件です。

リニア位相からの偏移とは何か

実際の電気/電子デバイスの多くは周波数によって遅延が異なり、ノンリニア位相シフト(複数の周波数成分からなる信号の歪み)が生じます。リニア位相からの偏移 を測定することにより、ノンリニア位相シフトを定量化することができます。

位相歪みの原因となるのはリニア位相からの偏移だけなので、位相応答のリニア部分を測定結果から除去する必要があります。このためには、本器の電気長機能を使って、被試験デバイスの電気長を数学的に除去します。これによって残るのがリニア位相からの偏移、すなわち位相歪みです。

なぜリニア位相からの偏移を測定するのか

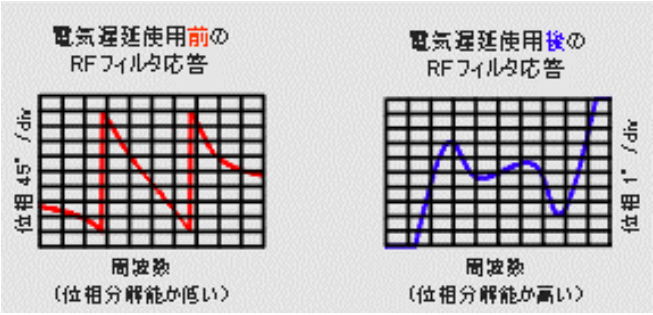
リニア位相からの偏移を測定することにより、下記の結果が得られます。

- データが秒単位(群遅延)でなく位相単位で表示されます。変調信号を通すデバイスの場合、位相単位の表示のほうが便利です。
- 群遅延測定よりもノイズの少ない測定結果が得られます。

電気長の使用

電気遅延機能は、従来のアナライザで用いられていた機械的な「ストレッチャ」の機能を電子的に実現したものです。この機能は下記のことを実現します。

- 基準信号経路に対して、仮想的な可変長無損失伝送ラインを追加または除去します。
- 被試験デバイスの電気長を補正します。
- 本器に表示される測定トレースを平坦化します。これにより、トレースを高分解能で表示し、位相ノンリニアリティの詳細を見ることができます。
- 被試験デバイスのリニア位相からの偏移を見るのに便利です。下の図を参照してください。

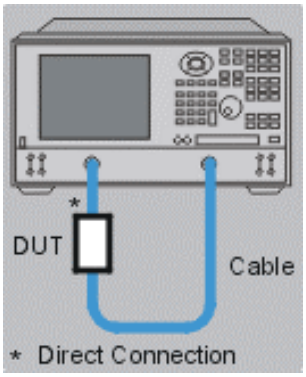


確度に関する考慮事項

テスト・セットアップの周波数応答は、リニア位相からの偏移測定における支配的な誤差です。スルー・レスポンス測定校正を行うことにより、この誤差を大幅に減らすことができます。さらに確度を上げるには、2ポート測定校正を実行します。

リニア位相からの偏移の測定方法:

1. アナライザを初期設定します。
2. 被試験デバイスが増幅器の場合、本器の信号源出力の調整が必要な場合があります。
   - 本器の信号源出力を、増幅器の出力応答の線形領域(通常は1 dB圧縮ポイントより10 dB下)に設定します。
   - 必要なら外部アッテネータを選択して、本器のポート2のレシーバに圧縮や損傷が生じないように増幅器の出力パワーを減衰させます。
3. 下の図のように被試験デバイスを接続します。



4. S21測定を選択します。
5. 被試験デバイスに対する設定を選択します。下記の設定を使用します。
   - フォーマット: 位相
   - スケール: オートスケール
6. デバイスを取り除き、校正を実行します。
7. デバイスを再接続します。
8. 測定結果の表示が見やすくなるようにスケールを調整します。
9. 電気遅延機能をオンにして、位相トレースを自動的に平坦化します。必要なら電気長を手動で微調整して、位相トレースのフラットネスを調整します。
10. マーカを使って、リニア位相からの偏移 の最大p-p値を測定します。
11. データをプリントするか、ディスクに保存します。

リバース・アイソレーション

概要　　概念　　リバース・アイソレーションの測定方法

リバース・アイソレーションとは、増幅器の逆方向伝送応答(出力から入力へ)を測定するものです。

ボタンをクリックすると下記事項についての説明が表示されます。

リバース・アイソレーションとは何か

リバース・アイソレーションとは、デバイス出力に印加された信号がどの程度よく入力から分離されるかを測定するものです。

リバース・アイソレーションの測定は順方向利得の測定と似ていますが、以下の点が異なります。

- スティミュラス信号が増幅器の出力ポートに適用されます。
- 応答が増幅器の入力ポートで測定されます。

対応するSパラメータはS12です。

なぜリバース・アイソレーションを測定するのか

理想的な増幅器のリバース・アイソレーションは無限大であり、出力から入力に信号は全く伝わりません。しかし、実際には反射信号が逆方向に増幅器内を通過します。この不要な逆方向伝送のために、必要な順方向の基本波信号に反射信号が干渉するおそれがあります。このため、リバース・アイソレーションの定量化は重要です。

確度に関する考慮事項

増幅器は逆方向の損失が大きい場合が多いので、フォワード伝送測定の場合と異なり、ポート2のレシーバを保護するための減衰は通常は不要です。減衰を取り除くことで、下記の結果が生じます。

- ダイナミック・レンジを広げることで、測定確度を上げることができます。
- 最高の測定確度を得るために新たな校正が必要になります。

RF信号源出力を増加させることにより、ダイナミック・レンジと確度を向上させることができます。

注記: 減衰を取り除き、RF信号源出力を増やした場合、順方向掃引を実行すると本器のポート2のレシーバが損傷されるおそれがあります。ポート2のレシーバに圧縮や損傷を引き起こさない程度に順方向パワーが小さい場合以外は、順方向掃引を使用しないか、2ポート校正を使用してください。

被試験増幅器のアイソレーションが非常に大きい場合､伝送された信号のレベルがレシーバのノイズ・フロアまたはクロストーク・レベルに近くなることがあります。ノイズ・フロアを下げるには次のようにします。

- 測定アベレージングを使用するか、アベレージング係数を増やします。
- 本器のIF帯域幅を狭めます。

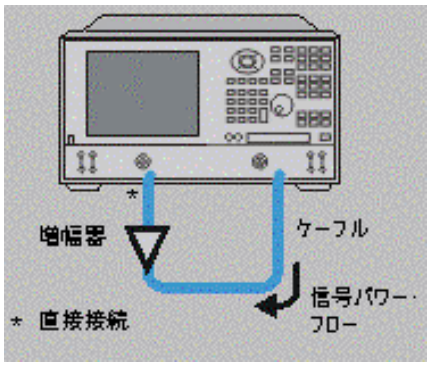
注記: IF帯域幅を狭めるかアベレージングを使用した場合、測定のダイナミック・レンジと確度は向上しますが、測定速度は低下します。

- クロストーク・レベルが測定確度に影響する場合、レスポンス/アイソレーション校正を実行してクロストーク誤差項を小さくします。校正のアイソレーション部分を実行する際には、測定に使用するのと同じアベレージング係数とIF帯域幅を使用すること

が重要です。

- テスト・セットアップの周波数応答は、リバース・アイソレーション測定における支配的な誤差です。スルー・レスポンス測定校正を行うことにより、この誤差を大幅に減らすことができます。この校正はレスポンス/アイソレーション校正の一部として実行することができます。
- 増幅器は温度が異なると非常に異なる応答を示す場合があります。テストは増幅器が目的の動作温度にあるときに行うべきです。

1. 増幅器を下の図のように接続します。



2. アナライザを初期設定します。
3. S12測定を選択します。
4. 被試験増幅器に合わせた設定を選択します。
5. 増幅器を取り除き、スルー・レスポンス校正またはレスポンス/アイソレーション校正を実行します。
6. 表示された測定結果が見やすくなるようにスケールを調整し、目的の周波数でのリバース・アイソレーションをマーカを使って測定します。
7. データをプリントするか、ディスクに保存します。

小信号利得およびフラットネス

概要　　概念　　測定方法

小信号利得とは、増幅器の線形動作領域の利得です。これは通常、一定の入力パワーで周波数を掃引して測定されます。利得フラットネスは、指定した周波数レンジでの利得の変動の度合いを表します。

ボタンをクリックすると下記の事項について知ることができます。

利得とは何か

RF増幅器の利得は、増幅器の出力信号と入力信号のパワーの差と定義されます。増幅器の入力と出力のインピーダンスはどちらもシステムの特性インピーダンスに等しいと仮定されます。

- Sパラメータの用語では、利得のことをS_{21}と呼びます。
- 利得はdB単位で表されます。すなわち、入力パワーに対する出力パワーの対数比です。
- 利得を計算するには、dBm(1 mWを基準としたパワー)単位で表した出力レベルから入力レベルを減算します。
- 増幅器利得は通常、特定の周波数レンジにおける最小値として指定されます。増幅器によっては、利得の最小値と最大値を規定している場合もあります。これは、システムの後の部分がアンダードライブにもオーバドライブにもならないようにするためです。

フラットネスとは何か

フラットネスは、特定の周波数レンジで増幅器の利得にどれだけの変動が許されるかを表します。増幅器の利得が変動すると、増幅器を通る信号に歪みが生じるおそれがあります。

なぜ小信号利得およびフラットネスを測定するのか

目的の帯域幅内で利得に偏移があると、各周波数成分が均等に増幅されないため、伝送信号に歪みが生じます。小信号利得は、50 Ωシステムにおける特定の周波数での増幅器の利得を定量化するために使います。フラットネスは、50 Ωシステムにおける特定の周波数レンジでの増幅器利得の偏移を知るために使います。

確度に関する考慮事項

- 増幅器は温度が異なると非常に異なる応答を示す場合があります。テストは増幅器が目的の動作温度にあるときに行うべきです。
- 必要なら、増幅器の出力パワーを適当に減衰します。出力パワーが大きすぎると下記のような結果が生じるおそれがあります。
    - 本器のレシーバが損傷されます。
    - 本器のレシーバの入力圧縮レベルを超過し、測定が不正確になります。

増幅器の出力パワーの減衰には、下記のような手段が使用できます。

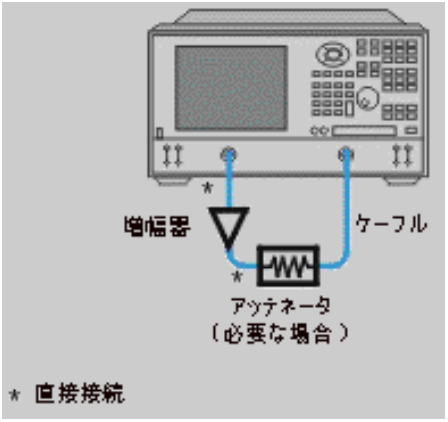
- アッテネータ
- 結合器

アッテネータと結合器はテスト・システムの一部なので、校正の際にその周波数応答と不整合の影響を考慮する必要があります。適切な誤差補正技法により、これらの影響を減らすことができます。

- 小信号利得およびフラットネスの測定セットアップにおける支配的な誤差は、周波数応答です。スルー・レスポンス測定校正により、この誤差を大きく減らすことができます。さらに高い確度が必要なら、2ポート測定校正を実行します。
- IF帯域幅を狭めるか、アベレージングを使うと、測定のダイナミック・レンジと確度は向上しますが、測定速度は低下します。

利得およびフラットネスの測定方法

1. 本器を初期設定します。
2. S21測定パラメータを選択します。
3. 本器の信号源出力を、増幅器の出力応答の線形領域(通常は1 dB圧縮ポイントより10 dB下)に設定します。
4. 必要なら外部アッテネータを選択して、本器のポート2のレシーバに圧縮や損傷が生じないように増幅器の出力パワーを減衰させます。



5. 下の図のように増幅器を接続し、DCバイアスを供給します。
6. 被試験増幅器に対するアナライザ設定を選択します。
7. 増幅器を取り除き、測定校正を実行します。増幅器の測定に使用するアッテネータやケーブルは必ず校正セットアップに含めます。
8. 機器ステートをメモリに保存します。
9. 増幅器を再接続します。
10. 測定結果の表示が見やすくなるようにスケールを調整し、目的の周波数での小信号利得をマーカを使って測定します。
11. マーカを使ってp-pリップルを表示することにより、特定の周波数レンジでの利得フラットネスを測定します。
12. データをプリントするか、ディスクに保存します。
13. このタイプの測定は自動化できます。

時間領域測定

時間領域オプション010が本器に装備されている場合、デバイス応答を時間の関数として表示することができます。 ボタンをクリックすると情報が表示されます。

- **時間領域の概要**
- **バンドパス/ローパス時間領域モード**
- **時間領域の分解能とレンジ**
- **表示データに対するウィンドウ・フィルタ**
- **時間ゲーティング・フィルタ**
- **時間領域測定データ**

時間領域の概要

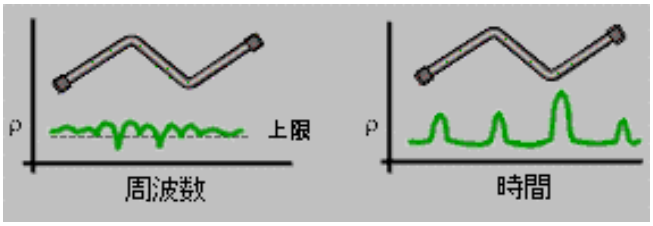
概要　　概念　　時間領域測定の実行方法

時間領域オプション010が本器に装備されている場合、デバイス応答を時間の関数として表示することができます。このトピックでは、時間領域オプションの動作の概要について説明します。

時間領域測定とは何か

- 本器の通常動作では、被試験デバイスのRF性能特性が周波数の関数として測定されます。
- 本器に時間領域オプション010が装備されている場合、数学的な技法(逆フーリエ変換)を使って、周波数領域データを時間領域に変換することが可能になります。
- 測定結果は、時間を水平軸として表示されます。
- 各応答値が時間軸上に分離されて示されるので、ネットワークの性能や限界に関する異なる解析方法が可能になります。

下の図は、同じケーブル反射測定データを周波数領域と時間領域で表示したものを示します。ケーブルには2つの屈曲部があり、それぞれが不整合すなわちライン・インピーダンスの変化をもたらします。

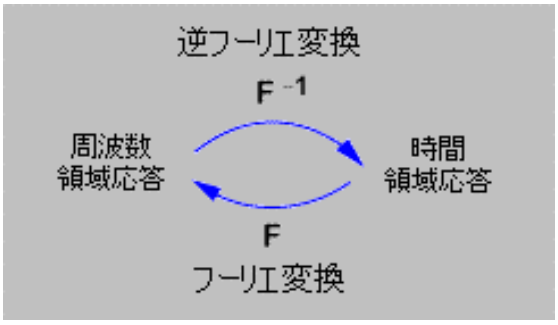


- 入力ポートで測定された**周波数応答**は、ケーブルの不整合同士の相互作用によって生じた複合反射応答(リップル)を示しますが、ケーブルのどの位置で不整合が発生しているかを知るのは困難です。
- **時間領域応答**を使えば、ケーブルの内部を調べて、1つ1つの不整合の位置と大きさを知ることができます。上の例では、2番目の屈曲部で大きな不整合が生じていることがわかります。

フーリエ変換

時間領域で応答を測定する場合、ゲーティング機能を使って不要な応答を除去することも可能です。その後、必要な応答だけを周波数領域に変換することができます。これにより、コネクタやアダプタなどの外部デバイスの影響を除去して、特定の信号や変調波形に対する回路の応答を調べることができます。このプロセスでは下記の変換が用いられます。

- **逆フーリエ変換—**周波数測定データを時間領域データに変換します。
- **フーリエ変換—**時間領域データを周波数領域に戻します。



タイム・ドメイン・リフレクトメトリとの類似性

- 時間領域モードは、以前からあるタイム・ドメイン・リフレクトメトリ(TDR)をシミュレートしています。これは、反射波形を観察することで電気ラインの特性を知る方法のことです。
- 応答の形状と位置を見ることで、不連続部のインピーダンスと距離を知ることができます。

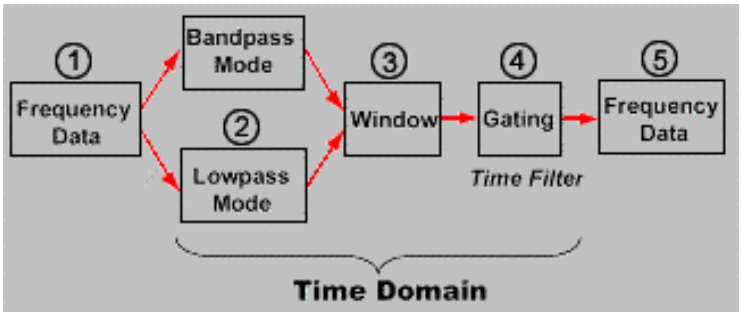
タイム・ドメイン・リフレクトメトリ解析

- 従来のTDR解析では、テスト・デバイスに実際のインパルスまたはステップを与えて、反射してきたエネルギーを観察します。
- TDR画面には、入射されたステップと反射波形とが表示されます。
- 反射波形の大きさ、持続時間、形状を調べることで、伝送システムのインピーダンス変化の性質を知ることができます。

アナライザの時間領域解析

- 本器は実際に入射インパルスやステップを与えるわけではなく、掃引周波数応答測定を実行します。
- 周波数応答データを元に、選択した変換に応じて、デバイスのステップ、インパルス、バンドパスのいずれかの応答が計算されます。
- 本器は時間領域応答(TDR形式の表示)の形でデータを示します。

時間領域測定のシーケンス



1. **周波数データ**—本器が周波数領域でのデバイス応答を測定したときに収集されます。
2. **バンドパス・モードとローパス・モード**—被試験デバイスの周波数特性と表示したい応答に基づいて、どちらかを選択します。
   - **バンドパス**・モードは、柔軟性の高いモードです。フィルタなど、一部の周波数を通過しないデバイスに適しています。インパルス応答だけが得られます。
   - **ローパス**・モードは、従来のTDRをシミュレートします。ケーブルなど、DCまでのすべての周波数を通すデバイスに適しています。掃引には高調波関係のある周波数が用いられます。インパルス応答とステップ応答の両方が得られます。
3. **ウィンドウ**——サンプル周期の端での0遷移をスムーズにすることで、時間領域応答を最適化します。これにより応答の不要なサイドローブが抑圧され、低レベルの応答が見えるようになります。最小ウィンドウを使うと、分解能は最高になりますが、サイドローブは最も大きくなります。
4. **ゲーティング**—不要な応答の影響を除去する時間フィルタを指定します。
5. **周波数データ**—を表示するために、時間領域応答を周波数領域に戻すことができます。この結果は、ゲーティングで残された特定の応答だけの効果を示します。

ボタンをクリックすると、時間領域測定の一般的な手順が表示されます。

測定条件設定

1. 周波数領域測定の設定を選択します。測定の設定を参照してください。
2. 使用する変換モードを決定します。
   - 被試験デバイスの測定応答がDCに達する場合は、ローパス・モードとステップ応答を使用します。
   - 被試験デバイスがDCで測定応答を示さない場合、バンドパス・モードを使用します。

   バンドパス・モードとローパス・モードを参照してください。
3. 被試験デバイスに対して必要な応答分解能と測定レンジを得るために、測定設定の値を計算します。
   - 周波数レンジ
   - ポイント数
   - ウィンドウ幅

     分解能とレンジを参照してください。
4. 前のステップで計算した周波数スパンとポイント数の値を設定します。測定の設定を参照してください。
5. ローパス・モードを選択した場合、下記のステップを実行します。
   a. スタート周波数は300 kHzのままにし、ストップ周波数とポイント数を入力します。
   b. **Trace**メニューで**Transform, Set Freq. Low Pass**をクリックします。これにより、すべての周波数が自動的に設定されます。
6. データを時間領域に変換し、前のステップで計算したウィンドウの値を入力します。応答分解能と測定レンジが測定アプリケーションの必要を満たすかどうかを調べます。
   a. 現在の設定でデバイス応答の測定ができない場合、周波数スパン、ポイント数、ウィンドウ幅を変更します。時間領域の分解能とレンジを参照してください。
7. 変換をオフにし、測定条件設定と必要な確度に適合する測定校正を実行します。測定校正の選択と実行を参照してください。
8. データを時間領域に変換します。

結果の最適化

1. 測定応答に対してオートスケールを実行します。
2. 最大の表示分解能を得るために、目的の応答を画面の中央に配置します。
3. 表示された測定結果から不要な応答を除去するため、ゲーティング(時間フィルタ)を設定してオンにします。

表示データの解釈

1. 不連続部のインピーダンスのタイプを解釈します。時間領域測定データを参照してください。
2. ゲーティング結果のデータを周波数領域で表示するには、ゲーティング機能をオンにし、変換をオフにします。

時間領域の分解能とレンジ

概要　　概念　　分解能とレンジの設定方法

このトピックでは、デバイス応答の時間領域データをすべて表示するための方法について説明します。最高の分解能と最大のレンジが得られる設定を考慮する必要があります。

応答分解能

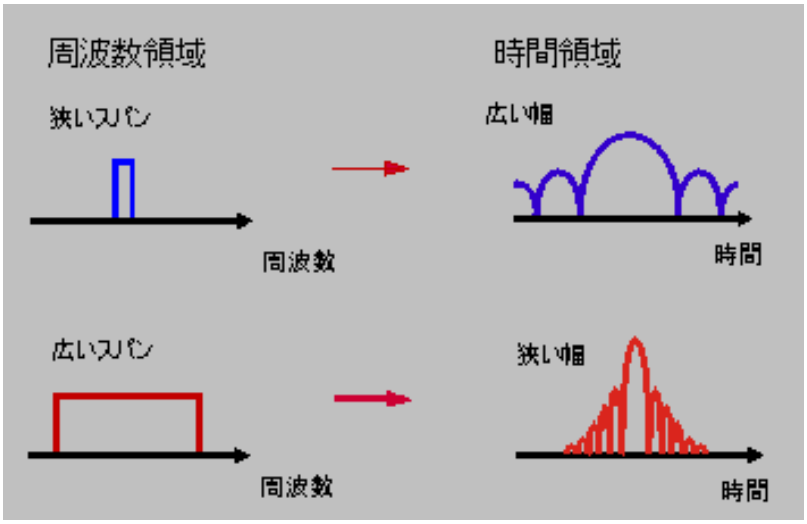
時間領域の応答分解能とは、時間的に近接した2つの応答を区別する能力のことです。時間領域での測定応答の分解能は、下記のようないくつかの要因で決まります。

応答分解能に対する周波数スパンの影響

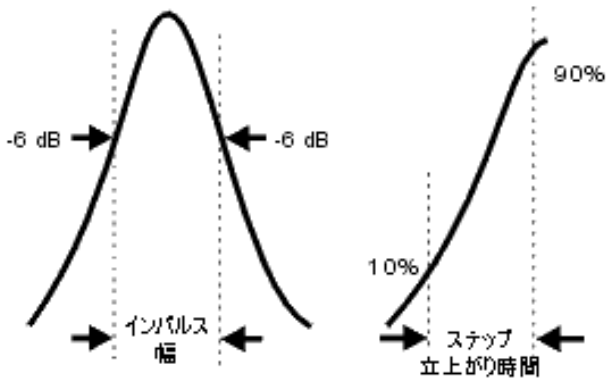
下の図は、周波数スパンが狭い場合と広い場合の応答分解能に対する影響を示します。

- 狭い周波数スパンで測定された応答は、比較的広い周波数スパンで測定された各コネクタの応答に重なるように表示されています。
- 周波数スパンが広い場合、2個のコネクタが別々の応答に分解されます。

- 周波数スパンとパルスの時間幅とは反比例の関係になります。
- 周波数スパンが広いほど、インパルスの幅またはステップの立上がり時間が小さくなります。

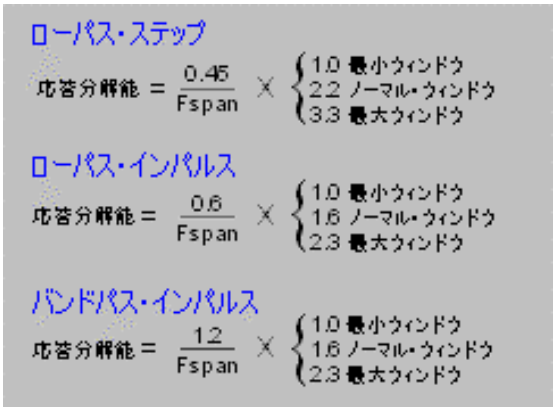


- 振幅が等しい応答の場合、応答分解能は50%(-6 dB)ポイントで定義されたインパルスの幅、または10%ポイントから90%ポイントの間で定義されたステップの立上がり時間に等しくなります。

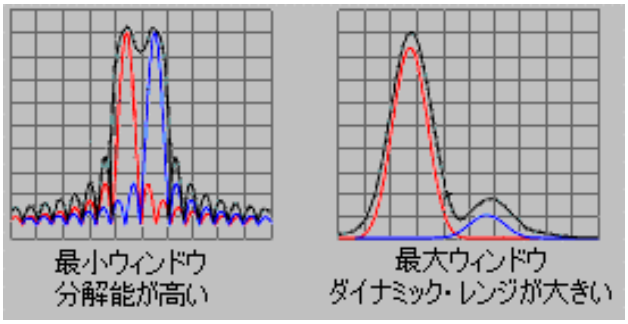


応答分解能に対するウィンドウ幅の影響

- 応答分解能は、変換に用いられるウィンドウの関数でもあります。
- 近似的な応答分解能を下記の式で計算できます。これらの式は振幅が等しい応答だけに当てはまり、50%のインパルス幅または10%-90%のステップ立上がり時間を計算します。



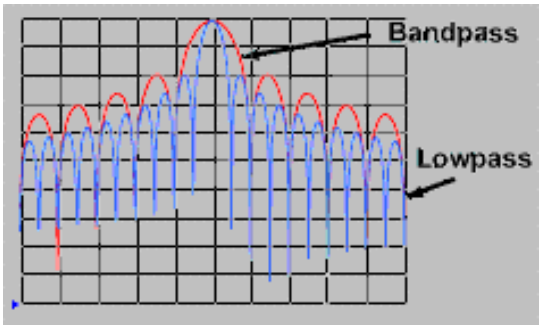
下の図は、ウィンドウ幅の選択がデバイス応答に依存することを示します。



- レベルが等しい信号に対しては、最小ウィンドウを使うと分解能を改善できます。
- レベルが異なる信号に対しては、最大ウィンドウを使うとダイナミック・レンジを改善できます。

応答分解能に対する時間領域モードの影響

- 時間領域のバンドパス・モードとローパス・モードでは、応答分解能に違いがあります。
- 周波数スパンとポイント数が等しい場合、ローパス・モードのほうが分解能が高く、バンドパス・モードに比べてインパルス幅が半分になります。



表示分解能

時間領域の表示分解能とは、1つの応答の時間的位置を正確に表示する能力のことです。表示分解能が高いほど、応答のピークやヌルの位置を正確に知ることができます。どれだけ正確になるかは、選択した時間スパンとポイント数に直接関係します。

$$\text{表示分解能} = T_{\text{スパン}} / \text{ポイント数} - 1$$

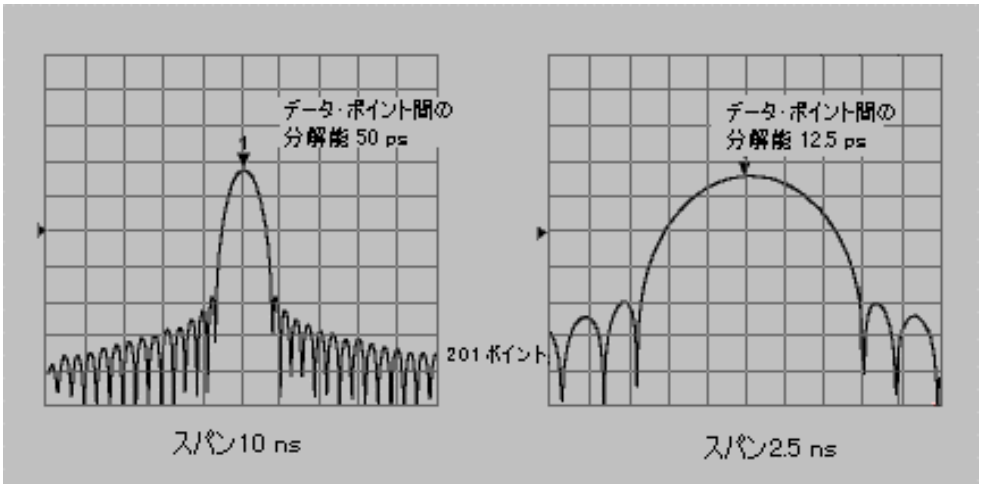
表示分解能を改善するには下記の2つの方法があります。

- 時間スパンを狭める
- ポイント数を増やす

注記: ポイント数を増やすと、測定校正が無効になる場合があります。

下の例では、どちらの測定も201ポイントで実行されています。

- 50 psの分解能に対応する時間スパンは10 nsです。
- 12.5 psの分解能に対応する時間スパンは2.5 nsです。



注記: 近接した2つの信号を分解する能力は変化していません。ただし、ローパス・モードで測定を行う場合、ポイント数を変更すると周波数スパンが変化する場合があります。

分解能最適化のヒント

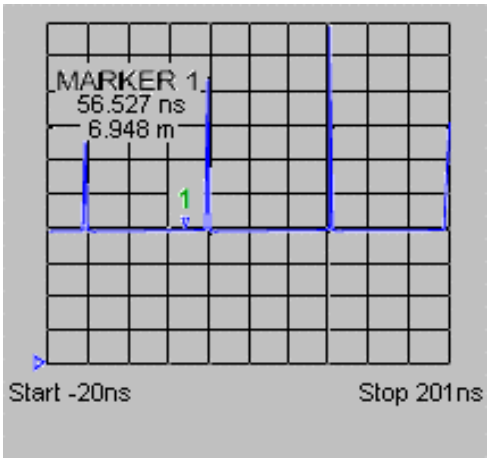
- ほとんどの場合、有効な最大の周波数スパンを選択することで、時間分解能を最高にするのが適切です。ただし、被試験デバイスの使用可能周波数レンジを超えないようにしなければなりません。
- 応答分解能は、周波数スパン、ウィンドウ幅、時間領域モードに依存します。

測定レンジ

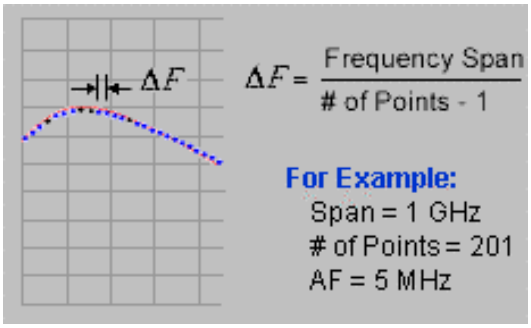
時間領域の測定レンジとは、応答の繰返しに遭遇せずに測定を行うことができる時間の長さと定義されます。

注記: レンジと応答分解能は反比例関係にあります。一方を改善するともう一方が悪化します。

応答の繰返し(エリアジング)は、一定の時間間隔で発生します。下の例では、0秒付近の実際の応答が60 ns付近と120 ns付近に繰り返されています。



測定レンジは、デバイスの周波数応答測定の周波数ポイント間隔と関係しています。



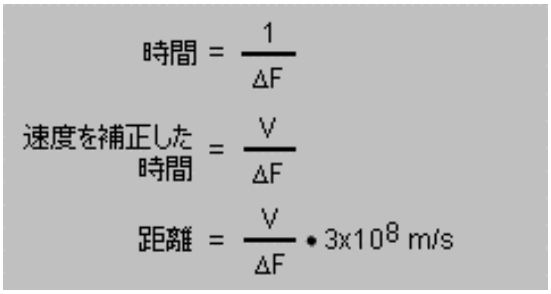
測定レンジは(ポイント数-1)に比例し、周波数スパンに反比例します。レンジを広げるには、下記の2つの設定を変更する方法があります。

- ポイント数を増やす
- 周波数スパンを狭める

注記: これらの変更はどちらも、測定校正の前に行う必要があります。

測定レンジをメートル単位で求めるには、時間間隔に自由空間中の光速(3×10^8 m/s)を掛けます。測定レンジの実際の物理長を計算するには、伝送媒体の相対速度をこの値に掛けます。

- ポリエチレンの相対速度 = 0.66
- テフロンの相対速度 = 0.70



注記: 可能な最大のレンジに本器を設定すると、時間領域の応答分解能が低下します。また、最大のレンジはケーブルの損失に依存します。返ってくる信号が小さすぎて測定できない場合、周波数スパンと無関係にレンジはこの減衰によって制限されます。

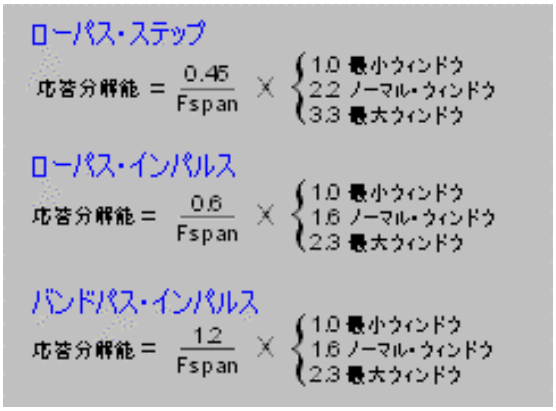
ボタンをクリックすると、時間領域の分解能とレンジを設定する手順が表示されます。

マウス手順

この手順では、時間領域測定がすでに設定されていることを前提とします。

応答分解能

1. 必要な応答分解能を実現するため、周波数スパンとウィンドウ幅を計算します。



2. **Trace**メニューで**Transform**をクリックします。
3. **Category**ボックスで下矢印をクリックして**Window**を選択します。
4. 測定アプリケーションに必要な分解能を実現するウィンドウ幅を選択します。

- 同じくらいの大きさの信号を測定する場合、最小ウィンドウを選択します。
- 大きさが非常に異なる信号を測定する場合、最大ウィンドウを測定します。

5. **Channel**メニューで**Start/Stop**をクリックし、周波数レンジを広げる値を入力します。
6. さらに応答分解能を高める必要がある場合、被試験デバイスの動作周波数がDCに達するなら、ローパス変換モードを使用します。時間領域モードを参照してください。

表示分解能

1. **Trace**メニューで**Transform**をクリックします。
2. **Center**ボックスに、測定対象の応答を画面の中央に配置する値を入力します。
3. **Span**ボックスに、時間スパンを狭める値を入力します。
4. さらに高い表示分解能が必要なら、ポイント数を増やします。**Sweep**メニューで**Number of Points**をクリックし、大きい値を選択します。

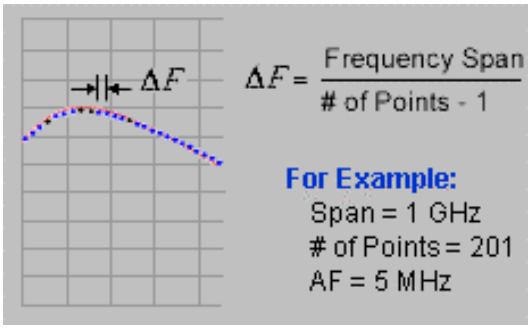
測定レンジ

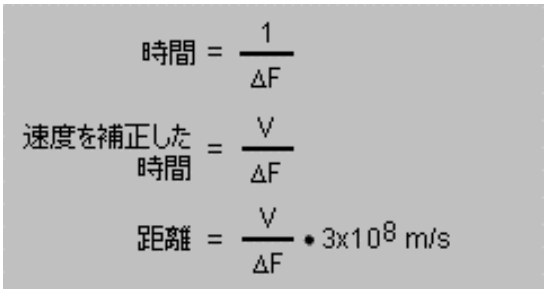
応答の繰返し(エリアジング)が発生した場合、実際の応答とエリアジング応答はテストを行わない限り見かけ上は区別できないのが普通です。

1. 時間スパンを広げて、十分な間隔をおいた明らかな繰返しが見られるかどうかを観察します。
2. それでも応答が実際のものかどうか疑問が残る場合、応答を画面中央に配置して、トレースを記憶します。
3. 周波数スパンを5%ほど小さくして、もう一度デバイスを測定します。

- 時間領域応答が正しいものなら、画面の中央から動かないはずです。
- 応答がエリアジングによるものなら、周波数レンジを変更すると時間軸に沿って移動します。

4. 必要な測定レンジを実現する周波数スパンとポイント数を計算します。





5. **Sweep**メニューで**Number of Points**をクリックして、大きい値を選択します。
6. **Channel**メニューで**Start/Stop**をクリックし、必要な測定レンジを実現するために計算した値を入力します。

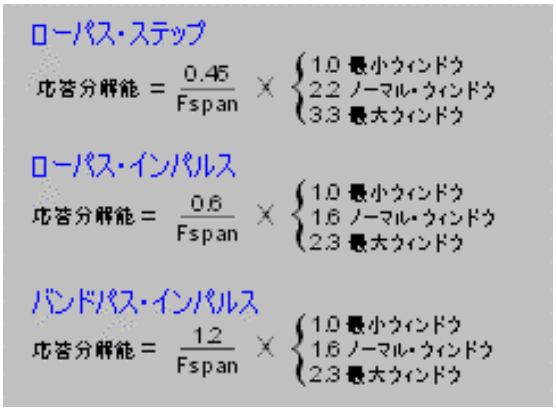
注記: 必要な分解能とレンジを実現するために測定設定を変更した後は、変換をオフにし、測定校正を実行してから、変換をオンにします。

フロントパネル・キー手順

この手順では、時間領域測定がすでに設定されていることを前提とします。

応答分解能

1. 必要な応答分解能を実現するため、周波数スパンとウィンドウ幅を計算します。



1. **COMMAND** ブロックで**Menu/Dialog**を押します。
2. **NAVIGATION**ブロックで右タブと矢印を押して**Trace**メニューに移動し、**Transform**まで移動します。**Click**を押します。
3. 右タブを押して**Category**ボックスに移動し、下矢印をクリックして**Window**を選択します。
4. 測定アプリケーションに必要な分解能を実現するウィンドウ幅を選択します。

- 同程度の大きさの信号を測定する場合、最小ウィンドウを選択します。
- 大きさが非常に異なる信号を測定する場合、最大ウィンドウを測定します。

5. **Channel**メニューで**Start/Stop**をクリックし、周波数レンジを広げる値を入力します。
6. さらに応答分解能を高める必要がある場合、被試験デバイスの動作周波数がDCに達するなら、ローパス変換モードを使用します。時間領域モードを参照してください。

表示分解能

1. **COMMAND** ブロックで**Menu/Dialog**を押します。
2. **NAVIGATION**ブロックで右タブと矢印を押して**Trace**メニューに移動し、**Transform**まで移動します。**Click**を押します。
3. 右タブを押して**Center**ボックスに移動し、測定対象の応答を画面の中央に配置する値を入力します。
4. 右タブを押して**Span**ボックスに移動し、時間スパンを狭める値を入力します。
5. さらに高い表示分解能が必要なら、ポイント数を増やします。
   a. **CHANNEL SETUP**ブロックで**Sweep Setup**を押します。
   b. **F2 (Points)**を押し、大きい値を選択します。

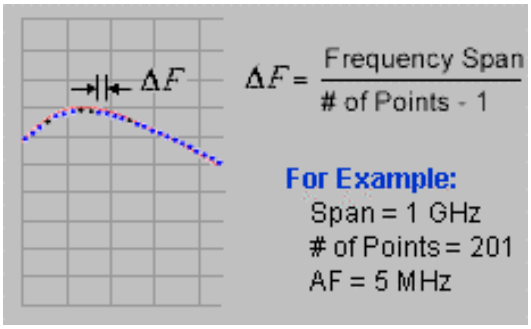
測定レンジ

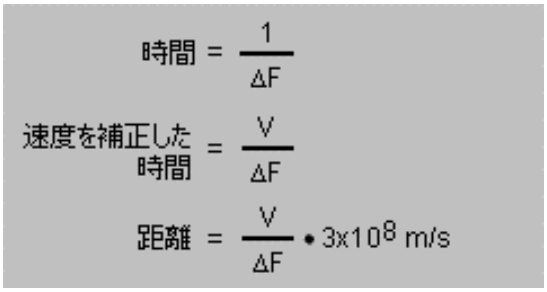
応答の繰返し(エリアジング)が発生した場合、実際の応答とエリアジング応答はテストを行わない限り見かけ上は区別できないのが普通です。

1. 時間スパンを広げて、十分な間隔をおいた明らかな繰返しが見られるかどうかを観察します。
   a. **COMMAND** ブロックで**Menu/Dialog**を押します。
   b. **NAVIGATION**ブロックで右タブと矢印を押して**Trace**メニューに移動し、**Transform**まで移動します。**Click**を押します。
   c. 右タブを押して**Span**ボックスに移動し、時間スパンを広げる値を入力します。
2. それでも応答が実際のものかどうか疑問が残る場合、応答を画面中央に配置して、トレースを記憶します。
   a. 右タブを押して**Center**ボックスに移動し、測定対象の応答を画面の中央に配置する値を入力します。
   b. **TRACE SETUP**ブロックで**Math/Memory**を押し、**F1 (Data>>Mem)**を押します。
3. 周波数スパンを5%ほど小さくして、もう一度デバイスを測定します。
   a. **CHANNEL SETUP**ブロックで**Stop/Span**を押します。
   b. アクティブ・エントリで**F4 (Span)**を押し、スパンを狭める値を入力します。
4. 時間領域応答が正しいものなら、画面の中央から動かないはずです。

- 応答がエリアジングによるものなら、周波数レンジを変更すると時間軸に沿って移動します。

4. 必要な測定レンジを実現する周波数スパンとポイント数を計算します。





5. **CHANNEL SETUP**ブロックで**Sweep Setup**、**F2 (Points)**を押し、計算した値を選択します。

6. **CHANNEL SETUP**ブロックで **Start/Stop**を押し、必要な測定レンジを実現するために計算した値を入力します。

注記: 必要な分解能とレンジを実現するために測定設定を変更した後は、変換をオフにし、測定校正を実行してから、変換をオンにします。

表示データに対するウィンドウ・フィルタ

概要　　概念　　ウィンドウ・フィルタの設定方法

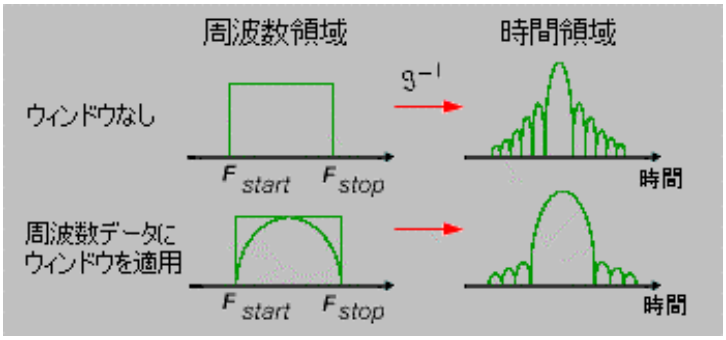
時間領域変換では、サイドローブやインパルス幅の影響によって、時間分解能が制限される場合があります。ウィンドウ・フィルタを使用すると、これらの影響を 低減できます。

アナライザにはウィンドウ関数機能が装備されており、時間領域測定で 個々の応答のアイソレートや識別をより簡単に行うことができます。ウィンドウ関数機能によって、時間領域結果のインパルス幅、サイドローブ・レベル、ステップ立上がり時間を 変更することが可能です。

ウィンドウ・フィルタリングの利点

周波数領域測定ではスタート周波数とストップ周波数に急な遷移があり、 時間領域応答でオーバシュートとリンギングが発生します。 ウィンドウ機能を使うと、周波数領域の遷移を 小さくすることができます。

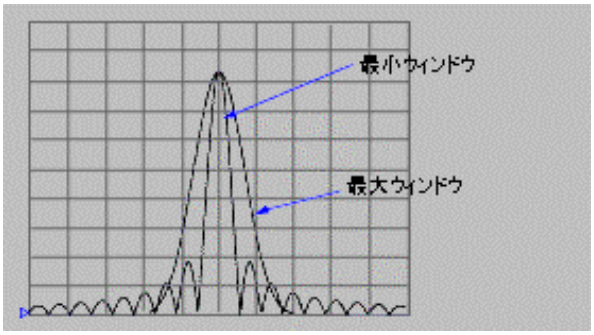
- **インパルス幅 またはステップ立上がり時間** -- テスト・システムの帯域制限性によって発生します。 2つの隣接する応答を分離する能力が制限されます。 インパルス幅は、測定の周波数スパンと反比例します。 インパルス幅を狭める唯一の方法は、 周波数スパンを広げることです。
- **インパルスのサイドローブ** --ストップ周波数における急なカットオフによって起こります。隣接の高レベル応答のサイドローブに低レベル応答が隠れるため、時間領域測定の ダイナミック・レンジが 制限されます。



ウィンドウ・フィルタリングの効果

- インパルス幅を小さくすると、**応答 分解能**が改善されるので、2つの近接する応答が分離可能になります。
- インパルス・サイドローブを小さくすると、**ダイナミック・レンジ**が改善されるので、低レベル応答が測定可能になります。

ウィンドウ関数値は変更可能です。



注記: 分解能は、適用されたウィンドウ関数に比例し、 測定された周波数スパンに反比例します。詳細については、応答とレンジ分解能を参照してください。

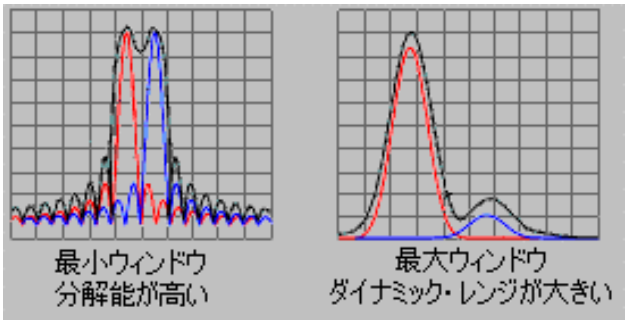
アナライザでは測定データをウィンドウ・フィルタに通すことにより、 デバイス応答を変更することができます。下記に示す効果があります。

- **インパルス・スティミュラスの場合 --** サイドローブとインパルス幅の影響により、時間分解能の制限値が変わります。
- **ステップ・スティミュラスの場合 --** (ローパスのみ) オーバシュートとリンギングが減少します。

適切なウィンドウ値

被試験デバイスから予測される応答の種類に一番あった ウィンドウを選択できます。

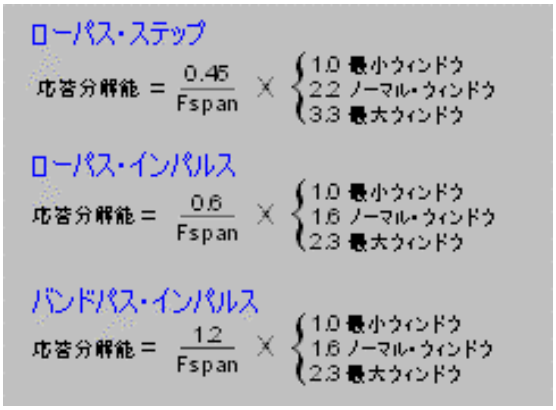
- **測定対象 デバイスの2つの応答の振幅レベルが等しい場合、**最小インパルス幅、最小ウィンドウを選択して 分解能を改善します。
- **測定対象 デバイスの2つの応答の振幅が異なる場合、**最低サイドローブ・レベル、最大ウィンドウを選択して ダイナミック・レンジを改善 します。



ウィンドウの特性

下記に示す依存関係が存在します。次のテーブルにはこの関係をサンプル値として示します。
サンプル・ウィンドウの特性を2.997 GHzの周波数スパンに対して示します。

応答分解能の式



下の図は、ショート回路反射測定の応答に各種ウィンドウを適用した場合の、 典型的な効果を示します。
ボタンをクリックすると、時間領域モードでウィンドウ・フィルタを設定する 手順が表示されます。

マウス手順

1. **Trace**メニューで、**Transform**をクリックします。
2. **Category** ボックスで、下矢印をクリックし、**Window**を選択します。
3. 測定アプリケーションに必要な分解能を得るために、適したウィンドウを選択します。
   - 振幅の等しい信号を測定する必要がある場合、応答分解能を高めるために最小ウィンドウを 選択します。
   - 振幅の大きく異なる信号を測定する必要がある場合、ダイナミック・レンジを広げるために最大ウィンドウを選択します。

注記: ウィンドウ値は、最小/最大、Kaiser Beta、またはインパルス幅/ステップ立上がり時間によって指定することができます。

フロントパネル・キー手順

1. **COMMAND** ブロックで、**Menu/Dialog**を押します。
2. **NAVIGATION** ブロックで、右タブと矢印を押して**Trace** メニューに移動し、**Transform**まで移動します。 **Click**を押します。
3. 右タブを押して **Category** ボックスに移動し、下矢印をクリックして**Window**を選択します。
4. 測定アプリケーションに必要な分解能を得るために、適した ウィンドウを選択します。
   - 振幅の等しい信号を測定する必要がある場合、 応答分解能を高めるために最小ウィンドウを 選択します。
   - 振幅の大きく異なる信号を測定する必要がある場合、 ダイナミック・レンジを広げるために最大ウィンドウを 選択します。

注記:ウィンドウ値は、 最小/最大、Kaiser Beta、またはインパルス幅/ステップ立上がり 時間によって指定することができます。

時間ゲーティング・フィルタ

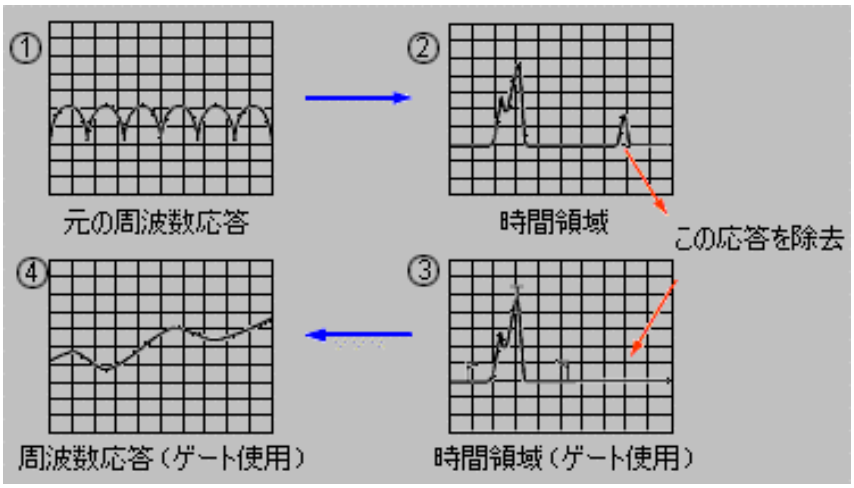
概要　　概念　　時間ゲーティング・フィルタの設定方法

ゲーティングを使うと、時間領域の応答を選択して数学的に除去できます(時間フィルタ)。残りの時間領域応答は周波数領域に再変換して表示できます。

時間領域応答にゲーティングを適用した後、変換をオフにして、ゲートがオンの状態でのデバイスの周波数応答を観察できます。例えば伝送測定で、複数の伝送経路の影響を除去することができます。また、それぞれの時間領域応答を個別に表示して、周波数領域における各応答の影響を解析することもできます。

ゲーティングの操作

1. 周波数領域でデバイスを測定します。
2. 時間領域変換をオンにします。時間領域応答(ローパスまたはバンドパス)が計算されます。
3. ゲート境界を設定します。
   - **バンドパス・ゲート・タイプの場合**—測定に含めたい応答を中心にゲートを配置します。ゲートをオンにすると、表示された測定結果からゲートの外側の応答が数学的に除去されます。
   - **ノッチ・ゲート・タイプの場合**—測定から除去したい応答を中心にゲートを配置します。ゲートをオンにすると、表示された測定結果からゲートの内側の応答が数学的に除去されます。
4. 時間領域変換をオフにして周波数領域で応答を表示し、ゲート・スパンの外側(ノッチ・ゲート・タイプの場合はゲート・スパンの内側)の応答を除去した結果を観察します。以下の例は、バンドパス形状ゲートの動作を示します。



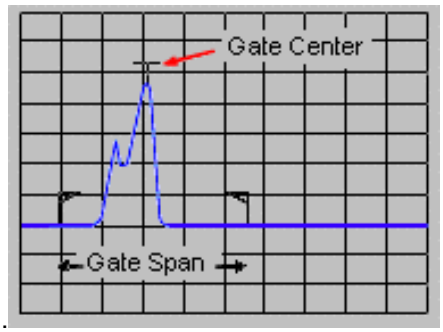
ゲート設定

ゲート境界とゲートの動作をこの設定で定義できます。

- **Start/Stop(スタート/ストップ)**—6 dBカットオフ時間を示します。
- **Center(中心)** —ゲートの中心時間を選択します。
- **Span(スパン)** —ゲートのストップ時間からスタート時間を引いたものと同じです。

- **ゲート・タイプ:**
  - **Bandpass(バンドパス)** —スタート/ストップまたはスパンの値の外側にある応答を除去します。
  - **Notch(ノッチ)—** スタート/ストップまたはスパンの値の内側にある応答を除去します。

この例ではバンドパス・ゲート・タイプが選択されており、スパンの外側の応答が除去されます。



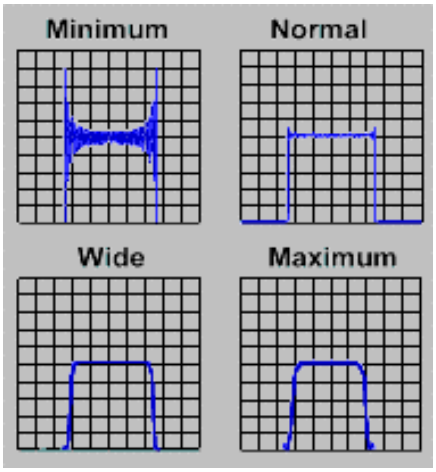
ゲート形状

ゲートのフィルタ形状は、表示データに対するウィンドウ・フィルタで説明しているウィンドウ・フィルタに似ています。一連のゲート形状(最小、ノーマル、ワイド、最大)の中から、測定に合わせて最適なものを選ぶことができます。

- **Minimum(最小)形状—**サイド・ローブが最も高く、ロールオフが最も鋭く、通過帯域のリップルは最大になります。
- **Maximum(最大)形状—** サイドローブが最も低く、ロールオフが最も遅く、通過帯域のリップルは最小になります。

ゲートの通過帯域リップル

ゲートはバンドパス形状のフィルタであり、4つのゲート形状はそれぞれ異なるフィルタ特性を示します。下の図は、4つのゲート形状のそれぞれの通過帯域リップルの違いを0.5 dB/divで示しています。最小形状ゲートが最もリップルが大きくなります。

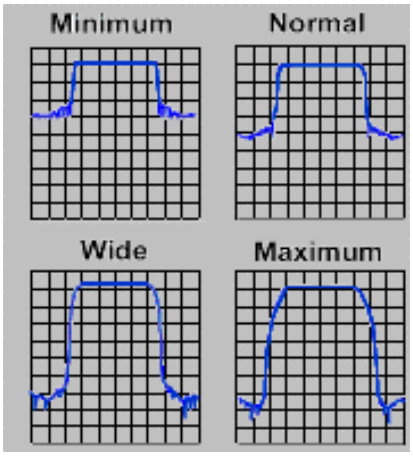


ゲートのサイドローブ

下の図は、4つのゲート形状のそれぞれのサイドローブ・レベルを比較したものです。サイドローブ・レベルを下げると、その代わりにカットオフ・レートが増加します。

- **最小形状ゲート**は最もサイドローブ・レベルが高く、カットオフ・レートが高速で、目的の応答と時間的に近い不要応答を除去するのに最も役立ちます。

- **最大形状ゲート**は最もサイドローブ・レベルが低く、カットオフ・レートが低速で、ゲート形状が広く、ゲート・スパン外部の応答に対する減衰率が最も大きくなります。



ゲート特性

4つのゲート形状はそれぞれ異なる特性を持っています。

- **通過帯域リップルとサイドローブ・レベル—**ゲート形状を記述します。
- **カットオフ時間—**ストップ時間(フィルタ・スカートの-6 dB位置)と最初のサイドローブのピークとの間の時間を表します。
- **最小ゲート・スパン—**カットオフ時間の2倍です。

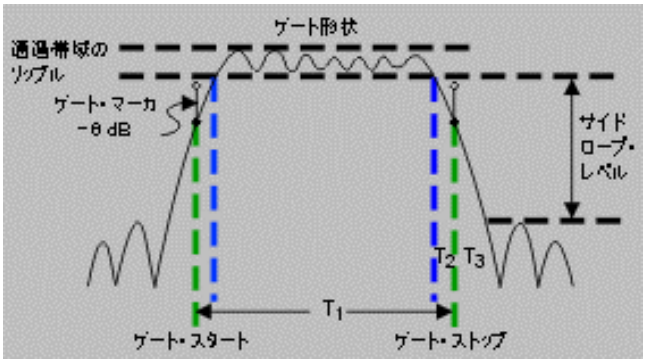
ゲート形状	通過帯域のリップル	サイドローブ・レベル	カットオフ時間	最小ゲート・スパン
最小	±0.10 dB	−48 dB	1.4/周波数スパン	2.8/周波数スパン
ノーマル	±0.01 dB	−68 dB	2.8/周波数スパン	5.6/周波数スパン
ワイド	±0.01 dB	−57 dB	4.4/周波数スパン	8.8/周波数スパン
最大	±0.01 dB	−70 dB	12.7/周波数スパン	25.4/周波数スパン

カットオフ時間

下の図は全体的なゲート形状を示し、それぞれの形状の特性を記します。

- T_1はゲートのスパンであり、ストップ時間からスタート時間を引いたものに等しくなります。
- T_2は通過帯域のエッジと-6 dBポイントとの間の時間であり、フィルタのカットオフ・レートを表します。
- T_3は-6 dBポイントとゲート阻止帯域のエッジとの間の時間です。

すべてのフィルタ形状で、T_2はT_3に等しく、中心時間の両側でフィルタは同じ形になります。



最小ゲート・スパン

各ゲート形状には、正しく動作するための最小推奨ゲート・スパンが存在します。これはゲートのカットオフ・レートが有限だからです。最小推奨ゲート・スパンは下記の式で与えられ、このスパンでフィルタの通過帯域が0 sになります。

$$T_{1MIN} = 2 \times T_2$$

最小スパンよりも小さいゲート・スパンを指定すると、下記の現象が生じます。

- 通過帯域のない歪んだゲート形状が生じます。
- 歪んだ形状が生じます。
- スタート時間とストップ時間が間違って表示されます。
- サイドローブ・レベルが上昇する場合があります。

ボタンをクリックすると、時間領域ゲートを設定する手順が表示されます。

マウス手順

1. **Trace**メニューで**Transform**をクリックします。
2. **Category**ボックスで下矢印をクリックし、**Gating**を選択します。
3. **Gate Type**ボックスでゲート・フィルタのタイプを選択します。
   - 表示された応答のうちゲート・スパンの外側にあるものを除去するには、**Bandpass**を選択します。
   - 表示された応答のうちゲート・スパンの内側にあるものを除去するには、**Notch**を選択します。
4. **Gate Shape**ボックスでゲート・フィルタの形状を選択します。各ゲート形状には、正しく動作するための最小推奨ゲート・スパンが存在します。

ゲート形状	通過帯域のリップル	サイドローブ・レベル	カットオフ時間	最小ゲート・スパン
最小	±0.10 dB	−48 dB	1.4/周波数スパン	2.8/周波数スパン
ノーマル	±0.01 dB	−68 dB	2.8/周波数スパン	5.6/周波数スパン
ワイド	±0.01 dB	−57 dB	4.4/周波数スパン	8.8/周波数スパン
最大	±0.01 dB	−70 dB	12.7/周波数スパン	25.4/周波数スパン

5. ゲート境界を、**Start**と**Stop**、または**Center**と**Span**を使って設定します。
6. **Gating**チェック・ボックスをクリックしてゲート機能をオンにします。

フロントパネル・キー手順

1. **COMMAND**ブロックで**Menu/Dialog**を押します。
2. **NAVIGATION**ブロックで右タブと矢印を押して**Trace**メニューに移動し、**Transform**まで移動します。**Click**を押します。
3. 右タブを押して**Category**ボックスに移動し、下矢印をクリックして**Gating**を選択します。
4. 右タブを押して**Gate Type**ボックスに移動し、ゲート・フィルタのタイプを選択します。
   - 表示された応答のうちゲート・スパンの外側にあるものを除去するには、**Bandpass**を選択します。
   - 表示された応答のうちゲート・スパンの内側にあるものを除去するには、**Notch**を選択します。

5. 右タブを押して**Gate Shape**ボックスに移動し、ゲート・フィルタの形状を選択します。各ゲート形状には、正しく動作するための最小推奨ゲート・スパンが存在します。ゲート特性を参照してください。

ゲート形状	通過帯域のリップル	サイドローブ・レベル	カットオフ時間	最小ゲート・スパン
最小	±0.10 dB	−48 dB	1.4/周波数スパン	2.8/周波数スパン
ノーマル	±0.01 dB	−68 dB	2.8/周波数スパン	5.6/周波数スパン
ワイド	±0.01 dB	−57 dB	4.4/周波数スパン	8.8/周波数スパン
最大	±0.01 dB	−70 dB	12.7/周波数スパン	25.4/周波数スパン

6. ゲート境界を、**Start**と**Stop**、または**Center**と**Span**を使って設定します。
7. 右タブを押して**Gating**チェック・ボックスに移動し、**Click**を押してゲート機能をオンにします。

時間領域測定データ

概要　　概念　　時間データの解釈方法

時間領域でのデバイス応答を理解するため、ここではさまざまなタイプの測定の例を示します。

マスキング

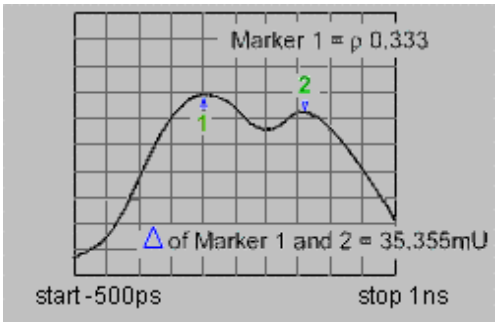
時間領域測定で表示される応答はマスキングによって変化するため、測定データを解釈する際にはマスキングについて理解しておくことが役立ちます。

- マスキングが発生するのは、基準面に最も近い不連続部または損失がその後の不連続部の応答に影響する場合です。
- 最初の不連続部で反射または吸収されたエネルギーは、2番目以降の不連続部には到達しません。
- この結果、2番目以降の不連続部の応答は、最初の応答がなかった場合に比べて小さくなります。

エネルギー反射によるマスキング

反射によるマスキングは、大きい不整合がある場合に発生します。このマスキング現象の例を下に示します。ここではラインのインピーダンスが変化しています。

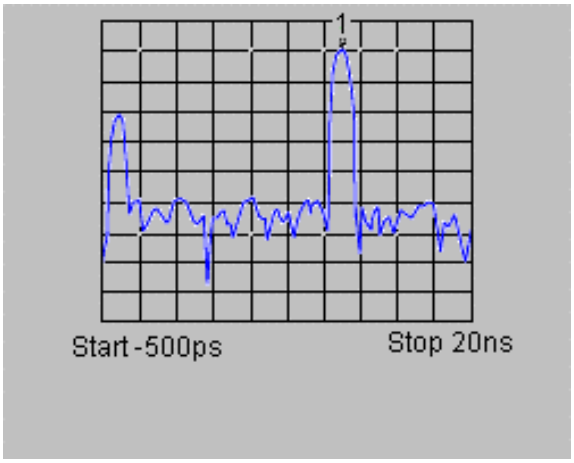
- 50Ωラインと25Ωエア・ラインと50Ωラインを順に接続したものを示します。
- 最初の不連続部の反射係数は約0.333です(これは25Ωのインピーダンスに対する正しい値です)。
- 25Ωセクションの末尾では、50Ωのインピーダンスから反射係数は0.333になるはずですが、そうなっていません。2個のデルタ・マーカが示すように、これらの振幅には35.355 mUの差があります。この理由は、最初の不整合でエネルギーが反射されたために、2番目の応答に入射するインパルスの振幅が1より小さくなっているからです。



エネルギー吸収によるマスキング

エネルギー吸収によるマスキングは、損失の大きい回路を測定する場合に発生します。このマスキング現象の例を下に示します。ここではケーブルをオープン回路で終端しています。

- オープン回路の応答は、マーカ1で-2.445 dBのリターン・ロスを示しています(対数振幅フォーマットで表示)。
- 期待される応答は、オープン回路の代表値(0 dB)です。
- マーカの値は、順方向経路と逆方向経路の損失の合計を示します。



バンドパス・モードでの反射測定

横軸のデータ

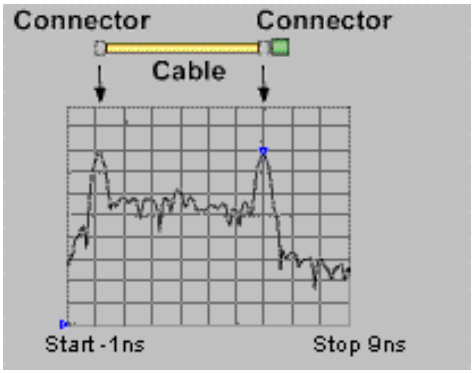
- テスト・ポートに入力されたインパルスが不連続部に到達してテスト・ポートに戻るまでの時間

縦軸のデータ

- 対数振幅フォーマットの場合はリターン・ロス(dB)
- リニア振幅フォーマットの場合は反射係数(ρ)

下の例は、バンドパス・モードでの反射測定を示します。

- 時間0のポイントにパルスがあります。これはテスト・ポート出力から見て最初の不連続部(コネクタ)の伝播時間を示します。
- 次のパルスは、インパルスが遭遇した次の不連続部(コネクタ)を示します。



バンドパス・モードでの伝送測定

横軸のデータ

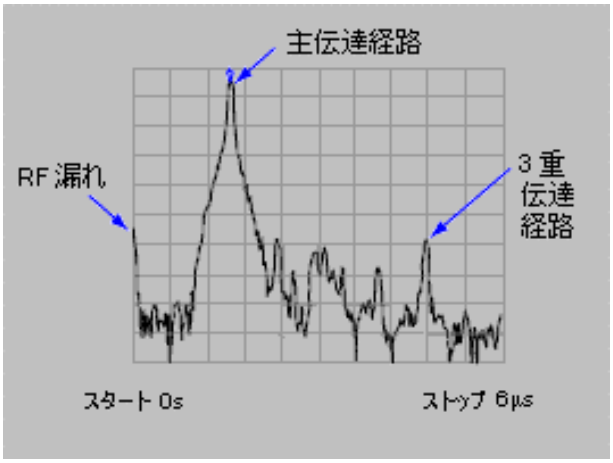
- 被試験デバイスによる伝搬遅延

縦軸のデータ

- 対数振幅フォーマットの場合は伝送損失または利得(dB)
- リニア振幅フォーマットの場合は伝送係数(ρ)

下の例は、バンドパス・モードの伝送測定を示します。

- これより、表面弾性波(SAW)フィルタ内の複数の伝送経路に関する情報がこのモードで得られることがわかります。
- 時間ポイント0に最も近いパルスは、回路の入力から出力までの最短経路を表します。これは必ずしも最大のパルスではなく、目的の経路に一致するとも限りません。
- その後のパルスはそれぞれ、入力から出力までの可能な別の経路を表します。これらはたまたま最短経路より長かったものです。



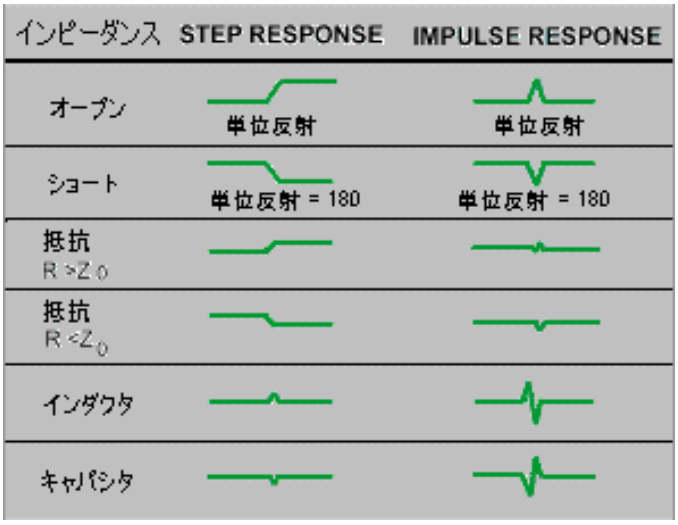
ヒント: SAWフィルタの主経路の応答だけを見るには、ゲートを使って主パルス以外を阻止する方法があります。その後、周波数領域で主経路の周波数応答だけを表示することができます。

ローパス・モードを使った障害位置検出

ローパス・モードでは、被試験デバイスのタイム・ドメイン・リフレクトメータ(TDR)をシミュレートすることができます。この応答から得られる情報は、不連続部のタイプを知るために役立ちます。

下の図は、既知の不連続部のローパス応答を示します。各種の回路素子に対応するローパス時間領域S_{11}応答波形のシミュレーションが示されています。ローパス・モードでは、ステップまたはインパルスをスティミュラスとする応答を見ることができます。

ヒント: ステップ応答を使ったほうが、不連続部の特性を見分けるのは容易になります。このモードは従来のTDR測定に最も近い結果を与えます。



ローパス・モードでの反射測定

横軸のデータ

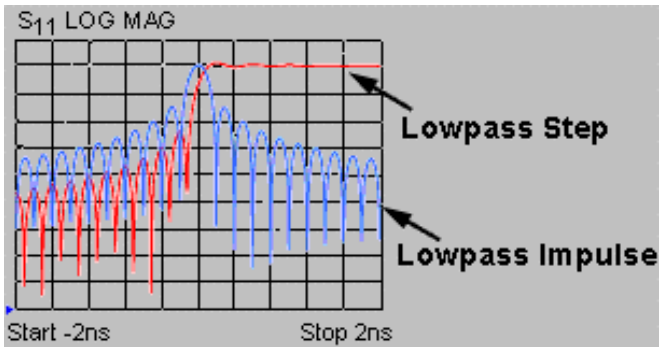
- テスト・ポートに入力されたインパルスが不連続部に到達してテスト・ポートに戻るまでの時間

注記: 実際の物理長を求めるには、適切な速度因子を入力します。

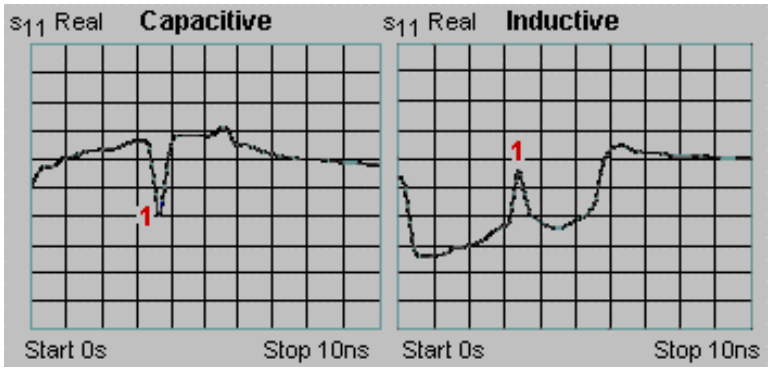
縦軸のデータ

- 実数フォーマットの場合は反射係数(ρ)

下の例は、ローパス・モードでの同じ反射測定のインパルス応答とステップ応答の結果を対数振幅フォーマットで重ねて表示したものです。



- 下の例は、2本の異なるケーブルのローパス反射測定結果を実数フォーマットで表示したものです。
- ローパス応答には、不連続部の位置と、存在するインピーダンスのタイプに関する情報が含まれます。



- **左の測定結果**は、クリンプしたケーブルの応答で、容量性の不連続部を示します。
- **右の測定結果**は、ほつれたケーブルの応答で、誘導性の不連続部を示します。

ローパス・モードでの伝送測定

横軸のデータ

- ある周波数レンジでのデバイスの平均通過時間
- デバイスの電気遅延時間

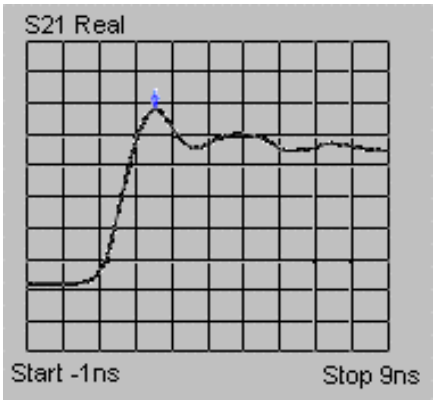
縦軸のデータ

- 実数フォーマットの場合は実数単位(Vなど)
- 対数フォーマットの場合は伝送損失または利得(dB)(インパルス・モードのみ)

下の例は、増幅器のローパス・ステップ応答を示します。

- 測定周波数レンジにおける平均群遅延は、ステップと増幅器応答との時間差です。
- ステップの立上がり時間は、周波数領域掃引の最高周波数に比例します。周波数が高いほど立上がり時間は高速になります。

- 増幅器応答のリンギングは、デザインの減衰が不足であることを示します。



ヒント: ローパス・ステップ応答の場合、最も有用なフォーマットは実数フォーマット(反射係数単位)です。インパルス応答の場合も実数フォーマットは使用できますが、ダイナミック・レンジを広げて大きい不連続部と小さい不連続部を同時に観察するには、対数振幅フォーマットを使用します。

ボタンをクリックすると、時間領域測定結果を表示する手順を見ることができます。

マウス手順

1. テスト・ポートを起点として応答を表示するために、スタート時間を-1 sに設定します。
   a. **Trace**メニューで**Transform**をクリックします。
   b. **Start**ボックスをクリックして-1 secと入力します。
2. **Trace**メニューで**Autoscale**をクリックして、測定応答全体を表示させます。
3. 表示分解能を最高にするために、目標の応答を画面中央に配置し、時間スパンを狭めます。
   a. **Trace**メニューで**Transform**をクリックします。
   b. **Start**ボックスをクリックして-1 secと入力します。
4. さらに応答分解能を上げたい場合は、ウィンドウと測定設定を変更します。時間領域の分解能とレンジを参照してください。
5. 応答までの時間と距離を正しく求めるには、速度因子を正しく設定します。
   a. **Scale**メニューで**Electrical Delay**をクリックします。
   b. **Velocity Factor**ボックスに0～1.0の値を入力します。
      - 真空中の光速 = 1.0
      - ポリエチレン(大部分のケーブル) = 0.66
      - テフロン = 0.7
   c. マーカを使って不連続部までの時間(の2倍)と電気長(の2倍)の値を表示します。

注記: 電気長でなく物理長を求めるには、被試験媒体の速度因子を設定します。不連続部までの往復距離でなく片道の距離を表示させるには、実際の速度因子の半分の値を入力します。

6. ローパス・モードを使用している場合、インピーダンス変化の振幅と位相を求めます。ローパス・モードを使った障害位置検出を参照してください。

フロントパネル・キー手順

1. テスト・ポートを起点として応答を表示するために、スタート時間を-1 sに設定します。
   a. **COMMAND**ブロックで、**Menu/Dialog**を押します。
   b. **NAVIGATION**ブロックで右タブと矢印を押して**Trace**メニューに移動し、**Transform**まで移動します。**Click**を押します。
   c. 右タブを押して**Start**ボックスに移動し、-1 secと入力します。
2. **TRACE SETUP**ブロックで**F1 (Autoscale)**を押して、測定応答全体を表示させます。
3. 表示分解能を最高にするために、目標の応答を画面中央に配置し、時間スパンを狭めます。
   a. **COMMAND**ブロックで、**Menu/Dialog**を押します。
   b. **NAVIGATION**ブロックで右タブと矢印を押して**Trace**メニューに移動し、**Transform**まで移動します。**Click**を押します。
   c. 右タブを押して**Center**ボックスに移動し、目標の応答を画面の中央に配置する値を入力します。
   d. 右タブを押して**Span**ボックスに移動し、スパンを狭める値を入力します。
4. さらに応答分解能を上げたい場合は、ウィンドウと測定設定を変更します。時間領域の分解能とレンジを参照してください。
5. 応答までの時間と距離を正しく求めるには、速度因子を正しく設定します。
   a. **COMMAND**ブロックで、**Menu/Dialog**を押します。
   b. **NAVIGATION**ブロックで右タブと下矢印を押して**Scale**メニューに移動し、**Electrical Delay**まで移動します。**Click**を押します。
   c. 右タブを押して**Velocity Factor**ボックスに移動し、0～1.0の値を入力します。
      - 真空中の光速 = 1.0
      - ポリエチレン(大部分のケーブル) = 0.66
      - テフロン = 0.7
   d. マーカを使って不連続部までの時間(の2倍)と電気長(の2倍)の値を表示します。

注記: 電気長でなく物理長を求めるには、被試験媒体の速度因子を設定します。不連続部までの往復距離でなく片道の距離を表示させるには、実際の速度因子の半分の値を入力します。

6. ローパス・モードを使用している場合、インピーダンス変化の振幅と位相を求めます。ローパス・モードを使った障害位置検出を参照してください。

バンドパス/ローパス時間領域モード

概要　　概念　　時間領域モードの設定方法

測定値を設定する前に、時間領域モードを選択する必要があります。選択するモードは主に被試験デバイスに依存します。

- 概念
- タイム・ドメイン・モードの設定方法
- モードの比較
  - バンドパス
  - ローパス
- モードの選択
- 周波数レンジとデータ・ポイント
- インパルス応答とステップ応答
- 有用なデータ・フォーマット

各モードの比較

バンドパス・モード

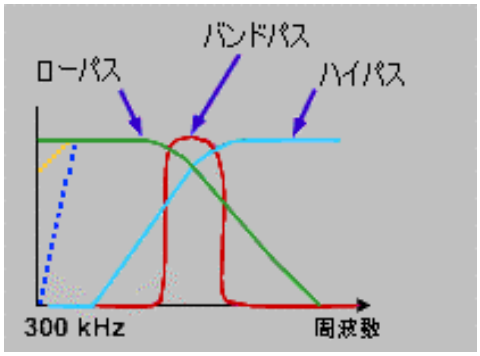
- 使いやすい
- DC(0 Hz)で動作しないバンドパス・デバイスの測定に有効
- 任意のスタート/ストップ周波数が使用可能
- 変換データにDC項が含まれないため、インパルス応答のみが対象
- 反射・伝送のどちらの測定にも使用可能
- 障害位置の検出が可能

ローパス・モード

- DCで動作するローパス・デバイスの測定に有効
- 従来のタイムドメイン・リフレクトメータ(TDR)をシミュレート
- 測定周波数は、周波数領域の最初のいくつかのデータ・ポイントから外挿されたDC項と高調波関係にあるものに限定
- 変換データにDC項が含まれるため、インパルス応答とステップ応答の両方が対象
- 障害位置の検出に加えて、不連続部のインピーダンスのタイプ(キャパシタンス性、インダクタンス性)が識別可能
- 反射・伝送のどちらの測定にも使用可能
- 同じ周波数スパンでバンドパス・モードに比べて高い応答分解能(2倍)を実現

被試験デバイスに応じたモードの選択

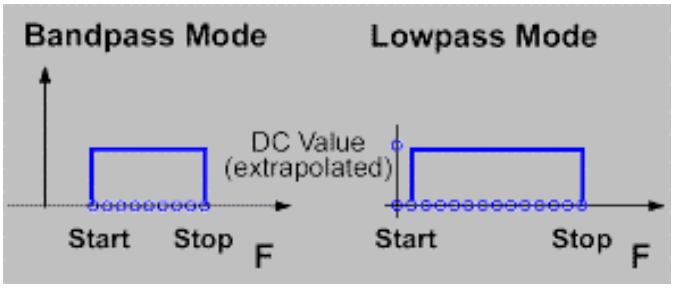
- **バンドパス・モードを選択**するのは、被試験デバイスがバンドパス、バンド・リジェクト、ハイパス・フィルタとして動作する場合です。
- **ローパス・モードを選択**するのは、被試験デバイスがDCまでのすべての周波数を通過する場合です。デバイス応答が周波数スパンの低域でロールオフを示す場合でも、ローパス・モードが使用できます。



周波数レンジとデータ・ポイント

バンドパス・モード

- 本器のレンジ内の任意のスタート/ストップ周波数が選択できます。
- 本器はスタート周波数とストップ周波数の間でデータを収集します。
- インパルス応答のみが対象です。

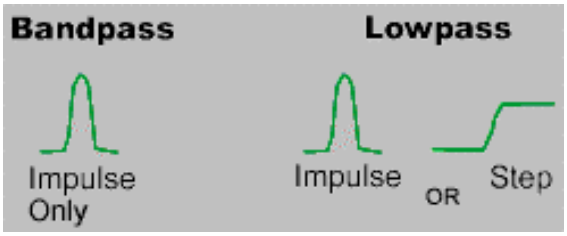


ローパス・モード

- 周波数領域の最初のいくつかのポイントからDC項を外挿する必要があります。
- データ・ポイントは等間隔でなければなりません。
- 測定周波数はすべて、スタート周波数の高調波に設定されます。
- ストップ周波数 = N × スタート周波数(N = ポイント数)という式により、Nの値に応じてストップ周波数が制限されます。

注記: スタート周波数の可能な最小値は300 kHzなので、ストップ周波数は最低N × 300 kHzでなければなりません。

インパルス応答とステップ応答



本器は、被試験デバイスにインパルス関数またはステップ関数を入力したときの時間領域応答を表示することができます。

- **インパルス信号**とは、0から1に遷移し、再び0に戻る電圧波形のことです。インパルスの幅は、周波数領域測定で用いられる周波数スパンに依存します。
- **ステップ信号**とは、0から1に遷移する電圧波形です。ステップの立上がり時間は、周波数領域測定で用いられる最高周波数によって決まります。

注記: ローパス・モードのステップ応答は、不連続部のタイプを調べるのに最も便利です。この組合わせではまた、インダクタンス性とキャパシタンス性の不連続部に関して、よく知られたTDR形式の表示が得られます。

- 周波数応答データに対して、選択した変換に応じてデバイスのステップ、インパルス、バンドパスのいずれかの応答が計算されます。
- データは時間領域応答(TDR形式の表示)として示されます。

インパルスの幅と立上がり時間が周波数スパンによってどのように変化するかの詳細については、分解能とレンジを参照してください。

有用なデータ・フォーマット

バンドパス・モードに有用なフォーマット

- **リニア振幅**
  - 反射測定の場合、このフォーマットは測定対象周波数レンジにおける反射係数(ρ)の平均リニア振幅を示します。
  - 伝送測定の場合、このフォーマットは測定周波数レンジにおける伝送経路の平均伝送係数(□)を示します。
  - このフォーマットは、互いに同程度の大きさの応答を観察するのに便利です。

- **対数振幅**
  - 反射測定の場合、このフォーマットはリターン・ロス(dB)の単位による応答を示します。表示される値は、測定周波数レンジにおける不連続部の平均リターン・ロスを表します。
  - 伝送測定の場合、このフォーマットは伝送損失または利得(dB)の単位による応答を示します。これは測定周波数レンジにおける伝送経路の平均損失を表します。
  - このフォーマットは、ダイナミック・レンジが広い伝送応答を観察するのに便利です。
- **定在波比**
  - 反射測定の場合、このフォーマットは測定周波数レンジにおける不連続部の平均定在波比(SWR)を示します。

ローパス・モードで有用なフォーマット

- **実数**
  - このフォーマットは、デバイス応答を実数の単位で示します。
  - 被試験デバイスの測定応答がDCに達する場合、ローパス・モードを使用します。
  - 被試験デバイスの測定応答がDCに存在しない場合、バンドパス・モードを使用します。

ローパス・モードの設定

マウス手順

1. 測定の設定を選択します。測定の設定を参照してください。
2. 周波数スパンとポイント数を計算します。

ストップ周波数 = N×スタート周波数(N = ポイント数)

注記: スタート周波数は300 kHzなので、ストップ周波数はN×300 kHz以上でなければなりません。

3. **Sweep**メニューで**Number of Points**をクリックし、値を入力します。
4. 周波数スパンの設定:
   a. **Channel**メニューで**Start/Stop**をクリックします。
   b. スタート周波数は300 kHzのままにし、ストップ周波数を入力します。
5. **Trace**メニューで**Transform**をクリックし、変換モードを選択します。
6. **Set Freq. Low Pass**をクリックします。これにより、すべての周波数が自動的に設定されます。
   a. ストップ周波数が指定した値から大幅に変化する場合は、初期値がローパス・モードの最低周波数レンジより小さかったことが原因です。この場合、ポイント数を減らし、周波数スパンを設定し直して、このステップを繰り返します。
7. **Transform**チェック・ボックスをクリックします。

ローパス・モードの設定

フロントパネル・キー手順

1. 測定の設定を選択します。測定の設定を参照してください。
2. 周波数スパンとポイント数を計算します。

ストップ周波数 = N×スタート周波数(N = ポイント数)

注記: スタート周波数は300 kHzなので、ストップ周波数はN×300 kHz以上でなければなりません。

3. **CHANNEL**ブロックで**Sweep Setup**を押します。
4. **F2 (Points)**を押して、前のステップで計算したポイント数を入力します。
5. 周波数スパンの設定:
   a. **CHANNEL**ブロックで**Start/Stop**を押します。
   b. スタート周波数は300 kHzのままにしておきます。
   c. **F2 (Stop)**を押してストップ周波数を入力します。
6. **COMMAND** ブロックで**Menu/Dialog**を押します。
7. **NAVIGATION**ブロックで右タブと矢印を押して**Trace**メニューに移動し、**Transform**まで移動します。**Click**を押します。
8. 右タブを押して**Transform Mode**に移動し、矢印を使ってモードを選択します。
9. 右タブを押して**Set Freq. Low Pass**に移動します。**Click**を押します。これにより、すべての周波数が自動的に設定されます。
   a. ストップ周波数が指定した値から大幅に変化する場合は、初期値がローパス・モードの最低周波数レンジより小さかったことが原因です。この場合、ポイント数を減らし、周波数スパンを設定し直して、このステップを繰り返します。
10. 右タブを押して**Transform**チェック・ボックスに移動し、**Click**を押します。

バンドパス・モードの設定

マウス手順

1. 測定の設定を選択します。測定の設定を参照してください。
2. **Trace**メニューで**Transform**をクリックし、**Band Pass**変換モードを選択します。

フロントパネル・キー手順

1. 測定の設定を選択します。測定の設定を参照してください。
2. **COMMAND** ブロックで**Menu/Dialog**を押します。
3. **NAVIGATION**ブロックで右タブと矢印を押して**Trace**メニューに移動し、**Transform**まで移動します。**Click**を押します。
4. 右タブを押して**Transform Mode**に移動し、矢印を使ってBand Passモードを選択します。
5. 右タブを押して**Transform**チェック・ボックスに移動し、**Click**を押します。

プログラミング

- Concepts
- COM
- Objects
- Properties
- Methods
- Events
- SCPI

Programming
COM Fundamentals

The following is terms are discussed in this topic:

- **Objects**
- **Collections**
- **Methods**
- **Properties**
- **Events**

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.

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.

For more information on working with objects, see Getting a Handle to an Object.

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()
    segs.DwellTime = 30e-3
 End Sub
```

Methods

A method is an action that is performed on an object. For example, **Add** is a method that applies to the Channel object. The following procedure uses the Add method to add a new channel named **NewChan.**

```
Sub AddChan(newChan as String)
    Chan.Add NewChan
 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.

To change the value of a property, follow the reference to an object with:

- a period (.)
- the property name
- an equal sign (=)
- the new property value.

For example, the following statement sets the IFBandwidth of a channel.

```
Chan.IFBandwidth = 1KHz
```

You can also read the current value of a property. The following statement reads the current IFBandwidth 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.

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**

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 in order to change or read its settings. The following analogy illustrates this:

A car could be called an object. Like all objects, it has many properties. One of its **properties** is "**Color**". You can read (by looking) or set (by painting) the color property of a car object. However, the color **value** (such as **Red** or **Green**) depends on what SPECIFIC car object you are referring to. "Car" is a class of objects. You can only read or set the properties of a specific car object; not the entire car class. Therefore, before reading or setting an object's properties, you need to get "a handle" to a specific object.

You can have handles to many objects at the same time. It does NOT have to be the Active or Selected object.

Note: This process is also called "getting an instance of an object", and "returning an object".

There are two steps for getting a handle to analyzer objects:

1. Declaring a Variable As an Object
2. Assigning an Object to the Variable

Note: Before doing this, you must first register the analyzer's type library on your PC. See Connecting to the Analyzer

Declaring a Variable As an Object

Note: The examples in these topics use the Visual Basic Programming Language. The **Green** text following an apostrophe (') is a comment.

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. For example, all three of the following declarations are valid:

- Dim RFNA ' Declare RFNA as Variant data type.
- Dim RFNA As Object ' Declare RFNA as Object data type.
- Dim RFNA As AgilentPNA835x.Application ' Declare RFNA As AgilentPNA835x.Application type

Note: If you use a variable without declaring it first, the data type of the variable is Variant by default.

If you know the specific object type, you should declare the object variable as that object type. Declaring specific object types provides automatic type checking, faster code, and improved readability.

Assigning an Object to a Variable

The first and most important object to assign to a variable is the Application object (the Network Analyzer). When assigning an object to a variable, use the **Set** keyword before the object variable that was declared previously. In the following example, "RFNA" is the variable we declared in the previous examples. So we assign the current AgilentPNA835x Application to "RFNA".

Set RFNA = AgilentPNA835x.Application

However, because the AgilentPNA835x object is the Application server, 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 Sharing Files between your PC and the Analyzer.

For example, the following statements would create an instance of the Analyzer object.

```
Dim RFNA AS AgilentPNA835x.Application
  Set RFNA = CreateObject("AgilentPNA835x.Application", "Analyzer46")
```

Note: These statements will start the Analyzer application if it is not already running on your instrument.

Once created, you can treat an object variable exactly the same as the object to which it refers. You can set or return the properties of the object or use any of its methods. For example:

```
RFNA.Visible = True 'Makes the Network Analyzer Application visible on the screen
```

Navigating the Object Hierarchy

To read and set properties of objects below the Analyzer Application, you do not have to "Create" the object as we did with the Application. But you DO have to navigate the object model hierarchy. (Refer to the Analyzer Object Model).

You could do refer to an object in the hierarchy directly, without declaring and assigning a variables. The following example navigates through the Application object to the Active Measurement which is a 'child' object of the Application. (The ACTIVE measurement is the measurement that is acted on if you change settings from the front panel.)

```
Application.ActiveMeasurement.SmoothingAperture = 10
```

You can see that this method makes for a very long statement. Making additional changes to the Active Measurement would require equally long statements.

The following example gets a handle to the Active Measurement object by assigning it to a variable.

The first step is to **Declare an object variable:**

```
Public meas AS Measurement
```

The next step is to **Set the object variable:**

We already assigned an instance of the (analyzer) Application to the variable **RFNA**. Therefore, we can use the RFNA variable to refer to a specific instance of the Application object.

```
Set meas = RFNA.ActiveMeasurement
```

The variable **meas** now contains a handle to the Application object (RFNA) **and** the ActiveMeasurement object. We can now set properties of the ActiveMeasurement as follows:

```
meas.SmoothingAperature = 10
```

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 'the collection of all measurements currently on the analyzer
  Set meass = RFNA.Measurements
```

Then you can iterate through the entire collection of measurements to read or set properties or execute methods.

```
meass.Format = naLinMag
```

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.

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-DCOMプログラミングの設定

COMプログラムの開発または実行前には、PCとアナライザとの通信を 確立する必要があります。通信を呼び出すと、アナライザへのアクセスが 得られます。次に、PC上でアナライザのタイプ・ライブラリを登録 します。

PNAは出荷時には、**Everyone(全員)** がCOM/DCOMを介したPNAアプリケーションの起動/アクセス許可を持つようにデフォルト設定されています。**Everyone**が指すユーザのレンジは、PNAが **ドメイン**のメンバであるか、**ワークグループ** のメンバであるか(どちらか一方を選ぶ必要があります)によって異なります。

注記: DCOM (Distributed Component Object Model)は、アナライザ・アプリケーションにリモートPCからアクセスする場合を いいます。
COMは、アナライザにアナライザのPCからアクセスする場合をいいます。

ワークグループ

ワークグループを設定するには、**PNA 管理者**がワークグループ名を宣言し、PNAを ワークグループのメンバとして宣言します。ワークグループの作成やメンバの管理に、 ネットワーク管理者は不要です。

Everyoneには、PNAでログオン・アカウントが 与えられたユーザのみが含まれます。

PNAはデフォルトでは、WORKGROUPという名前のワークグループのメンバとして構成されています。

注記: 新しいユーザの ログオン・アカウントを設定するには、ユーザの 追加を参照してください。
DCOMアクセスの場合、ユーザのアカウント名とパスワードは、 PCのログオン・アカウント名およびパスワードと完全に一致していなければなりません。

ドメイン

ドメインとは通常、コンピュータの組織化された大規模な グループです。ネットワーク管理者が、ドメインの保守を行い、 どのマシンがドメインのメンバであるかを管理します。

Everyoneには、ドメインのメンバである ユーザが含まれます。ログオン・アカウントを持つユーザも、アナライザに アクセスできます。

まとめ:

- **ワークグループ** は保守が不要ですが、ワークグループでDCOMアクセスが許可されるのは、PNAに対するログオン・アカウントを持つ ユーザだけです。
- **ドメイン** は管理者が必要ですが、ドメインの全メンバおよびアナライザに対する ログオン・アカウントを持つユーザに、PNAへのDCOMアクセス権があります。

セキュリティの次の段階として、 (**Everyone**でなく) **選ばれた**ドメインおよびワークグループの ユーザだけにアナライザのDCOM**アクセス**機能と**起動** 機能を許可します。

注記: この手順を 実行する前に、ユーザはまず、PNAのログオン・アカウントを取得する必要があります。ユーザの 追加を参照してください。

この手順は、以下に使用します。

- 選ばれたユーザのみ (全員でなく)に、アナライザへのリモート(DCOM)アクセスと起動 を許可する。アナライザのアプリケーションがまだオープンしていない場合、起動機能によって アプリケーションがスタートします。
- アナライザへのDCOMアクセスを持つか確認する。

アクセス機能と起動機能に対して以下を実行します。

1. PNAで、 Windows **Start**ボタンをクリックします。
2. **Run**をクリックします。
3. **Open:** ボックスに、**dcomcnfg**と入力します。
4. **OK**をクリックします。
5. Distributed COM Configuration Propertiesウィンドウで、 Applicationsリストの **Agilent PNA Series**を クリックします。次に、**Properties...** (ボタン)をクリックします。
6. **Security** タブをクリックします。

アクセス機能

特定のユーザにPNAアプリケーションへの**DCOM アクセス**を許可するには、以下の手順に従ってPNAを設定します。

Agilent PNA Series Properties ダイアログ・ ボックスで:

1. **Use custom access permissions**をクリックします。
2. **(Use custom access permissions)**の横の **Edit...**をクリックします。
3. Registry Value Permissionsで、 **Everyone**を選択します。
4. **Remove**をクリックします。
5. **Add**をクリックします。
6. PNAへのアクセス権を与える対象として、1つ以上のグループまたは 特定のユーザを選択できます。
    1. グループにアクセス権を与えるには、リストからグループを選択します。
    2. 特定のユーザにアクセス権を与えるには、**Show users**または**Members**をクリックした後、リストから名前を選択します。
7. **Add**をクリックします。
8. **OK**をクリックします。

起動許可

選ばれたユーザがPNAアプリケーションを**起動 (スタート)**できるようにするには、以下の手順に従ってPNAを設定します。 この手順は、PNAへのアクセス権を与える場合と同じです。

Agilent PNA Series Propertiesで:

1. **Use custom launch permissions**をクリックします。
2. **(Use custom launch permissions)**の横の **Edit...**をクリックします。
3. Registry Value Permissionsで、 **Everyone**を選択します。
4. **Remove**をクリックします。
5. **Add**をクリックします。
6. PNAの起動許可を与える対象として、1つ以上のグループまたは 特定のユーザを選択できます。
    1. グループに起動許可を与えるには、リストからグループを選択します。
    2. 特定のユーザに起動許可を与えるには、**Show users**または**Members**をクリックした後、リストから名前を選択します。
7. **Add**をクリックします。
8. **OKをクリックします。**

Agilent PNA Series Propertiesで:

1. **Identityタブ**をクリックします。
2. **The interactive user**をクリックします。 この機能は、PNA COMのイベントをサポートします。

PC上でアナライザのタイプ・ライブラリを登録する

タイプ・ライブラリには、アナライザのオブジェクト・モデルが含まれています。PCに、 オブジェクト・モデルの位置を記録しておくRegistryファイルがあります。 したがって、コードの開発やプログラムの実行に使用するアナライザのタイプ･ライブラリを PC上で登録する必要があります。 COMプログラムを実行する前に、デザイン時にタイプ・ライブラリを登録しておくと より効率的です。

先に進む前に以下の2つの項目を実行します。

1. PCとアナライザの両方を LANに接続します。
2. ドライブをアナライザにマップするか、フロッピー・ディスクや 他の媒体にタイプ・ライブラリ・ファイルをコピーできます。ドライブをマッピングするには、 ドライブ・マッピングを参照してください。

注記: PCでタイプ・ライブラリを 登録するには、PCの管理者としてログオンする 必要があります。

この手順では、ドライブがPCからアナライザにマッピングされているものとします。 この手順では、以下を実行します。

- PCにネットワーク・ アナライザ・アプリケーションを登録します。
- PCに proxystub (835xps.DLL)をコピーして、登録します。
- PCにタイプ・ライブラリ(835x.tlb)を コピーして、登録します。

1. PCでWindowsエクスプローラを使い、アナライザのC:ドライブを探します。 これは、ドライブをマッピングしたときに割り当てたドライブ名なので、ドライブ名が "C:" であるとは限りません。
2. **Program Files ¥ Agilent ¥ Network Analyzer ¥ Automation**に移動します。
3. **Automation.vbs**をダブルクリックします。

注記: "Open With"ダイアログ・ボックスが表示される場合は、PCに Windowsスクリプト・ホストがインストールされていません。Windowsスクリプト・ホストは、 Microsoftの次のウェブサイトから無料でダウンロードすることができます:
http://www.microsoft.com/msdownload/vbscript/scripting.asp

4. Visual Basicスクリプト・プログラムがアナライザ名をたずねてきます(アナライザ名は、 アナライザのControl Panel ¥ System ¥ ネットワークID ¥ コンピュータのフルネームでわかります)。プロンプトにアナライザ名を入力します。
5. スクリプトを実行した後、OKをクリックします。これで、アナライザのタイプ・ライブラリが PCに登録されました。

注記: プログラミング環境によっては、 PC上でタイプ・ライブラリへの参照を設定する必要があります。

ネットワーク・アナライザ・アプリケーションを本器にインストールすると、 アプリケーションおよび供給されたすべてのオブジェクトがマシンに登録されます。 本器を、リモート・サーバとして動作するよう設定しておく必要があります。

アナライザが正しく動作しない場合は、以下の手順に従って、 アプリケーションと付随するライブラリをアナライザに 手動で再登録してみてください。

注記: 以下の手順は、順番どおりに 実行する必要があります。

PNAネットワーク・アナライザにマッピングされた ドライブを持つPCから:

C:¥Program Files¥Agilent¥NetworkAnalyzerで835x.exeを探します。

1. **835x.exe**を選択して右クリックします。
2. **Register COM Server**を選択して、アプリケーションを登録します。
3. **835xps.DLL**を選択して右クリックします。
4. **Register COM Server**を選択します。

この手順を終了したら、「アクセスの 手順」(dcomcnfgの実行)に移ります。

COM Data Types

The Analyzer 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**

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 supply a data type, 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 analyzer provides and receives Variant data in several instances because there a programming languages that cannot send or receive "typed" data. Variant data transfers at a slower rate than "typed" data.

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**

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.

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Sev		**C**	**R**																	**Facility**										**Code**	

Sev	Severity Code
C	Customer code flag (not used)
R	Reserved bit (not used)
Facility	Category code
Code	Event ID

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.

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.

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¥Agilent¥Network Analyzer¥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), and

"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.

```
// An example program to illustrate the use of #import to bind to the
  // PNA type library.
  //
  #include "stdafx.h"
  #include "stdio.h"
  #include "math.h"

  ////////////////////////////////////////////
  // import the network analyzer type library
  ////////////////////////////////////////////
  #import "C:¥Program Files¥Agilent¥Network Analyzer¥835x.tlb" no_namespace,
named_guids
  ////////////////////////////////////////////
  // include the error definitions for the PNA so we can implement
  // error handling.
  ////////////////////////////////////////////
  #include "C:¥Program Files¥Agilent¥Network Analyzer¥errorsystemmessage.h"

  IApplicationPtr pNA; // top level application pointer
  float fScalarData [1601]; // global buffer for data retrieval
  float fScalarData2[1601];
```

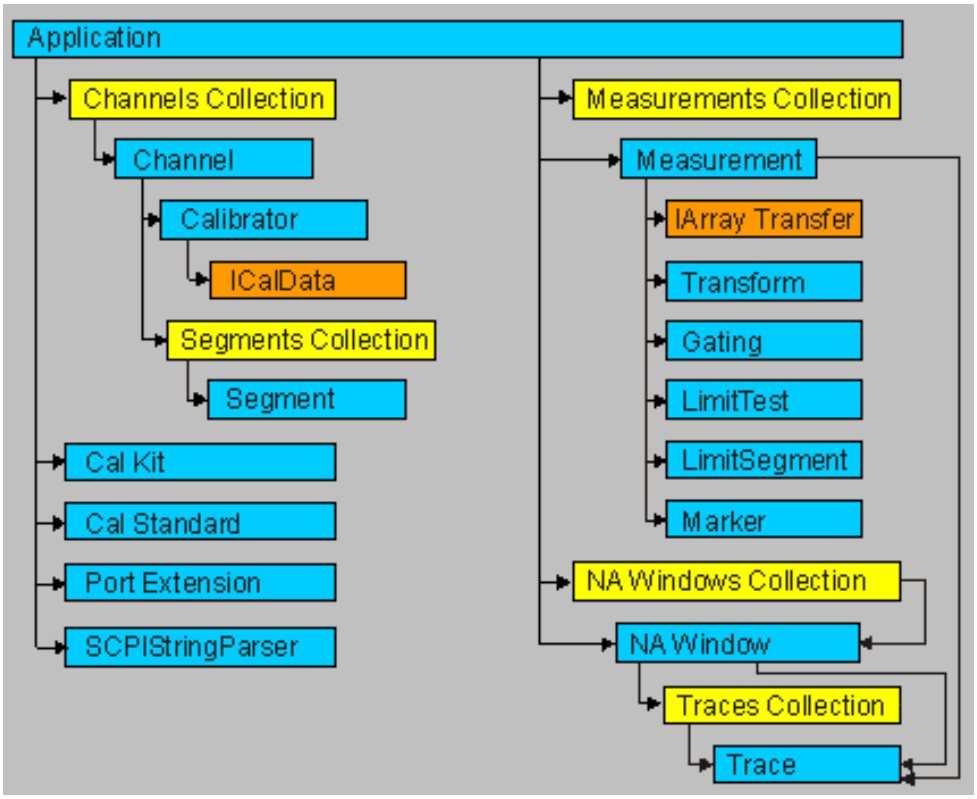
```
DWORD dwCookie;

////////////////////////////////////////////
// SetupChannel:
//
// input: pointer to the channel
//
// function: sets properties on the channel
////////////////////////////////////////////
void SetupChannel(IChannelPtr pChannel)
{
  pChannel->put_StartFrequency( 1.2E9 );
  pChannel->put_StopFrequency ( 4.2E9 );
  pChannel->put_NumberOfPoints ( 201);

}

////////////////////////////////////////////
// AcquireData:
//
// input: pointer to the channel
//
// function: single sweeps the channel
////////////////////////////////////////////
void AcquireData( IChannelPtr pChannel )
{
  pChannel->Single( TRUE );
}

////////////////////////////////////////////
// ReadData:
//
// input: pointer to the Measurement object
//
// function: reads data from the measurment's formatted
// result data buffer
////////////////////////////////////////////
void ReadScalarData(IMeasurementPtr pMeas )
{
  IArrayTransferPtr pDataTransfer;
  pDataTransfer = pMeas;
  long numVals = 1601;
  float* pData = fScalarData;

  if(pDataTransfer){

  pDataTransfer->getScalar( naMeasResult, naDataFormat_LogMag, &numVals, pData);

  for (int i = 0; i < numVals; i++)
  printf("%d¥t%f¥n",i,pData[i]);
  }
  TCHAR msg[100];
  BSTR param;
  pMeas->get_Parameter(&param);
  swprintf(msg,L"Review %s data",param);
  MessageBox(NULL,msg,L"User Message",0);
  ::SysFreeString(param);
}
```

```
void ReadComplexData(IMeasurementPtr pMeas )
{
  IArrayTransferPtr pDataTransfer;
  pDataTransfer = pMeas;
  long numVals = 1601;
  float* pReal= fScalarData;
  float* pImag = fScalarData2;

  if(pDataTransfer){

  pDataTransfer->getPairedData( naRawData, naRealImaginary, &numVals, pReal, pImag);

  for (int i = 0; i < numVals; i++)
  printf("%d¥t%f¥t%f¥n",i,pReal[i], pImag[i]);
  }
  TCHAR msg[100];
  BSTR param;
  pMeas->get_Parameter(&param);
  swprintf(msg,L"Review %s data",param);
  MessageBox(NULL,msg,L"User Message",0);
  ::SysFreeString(param);
}
////////////////////////////////////////////
// PutData:
//
// input: pointer to the Measurement object
//
// function: writes data to the measurment's raw data
// buffer
////////////////////////////////////////////
void PutData( IMeasurementPtr pMeas )
{
  IArrayTransferPtr pDataTransfer;
  pDataTransfer = pMeas;
  long numVals = 201;

  if(pDataTransfer){

  NAComplex* pComplex = new NAComplex[numVals];

  pComplex[0].Im = 0;
  pComplex[0].Re = 1;
  for (int i = 1; i < numVals; i++)
  {
  pComplex[i].Im = (float)sin(i)/i;
  pComplex[i].Re = (float)cos(i)/i;
  }

  pDataTransfer->putNAComplex( naRawData, numVals, pComplex, naDataFormat_Polar);
  delete [] pComplex;
  }
}

////////////////////////////////////////////
// printError
////////////////////////////////////////////
void printError( HRESULT hr)
{
  BSTR text;
```

```
  hr = pNA->get_MessageText ((NAEventID) hr, &text);
  MessageBox(NULL,text,L"Network Analyzer error",0);
  ::SysFreeString(text);
}


//////////////////////////////////////////////
// main
//////////////////////////////////////////////
int main(int argc, char* argv[])
{
  HRESULT hr;
  const long channel1 = 1;
  const long window1 = 1;
  const long srcport = 1;
  IMeasurementPtr pMeasurement;
  IChannelPtr pChannel;

  // initialize COM libraries
  CoInitialize(NULL);

  try {
  pNA = IApplicationPtr("AgilentPNA835x.Application.1");

  pNA->put_Visible(TRUE);
  pNA->Reset();


  pNA->CreateMeasurement (channel1, "S21",srcport, 5);
  hr = pNA->get_ActiveChannel( &pChannel);

  if (SUCCEEDED (hr))
  {
  SetupChannel( pChannel);
  AcquireData(pChannel);
  }

  hr= pNA->get_ActiveMeasurement( &pMeasurement);
  if (SUCCEEDED(hr))
  {
  pMeasurement->put_Format( naDataFormat_Polar);
  ReadScalarData( pMeasurement);
  ReadComplexData( pMeasurement);
  PutData(pMeasurement);
  }
  if (FAILED(hr))
  {
  printError(hr);
  }

  // make sure to release the remaining pointers
  // before calling CoUninitialize

  pMeasurement.Release();
  pChannel.Release();
  pNA.Release();
  }
  catch (_com_error err)
  {
  printError( err.Error() );
```

```
    }

    CoUninitialize();
    return 0;
}
```

PNA Object Model



Click on an object box to view details.

Application Object

Description

The Application object is the highest object in the analyzer 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.

This object is unique in that you must Create this object rather than just get a handle to it. See Getting a Handle to an Object.

Methods	Description
ActivateWindow	Makes a window object the Active Window.
AllowAllEvents	Monitors all events
AllowEventCategory	Monitors an event category
AllowEventMessage	Monitors an event
AllowEventSeverity	Monitors an event severity level
BuildHybridKit	Defines the user kit as port1kit + port2kit.
SetFailOnOverRange	
Channel (object)	
CreateMeasurement	Creates a new measurement.
CreateSParameter	Creates a new S-Parameter measurement.
DeleteShortCut	Removes a macro (shortcut) from the list of macros
DisallowAllEvents	Monitors NO events
DoPrint	Prints the screen to the active Printer.
ExecuteShortcut	Executes a macro (shortcut) stored in the analyzer.
GetShortcut	Returns the title and path of the specified macro (shortcut).
ManualTrigger	Triggers the analyzer when TriggerSignal = naTriggerManual.
Preset	Resets the analyzer to factory defined default settings.

PrintToFile	Saves the screen data to bitmap (.bmp) file of the screen.
PutShortcut	Puts a Macro (shortcut) file into the analyzer.
Quit	Ends the Network Analyzer application.
Recall	Restores all cal kits from disk.
RecallKits	Recalls the current state of the calibration kits on disk.
Reset	Removes all existing windows and measurements.
RestoreCalKitDefaults	Restores the factory defaults for the specified kit.
RestoreCalKitDefaultsAll	Restores the factory defaults for all kits.
Save	Saves all cal kits to disk.
SaveKits	Ends the Network Analyzer application.
SetFailOnOverRange	Causes over range values to return an error code
ShowStatusBar	Shows and Hides the Status Bar.
ShowStimulus	Shows and Hides Stimulus information.
ShowTitleBars	Shows and Hides the Title Bars.
ShowToolbar	Shows and Hides the specified Toolbar.

Properties	**Description**
ActiveCalKit	Returns a pointer to the kit identified by kitNumber.
ActiveChannel	Returns a handle to the Active Channel object.
ActiveMeasurement	Returns a handle to the Active Measurement object.
ActiveNAWindow	Returns a handle to the Active Window object.
ArrangeWindows	Sets or returns the arrangement of all the windows.
CalKitType	Sets the active cal kit.
Channels (collection)	
ExternalALC	Sets or returns the source of the analyzer leveling control.
GPIBMode	Makes the analyzer the system controller or a talker/listener.

IDString	Returns the model, serial number and software revision of the analyzer
Measurements (collection)	
MessageText	Returns a message for an eventID
NAWindows (collection)	
Options	Returns the options on the analyzer
PortExtension (object)	
ScpiStringParser (object)	
SourcePowerState	Turns Source Power ON and OFF.
SystemImpedanceZ0	Sets the analyzer impedance value
TriggerSignal	Sets or returns the trigger source.
TriggerType	Sets or returns the scope of a trigger signal.
VelocityFactor	Sets the velocity factor to be used with Electrical Delay and Port Extensions.
Visible	Makes the Network Analyzer application visible or not visible. **(Default property of this object)**
WindowState	Sets or returns the window setting of Maximized, Minimized, or Normal.

Events	Description
OnCalEvent	Triggered by a calibration event.
OnChannelEvent	Triggered by a channel event.
OnDisplayEvent	Triggered by a display event.
OnHardwareEvent	Triggered by a hardware event.
OnMeasurementEvent	Triggered by a measurement event.
OnSCPIEvent	Triggered by a SCPI event.
OnSystemEvent	Triggered by a system event.
OnUserEvent	For future use

Calibrator Object

Description

The Calibrator object is a child of the channel. It is a vehicle to perform calibration.

There must be a measurement present for the calibrator to use or you will receive an error (no measurement found). 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. And likewise for a response measurement.

With the calibrator object, you can take a number of approaches to calibration.You can collect data yourself and download it to the ACQUISITION buffer. The acquisition buffer holds the actual measured data for each standard.

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.AcquireCalStandard
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.

The ICalibrator methods can be used from VEE and VBScript as they use automation safe Variant Data. Use the ICalData interface for typed array passing.

Method	Description
AcquireCalStandard	Cause the analyzer to measure a calibration standard.
CalculateErrorCoeffecients	Generates Error Terms from standard and actual data in the error correction buffer.
DoECAL1Port	Completes a 1 port ECAL
DoECAL2Port	Completes a 2 port ECAL
GetECALModuleInfo	Returns information about the attached module
getErrorTerm	Retrieves error term data for the active calibration.
getStandard	Retrieves calibration data from the aquisition data buffer (before error-terms are applied).
putErrorTerm	Puts error term data into the error-correction buffer for the active calibration.
putStandard	Puts data into the acquisition data buffer (before error-terms are applied)
setCalInfo	Specifies the type of calibration and prepares the internal state for the rest of the calibration.

Property	Description
AcquisitionDirection	Specifies the direction in a 2-Port cal
Simultaneous2PortAcquisition	Allows the use of 2 sets of standards at the same time.

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 either the **application object (app**.calKitType) or the **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. However, to retrieve a pointer to the cal kit, use **app.ActiveCalKit**.

The calKit object behaves somewhat differently from other objects in the system in that you can only have a pointer to **one** cal kit (which is also the active calkit).

Therefore, when you change the calkitType (from either of these objects) you may also be changing the object to which you may have several references. This is different from the behavior for most other objects in the system.

For example, the following code specifies two calKitType and, in turn, assigns the "Active cal kit" to two different variables: ck1 and ck2.

```
Dim app As AgilentPNA835x.Application
 Dim ck1 As calKit
 Dim ck2 As calKit

 Private Sub Form_Load()
 Set app = CreateObject("AgilentPNA835x.Application", "analyzerName")
 app.CalKitType = naCalKit_85032B_N50
 Set ck1 = app.ActiveCalKit

 app.CalKitType = naCalKit_85038A_7_16
 Set ck2 = app.ActiveCalKit

 Print "ck1: " & ck1.Name
 Print "ck2: " & ck2.Name
 End Sub
```

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: 7-16 Model 85038
 ck2: 7-16 Model 85038
```

Method	Description
getCalStandard	Returns a handle to a calibration standard for modifying its definitions.

Property	Description
CalKitType	Sets and returns the cal kit type
Name	Sets and returns the name of the cal kit
PortLabel	Labels the ports for the kit; only affects the cal wizard annotation.
StandardForClass	Maps a standard device to a cal class.

CalStandard Object

Description

Contains all of the settings that are required to modify a calibration kit. Get a handle to a standard with the calkit.GetCalStandard Method.

Method	Description
None	

Property	Description
C0	Sets and Returns the C0 (C-zero) value (the first capacitance value) for the calibration standard.
C1	Sets and Returns the C1 value (the second capacitance value) for the calibration standard.
C2	Sets and Returns the C2 value (the third capacitance value) for the calibration standard.
C3	Sets and Returns the C3 value (the fourth capacitance value) for the calibration standard.
Delay	Sets and Returns the electrical delay value for the calibration standard.
L0	Sets and Returns the L0 (L-zero) value (the first inductance value) for the calibration standard.
L1	Sets and Returns the L1 value (the second inductance value) for the calibration standard.
L2	Sets and Returns the L2 value (the third inductance value) for the calibration standard.

L3	Sets and Returns the L3 value (the third inductance value) for the calibration standard.
Label	Sets and Returns the label for the calibration standard.
loss	Sets and Returns the insertion loss for the calibration standard.
Maximum Frequency	Sets and Returns the maximum frequency for the calibration standard.
Medium	Sets and Returns the media type of the calibration standard.
Minimum Frequency	Sets and Returns the minumum frequency for the calibration standard.
Type	Sets and Returns the type of calibration standard.
Z0	Sets and Returns the characteristic impedance for the calibration standard.

Channel Object

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.

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 chan as Channel
 '
  Set chan = pna.ActiveChannel
```

or

```
Set chan = pna.Channels( n )
```

The first method will return the channel object that is driving the active measurement. When you ask for the ActiveChannel, you get the channel 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. Here's one way:

```
Pna.CreateMeasurement( ch1, "S11", port1, window2)
```

Here's another:

```
Pna.Channels.Add (ch2)
```

The latter will have no visible effect on the analyzer. It will simply create channel 2 if it does not already exist.

Method	Description
Abort	Aborts the current measurement sweep on the channel.
AveragingRestart	Clears and restarts averaging of the measurement data.
Continuous	The channel continuously responds to trigger signals.
GetXAxisValues	Returns the channel's X-axis values into a dimensioned Variant array.
GetXAxisValues2	Returns the channel's X-axis values into a dimensioned NON-Variant array.
Hold	Puts the Channel in Hold - not sweeping.
Next_IFBandwidth	A function that returns the Next higher IF Bandwidth value.
NumberOfGroups	Sets the Number of trigger signals the channel will receive.
Preset	Resets the channel to factory defined settings.
PreviousIFBandwidth	Returns the previous IF Bandwidth value.
Single	Channel responds to one trigger signal from any source (internal, external, or manual). Then channel switches to Hold.

Property	Description
AlternateSweep	Sets sweeps to either alternate or chopped.
Attenuator	Sets or returns the value of the attenuator control for the specified port number.
AttenuatorMode	Sets or returns the mode of operation of the attenuator control for the specified port number.
Averaging	Turns trace averaging ON or OFF for all measurements on the channel.
AveragingFactor	Specifies the number of measurement sweeps to combine for an average.
Calibrator (object)	
centerFrequency	Sets or returns the center frequency of the channel.
channelNumber	Returns the Channel number of the Channel or Measurement object.
CouplePorts	Turns ON and OFF port power coupling.
CWFrequency	Set the Continuous Wave (CW) frequency.
DwellTime	Sets or returns the dwell time for the channel.
FrequencySpan	Sets or returns the frequency span of the channel.
IFBandwidth	Sets or returns the IF Bandwidth of the channel.
NumberOfPoints	Sets or returns the Number of Points of the channel.

Parent	Returns a handle to the parent object of the channel.
PowerSlope	Sets or returns the Power Slope value.
ReceiverAttenuator	Sets or returns the value of the specified receiver attenuator control.
Segments (collection)	
StartFrequency	Sets or returns the start frequency of the channel.
StartPower	Sets the start power of the analyzer when sweep type is set to Power Sweep.
StopFrequency	Sets or returns the stop frequency of the channel.
StopPower	Sets the Stop Power of the analyzer when sweep type is set to Power Sweep.
SweepGenerationMode	Sets the method used to generate a sweep: continuous ramp (analog) or discrete steps (stepped).
SweepTime	Sets the Sweep time of the analyzer.
SweepType	Sets the type of X-axis sweep that is performed on a channel.
TestPortPower	Sets or returns the RF power level for the channel.
TriggerMode	Determines the measurement that occurs when a trigger signal is sent to the channel.
UserRangeMax	Sets the stimulus stop value for the specified User Range.
UserRangeMin	Sets the stimulus start value for the specified User Range.

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. For more information, see Collections in the Analyzer.

Methods	Description
Add	An alternate way to create a measurement.
Item	Use to get a handle on a channel in the collection.

Properties	Description
Count	Returns the number of channels in the analyzer.
Parent	Returns a handle to the current Application.

Gating Object

Description

Contains the methods and properties that control Time Domain Gating

Methods	Description
None	

Property	Description
Center	Sets or returns the Center time.
Shape	Specifies the shape of the gate filter.
Span	Sets or returns the Span time.
Start	Sets or returns the Start time.
State	Turns an Object ON and OFF.
Stop	Sets or returns the Stop time.
Type	Specifies the type of gate filter used.

IArrayTransfer Object

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 default IMeasurement Interface.

Method	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.

Property	Description
None	

ICalData Object

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.

Method	Description
getErrorTermComplex	Retrieves error term data
getStandardComplex	Retrieves calibration data from the aquisition data buffer (before error-terms are applied).
putErrorTermComplex	Puts error term data
putStandardComplex	Puts calibration data into the aquisition data buffer (before error-terms are applied).

Property	Description
None	

Limit Test Collection

Description

Child of the **Measurement** Object. A collection that provides a mechanism for iterating through the Measurement's LimitSegment objects (Limit Lines). The collection has 100 limit lines by default.

The only way to get a handle to an individual limit line is by using the LimitTest collection. You can either **1)** set the property directly, or 2) set a variable a limit line in the LimitTest collection.

Examples **1)**LimitTest(4).BeginResponse=.5
2)Set lim2 = Application.Measurement.LimitTest(4)

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.

LimitSegment Object

Description

The LimitSegment object is an individual limit line. The only way to get a handle to an individual limit line is by using the LimitTest collection. You can either **1)** set the property directly, or **2)** set a variable a limit line in the LimitTest collection.

Examples **1)**LimitTest(4).BeginResponse=.5
2)Set lim2=Application.Measurement.LimitTest(4)

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.

Marker Object

Description

Contains the methods and properties that control Markers.

To turn ON a marker, you get a handle to the marker. You get a handle to a marker through the measurement object. (There is no markers collection).

If it is not already, this command will activate marker 1.

Measurement.marker(1).Format = naLinMag

You can also set the marker object to an object variable:

```
Dim m1 As Marker
  Set m1 = meas.marker(1)
```

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.
2. Marker object properties override the Measurement object properties. For example, **meas.MarkerFormat = naLinMag** applies formatting to all markers. You can then override that setting for an individual marker by specifying **mark.Format = naLogMag** on the marker object.

Methods	Description
Activate	Makes an object the Active Object.
SearchMax	Searches the marker domain for the maximum value.
SearchMin	Searches the marker domain for the minimum value.
SearchNextPeak	Searches the marker's domain for the next largest peak value.
SearchPeakLeft	Searches the marker's domain for the next VALID peak to the left of the marker.
SearchPeakRight	Searches the marker's domain for the next VALID peak to the right of the marker.
SearchTarget	Searches the marker's domain for the target value.
SearchTargetLeft	Moving to the left of the marker position, searches the marker's domain for the target value.
SearchTargetRight	Moving to the right of the marker position, searches the marker's domain for the target value.

SetCenter	Changes the analyzer's center frequency to the X-axis position of the marker.
SetCW	Changes the analyzer to sweep type CW mode and makes the CW frequency the marker's frequency.
SetElectricalDelay	Changes the measurement's electrical delay to the marker's delay value.
SetReferenceLevel	Changes the measurement's reference level to the marker's Y-axis value.
SetStart	Changes the analyzer's start frequency to the X-axis position of the marker.
SetStop	Changes the analyzer's stop frequency to the X-axis position of the marker.

Property	Description
Bucket Number	Marker data point number
DeltaMarker	Makes a marker relative to the reference marker
Format	Linear, SWR, and so forth
Interpolated	Turn marker interpolation ON and OFF
Number	Read the number of the active marker
PeakExcursion	Sets and reads the peak excursion value for the specified marker.
PeakThreshold	Sets peak threshold for the specified marker.
SearchFunction	Emulates the Tracking function in the marker search dialog box.
Stimulus	Sets and reads the X-Axis value of the marker.
Target Value	Sets the target value for the marker when doing Target Searches.

Tracking	The tracking function finds the selected search function every sweep.
Type	Sets and reads the marker type.
UserRange	Assigns the marker to the specified User Range.
UserRangeMax	Sets the stimulus stop value for the specified User Range.
UserRangeMin	Sets the stimulus start value for the specified User Range.
Value	Reads the Y-Axis value of the marker.

Measurement Object

Description

The Measurement object is probably the most used object in the 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 in that a channel 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 determines what ever else is active. The active measurement is best described as 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.

You can access two other objects through the Measurement object: markers and limit test. 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 = pna.ActiveMeasurement
  Meas.marker(1).Stimulus = 900e6
  Meas.LimitTest.State = true ' on
```

Methods	Description
Activate	Makes an object the Active Object.
ActivateMarker	Makes a marker the Active Marker.
ChangeParameter	Changes the parameter of the measurement.
DataToMemory	Stores the active measurement into memory.

Delete	Deletes the measurement object.
DeleteAllMarkers	Deletes all of the markers from the measurement.
DeleteMarker	Deletes a marker from the active measurement.
GetData	Retrieves Complex data from analyzer memory
GetFilterStatistics	Returns all four Filter Statistics
GetReferenceMarker	Returns a handle to the reference marker.
GetTraceStatistics	Returns the Trace Statistics.
InterpolateMarkers	Turns All Marker Interpolation ON and OFF for the measurement.
putDataComplex	Puts complex data into one of five data buffers.
putDataScalar	Puts formatted variant data into the measurement results buffer.
SearchFilterBandwidth	Searches the domain with the current BW target.

Properties	**Description**
ActiveMarker	Returns a handle to the Active Marker object.
BandwidthTarget	The insertion loss value at which the bandwidth of a filter is measured.
BandwidthTracking	Turns Bandwidth Tracking function ON and OFF.
CalibrationType	Set or get the calibration type for the measurement.
channelNumber	Returns the channel number of a measurement.
ElectricalDelay	Sets electrical delay.
ErrorCorrection	Set or get the state of error correction for the measurement.
FilterBW	Returns the results of the SearchBandwidth method.
FilterCF	Returns the Center Frequency result of the SearchBandwidth method.
FilterLoss	Returns the Loss value of the SearchBandwidth method.
FilterQ	Returns the Q (quality factor) result of the SearchBandwidth method.

Format	Sets display format.
Gating (object)	
InterpolateCorrection	Turns ON and OFF the calculation of new error terms when stimulus values change.
LimitTest (collection)	
LimitTestFailed	Returns the results of limit testing
marker (object)	
MarkerFormat	Sets or returns the format of all the markers in the measurement.
Mean	Returns the mean value of the measurement.
Name	Sets or returns the name of the measurement.
NAWindow (object)	
Number	Returns the number of the measurement.
Parameter	Returns the measurement Parameter.
PeakToPeak	Returns the Peak to Peak value of the measurement.
PhaseOffset	Sets the Phase Offset for the active channel.
ReferenceMarkerState	Turns the reference marker ON or OFF
ShowStatistics	Displays and hides the measurement statistics (peak-to-peak, mean, standard deviation) on the screen.
Smoothing	Turns ON and OFF data smoothing.
SmoothingAperture	Specifies or returns the amount of smoothing as a ratio of the number of data points in the measurement trace.
StandardDeviation	Returns the standard deviation of the measurement.
StatisticsRange	Sets the User Range number for calculating measurement statistics.

Trace (object)	
TraceMath	Performs math operations on the measurement object and the trace stored in memory.
Transform (object)	
View	Sets (or returns) the type of trace displayed on the screen.

Measurements Collection

Description

A collection object that provides a mechanism for iterating through the Application measurements. See Collections in the Analyzer.

Methods	**Description**
Add	Adds a Measurement to the collection.
Item	Use to get a handle to a channel 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 the contains the graticule, or what is written on the display.

Methods	Description
Autoscale	Autoscales all measurements in the window.
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.

Property	Description
ActiveTrace	Sets a trace to the Active Trace.
Title	Writes or reads a custom title for the window.
TitleState	Turns ON and OFF the window title.
Traces (collection)	
WindowState	Maximizes or minimizes a window.

NAWindows Collection

Description

A collection object that provides a mechanism for iterating through the Application windows. See Collections in the Analyzer.

Methods	Description
Add	Adds a window to the NAWindows collection.
Item	Use to get a handle to a channel 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 Extensions Object

Description

Contains the methods and properties that control Port Extensions.

Methods	Description
None	

Property	Description
Input A	Sets the Input A extension value.
Input B	Sets the Input B extension value.
Port 1	Sets the Port 1 extension value.
Port 2	Sets the Port 2 extension value.
State	Turns Port Extensions ON and OFF.

SCPIStringParser Object

Method	Description
Parse	Provides the ability to send a SCPI command from within the COM command.

Properties	Description
None	

Segment Object

Description

Contains the methods and properties that affect a sweep segment.

You can get a handle to a sweep segment through the segments collection.[**segments.item(n)**.]

Methods	Description
None	

Property	Description
centerFrequency	Sets or returns the center frequency of the segment.
DwellTime	Dwell time value.
FrequencySpan	Sets or returns the frequency span of the segment.
IFBandwidth	Sets or returns the IF Bandwidth of the segment.
NumberOfPoints	Sets or returns the Number of Points of the segment.
SegmentNumber	Returns the number of the current segment.
StartFrequency	Sets or returns the start frequency of the segment.
State	Turns On or OFF a segment.
StopFrequency	Sets or returns the stop frequency of the segment.
TestPortPower	Sets or returns the RF power level of the segment.

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. See Collections in the Analyzer

Methods	Description
Add	Adds an item to either the Segments collection.
Item	Use to get a handle to a segment in the collection..
Remove	Removes an item from a collection of objects.

Properties	Description
Count	Returns the number of items in a collection of objects.
IF Bandwidth Option	Enables the IFBandwidth to be set on individual sweep segments.
Parent	Returns a handle to the current naNetworkAnalyzer application..
Source Power Option	Enables setting the Source Power for a segment.

Trace Object

Description

The Trace object takes measurement data and control how the data is painted on the display. You can control scale, reference position, and reference line from the Trace Object.

Methods	Description
Autoscale	Autoscales the trace or all of the traces in the selected window.

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.

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.

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.

Methods	Description
SetFrequencyLowPass	Sets low frequencies for low pass.

Property	Description
Center	Sets or returns the Center time.
ImpulseWidth	Sets or returns the Impulse Width of Time Domain transform windows.
KaiserBeta	Sets or returns the Kaiser Beta of Time Domain transform windows.
Mode	Sets the type of transform.
Span	Sets or returns the Span time.
Start	Sets or returns the Start time.
State	Turns an Object ON and OFF.
StepRiseTime	Sets or returns the Rise time of the stimulus in Low Pass Step Mode.
Stop	Sets or returns the Stop time.

About Performing a Calibration

AcquisitionDirection Property

Read / Write

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

ActiveCalKit Property

Read-only

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	**(object)** - 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

About Channels

ActiveChannel Property

Read-only

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

About Markers

ActiveMarker Property

Read-only

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

About Measurement Parameters

ActiveMeasurement Property

Read-only

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

About Windows

ActiveNAWindow Property

Read-only

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

About Traces

ActiveTrace Property

Read-only

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

About Sweeping

AlternateSweep Property **Write/Read**

Description	Sets sweeps to either alternate or chopped.
VB Syntax	*chan*.**AlternateSweep** = *value*
Variable	**(Type) - Description**
chan	A Channel **(object)**
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

Application Property **Read-only**

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

ArrangeWindows Property

Write/Read

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

About Attenuation

AttenuatorMode Property

Write/Read

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

*chan.***AttenuatorMode(***portNum***)** = *value*

Variable	**(Type) - Description**
chan	A Channel **(object)**
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.Attenuator(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

About Attenuation

Attenuator Property **Write/Read**

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	*chan*.**Attenuator(***portNum***)** = *value*
Variable	**(Type) - Description**
chan	A Channel **(object)**
portNum	**(long integer)** - Port number (**1** or **2**) of attenuator control to be changed.
value	**(double)** - Attenuator value in dB in 10dB steps. Choose any Long Integer between **0** and **70** 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

About Averaging

AveragingFactor Property

Write/Read

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

About Averaging

Averaging Property

Write/Read

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 Marker Search

BandwidthTarget Property

Write/Read

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

About Marker Search

BandwidthTracking Property

Write/Read

Description	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

About Performing a Calibration

CalibrationType Property

Write/Read

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
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

About Gating

Center Property

Write/Read

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: **7 (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

About Frequency

CenterFrequency Property

Write/Read

Description	Sets or returns the center frequency of the channel **or** Sets or returns the center frequency of the segment.
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

ChannelNumber Property

Read-only

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

About Modifying Cal Kits

C0 Property **Write/Read**

Description Sets and Returns the C0 (C-zero) value (the first capacitance value) for the calibration standard.

To set the other inductance 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

About Modifying Cal Kits

C1 Property **Write/Read**

Description Sets and Returns the C1 value (the second capacitance value) for the calibration standard.

To set the other inductance 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

About Modifying Cal Kits

C2 Property

Write/Read

Description	Sets and Returns the C2 value (the third capacitance value) for the calibration standard. To set the other inductance 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

About Modifying Cal Kits

C3 Property

Write/Read

Description	Sets and Returns the C3 value (the fourth capacitance value) for the calibration standard. To set the other inductance 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

Count Property **Read-only**

Description	Returns the number of items in a collection of objects.
VB Syntax	*object*.**Count**

Variable	**(Type) - Description**
object	Any of the following **(objects):** Channels collection LimitTest collection Measurements collection NAWindows collection Traces collection Segments 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	IChannels ILimitTest IMeasurements NAWindows ITraces ISegments

---

About Power Coupling

CouplePorts Property **Write/Read**

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	*chan*.**CouplePorts** = *value*

Variable	**(Type) - Description**
chan	A Channel **(object)**
value	**(enum NAStates)** Choose from: NaOff - Turns coupling OFF NaOn - Turns coupling ON
Return 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

About CW Frequency

CW Frequency Property **Write/Read**

Description	Set the Continuous Wave (CW) frequency. Must first send chan.SweepType = naCWTimeSweep
VB Syntax	*chan*.**CWFrequency** = *value*

Variable	**(Type) - Description**
chan	A Channel **(object)**
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

About Modifying Cal Kits

Delay Property **Write/Read**

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
Examples	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

About Data Format

Format Property

Write/Read

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_**Double** **7** - naDataFormat_**Imaginary** **8** - naDataFormat_**SWR**

Return Type	Long Integer
Default	**1** - naDataFormat_LogMag
Examples	app.TriggerMode = **naTriggerModePoint** 'Write fmt = meas.Format 'Read
C++ Syntax	HRESULT get_Format(tagDataFormat *pVal) HRESULT put_Format(tagDataFormat newVal)
Interface	IMeasurement

About Dwell Time

DwellTime Property

Write/Read

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** A Segment **(object)**
value	**(double)** - Dwell Time in seconds. Choose any number between**:** **0** and **100e-3**
Return Type	Double
Default	0

Examples	chan.DwellTime = 3e-3 'sets the dwell time for the channel -Write segs(3).CenterFrequency = 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

About Electrical Delay

ElectricalDelay Property **Write/Read**

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 **Write/Read**

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

ExternalALC Property **Write/Read**

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

FilterBW Property

Read-only

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

FilterCF Property

Read-only

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

FilterLoss Property

Read-only

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

FilterQ Property

Read-only

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

About Frequency Range

FrequencySpan Property

Write/Read

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

About Gate Filter

Shape Property **Write/Read**

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

About Time Domain

Type Property **Write/Read**

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:
naGateTypeBandpass - Includes (passes) the range between the start and stop times.
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

About GPIB Fundamentals

GPIBMode Property **Write/Read**

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_ScpiMode(tagScpiModeEnum* eScpiMode) HRESULT put_ScpiMode(tagScpiModeEnum eScpiMode)
Interface	IApplication

IDString Property **Read-only**

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

About Segment Sweep

IFBandwidthOption Property **Write/Read**

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

About IF Bandwidth

IFBandwidth Property **Write/Read**

Description	Sets or returns the IF Bandwidth of the channel. Sets or returns the IF Bandwidth of the segment.
VB Syntax	*object*.**IFBandwidth** = *value*

Variable	(Type) - Description
object	A Channel **(object) or** A Segment **(object)**
value	**(double)** - IF Bandwidth in Hz. Choose from: **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 | 35000 | 40000 |** If a number other than these is entered, the analyzer will round up to the closest valid number (unless a number higher than the maximum in entered.)
Return Type	Double
Default	3500

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

About Time Domain

ImpulseWidth Property **Write/Read**

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

InputA Property

Write/Read

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

InputB Property

Write/Read

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

Interpolate Correction Property

Write/Read

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

About Time Domain

KaiserBeta Property **Write/Read**

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

About Modifying Cal Kits

L0 Property **Write/Read**

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

About Modifying Cal Kits

L1 Property

Write/Read

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

About Modifying Cal Kits

L2 Property

Write/Read

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

About Modifying Cal Kits

L3 Property	**Write/Read**
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

Label Property **Write/Read**

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

BeginStimulus Property **Write/Read**

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

About Limits

EndStimulus Property

Write/Read

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 Sweeps

BeginResponse Property

Write/Read

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

EndResponse Property

Write/Read

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

Type (limit) Property

Write/Read

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: **naLimitSegmentType_Maximum** - limit line fails with a data point ABOVE the line **naLimitSegmentType_Minimum** - limit line fails with a data point BELOW the line **naLimitSegmentType_OFF** - turns limit line OFF
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

About Limits

LineDisplay Property

Write/Read

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

About Modifying Cal Kits

Loss Property

Write/Read

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

About Markers

BucketNumber Property

Write/Read

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

About Marker Format

MarkerFormat Property

Write/Read

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

About Marker Format

Format Property (marker)

Write/Read

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 Markers

Interpolated Property **Write/Read**

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

About Markers

Number Property **Read-only**

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

About Marker Types

Type (Marker) Property **Write/Read**

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: **naMarkerType_Normal -** the X-axis value for a normal marker will always be determined by the measurement data of the marker. **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

About Markers

Stimulus Property **Write/Read**

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

About Markers

Value Property

Read-only

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: naMarkerFormat_**LinMag** (1) naMarkerFormat_**LogMag** (1) naMarkerFormat_**Phase** (1) naMarkerFormat_**Delay** (1) naMarkerFormat_**Real** (1) naMarkerFormat_**Imaginary** (1) naMarkerFormat_**SWR** (1) naMarkerFormat_**LinMagPhase** (2) naMarkerFormat_**LogMagPhase** (2) naMarkerFormat_**RealImaginary** (2) naMarkerFormat_**ComplexImpedance** (3) 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

About Modifying Cal Kits

MaximumFrequency Property

Write/Read

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

About Trace Statistics

Mean Property

Read-only

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

Write/Read

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

MinimumFrequency Property

Write/Read

Description	Sets and Returns the minumum 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

About Traces

Name (Measurement) Property **Write/Read**

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

About Traces

Name (trace) Property **Write/Read**

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

About Modifying Cal Kits

Name (CalKit) Property

Write/Read

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

About Measurements

Number (Measurement) Property

Read-only

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

NumberOfPoints Property

Write/Read

Description	Sets or returns the Number of Points of the channel. Sets or returns the Number of Points of the segment.
VB Syntax	*object*.**NumberOfPoints** = *value*
Variable	**(Type) - Description**
object	A Channel **(object) or** A Segment **(object)**
value	**(long)** - Number of Points. Choose any number from **1** to **1601**
Return Type	Long Integer
Default	201
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

About Options

Options Property

Read-only

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

Parameter Property — **About Measurement Parameters** — **Read-only**

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

Parent Property — **Read-only**

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 object
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)
Interface	IChannels IChannel IMeasurements INAWindows ITraces ISegments

About Marker Search

PeakExcursion Property

Write/Read

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

PeakThreshold Property

Write/Read

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

About Trace Statistics

PeakToPeak Property

Read-only

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

About Phase Offset

PhaseOffset Property

Write/Read

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

Port1 Property **Write/Read**

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

About Port Extensions

Port2 Property **Write/Read**

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 Modifying Cal Kits

PortLabel Property **Write/Read**

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

About Power Slope

PowerSlope Property **Write/Read**

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	*app.***PowerSlope** = *value*

Variable	**(Type) - Description**
app	An Application **(object)**
value	**(double)** - Power Slope. Choose any number between -2 and 2. **No slope = 0**
Return Type	Double
Default	0

Examples	app.PowerSlope = 2 'Write pwrslp = app.PowerSlope 'Read
C++ Syntax	HRESULT get_PowerSlope(double *pVal) HRESULT put_PowerSlope(double newVal)
Interface	IChannel

About Receiver Attenuation

ReceiverAttenuator Property **Write/Read**

Description	Sets or returns the value of the specified receiver attenuator control.
VB Syntax	*chan*.**ReceiverAttenuator(***rec***)** = *value*

Variable	**(Type) - Description**
chan	A Channel **(object)**

rec	**(long integer)** - Receiver with attenuator control to be changed. Choose from: **0** - Receiver A **1** - 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(rnum) 'Read
C++ Syntax	HRESULT get_ReceiverAttenuator(long lport, double *pVal) HRESULT put_ReceiverAttenuator(long lport, double newVal)
Interface	IChannel

About Reference Markers

ReferenceMarkerState Property **Write/Read**

Description	Turn ON or OFF the reference marker. (can you access marker10?)
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 Position

ReferencePosition Property

Write/Read

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

About Reference Level

ReferenceValue Property

Write/Read

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 Marker Search

SearchFunction Property **Write/Read**

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 Segment Sweep

SegmentNumber Property **Read-only**

Description

Returns the number of the current segment.

VB Syntax

seg.**SegmentNumber**

Variable	**(Type) - Description**
seg	A Segment **(object)**. Get a handle to a segment by refering to the item in the

segments collection.

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

About Trace Statistics

ShowStatistics Property

Write/Read

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

About Performing a Calibration

Simultaneous2PortAcquisition Property

Read / Write

Description Specifies whether a 2-port calibration will be done simultaneously or one port at a time.

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

SmoothingAperture Property

Write/Read

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

About Smoothing

Smoothing Property

Write/Read

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 Limits

SoundOnFail Property

Write/Read

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) ILimitTest

---

About Segment Sweep

SourcePowerOption Property

Write/Read

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)** **1 or True** - Enables variable TestPortPower to be set segment sweep

	0 or False - 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

About Source Power

SourcePowerState Property

Write/Read

Description	Turns Source Power ON and OFF
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

About Time Domain

Span Property

Write/Read

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

About Trace Statistics

StandardDeviation Property

Read-only

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

StandardForClass Property

Write/Read

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 As naCalClass)** Calibration standard class. Choose from: naClassA naClassB naClassC naClassD naClassE naReferenceRatioLine naReferenceRatioThru naSOLT_Isolation naSOLT_Load naSOLT_Open naSOLT_Short naSOLT_Thru naTRL_Isolation naTRL_Line_Reflection naTRL_Line_Tracking naTRL_Reflection naTRL_Thru
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	**(double)** - Calibration class number. Choose a number between **1** and **8.** The <*value*> numbers are associated with the following calibration classes:
Return Type	Double
Default	Not Applicable

Examples	calstd.StandardForClass = 7 'Write StdClass = calstd.StandardForClass 'Read
C++ Syntax	HRESULT put_StandardForClass(tagNACalClass stdclass, long port, long standardNumber) HRESULT get_StandardForClass(tagNACalClass stdclass, long port, long* standardNumber)
Interface	ICalKit

About Linear Frequency Sweep

StartFrequency Property **Write/Read**

Description	Sets or returns the start frequency of the channel **or** Sets or returns the start frequency of the segment.
VB Syntax	*object*.**StartFrequency** = *value*
Variable	**(Type) - Description**
object	A Channel **(object)** **or** A Segment **(object)**
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	ISegment

About Power Sweep

StartPower Property **Write/Read**

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	*chan*.**StartPower** = *value*
Variable	**(Type) - Description**
chan	A Channel **(object)**
value	**(double)** - Start Power in dBm. There is 40 dB of range in power sweep. The values of start and stop depend on the amount of attenuation that you specify. With 0 dB of attenuation, the range is -20 dBm to +20 dBm. With 10 dB of attenuation, the range is -30 dBm to +10 dBm, and so forth. 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

About Time Domain

Start Property

Write/Read

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: **7 (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

State Property **Write/Read**

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)
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

Statistics Range Property **Write/Read**

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

About Time Domain

StepRiseTime Property

Write/Read

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

About Linear Frequency Sweep

StopFrequency Property

Write/Read

Description	Sets or returns the stop frequency of the channel **or** Sets or returns the stop frequency of the segment.
VB Syntax	*object*.**StopFrequency** = *value*
Variable	**(Type) - Description**
object	A Channel **(object)** **or** A Segment **(object)**
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

About Power Sweep

StopPower Property

Write/Read

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	*chan*.**StopPower** = *value*
Variable	**(Type) - Description**
chan	A Channel **(object)**
value	**(double)** - Stop Power in dB. Start Power in dB. There is 40 dB of range in power sweep. The acceptable values of start and stop depend on the amount of attenuation that you specify. With 0 dB of attenuation, the range is -20 dBm to +20 dBm. With 10 of attenuation, the range is -30 dBm to +10 dBm, and so forth. 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

About Time Domain

Stop Property

Write/Read

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: **7 (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

About Stepped Sweep

SweepGenerationMode Property

Write/Read

Description	Sets the method used to generate a sweep: continuous ramp (analog) or discrete steps (stepped).
VB Syntax	*chan*.**SweepGenerationMode** = *value*

Variable	**(Type) - Description**
chan	A Channel **(object)**
value	**(enum NASweepGenerationModes)** - **Choose either:** **naSteppedSweep** - source frequency is

	CONSTANT during measurement of eah displayed point. More accurate than Analog. Dwell time can be set in this mode. **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

About Sweep Time

SweepTime Property **Write/Read**

Description	Sets the Sweep time of the analyzer. Sweep time is limited so that the analyzer only sweeps as fast as possible for the current frequency range, number of points, and IFbandwidth.
VB Syntax	*chan*.**SweepTime** = *value*
Variable	**(Type) - Description**
chan	A Channel **(object)**
value	**(double)** - Sweep time in seconds. Choose a number between: **0** and **100**
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

About Sweep Types

SweepType Property **Write/Read**

Description	Sets the type of X-axis sweep that is performed on a channel.
VB Syntax	*chan*.**SweepType** = *value*

Variable	**(Type) - Description**
chan	A Channel **(object)**
value	**(enum NASweepTypes)** - Choose from: **naLinearSweep** **naPowerSweep** **naCWTimeSweep** **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 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

About System Impedance

SystemImpedanceZ0 Property **Write/Read**

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

About Marker Search

TargetValue Property **Write/Read**

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 'W**rite target = mark.**TargetValue** 'Read

---

C++ Syntax	HRESULT get_TargetValue(float *pVal) HRESULT put_TargetValue(float newVal)
Interface	IMarker

About Power Level

TestPortPower Property **Write/Read**

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) -** to set coupled power, use chan.CouplePorts. If CouplePorts = False, then each port power can be set independently. Otherwise, chanTestPortPower (1) = value sets power level at both ports. **or** A Segment **(object)**
portNum	**(long integer)** - Port number of the source power. Choose from **1** or **2**
value	**(double)** - RF Power in dBm. Choose any number between **-90 and 20.** 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 ISegment

About Title

Title Property **Write/Read**

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

About Titles

TitleState Property **Write/Read**

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

About Math Operations

TraceMath Property **Write/Read**

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

About Marker Search

Tracking Property **Write/Read**

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 **Write/Read**

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

About Triggering

TriggerMode Property **Write/Read**

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: **naTriggerModeMeasurement** - all measurements in the channel are made with each trigger signal the channel receives. **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. **Note:** Point Mode is only available in Manual trigger and TriggerType set to naGlobalTrigger. If you change any channel to TriggerModePoint, TriggerType will be set to naChannelTrigger.
Return Type	Long Integer
Default	0 - naTriggerModeMeasurement

Examples	app.TriggerMode = **naTriggerModePoint** 'Write trigtyp = app.TriggerMode 'Read

---

C++ Syntax	HRESULT get_TriggerMode (tagNATriggerMode *pMode) HRESULT put_TriggerMode (tagNATriggerMode newMode)
Interface	IChannel

About Trigger Source

TriggerSignal Property **Write/Read**

Description	Sets or returns the trigger source.
VB Syntax	*app.***TriggerSignal** = *value*

Variable	**(Type) - Description**
app	An Application **(object)**
value	**(enum NATriggerSignal)** - Choose from: **naTriggerInternal** - free run **naTriggerExternalPositive** - an external source that supplies a positive going signal **naTriggerExternalNegative** - an external source that triggers on a negative signal **naTriggerManual** - manual trigger source; use app.ManualTrigger to send a trigger signal.
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

About Trigger

TriggerType Property **Write/Read**

Description	Sets or returns the trigger type which determines the scope of a trigger signal. **Note:** Point (Trigger Mode) is only available in Manual trigger and Channel (Trigger Type). If you change any channel to Global Trigger, Trigger Mode will be set to Measurement.
VB Syntax	*app.***TriggerType** = *value*

Variable	**(Type) - Description**
app	An Application **(object)**
value	**(enum NATriggerType)** - Trigger type. Choose from: **naGlobalTrigger** - a trigger signal is applied to all triggerable channels **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

About Modifying Cal Kits

Type (calstd) Property — **Write/Read**

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

About User Ranges

UserRange Property **Write/Read**

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

About User Ranges

UserRangeMax Property **Write/Read**

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: **naDomainFrequency** **naDomainPower** **naDomainTime**
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

About User Ranges

UserRangeMin Property

Write/Read

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: **naDomainFrequency** **naDomainPower** **naDomainTime**
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

About Port Extensions

VelocityFactor Property **Write/Read**

Description	Sets the velocity factor to be used with Electrical Delay and Port Extensions.
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) **Note**: to specify the electrical delay for reflection measurements (in both directions), double the velocity factor.
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

View Property **Write/Read**

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

Visible Property **Write/Read**

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

About Arranging Windows

WindowState Property **Write/Read**

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: **naMaximized** - Maximizes the window **naMinimized** - Minimizes the window to an Icon on the lower toolbar **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 Scale

YScale Property

Write/Read

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

Z0 Property

Write/Read

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

About Reference Markers

DeltaMarker Property **Write/Read**

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

About Modifying Cal Kits

CalKitType Property **Write/Read**

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**
VB Syntax	***object*.CalKitType = *value***

Variable	**(Type) - Description**
object	**Use either of these:** **A calkit (object).** **An Application (object).** **Note: app.CalKitType and calkit.calKitType perform exactly the same function.**
value	**(enum naCalKit) - Calibration Kit type. 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	naCalKit
Default	Not Applicable

Examples	calkit.CalKitType = naCalKit_85038A_7_16 kitype = app.CalKitType

C++ Syntax	HRESULT get_CalKitType(tagNACalKit *pVal) HRESULT put_CalKitType(tagNACalKit newVal)
Interface	IApplication ICalKit

LimitTestFailed Property

Read-only

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

InterpolateMarkers Method

Write-only

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 raw_InterpolateMarkers(VARIANT_BOOL bNewVal)
Interface	IMeasurement

Abort Method

Write-only

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 raw_Abort(VARIANT_BOOL bSynchronize);
Interface	IChannel

About Calibration Standards

AcquireCalStandard Method **Write-only**

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.
VB Syntax	*cal*.**AcquireCalStandard** *std[,index]*

Variable	**(Type) - Description**
cal	A Calibrator **(object)**
std	**(enum NACalClass)** Standard to be measured. Choose from:
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 raw_AcquireCalStandard(tagNACalClass enumClass, short standardNumber)
Interface	ICalibrator

Activate Method **Write-only**

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 raw_Activate()
Interface	IMeasurement IMarker

About Markers

ActivateMarker Method

Write-only

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 raw_ActivateMarker(long lMarkerNumber)
Interface	IMeasurement
Remarks	Use ReferenceMarkerState to control the Reference marker.

About Windows

ActivateWindow Method

Write-only

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 raw_ActivateWindow(long WindowNumber)
Interface	IApplication

About Channels

Add (channels) Method

Write-only

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 raw_Add(VARIANT numVal, IChannel** pChannel)
Interface	IChannels

Add (measurement) Method

Write-only

Description	Adds a Measurement to the collection.
VB Syntax	*meass.***Add** *channel,param,source[,window]*
meass	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 - if unspecified, value is set to 0 **-1**: creates a new window **0** : current window **n** : new window number (n) - creates if necessary
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 raw_Add(long ChannelNum, BSTR strParameter, long srcPort, VARIANT_BOOL bNewWindow)
Interface	IMeasurements

About Windows

Add (NAWindows) Method

Write-only

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 raw_Add(long windowNumber)
Interface	INAWindows

About Segment Sweep

Add (segment) Method

Write-only

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 raw_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

AllowAllEvents Method **Write/Read**

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

About Analyzer Events

AllowEventCategory Method **Write/Read**

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

AllowEventMessage Method **Write/Read**

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

About Analyzer Events

AllowEventSeverity Method

Write/Read

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

About Display Formatting

Autoscale Method

Write-only

Description	Autoscales the trace (Trace object) or all of the traces (NAWindow object).
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 raw_AutoScale()
Interface	INAWindow ITrace

About Averaging

AveragingRestart Method

Write-only

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 raw_AveragingRestart()
Interface	IChannel

About Modifying Cal Kits

BuildHybridKit Method

Write-only

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	**(enum NACalKit)** - Specifies the two kits to be

port2Kit	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 raw_BuildHybridKit¥(tagNACalKit port1Kit, tagNAPortSex port1Sex, tagNACalKit port2Kit, tagNAPortSex port2Sex, tagNAAdapter adapter, tagNACalKit userKit)
Interface	IApplication

CalculateErrorCoefficients Method

Write-only

Description	This method is the final call in a calibration process. It calculates error-correction terms and turns error-correction ON.
VB Syntax	*cal*.**CalculateErrorCoeffecients**
Variable	**(Type) - Description**
cal	Calibrator **(object)**
Return Type	Not Applicable
Default	Not Applicable
Examples	Cal.**CalculateErrorCoeffecients**
C++ Syntax	HRESULT raw_CalculateErrorCoefficients()

Interface	ICalibrator

ChangeParameter Method — **About Measurement Parameters** — **Write-only**

Description	Changes the parameter of the measurement.
VB Syntax	*meas*.**ChangeParameter**(*param,sourcePort*)

Variable	**(Type) - Description**
meas	A Measurement **(object)**
param	**(string)** - New parameter. Choose from: • **"S11"** • **"S22"** • **"S21"** • **"S12"** • **"A"** • **"B"** • **"R1"** • **"R2"** **or** • **combine any 2 of (A,B,R1,R2) in this format: "A/R1"**
sourcePort	**(long integer)** - Port number of the source. Used only for non-sparameter measurements.
Return Type	Not Applicable
Default	Not Applicable

Examples	meas.**ChangeParameter "S11",1**
C++ Syntax	HRESULT raw_ChangeParameter(BSTR parameter, long sourcePort)
Interface	IMeasurement

Continuous Method — **About Triggering** — **Write-only**

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 raw_Continuous()
Interface	IChannel

About Measurement Parameters

CreateS-Parameter Method

Write-only

Description	Creates a new S-Parameter measurement in an existing or new window.
VB Syntax	*app.***CreateSParameter** *chan,recvr,source,[window]*

Variable	**(Type) - Description**
app	Application **(object)**
chan	**(long integer)** - Channel number of the new measurement
recvr	**(long integer)** - Port number of the receiver (1 or 2)
source	**(long integer)** - Port number of the source (1 or 2)
window	**(long integer)** - 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 1,2,1 'Creates a new S21 measurement in channel 1 and in the active window**
C++ Syntax	HRESULT raw_CreateSParameter(long ChannelNum, long RcvPort, long SrcPort, long windowNumber)
Interface	IApplication

About Measurement Parameters

CreateMeasurement Method

Write-only

Description	Creates a new measurement.
VB Syntax	*app.***CreateMeasurement** *chanNum,param,src[,window]*

Variable	**(Type) - Description**
app	Application **(object)**
chanNum	**(long)** - Channel number of the new measurement; can exist or be a new channel
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"**
src	**(long)** - Source Port. Choose **1** or **2** (ignored if *param* = Sparameter)
window	**(long)** Optional argument. Number of the window the measurement will be in. Choose **1** to **4**. If unspecified, *window* is the active window.
Return Type	Not Applicable
Default	Not Applicable
Examples	app.CreateMeasurement(1,"A/R1",1,0)
C++ Syntax	HRESULT raw_CreateMeasurement(long ChannelNum, BSTR strParameter, long srcPort, long windowNumber)
Interface	IApplication

About Math Operations

DataToMemory Method

Write-only

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 raw_DataToMemory()
Interface	IMeasurement

About Measurement Parameters

Delete Method

Write-only

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 raw_Delete()
Interface	IMeasurement

About Markers

DeleteMarker Method

Write-only

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 raw_DeleteMarker(long lMarkerNumber)
Interface	IMeasurement

About Markers

DeleteAllMarkers Method

Write-only

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 raw_DeleteAllMarkers()
Interface	IMeasurement

About Macros

DeleteShortCut Method **Write-only**

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

About Analyzer Events

DisallowAllEvents Method **Write/Read**

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 raw_DisallowAllEvents()
Interface	IApplication

About Printing

DoPrint Method **Write-only**

Description	Prints the screen to the default Printer.

VB Syntax	*app.***DoPrint**
Variable	**(Type) - Description**
app	Application **(object)**
Return Type	Not Applicable
Default	Not Applicable
Examples	app.DoPrint
C++ Syntax	HRESULT raw_DoPrint()
Interface	IApplication

About Calibration

DoECAL1Port Method

Write-only

Description	Does a 1-Port calibration using an ECAL module. You must first have a 1-port measurement active to perform the calibration.
VB Syntax	*cal*.**DoECAL1Port***[,port][,module]*
Variable	**(Type) - Description**
app	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	**(enum NAEcalModule)** Optional argument - ECAL module. Choose from: naECALModule_A - (default if unspecified) naECALModule_B
Return Type	Not Applicable
Default	Not Applicable
Examples	*cal*.**DoECAL1Port,2,**naECALModule_B
C++ Syntax	HRESULT raw_DoECAL1Port(long port, tagNAECALModule ecalModule)
Interface	ICalibrator

About Calibration

DoECAL2Port Method

Write-only

Description	Does a 2-Port calibration using an ECAL module. You must first have a 2-port measurement active to perform the calibration.
VB Syntax	*cal*.**DoECAL2Port***[,portA][,portB][,module]*
Variable	**(Type) - Description**

app	A Calibrator **(object)**
portA	**(long integer)** Optional argument - Receiver ports to calibrate. Choose from: 1 - port 1 (default if unspecified) 2 - port 2
portB	**(long integer)** Optional argument - Source port to calibrate. Choose from: 1 - port 1 2 - port 2 (default if unspecified)
module	**(enum NAEcalModule)** Optional argument - ECAL module. Choose from: naECALModule_A - (default if unspecified) naECALModule_B
Return Type	Not Applicable
Default	Not Applicable
Examples	*cal*.**DoECAL2Port,1,2,**naECALModule_B
C++ Syntax	HRESULT raw_DoECAL2Port(long rcvport, long srcPort, tagNAECALModule ecalModule)
Interface	ICalibrator

About Macros

ExecuteShortcut Method **Write-only**

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	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 raw_ExecuteShortcut(long index)
Interface	IApplication

GetCalStandard Method

Write-only

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 **8;** (there are 8 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 raw_GetCalStandard(long standardNumber, ICalStandard **pCalStd)
Interface	ICalKit

GetComplex Method

Read-only

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 pointer to 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 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	HRESULT raw_getComplex(tagNADataStore DataStore, long* pNumValues, float* pReal, float* pImag)
Interface	IArrayTransfer

About Accessing Data

GetData Method

Read-only

Description	Retrieves variant data from the specified location in your choice of formats. **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 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: • naDataFormat_LinMag • naDataFormat_LogMag • naDataFormat_Phase • **naDataFormat_Polar *** • **naDataFormat_Smith *** • naDataFormat_Delay • naDataFormat_Real • naDataFormat_Imaginary • 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 - 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 raw_getData(tagNADataStore DataStore, tagDataFormat DataFormat, VARIANT *pData)
Interface	IMeasurement

GetECALModuleInfo Method **Read-only**

Description	Returns the following information about the connected ECAL module: model number, serial number, connector type,calibration date, min and max frequency.
VB Syntax	*module* = *cal*.**GetECALModuleInfo**

Variable	**(Type) - Description**
module	**(string)** - variable to store the module information
cal	A Calibrator **(object)**
Return Type	String - Returned information:
Default	Not Applicable

Examples	*info* = *cal*.**GetECALModuleInfo** **Example return string:** **ModelNumber: 85092-60007, SerialNumber: 01386, ConnectorType: N5FN5F RF2, Calibrated: 5 Jun 2000 , MinFreq: 50000000, MaxFreq: 9100000000**
C++ Syntax	HRESULT raw_GetECALModuleInfo(tagNAECALModule ecalModule, BSTR* info)
Interface	ICalibrator

About Accessing Data

GetErrorTermComplex Method **Read-only**

Description	Retrieves error term data from the error correction buffer.The data is in complex pairs. **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, src, rcv, 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.
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

GetErrorTerm Method

Read-only

Description	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. **Note:** This method returns a variant which is less efficient than getErrorTermComplex on the ICalData interface.
VB Syntax	*data = cal.***getErrorTerm***term, src, rcv*
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
Return Type	Variant
Default	Not Applicable
Examples	Dim varError As Variant varError =

	cal.getErrorTerm(naErrorTerm_Tracking,2, 1)
C++ Syntax	HRESULT raw_getErrorTerm(tagNAErrorTerm ETerm, long ReceivePort, long SourcePort, VARIANT* pData)
Interface	ICalibrator

About Marker Search

GetFilterStatistics Method

Read-only

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.GetFileterStatistics cf,bw,loss,q
C++ Syntax	HRESULT raw_GetFilterStatistics(double* centerFreq, float* bw, float* loss, float* quality)
Interface	IMeasurement

About Accessing Data

GetNAComplex Method

Read-only

Description	Retrieves complex data from the specified locations. See also getComplex and getData Method.
VB Syntax	*measData*.**getNAComplex** *location, numPts, data*

Variable	**(Type) - Description**
measData	An IArrayTransfer pointer to 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 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 raw_getNAComplex(tagNADataStore DataStore, long* pNumValues, TsComplex* pComplex)
Interface	IArrayTransfer

About Accessing Data

GetPairedData Method

Read-only

Description	Retrieves pairs of data from 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 pointer to 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 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 raw_getPairedData(tagNADataStore DataStore, tagNAPairedDataFormat PairFormat, long* pNumValues, float* pReal, float* pImag)
Interface	IArrayTransfer

About Reference Markers

GetReferenceMarker Method

Write/Read

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 raw_GetReferenceMarker(IMarker** refMarker)
Interface	IMeasurement

About Accessing Data

GetScalar Method

Read-only

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 pointer to 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 See the Data Access Map

format	**(enum DataFormat)** - Scalar format in which you would llike 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 values(0), values(1)
```

C++ Syntax	HRESULT raw_getScalar(tagNADataStore DataStore, tagDataFormat DataFormat, long* pNumValues, float* pVals)
Interface	IArrayTransfer

GetShortcut Method **Read-only**

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	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 raw_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

---

GetStandard Method

Read-only

Description	Retrieves data that was acquired for a specific cal standard. This method returns the actual measurement data - not the calculated error terms. **Note:** This method returns a variant which is less efficient than getStandardComplex on the ICalData interface.
VB Syntax	*data = cal.***getStandard***class,rcv,src*
Variable	**(Type) - Description**
data	Variant array to store the data.
cal	A Calibrator **(object)**
class	**(enum As naCalClass)** Calibration standard class. Choose from: naClassA naClassB naClassC naClassD naClassE naReferenceRatioLine naReferenceRatioThru naSOLT_Isolation naSOLT_Load naSOLT_Open naSOLT_Short naSOLT_Thru naTRL_Isolation naTRL_Line_Reflection naTRL_Line_Tracking naTRL_Reflection naTRL_Thru
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 varStd = **cal.getStandard(naSOLT_Thru,2,1)**
C++ Syntax	HRESULT raw_getStandard(tagNACalClass stdclass, long ReceivePort, long SourcePort, VARIANT* pData)
Interface	ICalibrator

GetStandardComplex Method

Write-only

Description	Retrieves data that was acquired for a specific cal standard. This method returns the actual measurement data - not the calculated error terms. **Note:** This method exists on a non-default interface. If you cannot access this method, use the GetStandard Method on ICalibrator.
VB Syntax	*ICalData.***getStandardComplex** *class, rcv, src, numPts, real(), imag()*

Variable	**(Type) - Description**
ICalData	An ICalData pointer to the Calibrator object
class	**(enum As naCalClass)** Calibration standard class. Choose from: naClassA naClassB naClassC naClassD naClassE naReferenceRatioLine naReferenceRatioThru naSOLT_Isolation naSOLT_Load naSOLT_Open naSOLT_Short naSOLT_Thru naTRL_Isolation naTRL_Line_Reflection naTRL_Line_Tracking naTRL_Reflection naTRL_Thru
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 rel(200) As Single
  Dim img(200) As Single
  Dim sdata As ICalData
  Set sdata = chan.Calibrator
  sdata.getStandardComplex naSOLT_Open,
```

1, 1, numpts, rel(0), img(0)

C++ Syntax HRESULT raw_getStandardComplex(tagNACalClass stdclass, long ReceivePort, long SourcePort, long* pNumValues, float* pReal, float* pImag)

Interface ICalData

About Limit Testing

GetTestResult Method

Read-only

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 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 0
 Print "No Test Result"

 Case 1
 Print "Fails"

 Case 2
 Print "Pass"

 End Select
```

C++ Syntax	HRESULT raw_GetTestResult(long lStart, long lSize, tagNALimitTestResult *pVal)
Interface	ILimitTest

About Trace Statistics

GetTraceStatistics Method

Read-only

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 raw_GetTraceStatistics(double* pp, double* mean, double* stdDeviation)
Interface	IMeasurement

About Segment Sweep

GetX-axisValues2 Method

Write-only

Description	Returns the channel's X-axis values into a dimensioned Typed array. 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 exists on a non-default interface. If you cannot access this method, use the GetXAxisValues Method on IChannels. **Note:** In Segment Sweep, chan.NumberofPoints will return the total number of data points for the combined segments.
VB Syntax	*chanData.***GetXAxisValues2** *numPts,data*

Variable	**(Type) - Description**
chanData	An IArrayTransfer pointer to the 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 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 raw_GetXAxisValues2(long* pNumValues, double* stimulus)
Interface	IChannel

GetXAxisValues Method

Read-only

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.numpts - 1
   Print varData(i)
  Next i
```

C++ Syntax HRESULT raw_GetXAxisValues (VARIANT* xData)

Interface IChannel

About Triggering

Hold Method

Write-only

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 raw_Hold(VARIANT_BOOL bWait)

Interface IChannel

Item Method

Write-only

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):** Channels collection LimitTest collection Measurements collection

	NaWindows collection Traces collection Segments 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 raw_Item(VARIANT index, <interface>** pItem)
Interface	IChannels ILimitTest IMeasurements INaWindows ITraces ISegments

---

About Triggering

ManualTrigger Method

Write-only

Description	Triggers the analyzer when TriggerSignal = naTriggerManual.
VB Syntax	*app.***ManualTrigger** *[sync]*

Variable	**(Type) - Description**
app	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.
Return Type	Not Applicable
Default	Not Applicable

Examples	Dim wait as Boolean wait = True app.ManualTrigger wait

C++ Syntax	HRESULT raw_ManualTrigger(VARIANT_BOOL bSynchronize, long timeout)
Interface	IApplication

About Analyzer Events

MessageText Method **Write/Read**

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 raw_MessageText(tagNAEventID msgID, BSTR* message)
Interface	IApplication

About Dynamic Range

NextIFBandwidth Method **Write-only**

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 raw_Next_IFBandwidth (double *pVal)

Interface IChannel

About Triggering

NumberOfGroups Method

Write-only

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 raw_NumberOfGroups(long count, VARIANT_BOOL bWait)

Interface IChannel

SCPI Command Tree

Parse Method

Write-Read

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 raw_Parse(BSTR SCPI_Command, BSTR *pQueryResponse)
Interface	IScpiStringParser

Factory Preset Settings

Preset Method **Write-only**

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.
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 raw_Preset()
Interface	IApplication IChannel

.

About Dynamic Range

PreviousIFBandwidth Method **Write-only**

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 raw_Previous_IFBandwidth (double *pVal)
Interface	IChannel

About Saving to File

PrintToFile Method — **Write-only**

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. If you don't provide a suffix, the analyzer will append **.bmp**. The file is saved to the current working directory unless a valid full path name is specified.
Return Type	Not Applicable
Default	Not Applicable

Examples

```
app.PrintToFile "myfile"
  app.PrintToFile "c:¥data¥myfile.bmp"
```

C++ Syntax	HRESULT raw_PrintToFile(BSTR bstrFile)
Interface	IApplication

PutComplex Method

Write-only

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. 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 pointer to the Measurement (object)
location	**(enum NADataStore)** Where the Data will be put. Choose from: 0 - naRawData 1 - naCorrectedData 2 - naMeasResult 3 - naRawMemory 4 - naMemoryResult 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 • naDataFormat_Delay • naDataFormat_Imaginary • naDataFormat_LinMag • naDataFormat_LogMag • naDataFormat_Phase • naDataFormat_Real • naDataFormat_SWR • naDataFormat_Smith • naDataFormat_Polar
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 raw_putComplex(tagNADataStore DataStore, long lNumValues, float* pReal, float* pImag, tagDataFormat displayFormat)
Interface	IArrayTransfer

About Accessing Data

PutDataComplex Method

Write-only

Description	Puts complex data into the specified location. This method forces the channel into Hold mode to prevent the input data from being overwritten. Data put in naRawData *(location*) will be re-processed whenever a change is made to the measurement attributes such as format or correction. Data put in naMeasurement *(location*) will be overwritten by any measurement attribute changes.
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 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	meas.putDataComplex naMeasResult, varData
C++ Syntax	HRESULT raw_putDataComplex(tagNADataStore DataStore, VARIANT complexData)
Interface	IMeasurement

PutDataScalar Method

Write-only

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.
VB Syntax	*meas*.**putDataScalar** *format, data*

Variable	**(Type) - Description**
meas	A measurement **(object)**
format	**(enum NADataFormat)** Format of the data. Choose from: 1 - naDataFormat_LinMag 2 - naDataFormat_LogMag 3 - naDataFormat_Phase **4 - naDataFormat_Polar *** **5 - naDataFormat_Smith *** 6 - naDataFormat_Delay 7 - naDataFormat_Real 8 - naDataFormat_Imaginary 9 - naDataFormat_SWR *** Smith and Polar** formats require a two-dimensional array **varData (numpts, 2)** to accomodate both real and imaginary data. All other formats are a single dimension **varData(numpts).**
data	**(variant)** - A **two-dimensional** complex variant data array. **Note:** 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.

Return Type	Not Applicable
Default	Not Applicable
Examples	measData.putDataScalar naDataFormat_Real, varData
C++ Syntax	HRESULT raw_putDataScalar(tagNADataStore DataStore, VARIANT complexScalar)
Interface	IMeasurement

PutErrorTerm Method

Write-only

Description	Puts variant error term data into the error-correction buffer. See Accessing data.
VB Syntax	*cal.***putErrorTerm**(*term,src,rcv*,*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).

Return Type	Not Applicable
Default	Not Applicable
Examples	Dim varError As Variant varError = **cal.putErrorTerm (naErrorTerm_Tracking,2,1,VarData)**
C++ Syntax	HRESULT raw_putErrorTerm(tagNAErrorTerm ETerm, long ReceivePort, long SourcePort, VARIANT varData)
Interface	ICalibrator

PutErrorTermComplex Method

Write-only

Description	Puts error term data into the error-correction data buffer.
VB Syntax	*data.***putErrorTermComplex** *term, src, rcv, 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.
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
raw_putErrorTermComplex(tagNAErrorTerm
ETerm, long ReceivePort, long SourcePort,
long* pNumValues, float* pReal, float* pImag)
```

Interface ICalData

About Accessing Data

PutScalar Method

Write-only

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 pointer to the Measurement (object)
format	**(enum NADataFormat)** Format of the data. Choose from: 1 - naDataFormat_LinMag 2 - naDataFormat_LogMag 3 - naDataFormat_Phase 6 - naDataFormat_Delay 7 - naDataFormat_Real 8 - naDataFormat_Imaginary 9 - naDataFormat_SWR **Note: Smith and Polar** formats are not allowed.
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 raw_putScalar¥(tagDataFormateFormat, long lNumValues, float* pArrayOfScalar)
Interface	IArrayTransfer

PutNAComplex Method

Write-only

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 pointer to the Measurement (object)
location	**(enum NADataStore)** Where the Data will be put. Choose from: 0 - naRawData 1 - naCorrectedData 2 - naMeasResult 3 - naRawMemory 4 - naMemoryResult 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
raw_putNAComplex(tagNADataStore
```

	DataStore, long lNumValues, TsComplex* pArrayOfComplex, tagDataFormat displayFormat)
Interface	IArrayTransfer

About Macros

PutShortcut Method

Write-only

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	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 raw_PutShortcut(long Number, BSTR title, BSTR pathname)
Interface	IApplication

PutStandard Method

Write-only

Description	Writes variant data to the error correction buffer holding measurement data acquired for a specific standard.
VB Syntax	*cal.***putStandard** *class,rcv,src,data*

Variable	**(Type) - Description**
cal	A Calibrator **(object)**
class	**(enum As naCalClass)** Calibration standard class. Choose from: naClassA naClassB naClassC naClassD naClassE naReferenceRatioLine naReferenceRatioThru naSOLT_Isolation naSOLT_Load naSOLT_Open naSOLT_Short naSOLT_Thru naTRL_Isolation naTRL_Line_Reflection naTRL_Line_Tracking naTRL_Reflection naTRL_Thru
rcv	**(long)** - Receiver Port
src	**(long)** - Source Port
data	**(variant)** Two dimensional array (0:1, 0:points-1)
Return Type	Not Applicable
Default	Not Applicable

Examples	Dim varStd (1,200) As Variant cal.putStandard naSOLT_Thru, 2, 1, varStd
C++ Syntax	HRESULT raw_putStandard(tagNACalClass stdclass, long ReceivePort, long SourcePort, VARIANT varData)
Interface	ICalibrator

PutStandardComplex Method

Write-only

Description	Writes data to the error correction buffer holding measurement data acquired for a specific standard.
VB Syntax	*ICalData.***putStandardComplex** *class,rcv,src, numPts,real(),imag()*

Variable	**(Type) - Description**
ICalData	A ICalData pointer to the Calibrator object
class	**(enum As naCalClass)** Calibration standard class. Choose from: naClassA naClassB naClassC naClassD naClassE
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 ICalData Set sdata = chan.Calibrator sdata.putStandardComplex naSOLT_Open, 1, 1, 201, rel(0), img(0)
C++ Syntax	HRESULT raw_putStandardComplex(tagNACalClass stdclass, long ReceivePort, long SourcePort, long lNumValues, float* pReal, float* pImag)
Interface	ICalData

Quit Method **Write-only**

Description Terminates the Network Analyzer application.

VB Syntax *app.***Quit**

Variable	**(Type) - Description**
app	Application **(object)**

Return Type Not Applicable

Default Not Applicable

Examples

```
app.Quit
```

C++ Syntax HRESULT raw_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.

About Save/Recall

Recall Method **Write-only**

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	Application **(object)**
filename.ext	**(string)** - Filename and extension of the file to be recalled.

Extensions:

- .sta - Instrument State
- .cal - Calibration file
- .cst - Both Instrument State and Calibration file

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 raw_Recall(BSTR bstrFile)

Interface IApplication

About Modifying Cal Kits

Recall Kits Method **Write-only**

Description Recalls the calibration kits definitions that were stored witht the SaveKits command.

VB Syntax *app.***RecallKits**

Variable	**(Type) - Description**
app	Application **(object)**

Return Type Not Applicable

Default Not Applicable

Examples

```
app.RecallKits
```

C++ Syntax HRESULT raw_RecallKits()

Interface IApplication

Remove Method **Write-only**

Description Removes an item from a collection of objects.

VB Syntax *Object.***Remove** *item*

Variable	**(Type) - Description**
Object	Any of the following **(objects)** Measurements collection NAWindows collection Segments collection **Note:** segments has 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 (3 - 22)

C++ Syntax	HRESULT raw_Remove(VARIANT index) //IMeasurements
	HRESULT raw_Remove(long windowNumber) //INaWindows
	HRESULT raw_Remove(VARIANT index, long size) //Segments

Interface	IMeasurements INAWindows ISegments

About Presetting the Analyzer

Reset Method **Write-only**

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	Application **(object)**
Return Type	Not Applicable
Default	Not Applicable

Examples	app.**Reset**
C++ Syntax	HRESULT raw_Reset()
Interface	IApplication

About Modifying Cal Kits

RestoreCalKitDefaults Method **Write-only**

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	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 raw_RestoreCalKitDefaults¥(tagNACalKit kit)
Interface	IApplication

About Modifying Cal Kits

RestoreCalKitDefaultsAll Method

Write-only

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	Application **(object)**
Return Type	Not Applicable
Default	Not Applicable
Examples	app.RestoreCalKitDefaultsAll
C++ Syntax	HRESULT raw_RestoreCalKitDefaultsAll()
Interface	IApplication

About Saving Files

Save Method

Write-only

Description	Saves the current measurement state, calibration state, or both, to the analyzer's hard drive. The saved state can be recalled using app.Recall.
VB Syntax	*app.***Save**(*filename.ext*)
Variable	**(Type) - Description**
app	Application **(object)**
filename.ext	**(string)** - Filename and extension of the file to be saved. Extensions: • .sta - Instrument State • .cal - Calibration file • .cst - Both Instrument State and Calibration file

	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 raw_Save(BSTR bstrFile)
Interface	IApplication

About Modifying Cal Kits

SaveKits Method

Write-only

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	Application **(object)**
Return Type	Not Applicable
Default	Not Applicable
Examples	app.SaveKits
C++ Syntax	HRESULT raw_SaveKits()
Interface	IApplication

About Marker Search

SearchFilterBandwidth Method

Write-only

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 raw_SearchFilterBandwidth()

Interface IMeasurement

About Marker Search

SearchMax Method

Write-only

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 raw_SearchMax()

Interface IMarker

About Marker Search

SearchMin Method

Write-only

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 raw_SearchMin()

Interface IMarker

About Marker Search

SearchNextPeak Method

Write-only

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 raw_SearchNextPeak()

Interface IMarker

About Marker Search

SearchPeakLeft Method

Write-only

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 raw_SearchPeakLeft()

Interface IMarker

About Marker Search

SearchPeakRight Method

Write-only

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 raw_SearchPeakRight()
Interface	IMarker

About Marker Search

SearchTarget Method

Write-only

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 raw_SearchTarget()
Interface	IMarker

About Marker Search

SearchTargetLeft Method

Write-only

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 raw_SearchTargetLeft()
Interface	IMarker

About Marker Search

SearchTargetRight Method

Write-only

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 raw_SearchTargetRight()
Interface	IMarker

About Performing a Calibration

SetCalInfo Method

Write-only

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.
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 **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 put_setCalInfo(tagNACalType calType,long portA, long portB,)
Interface	ICalibrator

About Marker Functions

SetCenter Method **Write-only**

Description	Changes the center stimulus to the stimulus value of the marker. The start stimulus stays the same and the stop is adjusted.
VB Syntax	*mark*.**SetCenter**
Variable	**(Type) - Description**
mark	A Marker **(object)**
Return Type	Not Applicable
Default	Not Applicable
Examples	mark.**SetCenter**
C++ Syntax	HRESULT raw_SetCenter()
Interface	IMarker

About Markers

SetCW Method **Write-only**

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 raw_SetCW()
Interface	IMarker

About Marker Functions

SetElectricalDelay Method

Write-only

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 raw_SetElectricalDelay()
Interface	IMarker

About Analyzer Events

SetFailOnOverRange Method

Write/Read

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

About Time Domain

SetFrequencyLowPass Method

Write-only

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 raw_SetFrequencyLowPass(void)
Interface	ITransform

About Marker Functions

SetReferenceLevel Method

Write-only

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 raw_SetReferenceLevel()
Interface	IMarker

SetStart Method

Write-only

Description	Changes the start stimulus to the stimulus value of the marker. The stop stimulus stays the same and the span is adjusted.
VB Syntax	*mark*.**SetStart**
Variable	**(Type) - Description**
mark	A Marker **(object)**
Return Type	Not Applicable
Default	Not Applicable
Examples	mark.**SetStart**
C++ Syntax	HRESULT raw_SetStart()
Interface	IMarker

SetStop Method

Write-only

Description	Changes the stop stimulus to the stimulus value of the marker. The start stimulus stays the same and the span is adjusted.
VB Syntax	*mark*.**SetStop**
Variable	**(Type) - Description**
mark	A Marker **(object)**
Return Type	Not Applicable
Default	Not Applicable
Examples	mark.**SetStop**
C++ Syntax	HRESULT raw_SetStop()
Interface	IMarker

ShowMarkerReadout Method

Write-only

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 raw_ShowMarkerReadout(VARIANT_BOOL bState)

Interface INAWindow

About Display Formatting

ShowStatusBar Method

Write-only

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	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 raw_ShowStatusBar (VARIANT_BOOL bState)

Interface IApplication

About Display Formatting

ShowStimulus Method

Write-only

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	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 raw_ShowStimulus(VARIANT_BOOL bState)
Interface	IApplication

About Display Formatting

ShowTable Method

Write-only

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 raw_ShowTable (tagNATableType table)
Interface	INAWindow

About Display Formatting

ShowTitleBars Method

Write-only

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	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 raw_ShowTitleBars¥(VARIANT_BOOL bState)
Interface	IApplication

About Display Formatting

ShowToolbar Method **Write-only**

Description	Shows and Hides the specified Toolbar.
VB Syntax	*app.***ShowToolbar** *toolbar,state*
Variable	**(Type) - Description**
app	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 raw_ShowToolbar(tagNAToolbarType toolbar, VARIANT_BOOL bState)
Interface	IApplication

About Triggering

Single Method **Write-only**

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 raw_Single(VARIANT_BOOL bWait)
Interface	IChannel

Events

OnCalEvent

Description	Triggered by a calibration event. See a list of CAL Events.
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
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

About Analyzer Events

OnDisplayEvent

Description Triggered by a display event. See a list of Display Events

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

About Analyzer Events

OnHardwareEvent

Description Triggered by a hardware event. See a list of Hardware Events

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.
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
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
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**

Examples

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**

```
'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", "analyzerName")
  Set measData = app.ActiveMeasurement
  Set chan = app.ActiveChannel
  numpts = chan.NumberOfPoints

  '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 MSchart
  'right click on the chart and select:
  ' - line chart
  ' - data in rows
  End Sub



Sub GetComplex_Click()
  ReDim Data(numpts) As NAComplex
  Dim Real(201)
  Dim Imag(201)
  '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
 ' 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
 '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 send continuous trigger signals
app.TriggerSignal = naTriggerInternal
'The channel will only accept one, then go into hold
chan.Single sync
End Sub

Sub Sweep()
'The channel goes back to accepting all triggers
chan.Continuous
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

To use this code, prepare a form with the following:

- None

```
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
```

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,Graphical User Interface accurately, and repeatably. This topic contains the following introductory information that explains GPIBhardware and software and how they fit together.

- **The GPIB Hardware Components**
- **The GPIB Programming Elements**
- **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.

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:

- **The Interface select code** (typically 7) indicates which GPIB port in the system controller is used to communicate with the device.
- **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. The analyzer address can be changed through **System ¥ Configure ¥ SICL-GPIB** menu
- **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. Becuase the analyzer cannot be both Talker/Listener and System Controller, you cannot have a GPIB program running on the analyzer (controller) and have the analyzer responding to commands. To accomplish this, use COM programming.

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 Programming Elements

The following software programming elements combine to become a GPIB program:

- **GPIB Commands**
- **Messages**
- **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:

- **IEEE 488.1 Bus-management Commands**
- **IEEE 488.2 Common Commands**
- **SCPI Commands**

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

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. For a complete list of the common commands that are recognized by the analyzer, see GPIB Command Finder.

Examples of IEEE 488.2 Common Commands

- ***OPC - Operation Complete**
- ***RST - Reset**
- *OPT? - Queries the option configuration

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.

The Rules and Syntax of SCPI Commands provides more detail of the SCPI command structure. For a complete list of the SCPI commands that are recognized by the analyzer, see GPIB Command Finder.

Example of SCPI Commands:

- CALCULATE:AVERAGE:STATE ON
- SENSE:FREQUENCY:START?

GPIB Messages

A message is made up of one or more GPIB commands, punctuation, data, and an appropriate termination. There are two types of messages:

- **Program messages** are sent from the **controller to an instrument**.
- **Response messages** are sent from an **instrument to the controller**.

Programming Statements

GPIB messages are included with the language specific I/O statements to form program statements. The programming language determines the syntax of the programming statements. GPIB programs can be written in a variety of programming languages such as VEE, HP BASIC, or C++. Because most programmers are familiar with the BASIC language, all of the programming examples in this help system use HP BASIC. To modify these examples to work with your programming language, just substitute your language I/O statements for the HP BASIC OUTPUT and ENTER statements.

The following is an example of a message included in a Visual Basic statement to form a programming statement.

GPIB.Write "SOURCE:FREQUENCY:FIXED 1000 MHz"	sets the source frequency to 1000 Megahertz.

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.

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**

Note: For background information on SCPI Commands, see GPIB Fundamentals / Commands

---

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 brach commands.

Sometimes the same keyword, such as START, 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.
  - You must use white space to separate parameters from commands. For example, the <WSP> between STATE and ON in the command CALCULATE:SMOOTHING:STATE ON is mandatory. Whitespace does not affect the current path.
- **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 numerous 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

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, the FORMat:BORDer command should be used. See GPIB Command Finder for details.

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

The analyzer sends ASCII character data that looks like numeric data. All numeric data sent over the GPIB is character 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. The analyzer always encloses data in double quotes when it returns string data.

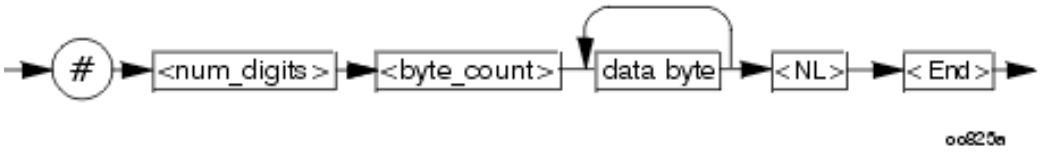
Example of String Data

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

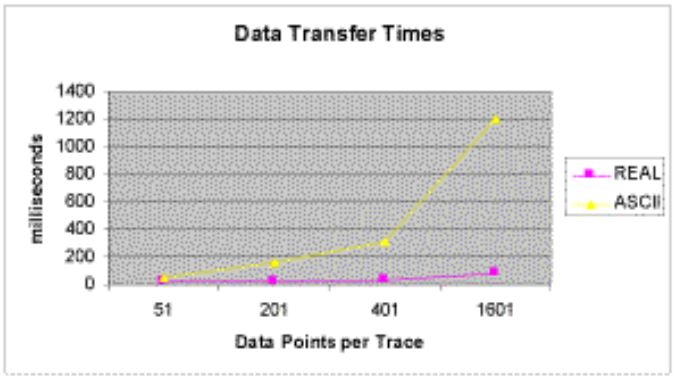
Transferring Measurement Data

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 the Data Arrays. Regardless of which measurement array is read, transferring measurement data is done the same.

When transferring measurement data, there are two data types to choose from:

- 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

These data types are selected using the FORMat SCPI command.

ASCII Data

The easiest and slowest way to transfer measurement data is to use ASCII data. 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

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**

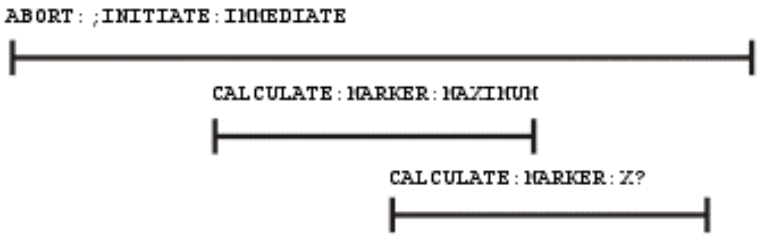
The *WAI command:

- **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:

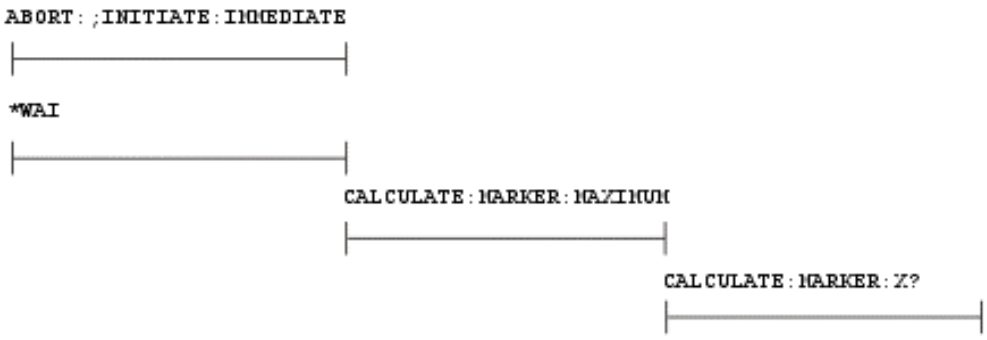


INITIATE:IMMEDIATE is an overlapped command; it allows the immediate processing of the sequential command, CALCULATE:MARKER:SEARCH:MAXIMUM. However, the INITIATE:IMMEDIATE is not considered complete until the measurement is complete. Therefore, the marker searches for maximum amplitude before the measurement 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**

This program determines which frequency contains the maximum amplitude.

```
GPIB.Write "ABORT; :INITIATE:IMMEDIATE"! Restart the measurement
GPIB.Write "*OPC?" 'Wait until complete
Meas_done = GPIB.Read 'Read output queue, throw away result
GPIB.Write "CALCULATE:MARKER:MAX" 'Search for max amplitude
GPIB.Write "CALCULATE:MARKER:X?" 'Which frequency?
Marker_x = GPIB.Read
PRINT "MARKER at " & Marker_x & " Hz"
```

***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.

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.

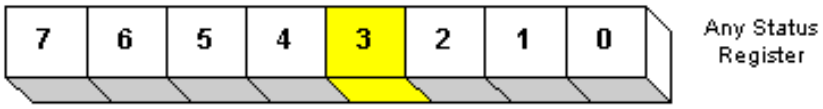
If the analyzer continues to measure and average the results after the average count is reached, use synchronization to determine when each subsequent measurement is complete.

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**

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
1. Read the weighted bit value for these bits	weights 16 and 32 (respectively)
2. Add these values together	16 + 32 = 48
3. Send this number as an argument in the appropriate command. (see Status Commands)	STAT:QUES:LIMIT1:ENAB 1026

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.

Agilent VEE Example

Application Configuration

For this example use Agilent VEE version 6.0 or above which contains the Variant data type used to transfer data from the PNA. The type library for the PNA should be referenced in the Agilent VEE development environment.

Using the Agilent VEE Object Browser the developer can see the classes and methods which are available for development of applications for the PNA Series analyzer.

Application Code

There is a runtime version of Agilent VEE that may be used if the application has been saved as 途untime□. A free version of Agilent VEE can be found on the following web site: http://www.agilent.com/find/vee/. The application may be run on a PC or on the PNA Series analyzer.

The application file is located at http://agilent.com/find/pna_applications.

Microsoft Visual Basic Example

Application Configuration

The type library for the PNA should be referenced in the Visual Basic development environment.

Using the Visual Basic Object Browser the developer can see what classes and methods are available for development of applications for the analyzer.

Application Code

The application code is contained below. To run the application, first generate the executable file. Once this is complete, it can be copied and executed on the analyzer or run on the PC. The application can also be run from the development environment.

```
Option Explicit
Dim app As AgilentPNA835x.Application
  Dim chan As AgilentPNA835x.Channel
  Dim meas As AgilentPNA835x.Measurement
  Dim result As Variant
  Dim i As Integer
  Dim num_points As Integer
  Dim message As String
Private Sub Main()
' Connect to the PNA application; change analyzerName to your analyzer's full computer name
Set app = CreateObject("AgilentPNA835x", "analyzerName")
' Reset the analyzer to instrument preset
  app.Reset
' Create S11 measurement
  app.CreateMeasurement 1, "S11", 1
' Set chan variable to point to the active channel
  Set chan = app.ActiveChannel
' Set meas variable to point to the active measurement
  Set meas = app.ActiveMeasurement
' Setup the channel for a single trigger
```

```
  chan.Hold True
  app.TriggerSignal = naTriggerManual
  chan.TriggerMode = naTriggerModeMeasurement
' Make the PNA application visible
  app.Visible = True
' Set channel parameters
  chan.NumberOfPoints = 11
  chan.StartFrequency = (1000000000#)
  chan.StopFrequency = (2000000000#)
' Send a manual trigger to initiate a single sweep
  chan.Single True
' Store the data in the "result" variable
  result = meas.GetData(naRawData,
  naDataFormat_LogMag)
' Display the result
  num_points = chan.NumberOfPoints
  For i = 0 To num_points - 1
  message = message & result(i) & vbCrLf
  Next
If MsgBox(message, vbOKOnly, "S11(dB) - VBS COM
  Example for PNA") Then
  Set chan = Nothing
  app.Quit
  End If
End Sub
```

Microsoft Visual Basic Script Example

Application Configuration

Some operating systems may require that the Visual Basic Scripting engine be installed before running the application on a PC. To download a free copy of a Visual Basic Scripting engine, visit the following web site: http://msdn.microsoft.com/scripting/

Application Code

The application code is contained below. The developer must save the file in a text file (i.e. using notepad) and save it with the ".vbs" extension. The ".vbs" extension will tell the operating system to execute the code using the Visual Basic Scripting engine.

In order to run the application, double-click on the saved .vbs file. The application can be run on a PC or copied and run on the PNA Series network analyzer.

```
Option Explicit
' Shell objects
  Dim app
  Dim chan
  Dim meas
  Dim result
  Dim message
  Dim num_points
  Dim i
' Connect to the PNA application; change analyzerName to your analyzer's full computer name
Set app = CreateObject("AgilentPNA835x", "analyzerName")
' Reset the analyzer to instrument preset
  app.Reset
' Create S11 measurement
  app.CreateMeasurement 1, "S11", 1
' Set chan variable to point to the active channel
  Set chan = app.ActiveChannel
' Set meas variable to point to the active measurement
  Set meas = app.ActiveMeasurement
' Setup the channel for a single trigger
  chan.Hold True
  app.TriggerSignal = 3
  chan.TriggerMode = 1
' Make the PNA application visible
  app.Visible = True
  ' Set channel parameters
  chan.NumberOfPoints = 11
  chan.StartFrequency = (1000000000)
  chan.StopFrequency = (2000000000)
' Send a manual trigger to initiate a single sweep
  chan.Single True
' Store the data in the "result" variable
  result = meas.GetData(0, 1)
' Display the result
  num_points = chan.NumberOfPoints
  For i = 0 To num_points - 1
  message = message & result(i) & vbCRLF
  Next
if MsgBox(message, vbOKOnly, "S11(dB) - VBS COM
  Example for PNA") then
  Set chan = Nothing
  app.quit
end if
```

Microsoft Visual C++ Example

Application Configuration

Microsoft Visual C++ version 6 was used for this example. In order to perform this example, create a new project in Microsoft Visual C++. Add a C++ file to the project and paste the following code into the file. The path for the type library in the code below should be changed to reference its location on the development PC.

Application Code

The application can be run on a PC or on the PNA.

```
#include "stdafx.h"
```

```
// import the Tsunami type library
  //--------------------------
  #import "C:¥Program Files¥Agilent¥Network Analyzer¥835x.tlb"
  no_namespace, named_guids
  int main(int argc, char* argv[])
  {
  // interface pointers to retrieve COM interfaces
  IUnknown* pUnk = 0;
  IApplication* pNA = 0;
  IChannel* pChan = 0;
  IMeasurement* pMeas = 0;
  IArrayTransfer* pTrans = 0;
  int i, num_points = 0;
  float* pScalarData;
HRESULT hr;
// Initialize the COM subsystem
  CoInitialize(NULL);
// Create an instance of the network analyzer
  // Request the NA's IUnknown interface
  hr = CoCreateInstance(CLSID_Application,0,
  CLSCTX_ALL,IID_IUnknown, (void**) &pUnk);
  if (!FAILED(hr)) {
// QueryInterface for the INetworkAnalyzer interface
  of the NetworkAnalyzer object
  hr = pUnk->QueryInterface(IID_IApplication,
  (void**)&pNA);
if (!FAILED(hr)) {
// Reset the analyzer to instrument preset
  pNA->Reset();
// Create S11 measurement
  pNA->CreateSParameter(1,1,1,1);
// Set pChan variable to point to the active
  channel
  pNA->get_ActiveChannel(&pChan);
if (pChan) {
// Set pMeas variable to point to the active
  measurement
  pNA->get_ActiveMeasurement(&pMeas);
if(pMeas) {
// Setup the channel for a single trigger
  pChan->Hold(true);
  pNA->TriggerSignal = naTriggerManual;
```

```
  pChan->TriggerMode =
  naTriggerModeMeasurement;
// Make the PNA application visible
  pNA->put_Visible(true);
// Set channel parameters
  pChan->NumberOfPoints = 11;
  pChan->StartFrequency = 1e9;
  pChan->StopFrequency = 2e9;
// Send a manual trigger to initiate a single
  sweep
  pChan->Single(true);
// QueryInterface for the IArrayTransfer
  interface of the NetworkAnalyzer object
  hr = pMeas->QueryInterface(IID_IArray
  Transfer, (void**)&pTrans);
if (!FAILED(hr)) {
// Store the data in the "result" variable
  num_points = pChan->NumberOfPoints;
  pScalarData = new float[num_points];
  pTrans->getScalar(naRawData, naData
  Format_LogMag, (long *)&num_points,
  pScalarData);
// Display the result
  printf("S11(dB) - Visual C++ COM
  Example for PNA¥n¥n");
  for (i = 0; i < num_points; i++)
  printf("%f¥n",pScalarData[i]);
}
  }
  }
  }
pUnk->Release();
  pMeas->Release();
  pChan->Release();
  pTrans->Release();
  pNA->Release();
  }
  CoUninitialize();
  return 0;
  }
```

Microsoft Word Example

Application Configuration

Microsoft® Office 2000 was used for this example. This version of Office contains Visual Basic for Applications (VBA) which allows developers to attach Visual Basic Macros to Word documents. The type library for the PNA Series network analyzer should be referenced in the Visual Basic development environment.

Application Code

The application code is contained below. The program inserts the data retrieved from the analyzer into a table in a Word document. To run the application, open the document using Microsoft Word. Enable the macros when prompted. Once this is complete, the application will execute and update the document. The application can be run on a PC or the analyzer.

```
Option Explicit
Dim app
  Dim chan
  Dim meas
  Dim result As Variant
  Dim i As Integer
  Dim num_points As Integer
Private Sub Document_Open()
' Connect to the PNA application; change analyzerName to your analyzer's full computer name
Set app = CreateObject("AgilentPNA835x", "analyzerName")
' Reset the analyzer to instrument preset
  app.Reset
' Create S11 measurement
  app.CreateMeasurement 1, "S11", 1
' Set chan variable to point to the active channel
  Set chan = app.ActiveChannel
' Set meas variable to point to the active measurement
  Set meas = app.ActiveMeasurement
' Setup the channel for a single trigger
  chan.Hold True
  app.TriggerSignal = naTriggerManual
  chan.TriggerMode = naTriggerModeMeasurement
' Make the PNA application visible
  app.Visible = True
' Set channel parameters
  chan.NumberOfPoints = 11
  chan.StartFrequency = (1000000000#)
  chan.StopFrequency = (2000000000#)
' Send a manual trigger to initiate a single sweep
  chan.Single True
' Store the data in the "result" variable
  result = meas.GetData(naRawData,
  naDataFormat_LogMag)
' Display the result
  num_points = chan.NumberOfPoints
  For i = 0 To num_points - 1
  ThisDocument.Tables(1).Cell(i + 2, 2).Range = result(i)
  Next
```

```
Set chan = Nothing
  app.Quit
End Sub
```

Microsoft Excel Example

Application Configuration

Microsoft Office 2000 was used for this example. This version of Office contains Visual Basic for Applications (VBA) which allows developers to attach Visual Basic Macros to Excel documents. The type library for the PNA network analyzer should be referenced in the Visual Basic development environment.

Application Code

The application code is contained below. The program inserts the data retrieved from the analyzer into cells in the Excel document. The cells are then used to update a graph in the Excel document. To run the application, open the document using Microsoft Excel. Enable macros when prompted by the application. Once this is complete, the application will execute and update the document. It can be run on a PC or the PNA analyzer.

```
Option Explicit
Dim app
  Dim chan
  Dim meas
  Dim result As Variant
  Dim i As Integer
  Dim num_points As Integer
Private Sub Workbook_Open()
' Connect to the PNA application; change analyzerName to your analyzer's full computer
name
Set app = CreateObject("AgilentPNA835x", "analyzerName")
' Reset the analyzer to instrument preset
  app.Reset
' Create S11 measurement
  app.CreateMeasurement 1, "S11", 1
' Set chan variable to point to the active channel
  Set chan = app.ActiveChannel
' Set meas variable to point to the active measurement
  Set meas = app.ActiveMeasurement
' Setup the channel for a single trigger
  chan.Hold True
  app.TriggerSignal = naTriggerManual
  chan.TriggerMode = naTriggerModeMeasurement
' Make the PNA application visible
  app.Visible = True
' Set channel parameters
  chan.NumberOfPoints = 11
  chan.StartFrequency = (1000000000#)
  chan.StopFrequency = (2000000000#)
' Send a manual trigger to initiate a single sweep
  chan.Single True
' Store the data in the "result" variable
  result = meas.GetData(naRawData,
  naDataFormat_LogMag)
```

```
' Display the result
  num_points = chan.NumberOfPoints 14
For i = 0 To num_points - 1
  Sheet1.Cells(3 + i, 1) = result(i)
  Next
Set chan = Nothing
  app.Quit
End Sub
```

National Instruments™ LabVIEW Example

Application Configuration

Use National Instruments(TM) Lab VIEW version 5.0 or above for this example. See the National Instruments LabVIEW documentation for information on using ActiveX objects in the LabVIEW development environment.

Application Code

National Instruments(TM) LabVIEW 5.0 or higher must be installed to run the application. The application can be run on a PC or on the PNA Series analyzer.

The application file is located at http://agilent.com/find/pna_applications.

See Other GPIB Example Programs

GPIB using Visual C++

```
/*
  * 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);
  }
}
```

See Other GPIB Example Programs

GPIB using Visual Basic

To run this program, you need:

- An established GPIB interface connection

```
Option Explicit

 Dim opc As Integer
 Dim result As String
 Dim i As Integer
 Dim num_points As Integer
 Dim commaPosition As Integer
 Dim message As String
 '
 ' Program executes when the 'Run' button is clicked.
 Private Sub cmdRun_Click()
   ' National Instruments ComponentWorks 'Instr' ActiveX Control, Version 2.0.1

   ' Default board number for National Instruments GPIB boards
   GPIB.BoardNumber = 0

   ' Default GPIB address of PNA
   GPIB.PrimaryAddress = 16

   ' Initialize the GPIB connection
   GPIB.Configure

   ' Reset the analyzer to instrument preset
   GPIB.Write "SYSTem:FPRESET"

   ' Create S11 measurement
   GPIB.Write "CALCulate1:PARameter:DEFine 'My_S11',S11"

   ' Turn on Window #1
   GPIB.Write "DISPlay:WINDow1:STATe ON"

   ' Put a trace (Trace #1) into Window #1 and 'feed' it from the measurement
   GPIB.Write "DISPlay:WINDow1:TRACe1:FEED 'My_S11'"

   ' Setup the channel for single sweep trigger
   GPIB.Write "INITiate1:CONTinuous OFF;*OPC?"
   opc = GPIB.Read
   GPIB.Write "SENSe1:SWEep:TRIGger:POINt OFF"


   ' Set channel parameters
   GPIB.Write "SENSe1:SWEep:POINts 11"
   GPIB.Write "SENSe1:FREQuency:STARt 1000000000"
   GPIB.Write "SENSe1:FREQuency:STOP 2000000000"


   ' Send a trigger to initiate a single sweep
   GPIB.Write "INITiate1;*OPC?"
   opc = GPIB.Read

   ' Must select the measurement before we can read the data
   GPIB.Write "CALCulate1:PARameter:SELect 'My_S11'"
```

```
    ' Read the measurement data into the "result" string variable
    GPIB.Write "FORMat ASCII"
    GPIB.Write "CALCulate1:DATA? FDATA"
    result = GPIB.Read

    ' Display the result
    GPIB.Write "SENSe1:SWEep:POINts?"
    num_points = GPIB.Read
    For i = 0 To num_points - 2
    commaPosition = InStr(result, ",")
    message = message & Left(result, commaPosition - 1) & vbCrLf
    result = Right(result, Len(result) - commaPosition)
    Next
    message = message & result

    MsgBox message, vbOKOnly, "S11(dB) - VB SCPI Example for PNA"

  End Sub
  Private Sub cmdExit_Click()
    End
  End Sub
```

8753 X-Reference

COM versus SCPI

There are two methods you can use to remotely control the analyzer: COM and SCPI. The following topics are intended to help you choose the method that best meets your needs:

- **Software Connection**
- **Physical Connection**
- **Selecting a Method**
- **Programming Languages**

Software Connection

COM uses a binary protocol, allowing the user to directly invoke a feature of the Network Analyzer. This is more efficient than SCPI. For example, the following statement calls directly into the Network Analyzer, 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 network analyzer:

IbWrite("*IDN?")

The network analyzer's SCPI parser would first decode this text string to determine that the user has asked for the network analyzer 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 analyzer'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 Analyzer's performance.

Using the SICL I/O Libraries, you can also connect to the Analyzer from a program running on the Analyzer.

External Control

You can control the analyzer from a remote PC using either COM or SCPI.

COM - (Component Object Model) can be used to access any program like the analyzer (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 Analyzer 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 an object called the ScpiStringParser. This object provides access to the SCPI parser (or command decoder) so that you can send SCPI text commands using automation.

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 the COM as the preferred programming paradigm. As new capability is developed, it may not be made available through SCPI.

But choosing a connection method depends on your situation. Here are some additional things to consider:

1. If you want to use the Analyzer to control other GPIB instruments, you may want to use COM as the means of talking to the instrument. In GPIB, the analyzer can not be configured as both **System Controller** and **talker/listener**. Because the Analyzer 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 to the Analyzer. A word of warning: the PNA uses a different platform than previous Agilent Network Analyzers. Therefore, not all commands have a direct replacement. See the 8753 command finder.

Programming Languages

You can program the Analyzer 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.

See Other GPIB Example Programs

Channels, Windows, and Measurements using GPIB

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"
```

See Other GPIB Example Programs

Create a Measurement using GPIB

This Visual Basic program creates a new S21 measurement and displays it on the display.

Use the links to see the command details.

To run this program, you need:

- An established GPIB interface connection

```
'Preset the analyzer
 GPIB.Write "SYSTem:PReset"

 ' Turn on window 1 - if new, creates it
 GPIB.Write "DISPlay:WINDow1:STATE ON"

 'Define a measurement name, parameter
 GPIB.Write "CALCulate:PARameter:DEFine 'MyMeas',S21"

 'Associate ("FEED") the measurement name ('MyMeas') to WINDow (1), and give the new
TRACe a number (1).
 GPIB.Write "DISPlay:WINDow1:TRACe1:FEED 'MyMeas'"
```

See Other GPIB Example Programs

Catalog Measurements using GPIB

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

```
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)
```

See Other GPIB Example Programs

Perform a Calibration using GPIB

This Visual Basic program does a Full 2-Port Calibration, including Isolation
To run this program, you need:

- An established GPIB interface connection
- A 2-port measurement set up with desired frequency range, power, and so forth, ready to be calibrated.

```
Sub SOLTCal()
 'Turn off continuous sweep
 GPIB.Write "INITiate:CONTinuous OFF"

 'Select 2-Port Calibration
 GPIB.Write "SENSe:CORRection:COLLect:METHod SPARSOLT"

 'Select a cal kit
 Calkitnum = 3
 GPIB.Write "SENSe:CORRection:COLLect:CKIT:SELect " & Val(Calkitnum)

 'Measure the standards
 MsgBox "Connect OPEN's to Ports 1 and 2; then press OK"
 Call Measurestandard("stan1")
 MsgBox "Connect SHORT's to Ports 1 and 2; then press OK"
 Call Measurestandard("stan2")
 MsgBox "Connect LOAD's to Ports 1 and 2; then press OK"
 Call Measurestandard("stan3")
 MsgBox "Connect THRU between Ports 1 and 2; then press OK"
 Call Measurestandard("stan4")
 MsgBox "Disconnect Ports 1 and 2 for isolation; then press OK"
 Call Measurestandard("stan5")

 'Compute the coefficients and turn on error correction
 GPIB.Write "SENSe:CORRection:COLLect:SAVE"

 'Resume continuous sweep.
 GPIB.Write "INITialize:CONTinuous ON"
 End Sub

 Sub Measurestandard(Std$)
 'Store the results of a sweep as correction data
 GPIB.Write "SENSe:CORRection:COLLect " & Std$

 'Take a sweep;return when complete
 GPIB.Write "INITiate:IMMediate;*OPC?"
 OPCreply = GPIB.Read

 End Sub
```

See Other GPIB Example Programs

ECALibrate Using GPIB

The following program does an Electronic Calibration using an Agilent ECAL module
These commands do the following:

- Acquire the standards
- Move the error terms back into the analyzer
- Enable the calibration

Note: A separate :SENS:CORR:COLL:SAVE is not needed.
To run this program, you need:

- An established GPIB interface connection

```
Private Sub Command5_Click()
  'Turn off continuous sweep
  GPIB.Write "INITiate:CONTinuous OFF"

  'ECal full 1 port and 2 port
  'This program assumes you have already set up the analyzer for an S11 measurement over
the frequency range, power, etc. that you want.

  'Select the Ecal "Kit"
  GPIB.Write "SENSe:CORRection:COLLect:CKIT 99"

  'Choose a Calibration Type (comment out one of these)
  GPIB.Write "SENSe:CORRection:COLLect:METHod refl3"
  GPIB.Write "SENSe:CORRection:COLLect:METHod SPARSOLT"

  'Enable or disable (comment out one) measurement of isolation
  GPIB.Write "SENSe:CORRection:ISOLation ON"
  GPIB.Write "SENSe:CORRection:ISOLation OFF"

  'Prompt for the ECal module
  MsgBox ("Connect ECal module to Port 1, then press enter")
'Acquire and store the calibration terms - return (*OPC) when finished
  GPIB.Write "SENSe:CORRection:COLLect:ACQuire ECALA;*OPC?"
  X = GPIB.Read
  MsgBox ("Done with calibration.")

  End Sub
```

See Other GPIB Example Programs

Modify a Calibration Kit using GPIB

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

```
'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"
```

See Other GPIB Example Programs

Setup Sweep Parameters using GPIB

This Visual Basic program sets up sweep parameters on the Channel 1 measurement.

To run this program, you need:

- An established GPIB interface connection

```
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
```

See Other GPIB Example Programs

Setup the Display using GPIB

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

```
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 GPIB Example Programs

Getting and Putting Data using GPIB

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

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 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
```

See Other GPIB Example Programs

Status Reporting using GPIB

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 interupt 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 SRQMethod
- 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'"
'slow 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
```

SCPI Command Tree

IEEE- 488.2 Common Commands

SCPI Commands

:ABORt	Stops all sweeps
CALCulate **__Click to show CALC commands**	
:CORRection	Sets Electrical Delay and Phase Offset
:DATA?	Queries measurement data.
:FILTer	Sets time domain gating
:FORMat	Sets the display format
:FUNCtion	Controls Trace Statistics
:LIMit	Controls limit lines for pass / fail testing
:MARKer	Controls the marker settings
:MATH	Performs math on the memory trace
:PARameter	Creates and deletes measurements
:RDATa?	Queries receiver data
:SMOothing	Controls point-to-point smoothing
:TRANsform	Controls time domain transform settings
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 show SENSe commands**	
:AVErage	Sets sweep averaging parameters
:BANDwidth	Specifies the IF filter bandwidth
:CORRection	Provides calibration capability.
:CORR:COLL:CKIT	Defines calibration standards
:COUPle	Sets sweep as Chopped or Alternate
:FREQuency	Controls frequency sweep 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.
SOURce	Controls the power to the DUT
STATus	Reads the analyzer status registers
SYSTem	Controls the analyzer defaults
TRIGger	Starts or ends a measurement

About Triggering

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

List of all commands in this block:
(Parameters in ***bold italics***)
:ABORt

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.

***TRG - Trigger**

Triggers a sweep on the active measurement channel when in trigger hold mode (ignored if the analyzer is in continuous sweep mode).

***TST? - Result of Self-test Query**

Returns the result of a query of the analyzer hardward 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	Overpower	too much power is detected at the input. This is from either using an amplifier, or a hardware failure.
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	OverTemp	the source temperature sensor exceeds the limit. It could result from restricted airflow or a broken fan

---

***WAI - Wait**

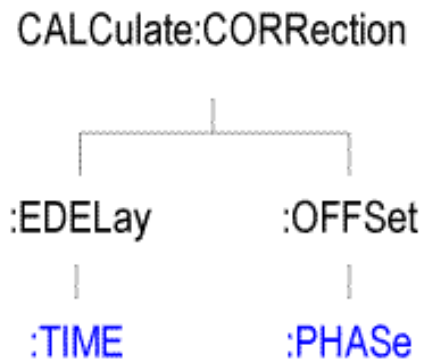
Prohibits the instrument from executing any new commands until all pending overlapped commands have been completed. See Understanding Command Synchronization

---

CALCulate

Calc:Correction Commands

Controls **Electrical Delay** and **Phase Offset**



- Click on a blue keyword to view the command details.
- 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. Therefore, you can have up to four measurements selected at the same time. To select the measurement use CALC<ChanNum>:PAR:SEL <MeasName>.

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	Character
Overlapped?	No
Default	0 seconds

CALCulate<cnum>:CORRection:OFFSet:PHASe <num>[<char>]

(Read-Write) 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	Character, returned value always in degrees
Overlapped?	No
Default	0 degrees

List of all commands in this block:
(Parameters in ***bold italics***)
:CALCulate***1***:CORRection:EDELay:TIME ***10e-12***
:CALCulate***1***:CORRection:EDELay:TIME?
:CALCulate***1***:CORRection:OFFSet:PHASe ***90deg***
:CALCulate***1***:CORRection:OFFSet:PHASe?

Calc:Data Command

CALCulate<cnum>:DATA <char>
Write Measurement data, Memory data, or Error terms.

CALCulate<cnum>:DATA? <char>
Read Measurement data, Memory data, or Error terms.

▼ **Return Data Format:**

FDATA - one number per trace point
SDATA - two numbers per trace point
FMEM - one number per trace point
SMEM - two numbers per trace point
All Error Terms - two numbers per trace point

Parameters
<cnum> - Channel number of the measurement. There must be a selected measurement on that channel. If unspecified, <cnum> is set to 1.

<char> - To write or read **Measurement and Memory Data**, choose from:

FDATA - formatted trace data from **measResult** location
SDATA - corrected complex trace data from **rawMeas** location
FMEM - formatted memory data from **memResult** location
SMEM - corrected complex data from **rawMemory** location

<char> - To write or read **Error Terms**...

▼For 2-Port SOLT and TRL calibrations,

▼For 1-Port calibrations:

▼For Response Open calibrations:

▼For Response Short calibrations:

▼For Response Thru calibrations:

▼For Response Thru and Isolation calibrations:

EXAMPLE - See an example using this command.
Overlapped? - No
Default - Not Applicable
Notes:

- When querying memory, you must first store a trace into memory using

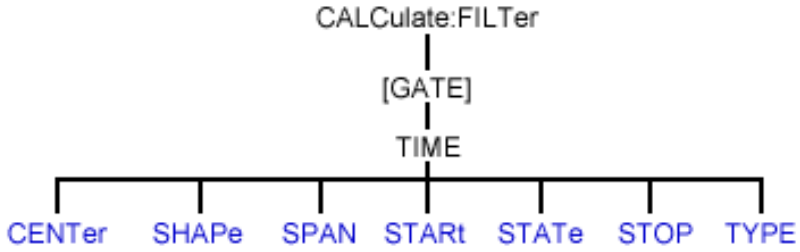
CALC:MATH:MEMorize.

- 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

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.

Note: 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.

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: 7 (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	Character

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 Character

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:
0 (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 Character

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:
0 (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 Character

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

List of all commands in this block:
(Parameters in ***bold italics***)

:CALCulate***1***:FILTer:GATE:TIME:CENTer ***10ms***
:CALCulate***1***:FILTer:GATE:TIME:CENTer?
:CALCulate***1***:FILTer:GATE:TIME:SHAPe ***MAX***
:CALCulate***1***:FILTer:GATE:TIME:SHAPe?
:CALCulate***1***:FILTer:GATE:TIME:SPAN ***10ms***
:CALCulate***1***:FILTer:GATE:TIME:SPAN?
:CALCulate***1***:FILTer:GATE:TIME:STARt ***10ms***
:CALCulate***1***:FILTer:GATE:TIME:STARt?
:CALCulate***1***:FILTer:GATE:TIME:STATe ***ON***
:CALCulate***1***:FILTer:GATE:TIME:STATe?
:CALCulate***1***:FILTer:GATE:TIME:STOP ***10ms***
:CALCulate***1***:FILTer:GATE:TIME:STOP?
:CALCulate***1***:FILTer:GATE:TIME:TYPE ***NOTCh***
:CALCulate***1***:FILTer:GATE:TIME:TYPE?

About Data Format

Calc:Format Command

Note: 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.

- See an example using this command.

CALCulate<cnum>:FORMat <char>

(Read-Write) Sets the display format for the measurement. **Critical Note:**

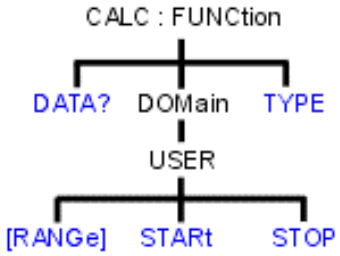
Parameters

<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

Examples	CALC:FORM MLIN calculate2:format polar
Query Syntax	CALCulate<cnum>:FORMat?
Return Type	Character
Overlapped?	No
Default	MLINear

List of all commands in this block:
(Parameters in ***bold italics***)
:CALCulate***1***:FORMat ***MLIN***
:CALCulate***1***:FORMat?

Calc:Function Commands



- Click on a blue keyword to view the command details.
- 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. Therefore, you can have up to four measurements selected at the same time. 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	Character
Example	.Culate2: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 Character

Overlapped? No
Default 0 - Full Span

CALCulate<cnum>:FUNCtion:DOMain:USER:STARt <range>, <start>

(Read-Write) Sets the start of the user-domain range. Must also set CALC:FUNC:DOM:USER and 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 Character

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 user-domain range. Must also set CALC:FUNC:DOM:USER and 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 Character

Overlapped?	No
Default	The analyzer's **Maximum** x-axis value

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

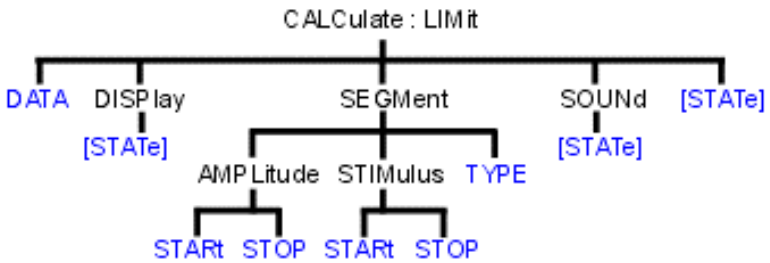
Examples	CALC:FUNC:TYPE PTP calculate2:function:type stdev
Query Syntax	CALCulate<cnum>:FUNCtion:TYPE?
Return Type	Character
Overlapped?	No
Default	PTPeak

List of all commands in this block:
(Parameters in ***bold italics***)

:CALCulate***1***:FUNCtion:DATA?
:CALCulate***1***:FUNCtion:DOMain:USER:RANGe ***0***
:CALCulate***1***:FUNCtion:DOMain:USER:RANGe?
:CALCulate***1***:FUNCtion:DOMain:USER:STARt ***1, 1e9***
:CALCulate***1***:FUNCtion:DOMain:USER:STARt? ***1***
:CALCulate***1***:FUNCtion:DOMain:USER:STOP ***4, 5e9***
:CALCulate***1***:FUNCtion:DOMain:USER:STOP? ***4***
:CALCulate***1***:FUNCtion:TYPE ***PTP***
:CALCulate***1***:FUNCtion:TYPE?

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.

Note: 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.

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**

_____

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	Definite length block - 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	OFF

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 Character

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 Character

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 Character

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 Character

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. **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

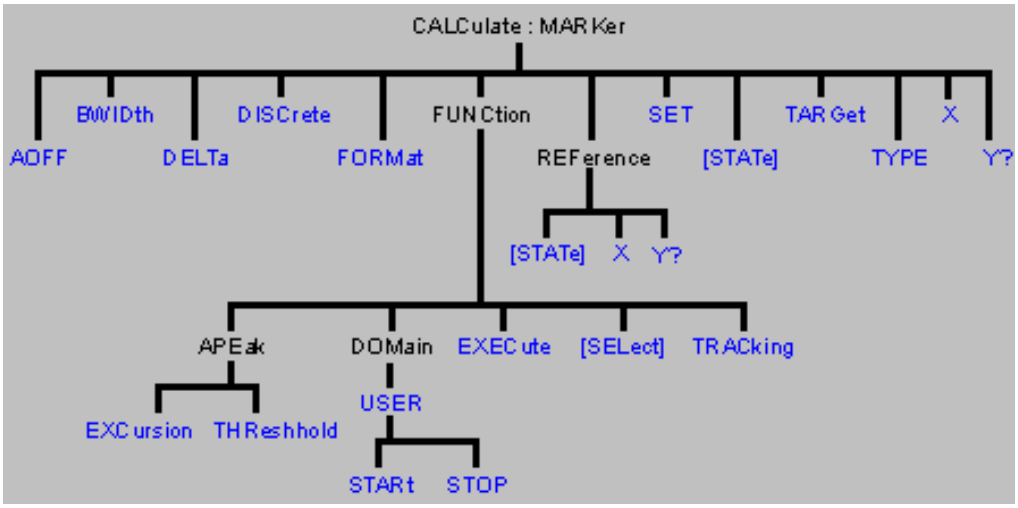
Query Syntax CALCulate<cnum>:LIMit:STATe?
Return Type Boolean (1 = ON, 0 = OFF)

Overlapped? No
Default OFF

List of all commands in this block:
(Parameters in ***bold italics***)
:CALCulate***1***:LIMit:DATA <*block*>
:CALCulate***1***:LIMit:DATA?
:CALCulate***1***:LIMit:DISPlay:STATe ***ON***
:CALCulate***1***:LIMit:DISPlay:STATe?
:CALCulate***1***:LIMit:SEGMent***1***:AMPLitude:STARt ***0***
:CALCulate***1***:LIMit:SEGMent***1***:AMPLitude:STARt?
:CALCulate***1***:LIMit:SEGMent***1***:AMPLitude:STOP ***0***
:CALCulate***1***:LIMit:SEGMent***1***:AMPLitude:STOP?
:CALCulate***1***:LIMit:SEGMent***1***:STIMulus:STARt ***1e9***
:CALCulate***1***:LIMit:SEGMent***1***:STIMulus:STARt?
:CALCulate***1***:LIMit:SEGMent***1***:STIMulus:STOP ***3e9***
:CALCulate***1***:LIMit:SEGMent***1***:STIMulus:STOP?
:CALCulate***1***:LIMit:SEGMent***1***:TYPE ***LMAX***
:CALCulate***1***:LIMit:SEGMent***1***:TYPE?
:CALCulate***1***:LIMit:SOUNd:STATe ***OFF***
:CALCulate***1***:LIMit:SOUNd:STATe?
:CALCulate***1***:LIMit:STATe ***ON***
:CALCulate***1***:LIMit:STATe?

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.

Note: 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.

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 Four Character values separated by commas: bandwidth, center Frequency, Q, loss.

Overlapped? No

Default -3

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>: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

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	Character
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	Character
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
	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 Character

_____

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 Character

_____

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 Character

_____

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]?

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 Character

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 Character

_____

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. **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 frequency span to the value of the marker's **domain** **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 xxx delay to the value of the marker

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	Character
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) Sets 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)
- 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.

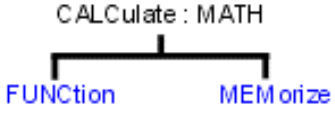
Examples	CALC:MARK:Y? calculate2:marker3:y?
Query Syntax	CALCulate<cnum>:MARKer<mkr>:Y?
Return Type	Character
Overlapped?	No
Default	Not applicable

List of all commands in this block:
(Parameters in ***bold italics***)

:CALCulate***1***:MARKer:AOFF
:CALCulate***1***:MARKer:BWIDth ***-2.513***
:CALCulate***1***:MARKer:BWIDth?
:CALCulate***1***:MARKer***1***:DELTa ***ON***
:CALCulate***1***:MARKer***1***:DELTa?
:CALCulate***1***:MARKer***1***:DISCrete ***ON***
:CALCulate***1***:MARKer***1***:DISCrete?
:CALCulate***1***:MARKer***1***:FORMat ***DEF***
:CALCulate***1***:MARKer***1***:FORMat?
:CALCulate***1***:MARKer***1***:FUNCtion:APEak:EXCursion ***0***
:CALCulate***1***:MARKer***1***:FUNCtion:APEak:EXCursion?
:CALCulate***1***:MARKer***1***:FUNCtion:APEak:THReshold ***0***
:CALCulate***1***:MARKer***1***:FUNCtion:APEak:THReshold?
:CALCulate***1***:MARKer***1***:FUNCtion:DOMain:USER ***1***
:CALCulate***1***:MARKer***1***:FUNCtion:DOMain:USER?
:CALCulate***1***:MARKer***1***:FUNCtion:DOMain:USER:STARt ***1e12***
:CALCulate***1***:MARKer***1***:FUNCtion:DOMain:USER:STARt?
:CALCulate***1***:MARKer***1***:FUNCtion:DOMain:USER:STOP ***1e12***
:CALCulate***1***:MARKer***1***:FUNCtion:DOMain:USER:STOP?
:CALCulate***1***:MARKer***1***:FUNCtion:EXECute ***MAX***
:CALCulate***1***:MARKer***1***:FUNCtion:SELect ***MAX***
:CALCulate***1***:MARKer***1***:FUNCtion:SELect?
:CALCulate***1***:MARKer***1***:TARGet ***-10.3***
:CALCulate***1***:MARKer***1***:TARGet?
:CALCulate***1***:MARKer***1***:FUNCtion:TRACking ***ON***
:CALCulate***1***:MARKer***1***:FUNCtion:TRACking?
:CALCulate***1***:MARKer:REFerence:STATe ***ON***
:CALCulate***1***:MARKer:REFerence:STATe?
:CALCulate***1***:MARKer:REFerence:X ***10***
:CALCulate***1***:MARKer:REFerence:X?
:CALCulate***1***:MARKer:REFerence:Y?
:CALCulate***1***:MARKer***1***:TYPE ***NORM***
:CALCulate***1***:MARKer***1***:TYPE?
:CALCulate***1***:MARKer***1***:SET ***CENT***
:CALCulate***1***:MARKer***1***:SET?
:CALCulate***1***:MARKer***1***:STATe ***ON***
:CALCulate***1***:MARKer***1***:STATe?
:CALCulate***1***:MARKer***1***:X ***1Ghz***
:CALCulate***1***:MARKer***1***:X?
:CALCulate***1***:MARKer***1***:Y?

Calc:Math Commands

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.

Note: 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.

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:

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.

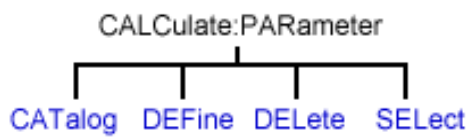
Examples	CALC:MATH MEM calculate2:math:memorize
Query Syntax	Not applicable
Overlapped?	No
Default	Not applicable

List of all commands in this block:
(Parameters in ***bold italics***)

:CALCulate***1***:MATH:FUNCtion ***NORM***
:CALCulate***1***:MATH:FUNCtion?
:CALCulate***1***:MATH:MEMorize

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.

Note: 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.

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?

Query Syntax CALCulate<cnum>:PARameter:CATalog?

Return Type String - "<measurement name>,<parameter>,[<measurement name>,<parameter>...]"

Overlapped? No

Default "CH1_S11_1,S11"

CALCulate<cnum>:PARameter:DEFine <Mname>,<param>

(Write-only) Creates a measurement but NOT display it.

- 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 measurment. 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 | S12 | S21 | S22** Or choose from the following for non S-Parameter measurements. Specify the source port with SENSe:SWEep:SRCPort <1|2> **A | B | R1 | R2 | A/B | B/A**

Examples CALC:PAR:DEF "Test,S12
calculate2:parameter:define "test,s22

Query Syntax Not Applicable; see Calc:Par:Cat?

Overlapped? No

Default Not Applicable

CALCulate<cnum>:PARameter:DELete <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<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

List of all commands in this block:
(Parameters in ***bold italics***)

:CALCulate***1***:PARameter:CATalog?
:CALCulate***1***:PARameter:DEFine ***"CH1_S21,S21"***
:CALCulate***1***:PARameter:DELete ***"CH1_S21"***
:CALCulate***1***:PARameter:SELect ***"CH1_S11"***
:CALCulate***1***:PARameter:SELect?

Calc:RData Command

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.

Note: 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.

CALCulate<cnum>:RDATA? <char>

(Read-only) Returns receiver data for the selected measurement. To query measurement data, see CALC: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 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	Character - Two numbers per data point

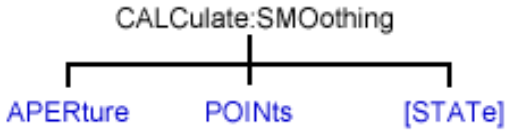
Overlapped?	No
Default	Not Applicable

List of all commands in this block:
(Parameters in ***bold italics***)
:CALCulate***1***:RDATA? ***A***

About Smoothing

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.

Note: 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.

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	Character

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	Character
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.

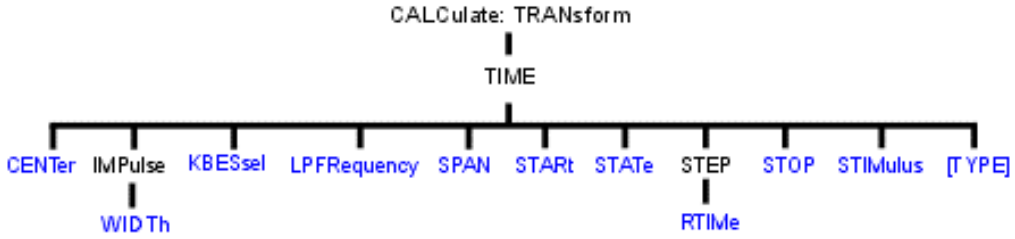
Examples	CALC:SMO ON calculate2:smoothing:state off
Query Syntax	CALCulate<cnum>:SMOothing[:STATe]?
Return Type	Boolean (1 = ON, 0 = OFF)
Overlapped?	No
Default	OFF

List of all commands in this block:
(Parameters in ***bold italics***)

:CALCulate***1***:SMOothing:APERture ***25***
:CALCulate***1***:SMOothing:APERture?
:CALCulate***1***:SMOothing:POINts ***10***
:CALCulate***1***:SMOothing:POINts?
:CALCulate***1***:SMOothing:STATe ***ON***
:CALCulate***1***:SMOothing:STATe?

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.

Note: 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.

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: **7** (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	Character
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	Character
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	Character
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: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	Character
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: **7** (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 Character

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:
7 (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 Character

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 Character

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

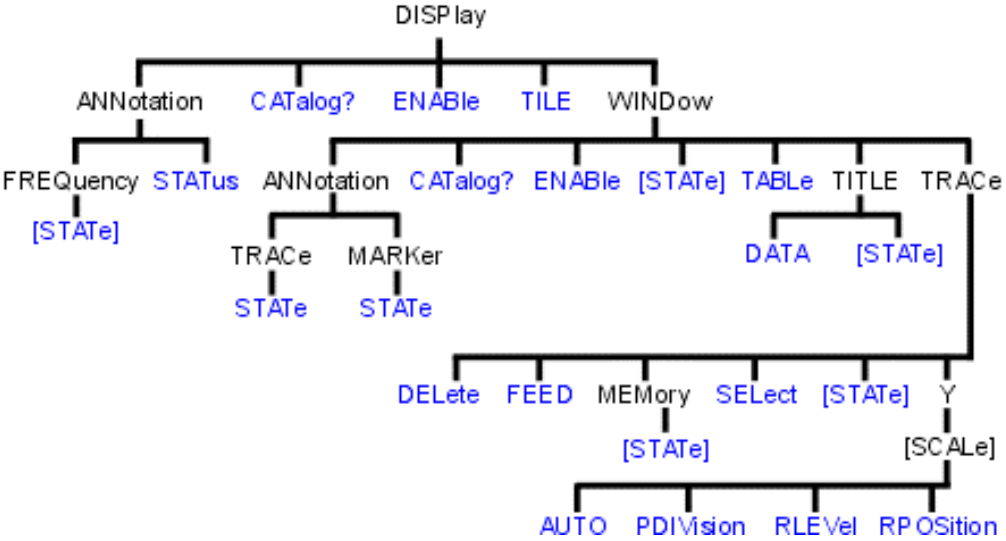
Query Syntax CALCulate<cnum>:TRANsform:TIME[:TYPE]?

Return Type	Character
Overlapped?	No
Default	BPAS

List of all commands in this block:
(Parameters in ***bold italics***)
:CALCulate***1***:TRANsform:TIME:CENTer ***10ms***
:CALCulate***1***:TRANsform:TIME:CENTer?
:CALCulate***1***:TRANsform:TIME:IMPUlse:WIDth ***110ps***
:CALCulate***1***:TRANsform:TIME:IMPUlse:WIDth?
:CALCulate***1***:TRANsform:TIME:KBESsel ***6***
:CALCulate***1***:TRANsform:TIME:KBESsel?
:CALCulate***1***:TRANsform:TIME:LPFREQuency
:CALCulate***1***:TRANsform:TIME:SPAN ***20ms***
:CALCulate***1***:TRANsform:TIME:SPAN?
:CALCulate***1***:TRANsform:TIME:STARt ***10ms***
:CALCulate***1***:TRANsform:TIME:STARt?
:CALCulate***1***:TRANsform:TIME:STATe ***ON***
:CALCulate***1***:TRANsform:TIME:STATe?
:CALCulate***1***:TRANsform:TIME:STEP:RTIMe ***110ps***
:CALCulate***1***:TRANsform:TIME:STEP:RTIMe?
:CALCulate***1***:TRANsform:TIME:STIMulus ***STEP***
:CALCulate***1***:TRANsform:TIME:STIMulus?
:CALCulate***1***:TRANsform:TIME:STOP ***20ms***
:CALCulate***1***:TRANsform:TIME:STOP?
:CALCulate***1***:TRANsform:TIME:TYPE ***LPASs***
:CALCulate***1***:TRANsform:TIME:TYPE?

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

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>

(Write-only) 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 **OFF**

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[:TILE]

(Write-only) Tiles the windows on the screen.

Examples
DISP
display:tile

Overlapped? No

Default Not Applicable

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 **4**); 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>:ANNotation:MARKer:STATe <ON|OFF>

(Read-Write) Specifies whether to show or hide the Marker data (when markers are ON) on the selected window.

Parameters

<wnum>	Any existing window number (**1** to **4**); 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>:CATalog?

(Read-only) Returns the trace numbers for the specified window.

Parameters

<wnum> Any existing window number (**1** to **4**); if unspecified, value is set to 1.

Return Type String of Character values, separated by commas
Example Window 1 with four traces:
DISPlay:WINDow1:CATalog?
Returns:
"1,2,3,4"

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 **4**); 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>[:STATe] <ON | OFF>

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 **4**
<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>

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 **4**); 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 **4**); 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 **4**); 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 **4**); 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 (feed) 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 **4**); 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 **4**); 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 **4**); 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 **4**); 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 **4**); 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 **4**); 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 Character

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 **4**); 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 SCPI Syntax 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 Character

Overlapped? No
Default 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 **4**); 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 SCPI Syntax 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 Character

Overlapped? No
Default 5

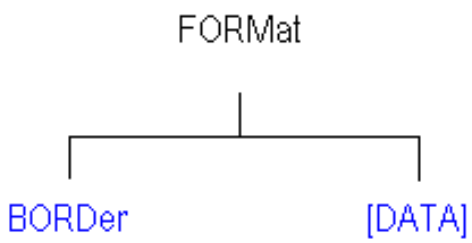
List of all commands in this block:
(Parameters in ***bold italics***)

:DISPlay:ANNotation:FREQuency:STATE ***ON***
:DISPlay:ANNotation:FREQuency:STATE?

:DISPlay:ANNotation:STATus ***ON***
:DISPlay:ANNotation:STATus?
:DISPlay:CATalog?
:DISPlay:ENABLE ***ON***
:DISPlay:ENABLE?
:DISPlay:TILE
:DISPlay:WINDow***1***:ANNotation:TRACe:STATe ***ON***
:DISPlay:WINDow***1***:ANNotation:TRACe:STATe?
:DISPlay:WINDow***1***:ANNotation:MARKer:STATe ***ON***
:DISPlay:WINDow***1***:ANNotation:MARKer:STATe?
:DISPlay:WINDow***1***:CATalog?
:DISPlay:WINDow***1***:ENABle ***ON***
:DISPlay:WINDow***1***:ENABle?
:DISPlay:WINDow***1***:STATe ***ON***
:DISPlay:WINDow***1***:STATe?
:DISPlay:WINDow***1***:TABLe ***OFF***
:DISPlay:WINDow***1***:TABLe?
:DISPlay:WINDow***1***:TITLe:DATA ***"hello"***
:DISPlay:WINDow***1***:TITLe:DATA?
:DISPlay:WINDow***1***:TITLe:STATe ***ON***
:DISPlay:WINDow***1***:TITLe:STATe?
:DISPlay:WINDow***1***:TRACe***1***:DELete
:DISPlay:WINDow***1***:TRACe***1***:FEED ***"test"***
:DISPlay:WINDow***1***:TRACe***1***:MEMory:STATe ***ON***
:DISPlay:WINDow***1***:TRACe***1***:MEMory:STATe?
:DISPlay:WINDow***1***:TRACe***1***:SELect
:DISPlay:WINDow***1***:TRACe***1***:STATe ***ON***
:DISPlay:WINDow***1***:TRACe***1***:STATe?
:DISPlay:WINDow***1***:TRACe***1***:Y:SCALe:AUTO
:DISPlay:WINDow***1***:TRACe***1***:Y:SCALe:PDIVision ***20***
:DISPlay:WINDow***1***:TRACe***1***:Y:SCALe:PDIVision?
:DISPlay:WINDow***1***:TRACe***1***:Y:SCALe:RLEVel ***-10***
:DISPlay:WINDow***1***:TRACe***1***:Y:SCALe:RLEVel?
:DISPlay:WINDow***1***:TRACe***1***:Y:SCALe:RPOSition ***-10***
:DISPlay:WINDow***1***:TRACe***1***:Y:SCALe:RPOSition?

Format Commands

Specifies the way that data will be transferred when moving large amounts of data. These commands will effect 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>

(Write-only) 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
Examples	FORM:BORD SWAP format:border normal
Query Syntax	?Mat:BORDer?
Overlapped?	No
Default	Normal

FORMat[:DATA] <char>

(Read-Write) Sets the data format for use during transfer of measurement data.

Parameters

<char>	Choose from: **REAL,32** - (default value for REAL) Best for transfering 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.
Examples	FORM REAL,64 **format:data ascii**
Query Syntax	FORMat:DATA?
Return Type	Character,Character
Overlapped?	No
Default	REAL,32

List of all commands in this block:
(Parameters in ***bold italics***)

FORMat:BORDer ***NORMal***
FORMat:BORDer?
FORMat:DATA ***ASCii,0***
FORMat:DATA?

Hardcopy Command

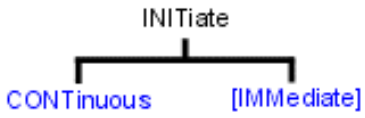
HCOPy[:IMMediate]

(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



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 ¥ Restart on the front panel.)

- 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.

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

List of all commands in this block:
(Parameters in ***bold italics***)
:INITiate:CONTinuous ***ON***
:INITiate:CONTinuous?
:INITiate***2***:IMMediate

Memory Commands

The memory commands control saving and loading instrument states to the hard drive.

- Click on a blue keyword to view the command details.
- See a List of all commands in this block.

All MMEM files have an extension according to their type.

- .sta - Instrument State
- .cal - Calibration file
- .cst - Both Instrument State and Calibration file

The default folder is "C:¥Program Files¥Agilent¥Network Analyzer¥Documents"

You can change the active directory using MMEMory:CDIRectory, or you can use an absolute path name to specify all MMEM files and folders.

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>	Any existing folder name. If unspecified C:¥Program Files¥Agilent¥Network Analyzer¥Documents is used.

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> Name of the file to be copied.

<file2> 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> Name of the file to be deleted.

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> Name of the file to be loaded. The default folder is used if unspecified in the filename.

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> Name of the folder to make.

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> Name of the file to be renamed.
<file2> 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> Name of the folder to remove.

Examples MMEM:RDIR "MyFolder"

Query Syntax Not applicable

Overlapped? No
Default Not applicable

MMEMory:STORe[:<char>] <file>
(Write-only) Stores the specified file.
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)
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> Name of any valid file that is not already in existance.

Examples
```
MMEM:STOR:STAT "myState"
  mmemory:store "c:¥bin¥myState.sta"
```

Query Syntax Not applicable

Overlapped? No

Default Not applicable

List of all commands in this block:
(Parameters in ***bold italics***)

:MMEMory:CATalog:***STATe***? ***"C:¥Program Files¥Agilent¥Network Analyzer¥Documents"***
:MMEMory:CDIRectory ***"¥documents"***
:MMEMory:CDIRectory?
:MMEMory:COPY ***"myState.sta"***,***"yourState"***
:MMEMory:DELete ***"myState.sta"***
:MMEMory:LOAD:***STATe "myState"***
:MMEMory:MDIRectory***"myFolder"***
:MMEMory:MOVE ***"myState.sta"***,***"yourState.sta"***
:MMEMory:RDIRectory ***"myFolder"***
:MMEMory:STORe:***STATe "myState"***

Output Command

OUTPut[:STATe] <ON | OFF>

(Read-Write) Turns RF power from the source ON or OFF.

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

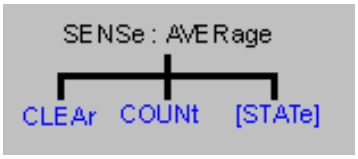
List of all commands in this block:
(Parameters in ***bold italics***)

:OUTPut:STATe ON?
:OUTPut:STATe?

SENSe

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.

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 Character

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

List of all commands in this block:
(Parameters in ***bold italics***)

:SENSe***1***:AVERage:CLEar
:SENSe***1***:AVERage:COUNt ***10***
:SENSe***1***:AVERage:COUNt?
:SENSe***1***:AVERage:STATe ***ON***
:SENSe***1***:AVERage:STATe?

About IF Bandwidth

Sense:Bandwidth Command

SENSe<cnum>:BANDwidth | BWIDth[:RESolution] <num>

(Read-Write) Sets the bandwidth of the digital IF filter to be used in the measurement. The keywords BAND or BWID are interchangeable.

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1
<num>	IF Bandwidth in Hz. Choose from: **1 | 2 | 3 | 5 | 7 | 10 | 15 | 20 | 30 | 50 | 70 | 100 | 150 | 200 | 300 | 500 | 700 | 1k | 1.5k | 2k | 3k | 5k | 7k | 10k | 15k | 20k | 30k | 35k | 40k |** If a number other than these is entered, the analyzer will round up to the closest valid number (unless a number higher than the maximum in entered.)

Examples SENS:BWID 1KHZ
sense2:bandwidth:resolution 1000

Query Syntax SENSe<cnum>:BANDwidth | BWIDth[:RESolution]?
Return Type Character

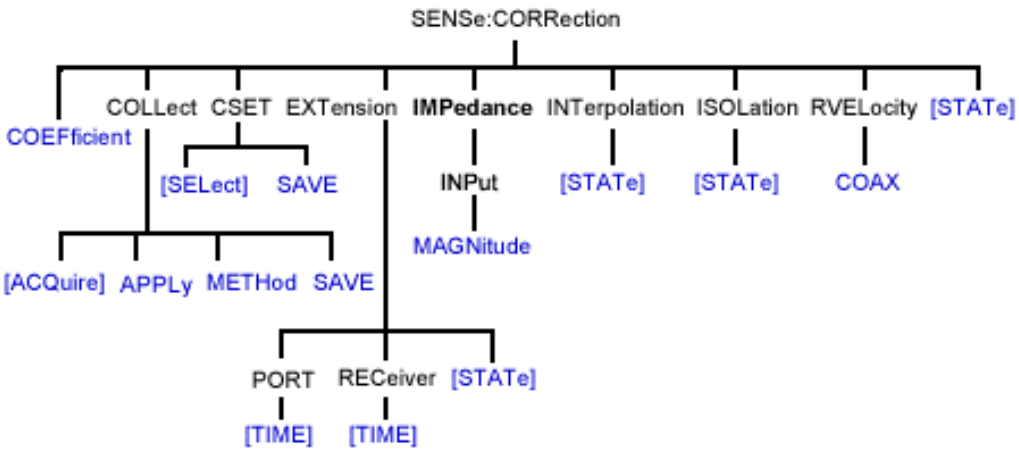
Overlapped? No
Default 35k

List of all commands in this block:
(Parameters in ***bold italics***)
:SENSe***1***:BANDwidth:RESolution ***1000***
:SENSe***1***:BANDwidth:RESolution?
CORRection

About Measurement Calibration

Sense:Correction Commands

Performs and applies measurement calibration and other error correction features.



- 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.

SENSe:CORRection:COEFficient <term>[,port]

(Write-only) Displays the selected error term as trace data. Correction must be turned on.

Note: This command displays error terms for only **Channel 1** measurements

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1
<term>	Error term to be displayed:
[port]	Optional Parameter - Port number of the source drive. If unspecified, value is set to 1.

Examples

```
SENS:CORR:COEF 1
 'Displays the directivity term for port 1 source
SENS:CORR:COEF 2,2
 'Displays the source match for port 2 source
```

Query Syntax Not applicable

Overlapped? No

Default Not applicable

SENSe<cnum>:CORRection:COLLect[:ACQuire] <char>

(Write-only) Measures the specified standard from the selected calibration kit. The calibration kit is selected using the Sense:Correction:Collect:CKIT command.

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1
<char>	Choose from:

Examples

```
SENS:CORR:COLL STAN1
 sense2:correction:collect:acquire stan5
```

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: 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
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.)

Parameters

<cnum> Any existing channel number. If unspecified, value is set to 1

<char> Choose from:

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.

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: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 filenaming 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>

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<cnum>:CORRection:EXTension:PORT<pnum>[:TIME] <num>

(Read-Write) Sets the extension value at the specified port. Must also set SENS:CORR:EXT ON.

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:
<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 Character

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**.**

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:

<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 Character

Overlapped? No
Default 0

SENSe<cnum>:CORRection:EXTension[:STATe] <ON | OFF>

(Read-Write) Turns port extensions ON or OFF.

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 Character

Overlapped? No

Default 50

SENSe<cnum>:CORRection:INTerpolation[:STATe] <ON | OFF>

(Read-Write) Turns correction interpolation ON or OFF.

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:interpolation:state off

Query Syntax SENSe<cnum>:CORRection:INTerpolation[: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 comand is not sent, the default state is to **disable** Isolation.

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<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)
Note: to specify the electrical delay for reflection measurements (in both directions), double the velocity factor.

Examples SENS:CORR:RVEL:COAX .66

sense2:correction:rvelocity:coax .70

Query Syntax SENSe<cnum>:CORRection:RVELocity:COAX?
Return Type Character

Overlapped? No
Default 1

SENSe<cnum>:CORRection[:STATe] <ON | OFF>

(Read-Write) Specifies whether or not correction data is applied to the measurement.

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)

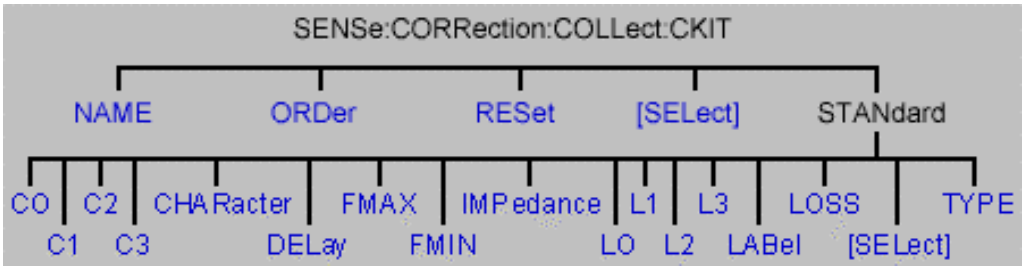
Overlapped? No
Default OFF

List of all commands in this block:
(Parameters in ***bold italics***)

:SENSe:CORRection:COEFficient ***2,1***
:SENSe***1***:CORRection:COLLect:ACQuire ***STAN1***
:SENSe***1***:CORRection:COLLect:ACQuire?
:SENSe***1***:CORRection:COLLect:APPLy
:SENSe***1***:CORRection:COLLect:AVERage:COUNt ***4***
:SENSe***1***:CORRection:COLLect:AVERage:COUNt?
:SENSe***1***:CORRection:COLLect:METHod ***SPARSOLT***
:SENSe***1***:CORRection:COLLect:METHod?
:SENSe***1***:CORRection:COLLect:SAVE
:SENSe***1***:CORRection:CSET:SELect ***USER05***
:SENSe***1***:CORRection:CSET:SELect?
:SENSe***1***:CORRection:CSET:SAVE ***USER03***
:SENSe***1***:CORRection:CSET:SAVE?
:SENSe***1***:CORRection:EXTension:PORT***2***:TIME ***2ms***
:SENSe***1***:CORRection:EXTension:PORT***2***:TIME?
:SENSe***1***:CORRection:EXTension:RECeiver***2***:TIME ***2ms***
:SENSe***1***:CORRection:EXTension:RECeiver***2***:TIME?
:SENSe***1***:CORRection:EXTension:STATe ***ON***
:SENSe***1***:CORRection:EXTension:STATe?
:SENSe***1***:CORRection:INTerpolation:STATe ***ON***
:SENSe***1***:CORRection:INTerpolation:STATe?
:SENSe***1***:CORRection:ISOLation:STATe ***ON***
:SENSe***1***:CORRection:ISOLation:STATe?
:SENSe***1***:CORRection:RVELocity:COAX ***.66***
:SENSe***1***:CORRection:RVELocity:COAX?
:SENSe***1***:CORRection:STATe ***ON***
:SENSe***1***:CORRection:STATe?

Sense:Correction:Collect:CKit Commands

Use to change the definitions of calibration kit standards.



Most of these commands act on the currently selected standard from the currently selected calibration kit.

- To select a Calibration **kit**, use SENS:CORR:COLL:CKIT:SEL.
- To select a Calibration **standard**, use SENS:CORR:COLL:CKIT:STAN:SEL
- Click on a blue keyword to view the command details.
- See a List of all commands in this block.

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: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:ORDer<class> <standard>

(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 Calibration Standards / Fabricating Calibration Standards / Assigning Standards to a Calibration Class

Parameters

<class>	Number of the calibration class that is assigned to <standard>. Choose a number between: **1** and **8** The <class> numbers are associated with the following calibration classes:
<standard>	Standard number between 1 and 8

Examples	Assigns standard 3 to S11A class: SENS:CORR:COLL:CKIT:ORD1 3 Assigns standard 2 to S21T class class: sense:correction:collect:ckit:order4 2
Query Syntax	SENSe:CORRection:COLLect:CKIT:ORDer?

Return Type Character; returns the <class> number of the selected standard.

Overlapped? No
Default Not Applicable

SENSe:CORRection:COLLect:CKIT:RESet <num>

(Write-only) 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. 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. The following are the default settings. Choose from:

Examples SENS:CORR:COLL:CKIT 2
sense2:correction:collect:ckit:select 7

Query Syntax SENSe:CORRection:COLLect:CKIT?
Return Type Character

Overlapped? No
Default 1

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 Character

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 Character

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 Character

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 Character

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 String

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 Character

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 Character

Overlapped? No
Default Not Applicable

SENSe:CORRection:COLLect:CKIT:STANdard:FMIN <num>

(Read-Write) Sets the minumum 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 Character

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 Character

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 Character

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 Character

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 Character

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 Character

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 Character

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 **8**

Examples SENS:CORR:COLL:CKIT:STAN 3
sense:correction:collect:ckit:standard:select 8

Query Syntax SENSe:CORRection:COLLect:CKIT:STANdard[:SELect]?
Return Type Character

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
THRU (Through)

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

List of all commands in this block:
(Parameters in ***bold italics***)
:SENSe:CORRection:COLLect:CKIT:NAME mytypen
:SENSe:CORRection:COLLect:CKIT:NAME?
:SENSe:CORRection:COLLect:CKIT:ORDer***4 2***
:SENSe:CORRection:COLLect:CKIT:ORDer?
:SENSe:CORRection:COLLect:CKIT:RESet ***1***
:SENSe:CORRection:COLLect:CKIT:SELect ***7***
:SENSe:CORRection:COLLect:CKIT?
:SENSe:CORRection:COLLect:CKIT:STANdard:C0 ***15***
:SENSe:CORRection:COLLect:CKIT:STANdard:C0?
:SENSe:CORRection:COLLect:CKIT:STANdard:C1 ***15***
:SENSe:CORRection:COLLect:CKIT:STANdard:C1?
:SENSe:CORRection:COLLect:CKIT:STANdard:C2 ***-15***
:SENSe:CORRection:COLLect:CKIT:STANdard:C2?
:SENSe:CORRection:COLLect:CKIT:STANdard:C3 ***15***
:SENSe:CORRection:COLLect:CKIT:STANdard:C3?
:SENSe:CORRection:COLLect:CKIT:STANdard:CHARacter ***COAX***
:SENSe:CORRection:COLLect:CKIT:STANdard:CHARacter?
:SENSe:CORRection:COLLect:CKIT:STANdard:DELay ***50ps***
:SENSe:CORRection:COLLect:CKIT:STANdard:DELay?
:SENSe:CORRection:COLLect:CKIT:STANdard:FMAX ***9Ghz***
:SENSe:CORRection:COLLect:CKIT:STANdard:FMAX?
:SENSe:CORRection:COLLect:CKIT:STANdard:FMIN ***1khz***
:SENSe:CORRection:COLLect:CKIT:STANdard:FMIN?
:SENSe:CORRection:COLLect:CKIT:STANdard:IMPedance ***50.3***
:SENSe:CORRection:COLLect:CKIT:STANdard:IMPedance?
:SENSe:CORRection:COLLect:CKIT:STANdard:L0 ***15***
:SENSe:CORRection:COLLect:CKIT:STANdard:L0?
:SENSe:CORRection:COLLect:CKIT:STANdard:L1 ***15***
:SENSe:CORRection:COLLect:CKIT:STANdard:L1?
:SENSe:CORRection:COLLect:CKIT:STANdard:L2 ***15***

:SENSe:CORRection:COLLect:CKIT:STANdard:L2?
:SENSe:CORRection:COLLect:CKIT:STANdard:L3 ***15***
:SENSe:CORRection:COLLect:CKIT:STANdard:L3?
:SENSe:CORRection:COLLect:CKIT:STANdard:LABel ***"Short2"***
:SENSe:CORRection:COLLect:CKIT:STANdard:LABel?
:SENSe:CORRection:COLLect:CKIT:STANdard:LOSS ***3.5e9***
:SENSe:CORRection:COLLect:CKIT:STANdard:LOSS?
:SENSe:CORRection:COLLect:CKIT:STANdard:SELECT ***8***
:SENSe:CORRection:COLLect:CKIT:STANdard:SELECT?
:SENSe:CORRection:COLLect:CKIT:STANdard:TYPE ***LOAD***
:SENSe:CORRection:COLLect:CKIT:STANdard:TYPE?

Sense:Couple Command

SENSe<cnum>:COUPle <ALL | NONE>

(Read-Write) Sets the sweep mode as Chopped or Alternate.

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

List of all commands in this block:
(Parameters in ***bold italics***)
:SENSe***1***:COUPle ***ALL***
:SENSe***1***:COUPle?

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.

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 Character

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 Character

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 Character

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

Note: 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?
Return Type Character

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?
Return Type Character

Overlapped? No
Default Maximum frequency of the analyzer

List of all commands in this block:
(Parameters in ***bold italics***)

:SENSe***1***:FREQuency:CENTer ***1mhz***
:SENSe***1***:FREQuency:CENTer?
:SENSe***1***:FREQuency:CW ***100e6***
:SENSe***1***:FREQuency:CW?
:SENSe***1***:FREQuency:SPAN ***10e6***
:SENSe***1***:FREQuency:SPAN?
:SENSe***1***:FREQuency:STARt ***10e6***
:SENSe***1***:FREQuency:STARt?
:SENSe***1***:FREQuency:STOP ***1e9***

:SENSe*1*:FREQuency:STOP?

Sense:Power Command

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 Character

Overlapped? No

Default 0

List of all commands in this block:
(Parameters in ***bold italics***)
:SENSe*1*:POWer:ATTenuation ***AREC,10***
:SENSe*1*:POWer:ATTenuation? ***AREC***

Sense:Roscillator Command

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

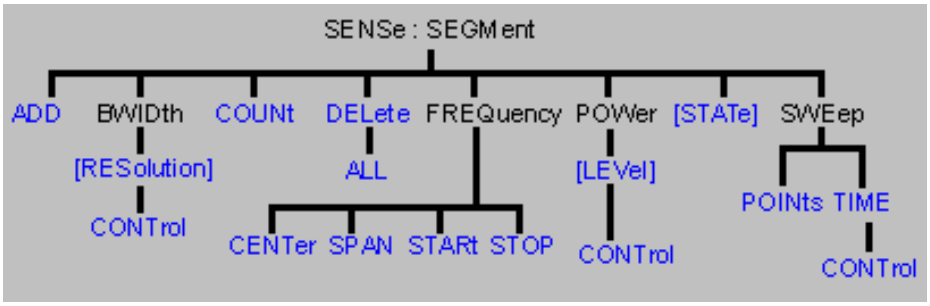
Overlapped? No

Default Not applicable

List of all commands in this block:
(Parameters in ***bold italics***)
:SENSe:ROSCillator:SOURce?

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.

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<snum>:BWIDth[:RESolution] <num>

(Read-Write) Sets the IFBandwidth 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. Choose from: **1 | 2 | 3 | 5 | 7 | 10 | 15 | 20 | 30 | 50 | 70 | 100 | 150 | 200 | 300 | 500 | 700 | 1k | 1.5k | 2k | 3k | 5k | 7k | 10k | 15k | 20k | 30k | 35k | 40k |** If a number other than these is entered, the analyzer will round up to the closest valid number (unless a number higher than the maximum in entered.)

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 Character

Overlapped? No
Default 35k

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 Character

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 Character

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 Character

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.

Note: All other segment Start and Stop Frequency values that are larger than this frequency are changed to this 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>	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 Character

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.

Note: All other segment's Start and Stop Frequency values that are larger than this frequency are changed to this 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>	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 Character

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. Choose from any number between: **-90** and **20**

Examples SENS:SEGM:POW 0
sense2:segment2:power1:level -10

Query Syntax SENSe<cnum>:SEGMent<snum>:POWer[<port>][:LEVel]?

Return Type Character

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>	Choose any number between **2** and **1601** **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 Character

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 Character

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

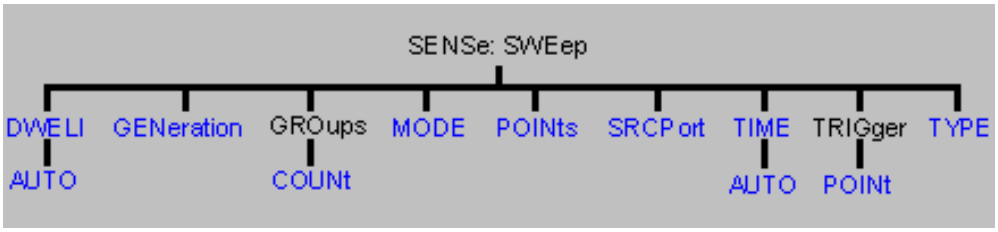
Default OFF

List of all commands in this block:
(Parameters in ***bold italics***)

:SENSe***1***:SEGMent***1***:ADD
:SENSe***1***:SEGMent***1***:BWIDth:RESolution ***1000***
:SENSe***1***:SEGMent***1***:BWIDth:RESolution?
:SENSe***1***:SEGMent:BANDwidth:CONTrol ***ON***
:SENSe***1***:SEGMent:BANDwidth:CONTrol?
:SENSe***1***:SEGMent:COUNt?
:SENSe***1***:SEGMent***1***:DELete
:SENSe***1***:SEGMent:DELete:ALL
:SENSe***1***:SEGMent***1***:FREQuency:CENTer ***1e6***
:SENSe***1***:SEGMent***1***:FREQuency:STARt?
:SENSe***1***:SEGMent***1***:FREQuency:SPAN ***1e6***
:SENSe***1***:SEGMent***1***:FREQuency:SPAN?
:SENSe***1***:SEGMent***1***:FREQuency:START ***1e6***
:SENSe***1***:SEGMent***1***:FREQuency:START?
:SENSe***1***:SEGMent***1***:FREQuency:STOP ***1e6***
:SENSe***1***:SEGMent***1***:FREQuency:STOP?
:SENSe***1***:SEGMent***1***:POWer***1***:LEVel ***-10***
:SENSe***1***:SEGMent***1***:POWer***1***:LEVel?
:SENSe***1***:SEGMent:POWer:LEVel:CONTrol ***ON***
:SENSe***1***:SEGMent:POWer:LEVel:CONTrol?
:SENSe***1***:SEGMent***1***:STATe ***ON***
:SENSe***1***:SEGMent***1***:STATe?
:SENSe***1***:SEGMent***1***:SWEep:POINts ***400***
:SENSe***1***:SEGMent***1***:SWEep:POINts?
:SENSe***1***:SEGMent***1***:SWEep:TIME ***.001***
:SENSe***1***:SEGMent***1***:SWEep:TIME?
:SENSe***1***:SEGMent:SWEep:TIME:CONTrol ***ON***
:SENSe***1***:SEGMent:SWEep:TIME:CONTrol?

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.

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 Character

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.

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 Character

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 **2** and **1601** **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 Character

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 time the analyzer takes to complete one sweep.

Parameters

<cnum>	Any existing channel number. If unspecified, value is set to 1
<num>	Sweep time in seconds. Choose a number between **0** and **86,400** (24hrs) **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 Character

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
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: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

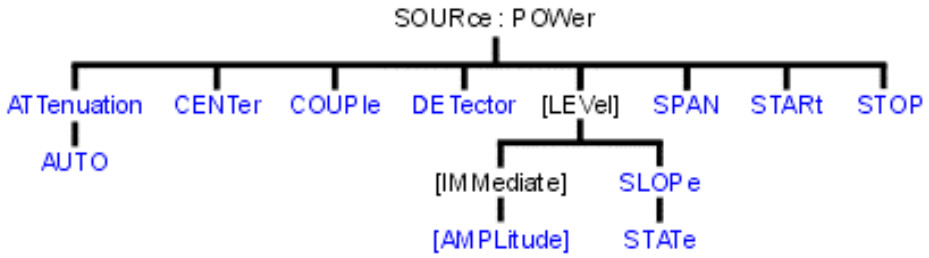
<cnum>	Any existing channel number. If unspecified, value is set to 1
<char>	Choose from: **LINear | POWer | CW | SEGMent** **Note**: SWEep TYPE cannot be set to SEGMent if there are no segments turned ON. A segment is automatically turned ON when the analyzer is started.
Examples	SENS:SWE:TYPE LIN sense2:sweep:type segment
Query Syntax	SENSe<cnum>:SWEep:TYPE?
Return Type	Character
Overlapped?	No
Default	LINear

List of all commands in this block:
(Parameters in ***bold italics***)

:SENSe***1***:SWEep:DWELL ***2ms***
:SENSe***1***:SWEep:DWELL?
:SENSe***1***:SWEep:DWELL:AUTO ***ON***
:SENSe***1***:SWEep:DWELL:AUTO?
:SENSe***1***:SWEep:GENeration ***STEP***
:SENSe***1***:SWEep:GENeration?
:SENSe***1***:SWEep:GROups:COUNt ***50***
:SENSe***1***:SWEep:GROups:COUNt?
:SENSe***1***:SWEep:MODE ***CONT***
:SENSe***1***:SWEep:MODE?
:SENSe***1***:SWEep:POINts ***201***
:SENSe***1***:SWEep:POINts?
:SENSe***1***:SWEep:SRCPort ***1***
:SENSe***1***:SWEep:SRCPort?
:SENSe***1***:SWEep:TIME ***2ms***
:SENSe***1***:SWEep:TIME?
:SENSe***1***:SWEep:TIME:AUTO ***ON***
:SENSe***1***:SWEep:TIME:AUTO?
:SENSe***1***:SWEep:TRIGger:POINt ***ON***
:SENSe***1***:SWEep:TRIGger:POINt?
:SENSe***1***:SWEep:TYPE ***LINear***
:SENSe***1***:SWEep:TYPE?

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.

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>	Choose a number between **0** and **70** dB, 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	Character
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. Choose a number between **-90** and **20** dBm (actual achievable leveled power depends on frequency)

Examples SOUR:POW:CENT -15
source2:power:center -7

Query Syntax SOURce<cnum>:POWer:CENTer?
Return Type Character

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 <INTernal | EXTernal>

(Read-Write) Sets the source leveling loop as Internal or External.

Parameters

<cnum> Any existing channel number. If unspecified, value is set to 1

<INTernal | EXTernal> **INTernal** - Internal leveling is applied to the source
EXTernal - External leveling is applied to the source through a rear-panel jack.

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. Choose any value between **-90** and **+20** dBm
Actual achievable leveled power depends on frequency.
Note: This command will accept **MIN** or **MAX** instead of a numeric parameter. See SCPI Syntax for more information.

Examples SOUR:POW1 5DB
source2:power:level
:immediate:amplitude maximum

Query Syntax SOURce<cnum>:POWer[:LEVel][:IMMediate][:AMPLitude]?
Return Type Character

Overlapped? No
Default 0 dBm

SOURce<cnum>:POWer[:LEVel]:SLOPe <int>

(Read-Write) Sets the RF power slope value.

Parameters

<cnum> Any existing channel number. If unspecified, value is set to 1

<int> 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 Character

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:
IS:SWE:TYPE POWer and SOURce:POWer:CENTer <num>**.**

Parameters

<cnum> Any existing channel number. If unspecified, value is set to 1
<num> Span power. Choose a number between:
-90 and **20** dBm
(actual achievable leveled power depends on frequency)

Examples SOUR:POW:SPAN -15
source2:power:span -7

Query Syntax SOURce<cnum>:POWer:SPAN?
Return Type Character

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. Choose a number between **-90** and **+20** dBm
(actual achievable leveled power depends on frequency)

Examples SOUR:POW:STAR -15
source2:power:start -7

Query Syntax SOURce<cnum>:POWer:STARt?
Return Type Character

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. Choose a number between **-90** and **+20** dBm
(actual achievable leveled power depends on frequency)

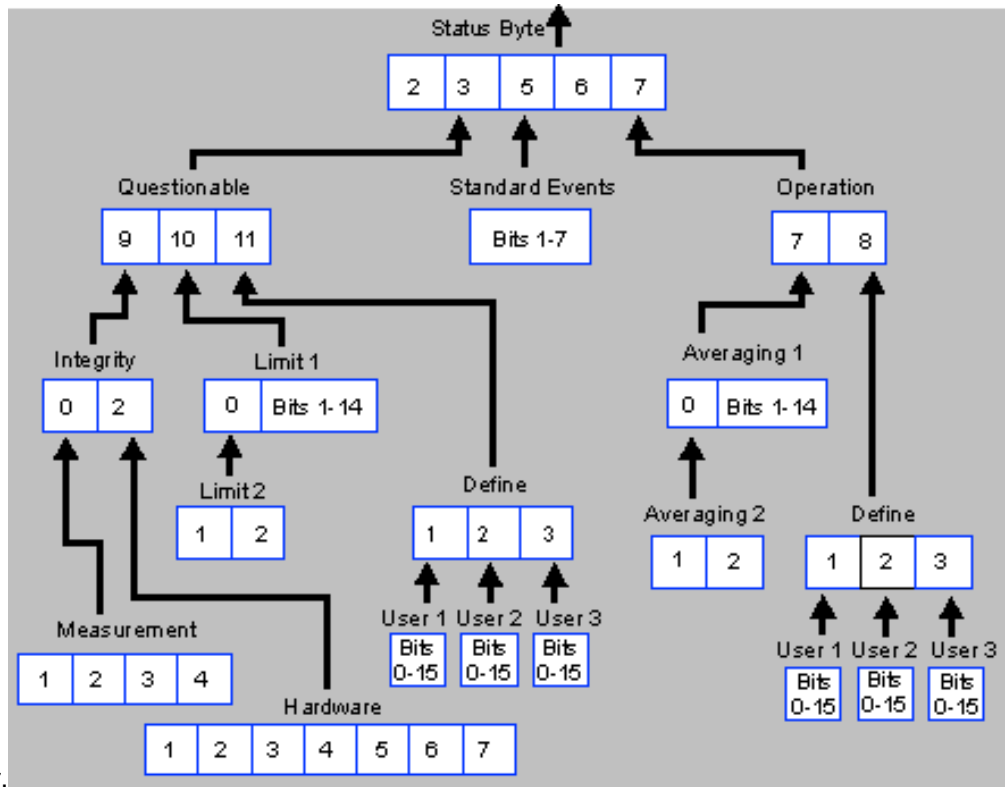
Examples SOUR:POW:STOP -15
source2:power:stop -7

Query Syntax	SOURce<cnum>:POWer:STOP?
Return Type	Character
Overlapped?	No
Default	0 dBm

List of all commands in this block:
(Parameters in ***bold italics***)
:SOURce***1***:POWer***1***:ATTenuation ***5***
:SOURce***1***:POWer***1***:ATTenuation?
:SOURce***1***:POWer***1***:ATTenuation:AUTO ***ON***
:SOURce***1***:POWer***1***:ATTenuation:AUTO?
:SOURce***1***:POWer:CENTer ***-7***
:SOURce***1***:POWer:CENTer?
:SOURce***1***:POWer:COUPle ***ON***
:SOURce***1***:POWer:COUPle?
:SOURce***1***:POWer:DETector ***INTernal***
:SOURce***1***:POWer:DETector?
:SOURce***1***:POWer***1***:LEVel:IMMediate:AMPLitude ***5***
:SOURce***1***:POWer***1***:LEVel:IMMediate:AMPLitude?
:SOURce***1***:POWer:LEVel:SLOPe ***-2***
:SOURce***1***:POWer:LEVel:SLOPe?
:SOURce***1***:POWer:LEVel:SLOPe:STATe ***ON***
:SOURce***1***:POWer:LEVel:SLOPe:STATe?
:SOURce***1***:POWer:SPAN ***-7***
:SOURce***1***:POWer:SPAN?
:SOURce***1***:POWer:STARt ***0***
:SOURce***1***:POWer:STARt?
:SOURce***1***:POWer:STOP ***5***
:SOURce***1***:POWer:STOP?

Status Register Commands

The status registers enable you to query the state of selected events that occur in the



analyzer.

Click on a blue register to view the details

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

Questionable Summary Register

Summarizes conditions that monitor the quality of measurement data.

STATus:QUEStionable:<keyword>

<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

Integrity Summary Register

Summarizes conditions in the Measurement Integrity register.

STATus:QUEStionable:INTegrity <keyword>

<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
0	2	Hardware Summary	any bit in the **Hardware** event register is set to 1

Hardware Register

Monitors the status of hardware failures.

STATus:QUEStionable:INTegrity:HARDware<keyword>

<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	Overpower	too much power is detected at the input. This is from either using an amplifier, or a hardware failure.
4	16	EE Write Failed	an attempted write to the EEPROM has failed, possibly caused by a hardware failure.
5	32	YIG Cal Failed	the analyzer was unable to calibrate the YIG. Either the phaselock has been lost or there has been a hardware failure.
6	64	Ramp Cal Failed	the analyzer was unable to calibrate the analog ramp generator due to a possible hardware failure.
7	128	OverTemp	the source temperature sensor exceeds the limit. It could result from restricted airflow or a

			broken fan

Measurement Register

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.

STATus:QUEStionable:INTegrity:MEASurement<keyword>

<keyword>	**Example**
:CONDition?	STAT:QUES**:INT:MEAS**:COND?
:ENABle <bits>	STAT:QUES**:INT:MEAS**:ENAB 1024
[:EVENt]?	STAT:QUES**:INT:MEAS**?
:NTRansition <bits>	STAT:QUES:**INT:MEAS**:NTR 1024
:PTRansition <bits>	STAT:QUES:**INT:MEAS**:PTR 0

Bit	**Weight**	**Description**	**Bit is set to 1 when the following conditions exist:**
0	1	Channel1	a channel1 setting change has occured and the data does not yet reflect that change.
1	2	Channel2	a channel2 setting change has occured and the data does not yet reflect that change.
2	4	Channel3	a channel3 setting change has occured and the data does not yet reflect that change.
3	8	Channel4	a channel4 setting change has occured and the data does not yet reflect that change.

Limit1 Register

Monitors the status of limit line failures and summarizes the Limit2 event register. When a trace fails, or any bit in the Limit2 event register fails, the representative bit is set to 1. These enable bits are set to 1 by default.

Note:The '**1**' on 'LIMit**1** is a parameter. If unspecified, the value is set to 1.

STATus:QUEStionable:LIMit1<keyword>

<keyword>	**Example**
:CONDition?	STAT:QUES:LIM1:COND?
:ENABle <bits>	STAT:QUES:LIM1:ENAB 1024
[:EVENt]?	STAT:QUES:LIM1?
:NTRansition <bits>	STAT:QUES:LIM1:NTR 1024
:NTRansition?	STAT:QUES:LIM1:NTR?
:PTRansition <bits>	STAT:QUES:LIM1:PTR 0
:PTRansition?	STAT:QUES:LIM1:PTR?

Bit	Weight	Description	Bit is set to 1 when the following conditions exist:
0	1	Limit 2 Reg summary	any point from limit2 event register fails
1	2	Trace 1	any point on trace fails the limit test
2	4	Trace 2	any point on trace fails the limit test
3	8	Trace 3	any point on trace fails the limit test
4	16	Trace 4	any point on trace fails the limit test
5	32	Trace 5	any point on trace fails the limit test
6	64	Trace 6	any point on trace fails the limit test
7	128	Trace 7	any point on trace fails the limit test
8	256	Trace 8	any point on trace fails the limit test
9	512	Trace 9	any point on trace fails the limit test
10	1024	Trace 10	any point on trace fails the limit test
11	2048	Trace 11	any point on trace fails the limit test
12	4096	Trace 12	any point on trace fails the limit test
13	8192	Trace 13	any point on trace fails the limit test
14	16384	Trace 14	any point on trace fails the limit test

Limit2 Register

Monitors the status of limit line failures. When trace 15 or 16 fails, the representative bit is set to 1. These enable bits are set to 1 by default.

Note: The '**2**' on 'LIMit**2** is a parameter. If unspecified, the value is set to 1.

STATus:QUEStionable:LIMit2<keyword>

<keyword>	**Example**
:CONDition?	STAT:QUES:LIM2:COND?
:ENABle <bits>	STAT:QUES:LIM2:ENAB 1024
[:EVENt]?	STAT:QUES:LIM2?
:NTRansition <bits>	STAT:QUES:LIM2:NTR 1024
:PTRansition <bits>	STAT:QUES:LIM2:PTR 0

Bit	Weight	Description	Bit is set to 1 when the following conditions exist:
1	2	Trace15	any point on trace fails the limit test
2	4	Trace16	any point on trace fails the limit test

Define Summary Register

Summarizes conditions in the Questionable:Define:User<1|2|3> event registers.

STATus:QUEStionable:DEFine<keyword>

<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

Questionable User Registers

Monitors conditions that you define and map in any of the three QUES:DEF:USER event registers.

STATus:QUEStionable:DEFine:USER<1|2|3><keyword>

<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.

Operation Summary Register

Summarizes conditions in the Averaging and Operation:Define:User<1|2|3> event registers.

STATus:OPERation<keyword>

<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	

Averaging1 Register

Monitors Averaging of Traces 1 to 14 and summarizes Traces 15 and 16.

Note:The '**1**' on 'AVERaging**1** is a parameter. If unspecified, the value is set to 1.

STATus:OPERation:AVERaging1<keyword>

<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

Bit	Weight	Description	Bit is set to 1 when the following conditions exist:
0	1	AVER2 summary	any enabled bit in AVERaging2 event register is set to 1
1	2	Trace 1	Averaging is complete
2	4	Trace 2	Averaging is complete
3	8	Trace 3	Averaging is complete
4	16	Trace 4	Averaging is complete
5	32	Trace 5	Averaging is complete
6	64	Trace 6	Averaging is complete
7	128	Trace 7	Averaging is complete
8	256	Trace 8	Averaging is complete
9	512	Trace 9	Averaging is complete
10	1024	Trace 10	Averaging is complete
11	2048	Trace 11	Averaging is complete
12	4096	Trace 12	Averaging is complete
13	8192	Trace 13	Averaging is complete
14	16384	Trace 14	Averaging is complete

Averaging2 Register

Monitors Averaging of Traces 15 and 16.

Note: The '**2**' on 'AVERaging**2** is a parameter. If unspecified, the value is set to 1.

STATus:OPERation:AVERaging2<keyword>

<keyword>	**Example**
:CONDition?	STAT:OPER:AVER2:COND?
:ENABle <bits>	STAT:OPER:AVER2:ENAB 1024
[:EVENt]?	STAT:OPER:AVER2?
:NTRansition <bits>	STAT:OPER:AVER2:NTR 1024
:PTRansition <bits>	STAT:OPER:AVER2:PTR 0

Bit	Weight	Description	Bit is set to 1 when the following conditions exist:
1	2	Trace 15	Averaging is complete
2	4	Trace 16	Averaging is complete

Define Summary Register

Summarizes conditions in the OPERation:Define:User<1|2|3> event registers.

STATus:OPERation:DEFine<keyword>

<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

Operation User Registers

Monitors conditions that you define and map in any of the three OPER:DEF:USER event registers.

STATus:OPERation:DEFine:USER<1|2|3><keyword>

<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

About Preset

System Commands

- Click on a blue keyword to view the command details.
- See a List of all commands in this block.

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: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 Default.

Examples SYST:PRES
system:preset

Overlapped? No

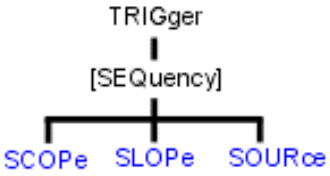
Default Not applicable

List of all commands in this block:
(Parameters in ***bold italics***)

:SYSTem:ERRor?
:SYSTem:ERRor:COUNt?
:STSTem:PRESet

Trigger Commands

Starts or ends a measurement sequence. These commands are an important part of synchronizing measurements.



- Click on a blue keyword to view the command details.
- See a List of all commands in this block.

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. You can use CURRent only when TRIG:SOURCE = MANual.

Examples TRIG:SCOP ALL
trigger:sequence:scope current

Query Syntax TRIGger[:SEQuence]:SCOPe?
Return Type Character

Overlapped? No
Default ALL

TRIGger[:SEQuence]:SLOPe <char>

(Read-Write) Triggers either on the positive going or negative going trigger signal edge. This setting only has an effect when TRIG:SOURce EXTernal is selected.

Parameters

<char> Choose from:
POSitive - external (rear panel) source
NEGative - internal source sends indefinite trigger signals

Examples TRIG:SLOP POS
trigger:sequence:source negative

Query Syntax TRIGger[:SEQuence]:SLOPe?
Return Type Character

Overlapped? No
Default NEGative

TRIGger[:SEQuence]:SOURce <char>

(Read-Write) Sets the source of the sweep trigger signal. This command is a superset of INITiate:CONTinuous, which can NOT set the source to External.

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

List of all commands in this block:
(Parameters in ***bold italics***)
:TRIGger:SEQuence:SCOPe ***ALL***
:TRIGger:SEQuence:SCOPe?
:TRIGger:SEQuence:SLOPe ***POS***
:TRIGger:SEQuence:SLOPe?
:TRIGger:SEQuence:SOURce ***EXT***
:TRIGger:SEQuence:SOURce?
Examples

技術仕様の理解

ここに示すのは、PNAシリーズ・ネットワーク・アナライザの技術仕様の一覧です。この内容をプリントするには、このリンクからPDF(Portable Document File)版をご利用ください。

ボタンをクリックすると下記の内容を見ることができます。

- **定義**
- **補正済みシステム性能**
- **未補正システム性能**
- **テスト・ポート出力(信号源)**
- **テスト・ポートおよびレシーバ入力**
- **一般情報**
- **測定スループット一覧**

定義

すべての仕様と特性は、25℃±5℃の範囲(特に注記しない限り)で、機器の電源をオンにしてから90分後に有効になります。

仕様: 保証される性能。仕様には、予測される性能の統計的分布、測定の不確かさ、環境条件による性能の変化を考慮したガードバンドが含まれています。

特性: この性能パラメータは、工場出荷時には満たされることが予想されますが、フィールドでは検証されておらず、製品保証の範囲外です。特性には仕様と同じガードバンドが含まれます。

代表値: 平均的なユニットの予想される性能で、ガードバンドを含みません。製品保証の範囲外です。

公称値: 説明のための一般的な用語で、性能レベルを示すものではありません。製品保証の範囲外です。

校正: 既知の標準を測定することにより、本器の系統誤差(再現性のある誤差)を評価するプロセス。

補正済み(残留): 誤差補正(校正)後の性能を示します。この性能を決める要因は、校正標準の品質と、標準定義の正確さ、およびシステムの再現性、安定性、ノイズです。

未補正(生): 誤差補正なしでの機器性能を示します。未補正性能は校正の安定性に影響します。

標準: アナライザについて言う場合は、特に注記しない限りすべてのオプションを含みます。

補正済みシステム性能

このセクションの仕様は、PNAシリーズ・アナライザで下記の条件の下に行われた測定に当てはまります。

- 10 Hz IF帯域幅
- データに対してアベレージングを行わない
- 環境温度25℃±5℃で、校正温度からの変化が1℃未満
- アイソレーション校正を省略しない

システム･ダイナミック･レンジ

N型コネクタ

3.5 mmコネクタ

表1. システム･ダイナミック･レンジ

説明	仕様(dB)	特性(dB)
ダイナミック・レンジ[a] (テスト・ポートでの)		
300 kHz～25 MHz[b]	125	
25 MHz～3 GHz[b]	128	
3 GHz～6 GHz	118	
6 GHz～9 GHz	113	
ダイナミック・レンジ[c] (レシーバ入力での)		
300 kHz～25 MHz[d]		140
25 MHz～3 GHz[d]		143
3 GHz～6 GHz		133
6 GHz～9 GHz		128

[a] テスト・ポートのダイナミック・レンジは、テスト・ポートの実効値ノイズ・フロアと信号源の最大出力パワーとの差として計算されます。有効ダイナミック・レンジを計算するには、測定の不確かさと干渉信号を考慮する必要があります。

[b] 750 MHz以下の特定の周波数では、レシーバの残留スプリアスのために100 dBに制限される場合があります。

[c] レシーバ入力のダイナミック・レンジは、レシーバの実効値ノイズ・フロアと信号源の最大出力パワーとの差として計算されます。有効ダイナミック・レンジを計算するには、測定の不確かさと干渉信号を考慮する必要があります。このセットアップが使用できるのは、レシーバ入力がレシーバの損傷レベルを超えないことが確実な場合に限ります。本器をセグメント掃引モードにすれば、大きいダイナミック・レンジが必要な部分(デバイス応答のうち挿入損失が大きい部分)では高いパワー・レベルのセグメントを定義し、レシーバの損傷のおそれがある部分(デバイス応答のうち挿入損失が小さい部分)ではパワーを下げることができます。

[d] 750 MHz以下の特定の周波数では、レシーバの残留スプリアスのために115 dBに制限される場合があります。

N型コネクタ使用時の補正済みシステム性能

- **85032F校正キット**
- **オプション015+85032F校正キット**
- **85092B校正キット**
- **オプション015+85092B校正キット**

表2. N型デバイス・コネクタ、85032F校正キット使用時の補正済みシステム性能

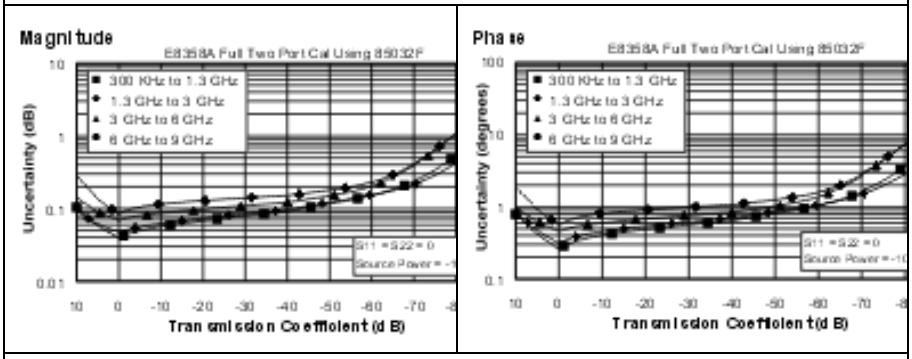
PNAシリーズ・アナライザ、85032F(N型、50Ω)校正キット、N6314Aテスト・ポート・ケーブル、フル2ポート校正の場合に該当。さらに、下記の条件の場合に該当。

- IF帯域幅 = 10 Hz
- データに対してアベレージングを行わない
- 環境温度25℃±5℃で、校正温度からの変化が1℃未満
- アイソレーション校正を省略しない

説明	仕様(dB)			
	300 kHz～1.3 GHz	1.3 GHz～3 GHz	3～6 GHz	6～9 GHz
方向性	49	46	40	38
信号源整合	41	40	36	35
負荷整合	49	46	40	38
反射トラッキング	±0.011	±0.021	±0.032	±0.054
伝送トラッキング	±0.011	±0.018	±0.040	±0.049

曲線をクリックすると拡大されます

伝送不確かさ(仕様)



反射不確かさ(仕様)

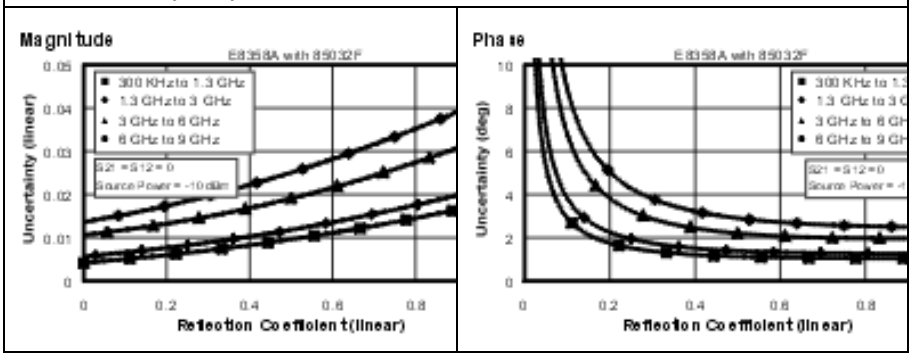


表3. N型デバイス・コネクタ、オプション015+85032F校正キット使用時の補正済みシステム性能

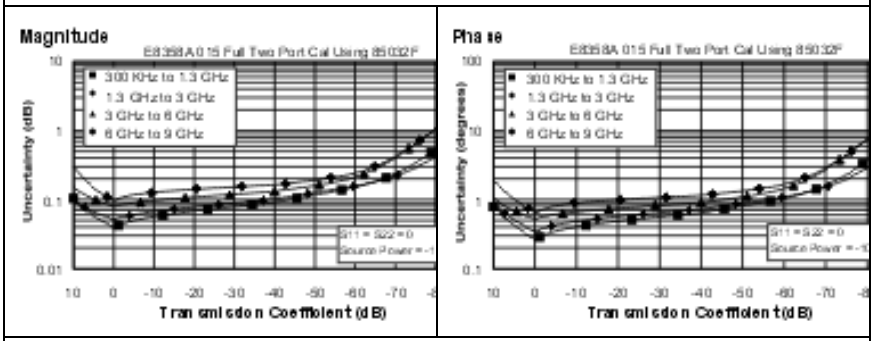
オプション015を装備したPNAシリーズ・アナライザ、85032F(N型、50□)校正キット、N6314Aテスト・ポート・ケーブル、フル2ポート校正の場合に該当。さらに、下記の条件の場合に該当。

- IF帯域幅 = 10 Hz
- データに対してアベレージングを行わない
- 環境温度25℃±5℃で、校正温度からの変化が1℃未満
- アイソレーション校正を省略しない

説明	仕様(dB)			
	300 kHz～1.3 GHz	1.3 GHz～3 GHz	3～6 GHz	6～9 GHz
方向性	49	46	40	38
信号源整合	41	40	36	35
負荷整合	49	46	40	38
反射トラッキング	±0.011	±0.021	±0.032	±0.054
伝送トラッキング	±0.011	±0.023	±0.050	±0.062

曲線をクリックすると拡大されます

伝送不確かさ(仕様)



反射不確かさ(仕様)

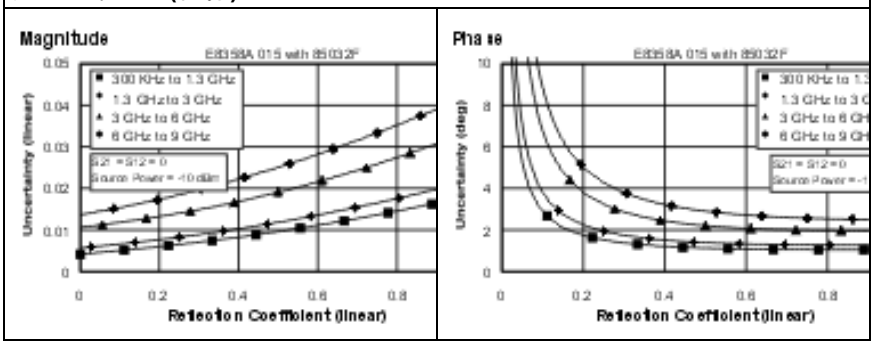


表4. N型デバイス･コネクタ、85092B電子校正モジュール使用時の補正済みシステム性能

PNAシリーズ・アナライザ、85092B(N型、50□)電子校正(ECal)モジュール、N6314Aテスト・ポート・ケーブル、フル2ポート校正の場合に該当。さらに、下記の条件の場合に該当。

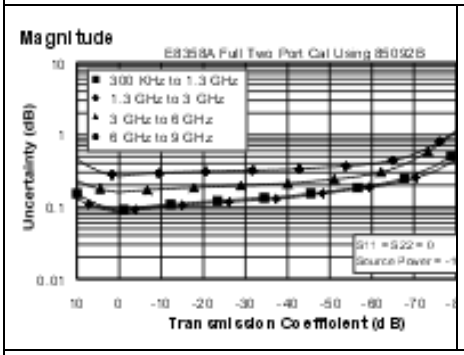
- IF帯域幅 = 10 Hz
- データに対してアベレージングを行わない
- 環境温度25℃±5℃で、校正温度からの変化が1℃未満
- アイソレーション校正を省略しない

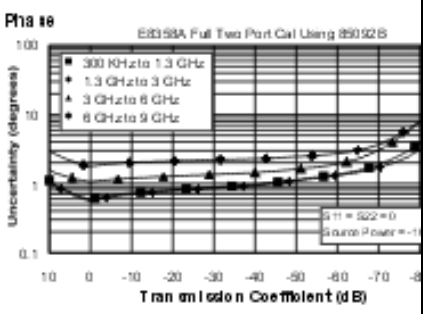
説明	仕様(dB)			
	300 kHz～1.3 GHz	1.3 GHz～3 GHz	3～6 GHz	6～9 GHz[a]
方向性	52	54	52	47
信号源整合	45	44	41	34
負荷整合	47	47	44	44
反射トラッキング	±0.037	±0.037	±0.068	±0.100
伝送トラッキング	±0.060	±0.055	±0.13	±0.23
伝送トラッキング代表値	±0.028	±0.045	±0.08	±0.105

[a] 湿度28%のときの仕様。これより湿度が高いと性能が低下する場合があります。

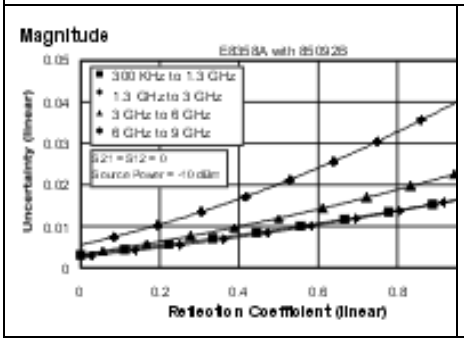
曲線をクリックすると拡大されます

伝送不確かさ(仕様)





反射不確かさ(仕様)



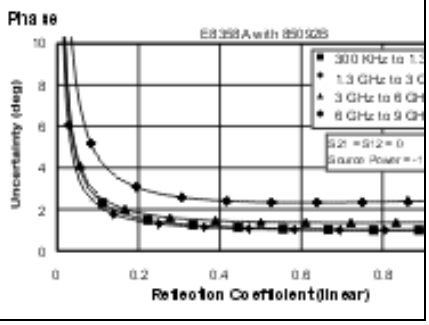


表5. N型デバイス・コネクタ、オプション015+85092B電子校正モジュール使用時の補正済みシステム性能

オプション015を装備したPNAシリーズ・アナライザ、85092B(N型、50□)電子校正(ECal)モジュール、N6314Aテスト・ポート・ケーブル、フル2ポート校正の場合に該当。さらに、下記の条件の場合に該当。

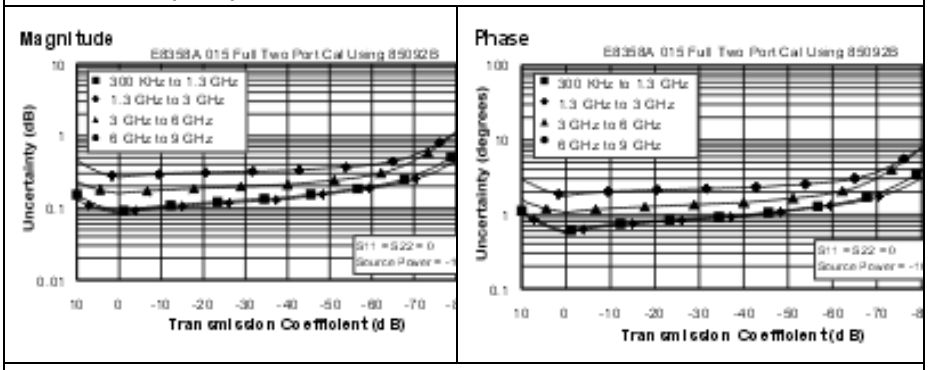
- IF帯域幅 = 10 Hz
- データに対してアベレージングを行わない
- 環境温度25℃±5℃で、校正温度からの変化が1℃未満
- アイソレーション校正を省略しない

説明	仕様(dB)			
	300 kHz～1.3 GHz	1.3 GHz～3 GHz	3～6 GHz	6～9 GHz[a]
方向性	52	54	52	47
信号源整合	45	44	41	34
負荷整合	47	47	44	39
反射トラッキング	±0.037	±0.037	±0.068	±0.100
伝送トラッキング	±0.060	±0.055	±0.13	±0.23
伝送トラッキング代表値	±0.028	±0.045	±0.08	±0.105

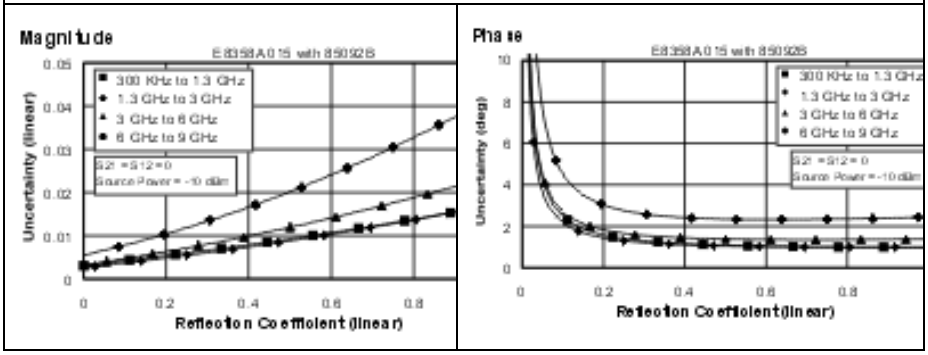
[a] 湿度28%のときの仕様。これより湿度が高いと性能が低下する場合があります。

曲線をクリックすると拡大されます

伝送不確かさ(仕様)



反射不確かさ(仕様)



3.5 mmコネクタ使用時の補正済みシステム性能

- **85033E校正キット**
- **オプション015+85033E校正キット**
- **85093B校正キット**
- **オプション015+85093B校正キット**

表6. 3.5 mmデバイス・コネクタ・タイプ、85033E校正キット使用時の補正済みシステム性能

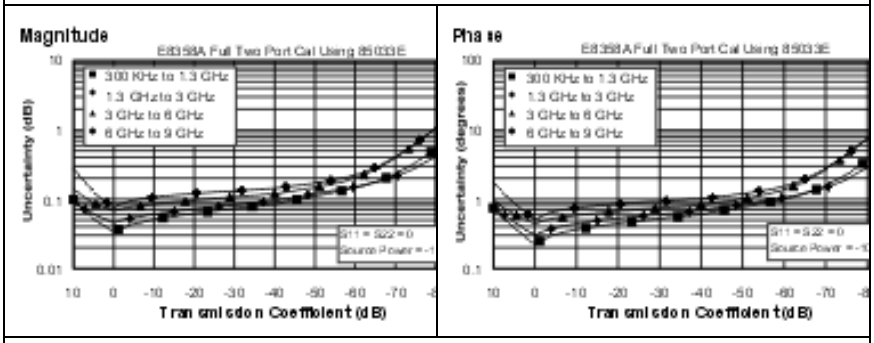
PNAシリーズ・アナライザ、85033E(3.5 mm、50□)校正キット、N6314Aテスト・ポート・ケーブル、フル2ポート校正の場合に該当。さらに、下記の条件の場合に該当。

- IF帯域幅 = 10 Hz
- データに対してアベレージングを行わない
- 環境温度25℃±5℃で、校正温度からの変化が1℃未満
- アイソレーション校正を省略しない

説明	仕様(dB)			
	300 kHz～1.3 GHz	1.3 GHz～3 GHz	3～6 GHz	6～9 GHz
方向性	46	44	38	38
信号源整合	43	40	37	36
負荷整合	46	44	38	38
反射トラッキング	±0.006	±0.007	±0.009	±0.010
伝送トラッキング	±0.010	±0.020	±0.041	±0.046

曲線をクリックすると拡大されます

伝送不確かさ(仕様)



反射不確かさ(仕様)

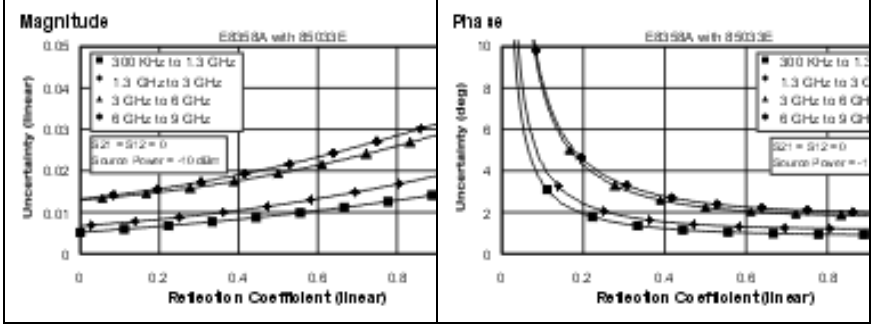


表7. 3.5 mmデバイス・コネクタ・タイプ、オプション015+85033E校正キット使用時の補正済みシステム性能

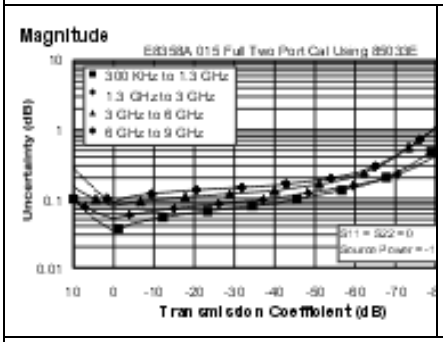
オプション015を装備したPNAシリーズ・アナライザ、85033E(3.5 mm、50□)校正キット、N6314Aテスト・ポート・ケーブル、フル2ポート校正の場合に該当。さらに、下記の条件の場合に該当。

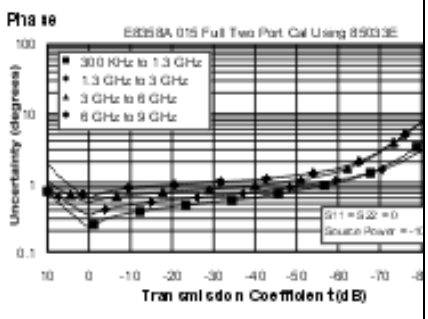
- IF帯域幅 = 10 Hz
- データに対してアベレージングを行わない
- 環境温度25℃±5℃で、校正温度からの変化が1℃未満
- アイソレーション校正を省略しない

説明	仕様(dB)			
	300 kHz～1.3 GHz	1.3 GHz～3 GHz	3～6 GHz	6～9 GHz
方向性	46	44	38	38
信号源整合	43	40	37	36
負荷整合	46	44	38	38
反射トラッキング	±0.006	±0.007	±0.009	±0.010
伝送トラッキング	±0.010	±0.025	±0.052	±0.058

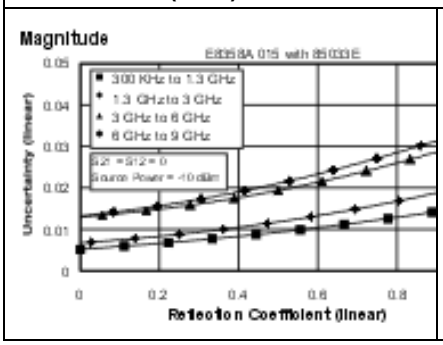
曲線をクリックすると拡大されます

伝送不確かさ(仕様)





反射不確かさ(仕様)



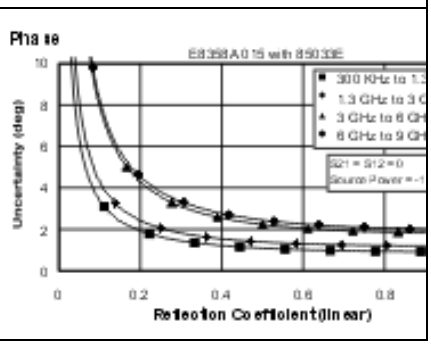


表8.3.5 mmデバイス･コネクタ･タイプ、85093B電子校正モジュール使用時の補正済みシステム性能

PNAシリーズ・アナライザ、85093B(3.5 mm、50□)電子校正(ECal) モジュール、N6314Aテスト・ポート・ケーブル、フル2ポート校正の場合に該当。さらに、下記の条件の場合に該当。

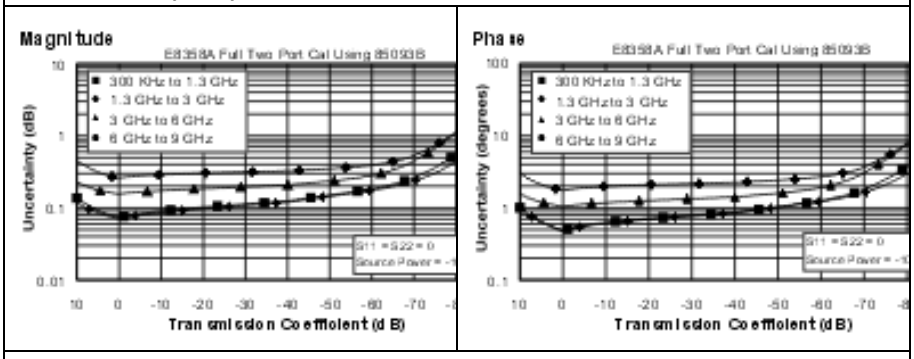
- IF帯域幅 = 10 Hz
- データに対してアベレージングを行わない
- 環境温度25℃±5℃で、校正温度からの変化が1℃未満
- アイソレーション校正を省略しない

説明	仕様(dB)			
	300 kHz～1.3 GHz	1.3 GHz～3 GHz	3～6 GHz	6～9 GHz[a]
方向性	50	52	51	45
信号源整合	45	43	40	34
負荷整合	47	47	44	39
反射トラッキング	±0.043	±0.043	±0.055	±0.100
伝送トラッキング	±0.050	±0.045	±0.13	±0.23
伝送トラッキング代表値	±0.028	±0.045	±0.08	±0.105

[a] 湿度28%のときの仕様。これより湿度が高いと性能が低下する場合があります。

曲線をクリックすると拡大されます

伝送不確かさ(仕様)



反射不確かさ(仕様)

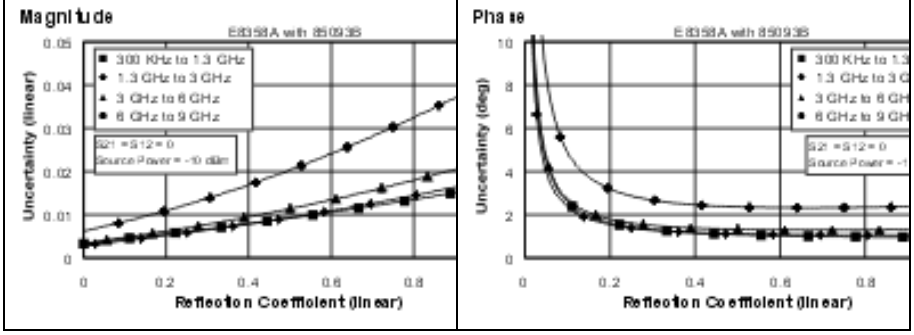


表9. 3.5 mmデバイス・コネクタ・タイプ、オプション015+85093B電子校正モジュール使用時の補正済みシステム性能

オプション015を装備したPNAシリーズ・アナライザ、85093B(3.5 mm、50□)電子校正(ECal)モジュール、N6314Aテスト・ポート・ケーブル、フル2ポート校正の場合に該当。さらに、下記の条件の場合に該当。

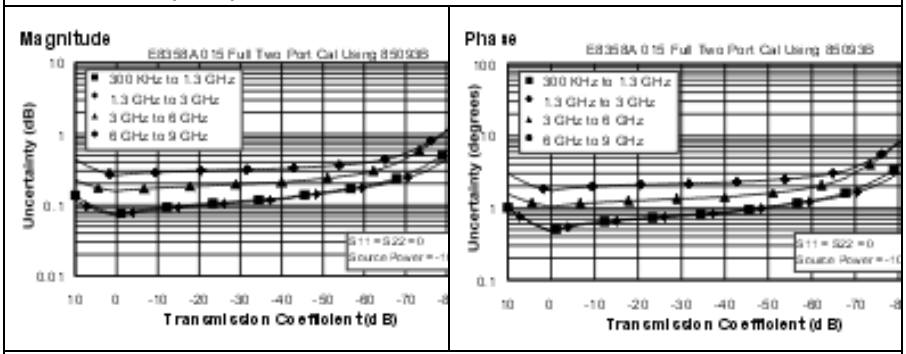
- IF帯域幅 = 10 Hz
- データに対してアベレージングを行わない
- 環境温度25℃±5℃で、校正温度からの変化が1℃未満
- アイソレーション校正を省略しない

説明	仕様(dB)			
	300 kHz～1.3 GHz	1.3 GHz～3 GHz	3～6 GHz	6～9 GHz[a]
方向性	50	52	51	45
信号源整合	45	43	40	34
負荷整合	47	47	44	39
反射トラッキング	±0.043	±0.043	±0.055	±0.100
伝送トラッキング	±0.050	±0.045	±0.13	±0.23
伝送トラッキング代表値	±0.028	±0.045	±0.08	±0.105

[a] 湿度28%のときの仕様。これより湿度が高いと性能が低下する場合があります。

曲線をクリックすると拡大されます

伝送不確かさ(仕様)



反射不確かさ(仕様)

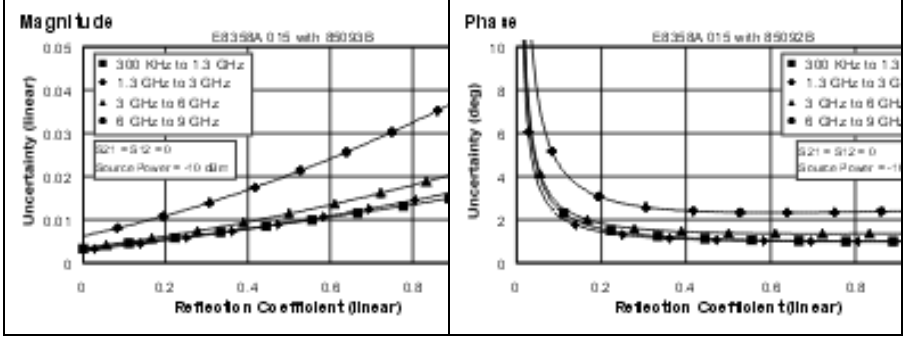


表10. 未補正システム性能

説明	仕様(dB)				
	300 kHz～1 MHz	1MHz～1.3 GHz	1.3 GHz～3 GHz	3～6 GHz	6～9 GHz
方向性	30	33	27	20	13
信号源整合	20	20	17	15	14
信号源整合(オプション015)	20	20	15	13	12
負荷整合	20	20	17	15	15
負荷整合 (オプション015)	20	20	15	13	13
反射トラッキング	±1.5	±1.5	±1.5	±2.5	±3.0
伝送トラッキング	±1.5	±1.5	±1.5	±2.5	±3.0

テスト・ポート出力特性(信号源)

- 出力周波数
- 出力パワー
- 出力信号純度

表11. テスト・ポート出力周波数

説明	仕様	補足情報
レンジ E8356A E8357A E8358A 分解能	 300 kHz～3.0 GHz 300 kHz～6.0 GHz 300 kHz～9.0 GHz 1 Hz	
信号源安定度		±1 ppm、0℃～40℃、代表値 ±0.2 ppm/年、代表値
CW確度	±1 ppm	

表12. テスト・ポート出力パワー[a]

説明	仕様	補足情報
レベル確度 300 kHz～6 GHz 6 GHz～9 GHz	 ±1.0 dB ±2.0 dB	パワー・レンジ0での0 dBmからの変動 (ステップ・アッテネータ0 dB位置) ±1.5dB、10 MHz未満
レベル・リニアリティ 300 kHz～9 GHz 300 kHz～1 MHz 1 MHz～6 GHz	 ±0.3 dB ±1.0 dB ±0.5 dB	パワー・レンジ0での0 dBmからの変動 -15～+5 dBm +5～+10 dBm +5～+10 dBm
レンジ[b]		

300 kHz～6 GHz 6 GHz～9 GHz	85～+10 dBm 85～+5 dBm	
掃引レンジ 300 kHz～6 GHz 6 GHz～9 GHz	 25 dB 20 dB	
レベル分解能	0.01 dB	

[a] ポート1のみの信号源出力性能。ポート2の出力性能は特性。

[b] 信号源を設定可能で、フェーズ・ロックが保証されるパワー。

表13. テスト・ポート出力信号純度

説明	仕様	補足情報
高調波(2次または3次) 出力パワー最大時 0 dBm出力時 10 dBm出力時		 < 25 dBc、特性[a] < 35 dBc、代表値、パワー・レンジ0 < 38 dBc、代表値、パワー・レンジ0 (ステップ・アッテネータ0 dB位置)
非高調波スプリアス 最大出力時 10 dBm出力時		 30 dBc、代表値 、オフセット周波数>1kHzの場合 50 dBc、代表値、オフセット周波数>1kHzの場合

[a] 25 MHzより下では代表値

テスト・ポートおよびレシーバ入力特性

- **入力レベル**
- **ノイズ・フロア**
- **トレース・ノイズ**
- **基準レベルおよび安定度**
- **ダイナミック確度**
- **群遅延**

表14. テスト・ポート入力レベル

説明	仕様	補足情報
最大テスト・ポート入力レベル		
300 kHz～25 MHz 25 MHz～3 GHz 3 GHz～6 GHz 6 GHz～9 GHz	+10 dBm +10 dBm +10 dBm +5 dBm	<0.6 dB圧縮 <0.4 dB圧縮 <0.7 dB圧縮 <0.7 dB圧縮
損傷レベル		
テスト・ポート1、2 R1、R2 IN A、B IN(標準) A、B IN(オプション015) カプラ入力(オプション015)		+30 dBmまたは±30 VDC、代表値 +15 dBmまたは±15 VDC、代表値 +15 dBmまたは±15 VDC、代表値 +15 dBmまたは±0 VDC、代表値 +36 dBmまたは±25 VDC、代表値

テスト・ポート・ノイズ・フロア[a]		
300 kHz～25 MHz[b] IF帯域幅10 Hz IF帯域幅1 kHz	 □≦115 dBm □≦95 dBm	
25 MHz～3 GHz[b] IF帯域幅10 Hz IF帯域幅1 kHz	 □≦118 dBm □≦98 dBm	
3 GHz～9 GHz IF帯域幅10 Hz IF帯域幅1 kHz	 □≦108 dBm □≦88 dBm	
レシーバ・ノイズ・フロア[a]		
300 kHz～25 MHz[c] IF帯域幅10 Hz IF帯域幅1 kHz		 □≦130 dBm、特性 □≦110 dBm、特性
25 MHz～3 GHz[c] IF帯域幅10 Hz IF帯域幅1 kHz		 □≦133 dBm、特性 □≦113 dBm、特性
6 GHz～9 GHz IF帯域幅10 Hz IF帯域幅1 kHz		 □≦123 dBm、特性 □≦103 dBm、特性
クロストーク		
300 kHz～1 MHz 1 MHz～25 MHz 25 MHz～3 GHz 3 GHz～6 GHz 6 GHz～9 GHz	<120 dB <125 dB <128 dB <118 dB <113 dB	テスト・ポート1と2の間、両ポートにショートを接続した状態
最大レシーバ入力レベル(A、B、R1、R2)		
300 kHz～6 GHz 6 GHz～9 GHz		6 dBm、代表値 11 dBm、代表値
基準入力レベル(R1、R2)[d]		
300 kHz～9 GHz		10～35 dBm、代表値
最大カプラ入力レベル(オプション015)		
300 kHz～9 GHz		+33 dBm、代表値

[a] dBm単位のリニア振幅トレースの実効値。

[b] 750 MHz以下の特定の周波数では、レシーバの残留スプリアスのために-90 dBmに制限される場合があります。

[c] 750 MHz以下の特定の周波数では、レシーバの残留スプリアスのために-105 dBmに制限される場合があります。

[d] フェーズ・ロックを維持できる入力レベル。

表15. テスト･ポート入力(トレース･ノイズ)

説明	仕様	補足情報
トレース・ノイズ[a]振幅		
IF帯域幅1 kHz 1F帯域幅10 kHz	< 0.002 dB rms < 0.005 dB rms	
トレース・ノイズ[a]位相		
IF帯域幅1 kHz 1F帯域幅10 kHz	< 0.010° rms < 0.035° rms	

[a] トレース・ノイズは、信号源を0 dBmに設定した状態でのスルーまたは全反射の比測定と定義されます。

表16. テスト・ポート入力(基準レベルおよび安定性)

説明	仕様	補足情報
基準レベル振幅		
レンジ 分解能	±200 dB 0.001 dB	
基準レベル位相		
レンジ 分解能	±500° 0.01°	
安定度振幅[a]		
300 kHz～3 GHz 3 GHz～6 GHz 6 GHz～9 GHz		0.02 dB/℃、代表値 0.04 dB/℃、代表値 0.06 dB/℃、代表値
安定度位相[a]		
300 kHz～3 GHz 3 GHz～6 GHz 6 GHz～9 GHz		0.2°/℃、代表値 0.3°/℃、代表値 0.6°/℃、代表値

[a] 安定度は、テスト・ポートでの比測定として定義されます。

表17. テスト・ポート入力(ダイナミック確度)

テスト・ポート入力パワー表示値の確度は、入力パワー・レベルを基準とします。下記の条件で入力1と2に該当します。

- IF帯域幅 = 10 Hz
- 環境温度25℃±5℃で、校正温度からの変化が1℃未満

曲線をクリックすると拡大されます

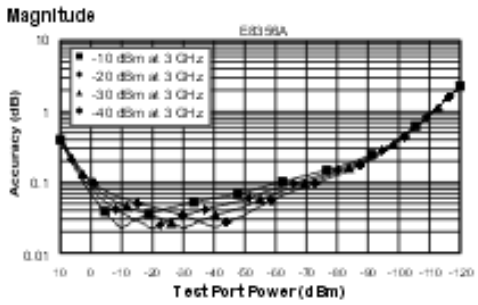
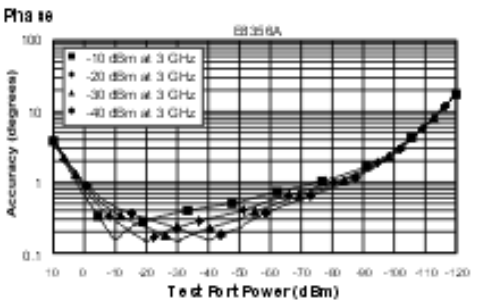
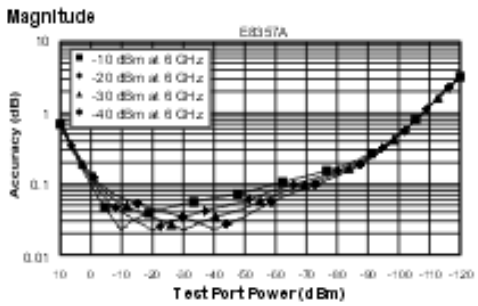
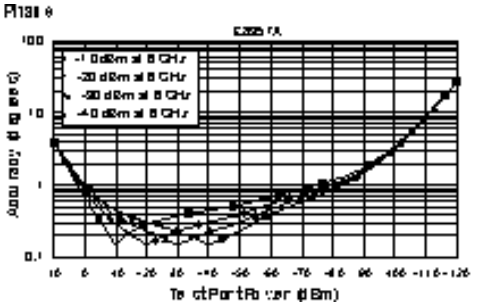
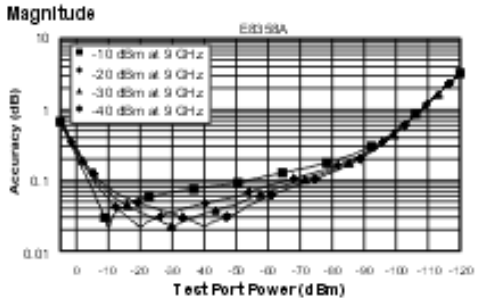
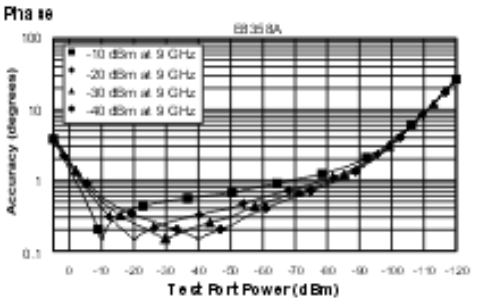
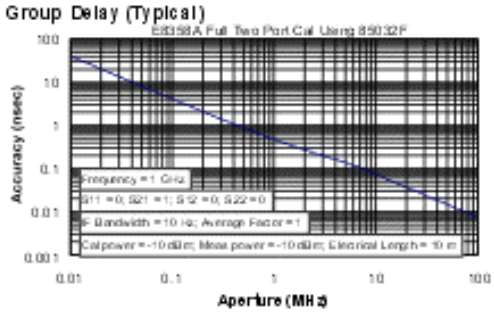
仕様	特性
300 kHz～3 GHz	300 kHz～3 GHz
	
300 kHz～6 GHz	300 kHz～6 GHz
	
300 kHz～9 GHz	300 kHz～9 GHz
	

表18. テスト・ポート入力(群遅延)[a]

説明	仕様	補足情報
アパーチャ(選択可能)	(周波数スパン)/(ポイント数1)	
最大アパーチャ	周波数スパンの20%	
レンジ	0.5×(1/最小アパーチャ)	
最大遅延		測定は最小アパーチャ内の180° 以内の位相変化に制限
確度		下図参照。特性。

下のグラフは、N型フル2ポート校正と10 HzのIF帯域幅を使用した場合です。挿入損失は2 dB未満、電気長は10 mと仮定します。

曲線をクリックすると拡大されます



一般に、個々の群遅延測定の確度(秒単位)は下記の式で求められます。

±位相確度(°)/[360×アパーチャ(Hz)]

アパーチャとデバイス長によって、使用する位相確度はインクリメンタル位相確度またはワーストケース位相確度のどちらかです。

[a] 群遅延は、指定された周波数ステップ(周波数スパンと掃引あたりのポイント数から決定)内の位相変化の測定結果から求められます。

一般情報

- システム帯域幅
- フロントパネル
- リアパネル
- 環境仕様と外形寸法

表19. システム帯域幅

説明	仕様	補足情報
IF帯域幅設定		
レンジ		1 Hz～40 kHz 公称設定値は下記の通り: 1、2、3、5、7、10、15、20、30、50、70、100、150、200、300、500、700 Hz 1k、1.5k、2k 3k、5k、7k、10k、15k、20k、30k、50k、35k、40kHz

表20. フロントパネル情報

説明	補足情報
RFコネクタ	
タイプ	N型、メス、50□、公称値
センター・ピン突出部長さ	0.204～0.207インチ、特性
プローブ電源	
コネクタ	3ピン・コネクタ、オス
正電源	+15 VDC±2%、最大400 mA、特性
負電源	12.6 VDC±5%、最大300 mA、特性
ディスプレイ	
サイズ	対角長21.3 cm(8.4インチ)アクティブ・マトリクスLCD。解像度640(水平)×480(垂直)。
リフレッシュ・レート	垂直59.83 Hz、水平31.41 Hz
表示範囲	
振幅	最大±200 dB(20 dB/divで)
位相	最大±180°
極座標	最小10 p単位 最大1000ユニット
表示分解能	
振幅	最小0.001 dB/div
位相	最小0.01°/div
マーカ分解能	
振幅	最小0.001 dB
位相	最小0.01°
極座標	最小0.01 m単位、最小0.01°

表21. リアパネル情報

説明	補足情報
テスト・ポート・バイアス入力	
コネクタ	BNC、メス
最大電圧	±30 VDC、代表値
最大電流(RF仕様の悪化を伴わない)	±200 mA、代表値
最大電流	±1 A、代表値
10 MHz基準入力	
コネクタ	BNC、メス
入力周波数	10 MHz ± 1 ppm、代表値
入力レベル	15 dBm～+20 dBm、代表値
入力インピーダンス	200 □、公称値
10 MHz基準出力	
コネクタ	BNC、メス
出力周波数	10 MHz ± 1 ppm、代表値
信号タイプ	正弦波、代表値
出力レベル	+10 dBm ± 4 dB、50 □に対して、代表値
出力インピーダンス	50 □公称値
高調波	<40 dBc、代表値
VGAビデオ出力	
コネクタ	15ピン・ミニD-sub。VGA互換モニタ用。
サポート対象デバイス フラット・パネル(TFT) フラット・パネル(DSTN) CRTモニタ	解像度 1024×768、800×600、640×480 800×600、640×480 1280×1024、1024×768、800×600、640×480 内蔵ディスプレイと外部ディスプレイの同時動作は640×480の解像度でのみ可能です。解像度を変更した場合、外部ディスプレイだけが見えるようになります(内蔵ディスプレイは空白表示になります)。
GPIB	24ピンD-sub(D-24型)、メス、IEEE-488互換
パラレル・ポート(LPT1)	36ピンD-sub(1284-C型)、メス、プリンタやその他のパラレル・ポート周辺機器との接続用
シリアル・ポート(COM1)	9ピンD-sub、オス、RS-232互換
USBポート	
接点1 接点2 接点3 接点4	Universal Serial Busジャック、タイプA構成(4接点インライン、接点1が左)、メス型 Vcc: 4.75～5.25 VDC、最大500 mA -データ +データ グランド
LAN	10/100BaseTイーサネット、8ピン構成。 2種類のデータ・レートを自動選択。
電源[a]	
周波数	48 Hz～66 Hz
115 V設定での電圧	90～132 VAC、公称値120 VAC

220 V設定での電圧	198～264 VAC、公称値240 VAC
VA最大値	最大600 VA

[a] 電源と別にアース線が必要。

表22. リアパネル情報

説明	補足情報
外部AM入力	
説明	テスト・ポート出力信号を低周波AM変調したり、テスト・ポート出力をシフトしたりするために使います。入力が0 Vなら本器で設定されたパワー・レベル、入力が正の電圧なら設定よりも高いレベル、負の電圧なら設定よりも低いレベルが出力されます。
コネクタ	BNC、メス
入力感度	8 dB/V、代表値
帯域幅	1 kHz、代表値
入力インピーダンス	1 k□、代表値
外部ディテクタ入力	
説明	外部の負極性ダイオード・ディテクタからの入力により、本器のフロントパネルから遠い位置にあるテスト・ポートのALCを可能にします。
コネクタ	BNC、メス
入力感度	500 mVでディテクタ入力約3 dBm、代表値
帯域幅	50 kHz、代表値
入力インピーダンス	1 k□、代表値
外部トリガ	
	トリガを参照

表23. 環境仕様と外形寸法

説明	補足情報
一般環境仕様	
RFI/EMI感受性	CISPR Pub. 11、Group 1、Class A、およびIEC 50082-1で定義
ESD	静電気防止のための作業手順を守り、静電気防止マットを使用することで防いでください。
埃	最高の信頼性を実現するため、なるべく防いでください。
動作環境	
温度	0 ℃～+40 ℃ この温度範囲内で、本器は起動し、フェーズロックし、エラー・メッセージを表示せずに動作します。
誤差補正温度範囲	25℃ ± 5℃ 校正温度からの変化が1℃以内
湿度	5%～95%、+40 ℃で
高度	0～4500 m
保管環境	
温度	-40 ℃～+70 ℃
湿度	0%～90%、+65 ℃で(非結露)
高度	0～15,240 m
キャビネット寸法	
高さ×幅×奥行き	222×425×426 mm、公称値。キャビネット寸法は前面と裏面の突起部を含みません。
質量	
正味	24 kg、公称値
輸送時	32 kg、公称値

測定スループット一覧

- 測定終了までのサイクル・タイム代表値
- IF帯域幅の変更を伴うサイクル・タイム
- ポイント数の変更を伴うサイクル・タイム
- データ転送時間
- リコールおよび掃引速度

表24. サイクル・タイム代表値[a,b] (ms)

	ポイント数			
	51	**201**	**401**	**1601**
スタート800 MHz、ストップ1000 MHz、IF帯域幅35 kHz				
未補正、 1ポート校正	6	10	17	53
2ポート校正	18	27	39	113
スタート300 kHz、ストップ3 GHz、IF帯域幅35 kHz				
未補正、 1ポート校正	32	43	52	93
2ポート校正	73	97	117	201
スタート300 MHz、ストップ6 GHz、IF帯域幅35 kHz				
未補正、 1ポート校正	40	50	57	98
2ポート校正	88	109	125	210
スタート300 kHz、ストップ9 GHz、IF帯域幅35 kHz				
未補正、 1ポート校正	45	55	61	99
2ポート校正	99	119	133	212
時間領域[c] (未補正掃引時間に対する増加分)				
変換	<1	<1	4	13
ゲーティング	<1	<1	4	17

[a] 代表値

[b] 掃引時間、再トレース時間、帯域クロス時間を含む。DISPLAY:ENABLE OFFで表示をオフにした状態。表示オンでは21 msを加算。1トレース(S11)測定のデータ。

[c] オプション010のみ。DISPLAY:ENABLE OFFで表示をオフにした状態。表示オンでは21 msを加算。1トレース(S11)測定。

表25. サイクル・タイムと IF帯域幅[a]

初期設定状態(201ポイント、補正オフ))から下記の変更だけを行った状態の場合に該当。

- CF = 1 GHz
- スパン = 100 MHz
- 表示オフ(表示オンでは21 msを加算

IF帯域幅(Hz)	サイクル・タイム (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] 代表値

[b] サイクル・タイムは掃引時間と再トレース時間を含む。

表26. サイクル・タイムとポイント数[a]

初期設定状態(IF帯域幅35 kHz、補正オフ)から下記の変更だけを行った状態の場合に該当。

- CF = 1 GHz
- スパン = 100 MHz
- 表示オフ(表示オンでは21 msを加算

ポイント数	サイクル・タイム (ms)[b]
3	4
11	4
51	5
101	6
201	9
401	16
801	29
1601	52

[a] 代表値

[b] サイクル・タイムは掃引時間と再トレース時間を含む。

表27. データ転送時間[a] (ms)

	ポイント数			
	51	**201**	**401**	**1601**
SCPI、GPIB経由 (プログラムを外部PCで実行)[b]				
32ビット浮動小数点数	4	7	13	41
64ビット浮動小数点数	7	14	24	81
ASCII	25	98	189	804
SCPI、10 Mビット/s LAN経由 (プログラムを外部PCで実行)[b]				
32ビット浮動小数点数	5	6	8	21
64ビット浮動小数点数	5	9	22	38
ASCII	18	53	98	362
SCPI、100 Mビット/s LAN経由 (プログラムを外部PCで実行)[c]				
32ビット浮動小数点数	3	5	6	12
64ビット浮動小数点数	4	6	9	20
ASCII	17	51	92	339
SCPI(プログラムを本器で実行)[d]				
32ビット浮動小数点数	2	3	4	7
64ビット浮動小数点数	4	5	6	15
ASCII	26	99	198	781
COM(プログラムを本器で実行)[e]				
32ビット浮動小数点数	1	1	1	2
バリアント形	1	3	4	19
DCOM、10 Mビット/s LAN経由 (プログラムを外部PCで実行)[f]				
32ビット浮動小数点数	2	3	5	14
バリアント形	5	14	26	100
DCOM、100 Mビット/s LAN経由 (プログラムを外部PCで実行)[f]				
32ビット浮動小数点数	2	2	2	4
バリアント形	3	5	9	35

[a] 代表値

[b] 600 MHzのHP Kayak上のVEE 5.0プログラム、National InstrumentsTM GPIBカードを使って測定。"CALC:DATA?SDATA"を使って複素S11データを転送。

[c] 600 MHzのHP Kayak上のVEE 5.0プログラムを使って測定。"CALC:DATA?SDATA"を使って複素S11データを転送。ネットワークに接続されている場合、速度はLANトラヒックに依存します。

[d] PNAシリーズ・アナライザ内部で動作するVEE 5.0プログラムを使って測定。"CALC:DATA?SDATA"を使って複素S11データを転送。

[e] PNAシリーズ・アナライザ内部で動作するVisual Basic 6.0プログラムを使って測定。複素S11データを転送。

[f] 600 MHzのHP Kayak上のVisual Basic 6.0プログラムを使って測定。複素S11データを転送。ネットワークに接続されている場合、速度はLANトラヒックに依存します。

[g] 32ビット浮動小数点数に対してはIArray Transfer.getComplexメソッドを使用。

[h] バリアント・データ型に対してはmeas.getDataメソッドを使用

表28. リコールおよび掃引速度

動作	ウィンドウ数	トレース数	リコール時間(ms)
リコール	1	1	140
リコールおよび掃引	1	1	198
リコール	1	2	259
リコールおよび掃引	1	2	341
リコール	1	4	588
リコールおよび掃引	1	4	723
リコール	2	2	282
リコールおよび掃引	2	2	373
リコール	3	4	610
リコールおよび掃引	3	4	788
リコール	4	4	613
リコールおよび掃引	4	4	790
リコール	4	8	1236
リコールおよび掃引	4	8	1463

製品サポートの取得

- その他のリソース
- アナライザ・オプション
- テクニカル・サポート
- アナライザのトラブル・シューティング
- アクセサリ
- ファームウェアおよびヘルプのアップグレード

その他のリソース

ネットワーク解析に関するその他のリソースを以下に示します。

ドキュメント・リソース

サード・パーティ・リソース

ドキュメント・リソース

アプリケーション・ノート

以下のアプリケーション・ノートのいくつかは、本器に収録されています。

1. **Help**メニューで**Product Overview**をクリックします。
2. Product Overviewはイントロダクションから始まります。これを飛ばして先に進むには、**Main Menu**をクリックします。
3. メイン・メニューで**Literature**をクリックし、文献タイトルをクリックするとドキュメントが表示されます。

また、以下のアプリケーション・ノートとプロダクト・ノートは、下記のURLからアクセスできます。

http://www.agilent.com/find/PNA

- 『Application Development with the Agilent PNA Series of Network Analyzers』(カタログ番号5980-2666ENUS)(英文)
- 『ネットワーク・アナライザのダイナミック・レンジの意義と改善』Application Note 1363-1 (カタログ番号5980-2778J)
- 『The Need for Speedin Component Manufacturing Test』(カタログ番号5980-2783EN)(英文)
- 『Generate Component Data Sheets with Agilent's BenchLinkXL』(カタログ番号5980-2781EN)(英文)
- 『Agilentの最先端コネクティビティ—コンポーネント・テスト向け—』(カタログ番号5980-2782J)
- 『Agilentベクトル・ネットワーク・アナライザを使用したSパラメータ・ネットワークのディエンベディングおよびエンベディング』 Application Note 1364-1 (カタログ番号5980-2784J)
- 『Advanced Filter Tuning Using TIme Domain』Application Note 1287-10 (カタログ番号5980-2785EN)(英文)
- 『ベクトル・ネットワーク解析の基礎』Application Note 1287-1 (カタログ番号5965-7707J)
- 『ネットワーク・アナライザのアーキテクチャ』Application Note 1287-2 (カタログ番号5965-7708J)
- 『ネットワーク・アナライザ測定に対する 誤差補正の適用』Application Note 1287-3 (カタログ番号5965-7709J)
- 『ネットワーク・アナライザ測定 : フィルタとアンプ』Application Note 1287-4 (Agilent カタログ番号5965-7710J)
- 『ネットワーク・アナライザのアプリケーションにおけるスループットの改善』Application Note 1287-5 (カタログ番号 5966-3317J)

- 『ネットワーク・アナライザによるハイ・パワー・デバイスの評価』Application Note 1287-6 (カタログ番号 5966-3319J)
- 『Agilent Technologiesネットワーク解析ソリューション　タイム・ドメインを使ったフィルタ調整の簡略化』Application Note 1287-8 (カタログ番号5968-5328J)
- 『Agilent Technologiesネットワーク解析ソリューション　ベクトル・ネットワーク・アナライザを使用したインフィクスチャ測定』Application Note 1287-9 (カタログ番号5968-5329J)
- 『ネットワーク・アナライザ測定を成功させる8つのヒント』Application Note 1291-1 (カタログ番号 5965-8166J)
- 『Specifying Calibration Standards for the Agilent 8510 Network Analyzer』Agilent Product Note 8510-5A (カタログ番号5956-4352、パーツ番号08510-90352、1988年2月)(英文)
- 『In-Fixture Microstrip Device Measurements Using TRL* Calibration』Agilent Product Note 8720-2 (カタログ番号5091-1943E、1991年8月)(英文)

サード・パーティ・リソース

テスト・フィクスチャおよびパーツ・ハンドラに関するお問い合わせ先:

Inter-Continental Microwave
1515 Wyatt Drive
Santa Clara, CA 95054-1524 USA
電話: (408) 727-1596
Fax: (408) 727-0105
Webサイト: www.icmicrowave.com
電子メール: icmfixture@aol.com

プロービング機器およびアクセサリに関するお問い合わせ先:

Cascade Microtech, Inc.
2430 NW 206th Avenue
Beaverton, OR 97006 USA
電話: (503) 601-1000
Fax: (503) 601-1002
Webサイト: www.cascademicrotech.com
電子メール: sales@cmicro.com

平衡測定、マルチポート測定に関するお問い合わせ先:

ATN Microwave, Inc.
85 Rangeway Road
N. Billerica, MA 01862-2105 USA
電話 (978) 667-4200
Fax: (978) 667-8548
Webサイト: www.atnmicrowave.com

アナライザ・オプション

本器には、マウス、キーボード、操作マニュアルとプログラミング・マニュアルを収録したCD-ROM、ミニ・パラレル-パラレル・アダプタ、3年間のAgilentへの返送サービス保証が付属しています。このトピックでは、別売のオプションについて説明します。

- **機器オプション**
- **オプション・アップグレード**
- **ドキュメンテーションとローカライゼーション**
- **サービスとサポート**

機器オプション

オプション010　時間領域機能
時間領域での反射・伝送応答測定用
時間領域の概要参照

オプション015　構成可能テスト・セット
ポート1と2の信号源出力と結合器入力にフロントパネルからアクセス可能になります。これにより、結合器とレシーバとの間に35 dBのステップ・アッテネータが追加されます。このオプションを使うことで、低レベル信号の測定感度の改善と、さまざまな測定アプリケーションのためのコンポーネントや他の周辺機器の追加が可能になります。
「ハイパワー・コンポーネント測定」を参照してください。

オプション1CM　ラック・マウント・キット
ラック・マウント(パーツ番号5063-9216)およびレール・キット(E3663AC)を追加します。フロント・ハンドルなしで使用します。

オプション1CP　ラック・マウント・キット
ラック・マウント(パーツ番号5063-9236)およびレール・キット(E3663AC)を追加します。購入済みのフロント・ハンドルを付けて使用します。

オプションAM8　CD-RWドライブ
USBケーブル付き外付けCD-RWドライブ。

オプションB30　USBハブ
USB周辺機器を増設するための4ポートUSBハブ。

オプション・アップグレード

本器の購入後にオプションを追加するためのアップグレード・キットが用意されています。E8356A、E8357A、E8358A用アップグレード・キットのご注文の際には、アナライザのモデル番号の後に"U"を付けて、その後に追加するオプションをご指定ください。

オプション010　時間領域アップグレード・キット(パーツ番号E8356-60101)
このキットをご注文の際には、E8356A、E8357A、またはE8358Aのシリアル番号が必要です。
インストールは含まれません。

オプション015　構成可能テスト・セット・アップグレード・キット(パーツ番号E8356-60102)
Agilentサービス・センタでのインストールを含みます。

オプションAM8　CD-RWドライブ
USBケーブル付き外付けCD-RWドライブ。

オプションB30　USBハブ
USB周辺機器を増設するための4ポートUSBハブ。

オプション099　ファームウェア・アップグレード・キット(パーツ番号E8356-60103)
PNAシリーズ・アナライザ用ファームウェアの最新リビジョンを収録した

CD-ROM。ファームウェアはユーザがインストール可能です。インストールには、USB接続のCD-ROMドライブか、LAN接続の外部コンピュータが必要です。最新ファームウェアはAgilentのウェブ・サイトからも入手できます。ウェブ・ページ:
www.agilent.com/find/pna
インストール手順については、ファームウェア・アップグレードを参照してください。

モデル番号の変更が伴うアップグレードの詳細については、Agilent計測お客様窓口にお問い合わせください。Agilent連絡先を参照してください。

ドキュメンテーションとローカライゼーション

オプションAVK ヘルプ内容の印刷物、すなわちユーザーズ・ガイドとプログラマーズ・ガイドを追加します。
(パーツ番号E8356-90028)

オプション0BW 印刷物のアセンブリ・レベル・サービス・マニュアルを追加します。
(パーツ番号E8356-90002)

オプションABD 印刷物のドイツ語ヘルプを追加します。
(パーツ番号E8356-90032)

オプションABE 印刷物のスペイン語ヘルプを追加します。
(パーツ番号E8356-90031)

オプションABF 印刷物のフランス語ヘルプを追加します。
(パーツ番号E8356-90029)

オプションABJ 印刷物の日本語ヘルプを追加します。
(パーツ番号E8356-90032)

サービス・サポート・オプション

オプションW01 3年間のAgilent返送サービス保証を1年間のオンサイト・サービスに変更します(お使いの地域で可能かどうかについては最寄りのAgilent営業所までお問い合わせください)。Agilent連絡先を参照してください。

オプションW31 3年間のAgilent返送サービス保証を3年間のオンサイト・サービスに変更します(お使いの地域で可能かどうかについては、Agilent計測お客様窓口にお問い合わせください)。Agilent連絡先を参照してください。

オプションW32 3年間のAgilent返送商用校正契約を追加します。

オプションW34 3年間のAgilent返送標準準拠校正契約を追加します。

オプションW50 Agilent返送サービス保証を5年間に延長します。

オプションW51 3年間のAgilent返送サービス保証を5年間のオンサイト・サービスに変更します(お使いの地域で可能かどうかについては、Agilent計測お客様窓口にお問い合わせください)。Agilent連絡先を参照してください。

オプションW52 5年間のAgilent返送商用校正契約を追加します。

オプションW54 5年間のAgilent返送標準準拠校正契約を追加します。

テクニカル・サポート

知りたい項目をクリックしてください。

- テストや測定に役立つ情報については、 **www.agilent.com/find/assist**をご覧ください。
- または、Agilent計測お客様窓口にお問い合わせください。

米国:

(電話) 1 800 452 4844

カナダ

(電話) 1 877 894 4414
(fax) 1 (905) 206 4120

欧州:

(電話) (31 20) 547 2000

日本:

(電話) (81) 426 56 7832
(fax) (81) 426 56 7840

中南米:

(電話) (305) 267 4245
(fax) (305) 267 4286

オーストラリア:

(電話) 1 800 629 485
(fax) (61 3) 9272 0749

ニュージーランド

(電話) 0 800 738 378
(fax) 64 4 495 8950

アジア太平洋地域:

(電話) (852) 3197 7777
(fax) (852) 2506 9284

アナライザのトラブルシューティング

いくつかのチェックを実行することで、本器が故障しているかどうかを確認できます。Agilent Technologiesに問い合わせたり、本器を修理のために返送したりする前に、下記のチェックを実行してみてください。

- **基本チェック**
- **誤差項チェック**
- **サービス・ガイドのチェック**

基本チェック

問題が発生したときに実行していた手順を繰り返すことで、問題が解決することがよくあります。Agilent Technologiesに問い合わせたり、本器を修理のために返送したりする前に、下記のチェックを実行してみてください。

1. 電源ソケットは通電していますか。本器のプラグは差し込んでありますか。
2. 本器の電源をオンにしていますか。フロントパネルの電源スイッチと、テスト・ポートを囲むLEDのうち少なくとも1個が緑に点灯しているかどうかを見ます。点灯していれば電源がオンになっていることがわかります。
3. 他の機器、ケーブル、コネクタなどを本器に接続している場合、接続が正しいかどうか、正常に動作しているかどうかを確認します。
4. 問題が発生したときの測定手順を再確認します。設定はすべて正しいですか。
5. 本器が意図したとおりに動作しない場合、Presetキーを押して既知の状態に戻します。
6. 実行しようとしている測定と期待される結果は、本器の仕様と能力の範囲内ですか。
7. 問題がファームウェアにあると思われる場合、トラブルシューティング手順を開始する前に、本器に最新のファームウェアが搭載されているかどうかを調べます。
8. 測定校正が有効かどうか確認します。詳細については、正確な測定校正を参照してください。
9. 必要なテスト機器が揃う場合、PNAサービス・ガイド第2章の「システムのテスト、検査、調整」に記載されているオペレータ・チェックとシステム検査を実行してみます。サービス・ガイドは本器に付属するCD-ROMに収録されています。

誤差項チェック

一定の間隔(毎週、毎月など)で誤差項をプリントしておけば、現在の誤差項をプリントした記録と比較することができます。再現性のある安定したシステムの場合、6ヶ月程度の長期間にわたって再現性のある誤差項が生成されるはずです。微妙な障害やわずかな性能低下が疑われる場合、誤差項の大きさを、同じ測定器と校正キットで以前に生成された値と比べてみます。誤差項をモニタする手順を参照してください。

- 長期的な傾向を見ると、ドリフト、コネクタやケーブルの摩耗、漸進的な劣化などが見られることがあります。この場合、さらに調査して予防保守を行う必要があります。それでもシステムが仕様の範囲内である場合もあります。コネクタの清掃やゲージング、あるいはケーブルの検査だけで問題が解決することもしばしばあります。
- 誤差項が突然変化している場合は、系統誤差に突然の変化があったことを示すので、詳しい調査が必要です。

トラブルシューティングの際には下記のことを考慮します。

- システムのすべての部品は、ケーブルや校正デバイスも含めて、系統誤差に寄与し、誤差項に影響を与えます。
- 誤差項解析が意味を持つためには、コネクタを清掃してゲージングし、仕様の範囲内にあることを確認する必要があります。コネクタの清掃とゲージングについては、PNAサービス・ガイドの第2章を参照してください。
    - 測定校正の後では、なるべくケーブルを曲げたりたわめたりしないようにします。これはケーブルの不安定性による誤差を防ぐためです。
    - 測定校正中には正しい接続方法を使用します。コネクタ・インタフェースには再現性がなければなりません。接続方法についてはPNAサービス・ガイドを参照してください。
- 再現性の程度を確認するため、手順を2回(別々の測定校正を使って)実行してみると役立つことがあります。結果に再現性がない場合、すべてのコネクタとケーブルをチェックします。
- 微妙な性能上の問題を調べるには、誤差項解析を使います。明らかな故障や重大な測定誤差が見られる場合は、PNAサービス・ガイドの第3章「トラブルシューティング」を参照してください。

サービス・ガイドのチェック

問題解決のための具体的なトラブルシューティング手順については、PNAサービス・ガイドを参照してください。サービス・ガイドは、本器に付属するCD-ROMに収録されています。また、当社のWebサイトからもダウンロードできます(**http://www.agilent.com/find/pna**)。

アクセサリ

- 校正キット
- ベリフィケーション・キット
- コネクタ・アダプタおよびアクセサリ・キット
- テスト・ポート・ケーブル

RFテスト・アクセサリの詳細については、Agilent RF and Microwave Test Accessories Catalog(カタログ番号5968-4314EN)またはWebサイトwww.agilent.com/find/mtaを参照してください。

以下のコネクタ・タイプに対してアクセサリが用意されています。

- 50 ΩN型
- 3.5 mm
- 7 mm
- 7-16

測定システムを完成するには、テスト・ポート・ケーブルと、校正キットが必要です。補正済みシステム性能の検証にはベリフィケーション・キットが用いられます。

校正キット

N型、50Ω メカニカル	Agilent 85032F
N型、50Ω RF電子校正(ECal)モジュール	Agilent 85092B
3.5 mm メカニカル	Agilent 85033E
3.5 mm RF電子校正(ECal)モジュール	Agilent 85093B
7mm	Agilent 85031B
7mm RF電子校正(ECal)モジュール	Agilent 85091B
7-16	Agilent 85038A
7-16 RF電子校正(ECal)モジュール	Agilent 85098B

ベリフィケーション・キット

N型	Agilent 85055A
3.5 mm	Agilent 85053B

コネクタ・アダプタおよびアクセサリ・キット

3.5 mm アダプタ・キット	Agilent 11878A
APC-7-N型(メス) 同軸アダプタ	Agilent 11524A
APC-7-N型(オス) 同軸アダプタ	Agilent 11525A
7-16-7-16 アダプタ・キット	Agilent 11906A
7-16-N型 アダプタ・キット	Agilent 11906B
N型、50 Ωアクセサリ・キット	Agilent 11853A
BNC、50 Ωアクセサリ・キット	Agilent 11854A

テスト・ポート・ケーブル

N型(オス-オス)、50 Ω、16インチ	Agilent N3839A
N型(オス-オス)、50 Ω、24インチ	Agilent N6314A
N型(オス-メス)、50 Ω、16インチ	Agilent N6315A

ファームウェアとヘルプのアップグレード

PNAシリーズ・アナライザの最新のファームウェア・アップグレードは、AgilentのWebサイトから入手できます。アップグレードへのアクセスとダウンロードは無料です。インターネットにアクセスできない場合、CD-ROMによるアップグレードもご提供しています。オーダ方法の詳細についてはアナライザ・オプションを参照してください。

アップグレードは**自己解凍型**のInstallShieldファイルです。下記の内容があります。

- ネットワーク・アナライザ測定アプリケーション
- オンライン・ヘルプ
- サービス・ユーティリティ

ファイル容量は全部で約20 MBです。アップグレードをダウンロードするには、PNAシリーズ・アナライザを使う方法と、外部PCを使う方法があります。

PNAシリーズ・アナライザを使用する手順

PNAシリーズ・アナライザ、ウェブ・ブラウザ、内蔵LAN(Local Area Network)を使ってAgilentウェブサイトに接続します。

1. PNAウェブ・ページ(**http://www.agilent.com/find/PNA**)に接続します。
2. **Support**をクリックします。
3. **Download Firmware**をクリックします。
4. 以下のどちらかを選択します。

   - 現在の場所(サーバ上)からプログラムを実行[推奨]
   - プログラムをディスク(アナライザのハード・ドライブ)に保存

アップグレードをディスクに保存した場合、アナライザ上でファイルを開き、指示に従ってください。ファイルは自己解凍型です。

外部PCを使用する手順

アナライザから直接インターネットにアクセスできない場合、インターネットにアクセスできる外部PCを使ってAgilentウェブサイトからファイルをダウンロードすることもできます。その後、ファイルをPCからLAN経由でアナライザに転送します。

1. PNAウェブ・ページ(**http://www.agilent.com/find/PNA**)に接続します。
2. **Support**をクリックします。
3. **Download Firmware**をクリックします。
4. プログラムをディスク(PCのハード・ドライブ)に保存します。
5. PCからLAN経由でアナライザにファイルを転送します。

   - プログラムを現在の場所(外部PC上)で実行します。
   - プログラムをディスク(アナライザのハード・ドライブ)に保存します。

警告: アップグレード・ファイルを保存することはできますが、ネットワーク・アナライザ・アプリケーションを外部PCにインストールすることは避けてください。システム設定が変更され、システムに障害が発生するおそれがあります。

---

10 MHz基準 33
1つを選択 168
1ポート(反射) 168
1個のオブジェクトに対するハンドル 300
2ポート・デバイスの反射確度 109
2ポート校正のアイソレーション部分 180
AGCループを持つ増幅器のテスト 235
AM-PM変換 223
AM入力、外部コネクタ 32
ASCIIファイル・タイプ 210
C++とCOM 312
CDMA基地局増幅器 235
COMイベント 309
COMオブジェクト・モデル 319
COMデータ・タイプ 307
COMの基本 298
COMの構成 304
COM対SCPI 539
CW時間 82
CW時間掃引 82
ECal 159
　手順 168
ECal 1ポート 168
ECalフル2ポート 168
F1、F2、F3、F4キー 26
Formatキー 28
GPIBコネクタ 34
GPIBの基本 503
Helpキー 21
IF帯域幅、トレース・ノイズの低減 124
IF帯域幅、独立した 82
LANコネクタ 34
Macro/Localキー 29
Math/Memoryキー 28
Measure Setupsキー 22
Measureキー 28
Menu/Dialogキー 21
OKキー 21
Presetキー 29
Printキー 29
Recallキー 29
RESTART 90
Restore Up/Downキー 29
Save as 210
Saveキー 29
Scaleキー 28
SCPIによるデータの取得 512
SCPIの構文 508
SCPIの例 548
SOLT校正 168
SRQ 522
SWR 69
Sパラメータ 69
Traceキー 22
TRL校正 168
Unitsキー 25
USBコネクタ 29
Utilityキー 29
VGA出力コネクタ 35
Windowキー 22
X軸スケール 60
Y軸スケーラ 60
アイソレーション 151
アイソレーション(クロストーク) 127
アイソレーション、2ポート校正 180
アイソレーション、リバース 261
　確度に関する考慮事項 261
　説明 261
　測定する理由 261
アイソレーション誤差 151
アクセサリ、影響 107
アクセサリの影響 107
アクセス、フロントパネル・インタフェース 38
アクティブ・エントリ・キー 23, 24
アクティブ・チャネル 20
アダプタ 180
　スルー 180
　パラレル-セントロニクス 34
　校正に使用 180
　等価交換法 180
アダプタの使用法 180
アッテネータ、レシーバ 69
アナライザの動作モデル 36
アナライザ画面のカスタマイズ 56
　ステータス・バー 56
　タイトル・バー 56
　ツールバー 56
　データ・トレース 56
　テーブル 56
　メモリ・トレース 56
　測定画面 56
アパーチャ、群遅延 228
アパーチャとは何か 228
アベレージング、掃引 124
アベレージング・キー 20
アンダーサンプリングされたデータ 131
イーサネット 34
インタフェース、フロントパネル 38
インピーダンス 69
ウィザード 168
ウィザード、校正 20
ウィンドウ 74
ウィンドウ、動作モデル 36
ウィンドウ、配列機能 74
　2分割 74
　3分割 74
　4分割 74
　オーバレイ 74
　ソート・アルゴリズム 74
　配置キー 22
ウィンドウが重なり合う 22
ウィンドウの配置 74
ウィンドウの表示 22

ウィンドウの分割.....22
ウィンドウを積み重ねる.....22
エクスカーション.....190
オートセーブ.....210
オープンまたはショート.....168
オブジェクトの階層.....300
オプション010、時間領域.....265
オプション015.....16
　テスト・ポート・ジャンパ、ポート1.....16
　テスト・ポート・ジャンパ、ポート2.....18
　レシーバ減衰.....96
オン/オフ.....168
カプラのロールオフ、補正.....107
リニア位相.....259
キー、チャネル設定.....20
キット.....159
キャンセル・キー.....21
クラス、校正標準.....159
グリッド、画面.....23, 24
グローバル・トリガ・スコープ.....90
クロストーク.....127
クロストーク、レシーバ.....127
クロストークの改善.....127
ゲーティング.....280
　伝送応答.....107
　反射応答.....107
コネクタ.....29
　10 MHz基準.....33
　GPIB.....34
　LAN.....34
　USB.....29
　VGA出力.....35
　シリアル・インタフェース・ポート(COM1).....34
　バイアス入力.....31
　パラレル・インタフェース・ポート.....34
　外部AM入力.....32
　外部ディテクタ入力.....32
　注意.....131
コネクタの再現性.....151
コマンド・キー.....21
コマンドの同期.....516
サービス要求.....522
サポート.....679
シーケンス、基本測定.....38
しきい値.....190
シリアル・インタフェース・ポート(COM1)コネクタ.....34
スイッチの再現性.....151
スケール.....60
スケール・ファクタ.....60
　オート.....60
　すべてオート.....60
　基準レベル.....60
　基準位置.....60
スコープ、トリガ.....90
　グローバル.....90
　チャネル.....90
スタート.....190
スティミュラス、デフォルト.....54
ステータス・バー.....56
ステータス・レジスタ.....522
ステート、トリガ.....90
　グループ.....90
　シングル.....90
　ポイント.....90
　ホールド.....90
　連続.....90
ステップ掃引.....116
ストップ.....190
スパン.....190
スムージング、トレース.....124
スルー・レスポンス.....168
スルー・レスポンスとアイソレーション.....168
スルー・レスポンス校正.....168
スルーの特性評価.....180
スループット、上げる.....140
スロープ、パワー.....96
セグメント.....199
セグメント、リミット.....199
セグメント・テーブル.....56
セグメント掃引.....82
セグメント掃引の使用.....140
セット内部の配置.....140
ソース.....151
ソース、トリガ.....90
　外部.....90
　手動.....90
　内部.....90
ターゲット.....190
ダイアログ・ボックス、クローズ.....21
タイトル.....56
タイトル・バー.....56
ダイナミック・レンジ.....112
　IF帯域幅を狭める.....112
　ノイズ・フロアを下げる.....112
　掃引アベレージングの追加.....112
タイプ.....220
タイプ、マーカ.....190
　ノーマル.....190
　固定.....190
チャネル.....36
チャネル、複数のトレースと.....74
チャネルのトリガ・ソース.....90
チャネルの動作モデル.....36
チャネル設定.....20
ツアー.....16
　フロントパネル.....16
　リアパネル.....31
ツールバー.....56
　アクティブ入力.....56
　スティミュラス.....56
　すべてオフ.....56
　ツアー.....23, 24
　トレース・ステータス.....56
　マーカ.....56

掃引制御 56
測定 56
ディスク・ドライブ、フロッピー 26
ディスプレイ 56
ディスプレイ・キー 22
タイトル 56
トレース・ステータス 56
マーカ読み値 56
リミット・テスト結果 56
リミット・ライン 56
ディスプレイ仕様
周波数スティミュラス 56
ディテクタ入力、外部 32
データ、アンダーサンプリングされた 131
データ・トレース表示 56
データ・ポイント 129
データの表示 225
データの変更 168
データ記憶 26
データ転送速度 146
データ表示 60
テーブル 56
セグメント 56
マーカ 56
リミット 56
テクニカル・サポート 679
テスト 199
テスト・ポート 69
デバイス接続 131
デバイス遅延 116
デフォルト 54
デフォルト条件 54
デルタ・マーカ 190
トリガ 90
トリガ・スコープ 90
グローバル 90
チャネル 90
トリガ・ステート 90
グループ 90
シングル 90
ポイント 90
ホールド 90
連続 90
トリガ・ソース 90
デフォルト 54
外部 90
手動 90
内部 90
ドリフト 151
温度 131
誤差 151
周波数 131
トレース 74
トレース、動作モデル 36
トレース・ステータス 56
トレース・ノイズ、低減 124
IF帯域幅 124
トレース・スムージング 124
掃引アベレージ 124
トレース・ノイズの低減 124
IF帯域幅 124
トレース・スムージング 124
掃引アベレージ 124
トレースの作成 22
トレース演算 203
トレース設定キー 28
トレース統計 203
なし 168
なぜ難しいか 235
なぜ必要なのか 149
ナビゲーション・キー 26
ノイズ、トレース 124
ノイズ・フロア 127
ノイズの低減 134
ノーマル・マーカ・タイプ 190
ノブ、フロントパネル 25
ノンインサータブル・デバイス 180
ノンインサータブル・デバイスの校正 180
ハード・ディスクへのファイルの保存 210
バイアス 31
バイナリ・ファイル・タイプ 210
ハイバネーション 54
ハイパワー・コンポーネント測定 235
AGCループを持つ増幅器のテスト 235
CDMA基地局増幅器 235
なぜ難しいか 235
外部レベリング 235
確度に関する考慮事項 235
構成選択ガイド 235
測定校正 235
パラメータ、増幅器 223
AM-PM変換 223
リターン・ロス 223
リニア位相からの偏移 223
リバース・アイソレーション 223
群遅延 223
複素インピーダンス 223
利得 223
利得ドリフト対時間 223
利得フラットネス 223
利得圧縮 223
パラメータ、測定 69
Sパラメータ 69
任意比 69
比でないパワー 69
パラレル・インタフェース・ポート・コネクタ 34
パワー 82
スロープ 96
ポート間の結合 96
レシーバ減衰 96
レベル 96
レンジ 96
減衰 96
最適 96
絶対出力 233

掃引 96
独立したセグメント 82
比でない 69
ピーク 190
ピーク、マーカ 190
ファイルのストア 26
ファイルのタイプ 210
ファイルのリコール 210
ファイルの記憶 26
ファイルの保存 210
ファイルの保存とリコール 210
ファイルへのプリント 210
ファイル名の編集 210
フォーマット 60
SWR 60
スミス・チャート 60
リニア振幅 60
位相 60
虚数 60
極座標 60
群遅延 60
実数 60
対数振幅 60
直交座標 60
フラットネスと利得、小信号 263
確度に関する考慮事項 263
説明 263
測定する理由 263
プリセット 54
プリセット条件 54
グローバル表示 54
セグメント 54
データ表示 54
トリガ 54
パワー 54
マーカ 54
リミット・ライン 54
応答 54
校正 54
時間領域 54
周波数 54
掃引 54
測定パラメータ 54
フル2ポート校正 168
フルSOLT 2ポート 168
フルTRL 2ポート 168
フロー・グラフ 69
プローブ電源 27
フロッピー・ディスク・ドライブ 26
ファイルの保存 210
プロパティ 168
プロパティ、校正 168
フロントパネル・ツアー 16
チャネル設定キー 20
トレース設定キー 28
入力キー 25
表示画面 23, 24
ヘルプ、使用法 41
ウィンドウのサイズや位置の変更 44
ヘルプ・ウィンドウのクローズ 45
現在のトピックのプリント 44
ヘルプ・ウィンドウ、クローズ 45
ヘルプ・ウィンドウのクローズ 45
ヘルプ・ウィンドウのサイズや位置の変更 .. 44
ポイント、数 129
ポイント数 129
ポート 27
A 16
B 18
R1 16
R2 18
テスト 27
レシーバ直接アクセス 16
ポート1 16
ポート2 18
ポート延長 119
ホールド・モード 96
マーカ 190
マーカ・キー 28
マーカ・タイプ 190
テーブル 56
デフォルト 54
デルタ 190
ノーマル 190
フォーマット 190
基準 190
固定 190
マーカ機能 190
スタート 190
ストップ 190
スパン 190
基準レベル 190
遅延 190
中心 190
マーカ検索 190
ターゲット 190
タイプ 190
ピーク 190
実行 190
帯域幅 190
表示 190
離散 190
領域 190
マーカ読み値 56
マウス・インタフェース 38
メニュー 159
メモリ・トレース表示 56
モデルの境界 36
モニタ、外部 35
ユーザ・インタフェース 38
ユーザ・スパン 190
ユーザ・レンジ 203
コネクタの再現性 151
スイッチの再現性 151
本器のノイズ 151
ランダム 151

ランダム測定誤差.....151
リアパネル・ツアー.....31
リコール.....168
リターン・ロス.....223
リニア位相からの偏移.....223
リニア位相シフト、補正.....119
リニア周波数.....82
リバース・アイソレーション.....261
リミット.....199
リミット・テーブル.....56
リミット・テーブル・キー.....28
リミット・テスト結果.....56
リミット・ライン.....56
リミット・ライン、デフォルト.....54
レシーバ.....16
レシーバ減衰.....96
レシーバ入力.....16
レスポンス校正.....168
レンジ.....269
レンジ、ダイナミック.....112
レンジ、パワー.....96
レンジ、周波数.....67
ロールオフ、カプラ.....107
圧縮、利得.....255
安定度、測定.....131
位相、リニア.....259
位相オフセット.....119
位相シフト.....119
位相シフト成分.....228
位相測定.....220
位相測定確度.....119
延長、ポート.....180
演算機能、データ/メモリ.....203
温度ドリフト.....131
画面、表示.....23, 24
画面の配置.....22
外部AM入力.....32
外部CD-RWドライブへのファイルの保存..210
外部ディテクタ入力.....32
外部トリガ.....90
外部レベリング.....235
概要.....149
確度.....168
確度、位相測定.....119
確度、改善.....107
確度、低損失2ポート・デバイスの.....109
確度、反射.....109
確度に関する考慮事項.....259
基準チャネル・ジャンパ.....16
基準マーカ.....190
基準レベル.....60
基準位置.....60
基準面.....180
機能、マーカ.....190
境界、測定.....74
群遅延.....223
係数、伝送.....69
係数、反射.....69
系統.....151
系統測定誤差.....151
検索、マーカ.....190
減衰.....96
現在のヘルプ・トピックのプリント.....44
呼出しの高速化.....140
固定マーカ・タイプ.....190
誤差、ランダム.....151
誤差、系統.....151
誤差、測定.....151
誤差補正.....151
交互掃引、クロストーク.....127
校正.....168
校正キットの変更.....159
校正確度の検証.....186
校正方法の確度.....168
構成.....168
構成可能テスト・セット、オプション015.235
構成選択ガイド.....235
時間、待ち.....82
時間領域.....265
時間領域の窓関数.....277
実行.....190
収集.....303
周波数、CW.....67
周波数スティミュラス.....56
周波数ドリフト.....131
周波数ポイント間隔.....119
周波数レンジ.....67
周波数基準、10 MHz.....131
周波数分解能.....67
終端.....109
出力、10 MHz基準.....33
出力パワー、絶対.....233
小信号利得とフラットネス.....263
条件設定A.....74
条件設定B.....74
条件設定C.....74
条件設定D.....74
信号源整合.....151
信号源整合誤差.....151
推奨する条件.....149
正確な.....180
正確な測定校正.....180
製品サポート.....679
切替えを自動化.....140
接続、フロントパネル.....69
設定.....134
設定、測定.....74
説明.....259
絶対出力パワー.....233
選択と実行.....168
選択肢の概要.....168
選択的にトリガ.....140
漸進的な劣化.....186
掃引.....134
掃引アベレージ.....20
掃引タイプ.....82

掃引速度 134
窓関数 277
送信 69
増幅器パラメータ 223
測定、アベレージ 20
測定、ハイパワー・コンポーネント 235
測定、位相 220
測定、複数の 140
測定シーケンス 38
測定する理由 228
測定の安定度、上げる 131
測定の呼出しの高速化 140
測定パラメータ 69
測定画面 56
測定基準面 180
測定境界 74
測定誤差 151
測定校正 127
測定校正、概要 149
測定校正、通常 168
測定校正方法 168
測定条件設定、構成済み 74
測定前の準備 15
速度、データ転送 146
速度、掃引 134
損失 69
帯域幅 190
待ち行列 512
待ち時間 82
遅延 190
遅延、群 228
遅延、電気 69
中心 190
直接アクセス、ポート1 16
直接アクセス、ポート2 18
低損失2ポート・デバイス 109
定義 223
定義の変更 159
転送レート、フロッピー・ドライブ 26
転送速度、データ 146
伝送クラス、校正標準 159
伝送トラッキング 151
伝送トラッキング誤差 151
電気遅延 119
電気遅延、使用 259
電気遅延の使用 259
電気長、追加 116
電気長の長いデバイス 116
電気長の長いデバイスの測定 116
電源、プローブ 27
電源スイッチ 27
電源投入時状態 54
電子校正(ECal) 159
統計 203
動作モデル 36
入力、10 MHz基準 33
入力、レシーバ 16
入力キー 25
任意比 69
反射 69
反射クラス、校正標準 159
反射トラッキング 151
反射トラッキング誤差 151
反射確度 109
比、定在波 69
比、任意 69
比でないパワー 69
標準 159
標準、校正 159
標準デバイス 159
表示、配置 74
表示の配列 74
表示フォーマット 60
表示画面 23, 24
負荷整合 151
負荷整合誤差 151
複数のチャネル、トレース、ウィンドウ 74
複数の測定 140
複素インピーダンス 223
分解能 269
補間、誤差補正に対する影響 186
補間されたデータ・ポイント 190
補助出力 16
方向性 151
方向性誤差 151
方法 168
本器のノイズ 151
未補正 186
有効性 186
利得 223
利得とフラットネス、小信号 263
利得ドリフト対時間 223
利得ドリフト対時間、定義 223
利得の定義 223
利得フラットネス 223
利得フラットネス、定義 223
利得圧縮 223
離散マーカ 190
領域 190
漏れ、クロストーク 127