NI-VNA

2025-06-18

Contents

NI Vector Network Analyzers Help . . . 21
Overview of Vector Network Analyzers . . . 21
Network Analysis Principles . . . 21
Vector Network Analyzer Measurements . . . 23
Time Domain Measurements . . . 24
Understanding How Resolution and Aliasing Impact Time-Domain Measurements . . . 26
Configuring Your Time-Domain Measurement . . . 27
Choosing Between Lowpass and Bandpass . . . 28
Windowing . . . 29
Gating . . . 31
Vector Network Analyzer Architectures . . . 31
VNA Measurement Best Practices . . . 34
NI-VNA State Diagram . . . 36
Calibration . . . 37
Factory Calibration . . . 38
User Calibration . . . 38
RF Calibration . . . 41
Automatic Calibration (AutoCal) . . . 43
Short-Open-Load-Through (SOLT) Calibration . . . 44
Short-Open-Load-Reciprocal (SOLR) Calibration . . . 45
Through-Reflect-Line (TRL) Calibration . . . 46
Power Calibration . . . 48
Source Power Calibration . . . 48
Test and Reference Receiver Calibration . . . 49
Error and Uncertainty . . . 49
Reference Plane . . . 51
Configuring Hardware in MAX . . . 52
Renaming a Hardware Module in MAX . . . 53
Activating Time Domain in MAX . . . 54
NI-VNA Examples . . . 54
Devices . . . 55

NI PXIe-5630 55
NI 5630 Block Diagram and Theory of Operation 56
PXI Trigger Lines 58
System Reference Clock 59
PFI Lines 59
NI 5630 Front Panel and Connectors 60
Hardware Integration and Maintenance 61
PXI Express Modules 61
Maintaining PXI Express Hardware 62
Connector Care 63
Uninstalling the Hardware 66
NI 5630 High-Sensitivity Mode 66
Accessories 68
NI PXIe-5632 68
NI 5632 Block Diagram and Theory of Operation 69
PXI Trigger Lines 72
System Reference Clock 72
PFI Lines 73
NI 5632 Front Panel and Connectors 73
Hardware Integration and Maintenance 75
PXI Express Modules 75
Maintaining PXI Express Hardware 76
Connector Care 77
Uninstalling the Hardware 80
NI 5632 Adapter Removal Calibration 80
NI 5632 De-Embedding 82
NI 5632 Power Sweeps 83
NI 5632 Segmented Sweep 83
NI 5632 Individual Control of Source and Receiver Frequencies 84
NI 5632 Raw Receiver Measurements 85
Accessories 85
NI-VNA Soft Front Panel 86
NI-VNA SFP Menu 87
NI-VNA SFP Toolbar 89
NI-VNA SFP Softkey Menu 90
NI-VNA SFP Display Window 92

Performing an Automatic SOLT Calibration (AutoCal) Using the NI-VNA SFP 92
Performing an Automatic SOLT Calibration (AutoCal) of the NI 5630 Using the NI-VNA SFP 92
Performing an Automatic SOLT Calibration (AutoCal) of the NI 5632 Using the NI-VNA SFP 95
Performing a Manual SOLT Calibration Using the NI-VNA SFP 97
Performing a Manual SOLT Calibration of the NI 5630 Using the NI-VNA SFP . . 98
Performing a Manual SOLT Calibration of the NI 5632 Using the NI-VNA SFP . . 100
Programming Reference 103
Using Properties and Attributes with NI-VNA 103
Using the Standard Functionality for error in Parameters 106
Using the Standard Functionality for error out Parameters 107
Using the Sweep Settings Configuration File 108
Specifying Sweep Settings 109
Specifying Setting Parameters 109
Segment Types 116
Possible Errors 119
Performing a User Calibration Using the NI-VNA API 120
Performing a Manual SOL (1-Port, 1-Path) Calibration 121
Performing a Manual SOLT (2-Port, 1-Path) Calibration 122
Performing a Manual TRL (2-Port, 2-Path) Calibration 123
Performing a Reflection Response Calibration 125
Performing a Transmission Response Calibration 125
Performing an Auto Calibration Using the NI-VNA API 126
Performing an Auto Calibration with a True Through Using the NI-VNA API . . . 127
Reference 128
NI-VNA LabVIEW Reference 128
NI-VNA VIs 128
niVNA Initialize 130
niVNA Close 131
niVNA Initialize With Options 133
niVNA Property Node 135
Configuration 137

niVNA Configure Connector 138
niVNA Perform Auto Reference Plane Extension 140
Sweep 142
niVNA Configure Sweep 143
niVNA Configure Frequency 145
niVNA Configure Power Start Stop 149
niVNA Configure Frequency Offset Equation 152
niVNA Load Sweep Settings Data From File 154
Trigger 156
niVNA Configure Trigger 157
niVNA Send Software Edge Trigger 166
niVNA Export Signal 168
Time Domain 170
niVNA Configure Time Domain 171
niVNA Configure Time Domain Gate 179
Measurement 187
niVNA Initiate 189
niVNA Fetch 191
niVNA Fetch Time Domain 201
niVNA Abort 207
niVNA Commit 209
niVNA Wait For Sweep Done 211
niVNA Is Sweep Done 213
niVNA Fetch Receiver Measurement 215
Calibration 224
niVNA Init User Cal 226
niVNA Perform Calibration Action 228
niVNA Abort Calibration Action 234
niVNA Close User Cal 236
niVNA Is Calibration Action Done 238
niVNA Wait For Calibration Action Done 240
niVNA Get Calibration State 242
niVNA Apply Adapter Removal Calibration 244
niVNA Load Calibration 246
niVNA Save Calibration 248
niVNA Load User Defined Manual Calibration Kit Data From

File 250
AutoCal 252
niVNA Load AutoCal Characterization Data From File . . 253
niVNA Query AutoCal Box Info 255
Utility 257
niVNA Reset 258
niVNA Reset With Defaults 259
niVNA Disable 261
niVNA Self Test 263
niVNA Revision Query 264
niVNA Error Message 266
niVNA Activate Feature 268
niVNA Query Feature Activation Status 270
niVNA Properties 272
Active Channel 288
Trigger 289
Start Trigger Type 289
Start Trigger Source 290
Start Trigger Edge 292
Step Trigger Type 293
Step Trigger Source 294
Step Trigger Edge 296
Test Signal 297
IF Bandwidth 297
Output Power 298
Reference Frequency 299
Source 299
Reference Frequency 301
Ref Plane Extension 302
Distance 302
Time 303
Dielectric Type 304
Dielectric User Value 305
Insertion Loss 306
Sweep 307

Power Sweep Frequency 307
Sweep Mode 308
Sweep Number of Points 309
Sweep Step Delay 310
Format Phase Offset 311
Sweep Frequency Start 312
Sweep Frequency Stop 313
Sweep Power Start 314
Sweep Power Stop 315
S11 Enabled 316
S21 Enabled 317
Driver State 318
S12 Enabled 320
S22 Enabled 321
Averaging And Smoothing 322
Averaging Count 322
Averaging Type 323
Smoothing Aperture 324
Individual Sources/Receiver 325
Source Driving State 325
Source Frequency Multiplier 326
Source Frequency Offset 327
Receiver Frequency Multiplier 328
Receiver Frequency Offset 329
Calibration Settings 330
RF Calibration Status 330
Source Power Calibration Status 331
Test Receiver Calibration Status 332
Reference Receiver Calibration Status 333
Source Calibration Power Offset 334
De-embedding Enabled 335
Correction Enabled 336
Calibration State 338
RF Calibration Type 339
Allow Changes While Calibrated 342
High Sensitivity Enabled 342

Connectors 343
Connector Gender 343
Connector Type 344
Calibration Thru Line Type 346
Calibration Thru Line Length 347
Calibration Thru Line Impedance 348
Calibration Thru Line Loss 349
Calibration Thru Reference Frequency 350
AutoCal 350
AutoCal True Thru Length 351
AutoCal True Thru Impedance 352
AutoCal True Thru Line Loss 353
AutoCal True Thru Reference Frequency 354
AutoCal Box Orientation 355
AutoCal Box Communication Port 356
AutoCal Assurance Enabled 357
TRL Calibration 358
Reference Plane Location 358
Number of Bands 359
Standard Type 360
Standard Length 361
Standard Line Loss 362
Standard Line Loss Frequency 364
List of Standards 365
Band Breakpoint Type 366
Band Breakpoint Frequency 367
Power Meter Name 368
Calibration File Intended For SFP 369
External Calibration Information 370
External Calibration Year 370
External Calibration Month 371
External Calibration Day 371
External Calibration Hour 372
External Calibration Minute 373
External Calibration Temperature 374
External Calibration Interval 375

User Calibration Information 375
User Calibration Year 376
User Calibration Month 377
User Calibration Day 378
User Calibration Hour 379
User Calibration Minute 380
User Calibration Temperature 381
Time Domain 381
Transform Type 381
Trip Type 383
Impulse Width at 3 dB 384
Alias Free Range 385
Range 386
Units 386
Start Point 387
Stop Point 388
Window 389
Shape 389
Kaiser-Bessel Beta 390
Dolph-Chebyshev Gamma 391
Gate 392
Shape 392
Enable 394
Inclusive 395
Range 395
Start Point 396
Stop Point 397
Kaiser-Bessel Beta 398
Dolph-Chebyshev Gamma 399
DC Term Extrapolation 399
Type 400
Method 401
Coefficient 402
User Value 403
Device Information 404
PXI Chassis 404

PXI Slot 404
Device Temperature 405
Device Serial Number 406
Number of Ports 407
Inherent IVI Properties 408
User Options 408
Cache 408
Interchange Check 409
Driver Setup 410
Query Instrument Status 411
Range Check 412
Record Value Coercions 413
Simulate 414
Driver Identification 414
Description 415
Driver Prefix 415
Driver Vendor 416
Revision 417
Class Specification Major Version 418
Class Specification Minor Version 419
Driver Capabilities 419
Class Group Capabilities 420
Supported Instrument Models 421
Instrument Identification 421
Manufacturer 422
Model 422
Firmware Revision 423
Advanced Session Information 424
Logical Name 424
I/O Resource Descriptor 425
Events 426
Ready For Start Event 426
Output Terminal 426
Active Level 428
Ready For Step Event 428
Output Terminal 429

Active Level 430
End Of Step Event 431
Output Terminal 431
Edge 433
End Of Sweep Event 434
Output Terminal 434
Edge 435
NI-VNA C Reference Help 436
NI-VNA Functions 436
niVNA_init 440
niVNA_InitWithOptions 442
Configure 445
Sweep 447
niVNA_ConfigureFrequencyCenterSpan 447
niVNA_ConfigureFrequencyStartStop 449
niVNA_ConfigureFrequencyOffsetEquation 450
niVNA_ConfigurePowerStartStop 452
niVNA_ConfigureSweep 453
niVNA_LoadSweepSettingsDataFromFile 455
Time Domain 456
niVNA_ConfigureTimeDomainCenterSpanTime . . . 457
niVNA_ConfigureTimeDomainCenterSpanDistance 458
niVNA_ConfigureTimeDomainStartStopTime 459
niVNA_ConfigureTimeDomainStartStopDistance . . 461
Gate 462
niVNA_ConfigureTimeDomainGateCenterSpanTime 462
niVNA_ConfigureTimeDomainGateCenterSpanDistance 464
niVNA_ConfigureTimeDomainGateStartStopTime 465
niVNA_ConfigureTimeDomainGateStartStopDistance 467
Trigger 468
Start Trigger 469

niVNA_ConfigureDigitalEdgeStartTrigger 469
niVNA_ConfigureSoftwareEdgeStartTrigger .. 471
niVNA_DisableStartTrigger 473
Step Trigger 474
niVNA_ConfigureDigitalEdgeStepTrigger..... 474
niVNA_DisableStepTrigger 476
niVNA_SendSoftwareEdgeTrigger..... 477
niVNA_ExportSignal..... 478
Set/Get Attribute..... 481
Set Attribute..... 481
niVNA_SetAttributeViInt32 482
niVNA_SetAttributeViInt64 483
niVNA_SetAttributeViReal64..... 485
niVNA_SetAttributeViString 486
niVNA_SetAttributeViBoolean 488
niVNA_SetAttributeViSession..... 489
Get Attribute 490
niVNA_GetAttributeViInt32..... 491
niVNA_GetAttributeViInt64..... 492
niVNA_GetAttributeViReal64 494
niVNA_GetAttributeViString..... 495
niVNA_GetAttributeViBoolean..... 497
niVNA_GetAttributeViSession 498
niVNA_InvalidateAllAttributes..... 500
Connector Configuration 501
niVNA_ConfigureConnector..... 501
niVNA_PerformAutoReferencePlaneExtension 503
Measurement 504
Sweep 505
niVNA_Initiate 506
niVNA_Commit 507
niVNA_Abort 508
niVNA_IsSweepDone 509
niVNA_WaitForSweepDone 511
niVNA_FetchSweepMeasurementF64 512
niVNA_FetchSweepMeasurementComplex 515

niVNA_FetchSweepMeasurementIntoFile 518
niVNA_FetchTimeDomainTransformF64 520
niVNA_FetchTimeDomainGateShapeF64 523
niVNA_FetchReceiverMeasurementF64 526
niVNA_FetchReceiverMeasurementComplex 530
Calibration 533
niVNA_InitUserCal 534
niVNA_PerformCalibrationAction 536
niVNA_PerformTRLCalibrationAction 540
niVNA_ApplyAdapterRemovalCalibration 541
niVNA_GetCalibrationState 543
niVNA_IsCalibrationActionDone 545
niVNA_WaitForCalibrationActionDone 546
niVNA_AbortCalibrationAction 547
niVNA_CloseUserCal 548
niVNA_LoadCalibrationAndConfigurationDataFromFile 550
niVNA_SaveCalibrationAndConfigurationDataToFile . . . 551
niVNA_LoadUserDefinedManualCalibrationKitDataFromFile 553
AutoCal 554
niVNA_QueryAutoCalBoxInfo 554
niVNA_LoadAutoCalCharacterizationDataFromFile 556
Utility 557
niVNA_self_test 558
niVNA_reset 559
niVNA_ResetWithDefaults 560
niVNA_Disable 561
niVNA_revision_query 562
Feature Activation 564
niVNA_ActivateFeature 564
niVNA_QueryFeatureActivationStatus 565
Error Info 567
niVNA_error_query 567

niVNA_error_message 569
niVNA_GetError 570
niVNA_ClearError 572
niVNA_GetNextCoercionRecord 573
Interchangeability Info 574
niVNA_GetNextInterchangeWarning 575
niVNA_ResetInterchangeCheck 577
niVNA_ClearInterchangeWarnings 578
Locking 579
niVNA_LockSession 579
niVNA_UnlockSession 581
niVNA_close 584
Attributes 584
Events 584
Ready For Start Event 584
NIVNA_ATTR_EXPORTED_READY_FOR_START_EVENT_OUTPUT_T
585
NIVNA_ATTR_READY_FOR_START_EVENT_LEVEL . . 586
Ready For Step Event 586
NIVNA_ATTR_EXPORTED_READY_FOR_STEP_EVENT_OUTPUT_TE
586
NIVNA_ATTR_READY_FOR_STEP_EVENT_LEVEL . . . 588
End Of Step Event 588
NIVNA_ATTR_EXPORTED_END_OF_STEP_EVENT_OUTPUT_TERM
588
NIVNA_ATTR_END_OF_STEP_EVENT_EDGE 589
End Of Sweep Event 590
NIVNA_ATTR_EXPORTED_END_OF_SWEEP_EVENT_OUTPUT_TER
590
NIVNA_ATTR_END_OF_SWEEP_EVENT_EDGE 591
Trigger 592
NIVNA_ATTR_START_TRIGGER_TYPE 592
NIVNA_ATTR_START_TRIGGER_SOURCE 593
NIVNA_ATTR_START_TRIGGER_EDGE 594
NIVNA_ATTR_STEP_TRIGGER_TYPE 594
NIVNA_ATTR_STEP_TRIGGER_SOURCE 595

NIVNA_ATTR_STEP_TRIGGER_EDGE 596
Averaging and Smoothing 597
NIVNA_ATTR_AVERAGING_COUNT 597
NIVNA_ATTR_AVERAGING_TYPE 597
NIVNA_ATTR_SMOOTHING_APERTURE. 598
Test Signal 599
NIVNA_ATTR_IF_BANDWIDTH 599
NIVNA_ATTR_OUTPUT_POWER 599
Frequency Reference 600
NIVNA_ATTR_REFERENCE_FREQUENCY_SOURCE 600
NIVNA_ATTR_REFERENCE_FREQUENCY 601
Reference Plane Extension 601
NIVNA_ATTR_REFPLANE_EXTENSION_DISTANCE. . 602
NIVNA_ATTR_REFPLANE_EXTENSION_TIME 602
NIVNA_ATTR_REFPLANE_EXTENSION_DIELECTRIC_TYPE 603
NIVNA_ATTR_REFPLANE_EXTENSION_DIELECTRIC_USER_VALUE 604
NIVNA_ATTR_REFPLANE_EXTENSION_INSERTION_LOSS 604
Sweep 604
NIVNA_ATTR_SWEEP_MODE 605
NIVNA_ATTR_SWEEP_NUMBER_OF_POINTS. 606
NIVNA_ATTR_SWEEP_FREQUENCY_START 606
NIVNA_ATTR_SWEEP_FREQUENCY_STOP 607
NIVNA_ATTR_POWER_SWEEP_FREQUENCY 608
NIVNA_ATTR_SWEEP_POWER_START 608
NIVNA_ATTR_SWEEP_POWER_STOP. 609
NIVNA_ATTR_SOURCE_CALIBRATION_POWER_OFFSET. 609
NIVNA_ATTR_SWEEP_STEP_DELAY 610
NIVNA_ATTR_FORMAT_PHASE_OFFSET 611
NIVNA_ATTR_S11_ENABLED 611
NIVNA_ATTR_S21_ENABLED 612
NIVNA_ATTR_S12_ENABLED 612

NIVNA_ATTR_S22_ENABLED 613
NIVNA_ATTR_DRIVER_STATE 614
Calibration Settings 615
NIVNA_ATTR_CORRECTION_ENABLED 615
NIVNA_ATTR_CALIBRATION_CURRENT_STATE 616
NIVNA_ATTR_RF_CALIBRATION_TYPE 617
NIVNA_ATTR_ALLOW_CHANGES_WHILE_CALIBRATED . . 619
NIVNA_ATTR_CALIBRATION_HIGH_SENSITIVITY_ENABLED . . 620
NIVNA_ATTR_POWER_METER_NAME 620
NIVNA_ATTR_POWER_COMPENSATION_SLOPE 621
NIVNA_ATTR_DE_EMBEDDING_ENABLED 621
NIVNA_ATTR_RF_CALIBRATION_STATUS 622
NIVNA_ATTR_SOURCE_POWER_CALIBRATION_STATUS 623
NIVNA_ATTR_TEST_RECEIVER_CALIBRATION_STATUS . . 623
NIVNA_ATTR_REFERENCE_RECEIVER_CALIBRATION_STATUS 624
NIVNA_ATTR_CALIBRATION_THRU_LINE_TYPE 625
NIVNA_ATTR_CALIBRATION_THRU_LINE_LENGTH 625
NIVNA_ATTR_CALIBRATION_THRU_LINE_IMPEDANCE . . 626
NIVNA_ATTR_CALIBRATION_THRU_LINE_LOSS 626
NIVNA_ATTR_CALIBRATION_THRU_REF_FREQUENCY . . 627
Connectors 628
NIVNA_ATTR_CONNECTOR_GENDER 628
NIVNA_ATTR_CONNECTOR_TYPE 628
AutoCal 629
NIVNA_ATTR_AUTOCAL_TRUE_THRU_LENGTH . . . 629
NIVNA_ATTR_AUTOCAL_TRUE_THRU_IMPEDANCE 630
NIVNA_ATTR_AUTOCAL_TRUE_THRU_LINE_LOSS 630
NIVNA_ATTR_AUTOCAL_TRUE_THRU_REF_FREQUENCY 631
NIVNA_ATTR_AUTOCAL_BOX_ORIENTATION 631
NIVNA_ATTR_AUTOCAL_BOX_COMMUNICATION_PORT

633
NIVNA_ATTR_AUTOCAL_ASSURANCE_ENABLED .. 633
TRL Calibration 633
NIVNA_ATTR_TRL_CALIBRATION_REFERENCE_PLANE_LOCATION
634
NIVNA_ATTR_TRL_CALIBRATION_NUMBER_OF_BANDS
634
NIVNA_ATTR_TRL_CALIBRATION_STANDARD_TYPE ...
635
NIVNA_ATTR_TRL_CALIBRATION_STANDARD_LENGTH
636
NIVNA_ATTR_TRL_CALIBRATION_STANDARD_LINE_LOSS
637
NIVNA_ATTR_TRL_CALIBRATION_STANDARD_LINE_LOSS_FREQU
637
NIVNA_ATTR_TRL_CALIBRATION_BAND_STANDARDS_LIST
638
NIVNA_ATTR_TRL_CALIBRATION_BAND_BREAK_POINT_TYPE
639
NIVNA_ATTR_TRL_CALIBRATION_BAND_BREAK_POINT_FREQUE
640
NIVNA_ATTR_CALIBRATION_FILE_INTENDED_FOR_SFP ....
640
External Calibration Information 641
NIVNA_ATTR_EXT_CAL_YEAR..... 641
NIVNA_ATTR_EXT_CAL_MONTH 641
NIVNA_ATTR_EXT_CAL_DAY..... 642
NIVNA_ATTR_EXT_CAL_HOUR..... 642
NIVNA_ATTR_EXT_CAL_MINUTE..... 643
NIVNA_ATTR_EXT_CAL_TEMPERATURE 643
NIVNA_ATTR_EXT_CAL_INTERVAL 644
User Calibration Information..... 644
NIVNA_ATTR_USER_CAL_YEAR..... 644
NIVNA_ATTR_USER_CAL_MONTH 645
NIVNA_ATTR_USER_CAL_DAY 645
NIVNA_ATTR_USER_CAL_HOUR 645

NIVNA_ATTR_USER_CAL_MINUTE 646
NIVNA_ATTR_USER_CAL_TEMPERATURE 646
Time Domain 647
NIVNA_ATTR_TIME_DOMAIN_TYPE 647
NIVNA_ATTR_TIME_DOMAIN_TRIP_TYPE 648
NIVNA_ATTR_TIME_DOMAIN_IMPULSE_WIDTH_3DB . . . 649
NIVNA_ATTR_TIME_DOMAIN_ALIAS_FREE_RANGE 649
Range 650
NIVNA_ATTR_TIME_DOMAIN_UNITS 650
NIVNA_ATTR_TIME_DOMAIN_START 650
NIVNA_ATTR_TIME_DOMAIN_STOP 651
Window 651
NIVNA_ATTR_TIME_DOMAIN_WINDOW_SHAPE . . . 652
NIVNA_ATTR_TIME_DOMAIN_WINDOW_KAISER_BESSEL_BETA 653
NIVNA_ATTR_TIME_DOMAIN_WINDOW_DOLPH_CHEBYSHEV_GA 653
Gate 654
NIVNA_ATTR_TIME_DOMAIN_GATE_SHAPE 654
NIVNA_ATTR_TIME_DOMAIN_GATE_ENABLE 655
NIVNA_ATTR_TIME_DOMAIN_GATE_INCLUSIVE . . . 656
Range 656
NIVNA_ATTR_TIME_DOMAIN_GATE_START . . 656
NIVNA_ATTR_TIME_DOMAIN_GATE_STOP . . . 657
NIVNA_ATTR_TIME_DOMAIN_GATE_KAISER_BESSEL_BETA 658
NIVNA_ATTR_TIME_DOMAIN_GATE_DOLPH_CHEBYSHEV_GAMMA 658
DC Term Extrapolation 659
NIVNA_ATTR_TIME_DOMAIN_DC_TERM_EXTRAPOLATION_TYPE 659
NIVNA_ATTR_TIME_DOMAIN_DC_TERM_EXTRAPOLATION_METH 660
NIVNA_ATTR_TIME_DOMAIN_DC_TERM_EXTRAPOLATION_COEFF 660
NIVNA_ATTR_TIME_DOMAIN_DC_TERM_EXTRAPOLATION_USER_

661
Device Information. 661
NIVNA_ATTR_DEVICE_PXI_CHASSIS. 661
NIVNA_ATTR_DEVICE_PXI_SLOT 662
NIVNA_ATTR_DEVICE_TEMPERATURE 662
NIVNA_ATTR_DEVICE_SERIAL_NUMBER 663
NIVNA_ATTR_NUMBER_OF_PORTS 663
Inherent IVI Attributes 663
User Options 663
NIVNA_ATTR_CACHE 664
NIVNA_ATTR_INTERCHANGE_CHECK 664
NIVNA_ATTR_DRIVER_SETUP 665
NIVNA_ATTR_QUERY_INSTRUMENT_STATUS 665
NIVNA_ATTR_RANGE_CHECK. 666
NIVNA_ATTR_RECORD_COERCIONS. 666
NIVNA_ATTR_SIMULATE 667
Driver Identification 667
NIVNA_ATTR_SPECIFIC_DRIVER_DESCRIPTION . . . 668
NIVNA_ATTR_SPECIFIC_DRIVER_PREFIX. 668
NIVNA_ATTR_SPECIFIC_DRIVER_VENDOR 668
NIVNA_ATTR_SPECIFIC_DRIVER_REVISION 669
NIVNA_ATTR_SPECIFIC_DRIVER_CLASS_SPEC_MAJOR_VERSION 669
NIVNA_ATTR_SPECIFIC_DRIVER_CLASS_SPEC_MINOR_VERSION 670
Driver Capabilities 670
NIVNA_ATTR_GROUP_CAPABILITIES 670
NIVNA_ATTR_SUPPORTED_INSTRUMENT_MODELS . . . 671
Instrument Identification 671
NIVNA_ATTR_INSTRUMENT_MANUFACTURER 671
NIVNA_ATTR_INSTRUMENT_MODEL 671
NIVNA_ATTR_INSTRUMENT_FIRMWARE_REVISION 672
Advanced Session Information 672
NIVNA_ATTR_LOGICAL_NAME 672

NIVNA_ATTR_IO_RESOURCE_DESCRIPTOR 673
Individual Sources/Receiver 673
NIVNA_ATTR_SOURCE_DRIVING_STATE 673
NIVNA_ATTR_SOURCE_FREQUENCY_MULTIPLIER 674
NIVNA_ATTR_SOURCE_FREQUENCY_OFFSET 675
NIVNA_ATTR_RECEIVER_FREQUENCY_MULTIPLIER 675
NIVNA_ATTR_RECEIVER_FREQUENCY_OFFSET 676

September 2014, 373153D-01

This help file contains hardware and software information for NI RF vector network analyzers. This file contains an introduction to fundamental RF vector network analyzer concepts, guidelines for accurate measurements, and programming reference information for LabVIEW, C, and LabWindows™/CVI™.

Overview of Vector Network Analyzers

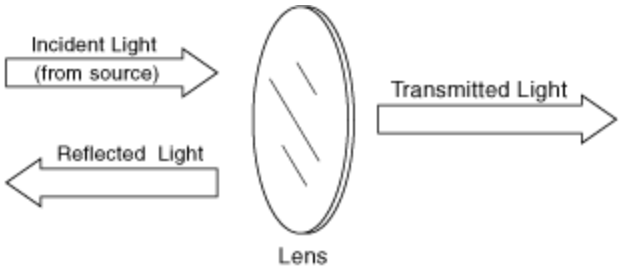
Network analyzers are powerful instruments that, when properly used, provide unparalleled measurement accuracy. Network analyzers are useful in a number of applications and industries, including measuring linear characteristics of radio frequency (RF) components and devices. You can also use modern network analyzers in applications such as signal integrity and materials measurement.

You can use a vector network analyzer (VNA) to measure magnitude, phase, and impedance as seen at the terminals of a network. A VNA compares the incident signal that leaves the analyzer with either the signal transmitted through the device under test (DUT) or the signal that is reflected from its input. A network analyzer is a calibrated closed stimulus-response system, so you can measure RF characteristics with exceptional precision.

The ability to precisely measure magnitude and phase parameters drives the need for exceptional measurement practices to ensure that significant errors do not enter the measurement. While a small error may be inconsequential in contrast to the measurement uncertainty of some RF instruments, it can be substantial with a precise instrument such as a network analyzer.

Network Analysis Principles

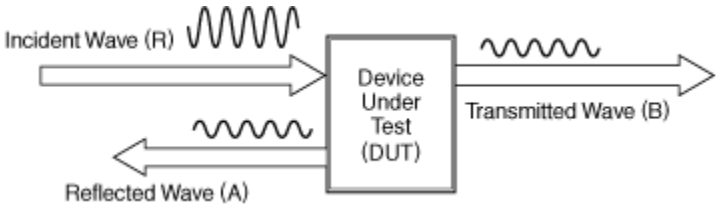
With respect to network analysis, a ***network*** is a group of interconnected electrical components. A network analyzer can quantify the impedance mismatch between two RF components to maximize power efficiency and signal integrity. Each time an RF signal leaves one component and enters another, portions of the signal are reflected and transmitted, similar to the light through the lens, shown in the following figure.



In this figure, light from a source directs an incident signal at an optical device, such as a lens. As light hits the lens, depending on the lens properties, some light is reflected back to the source, and some light is transmitted through the lens. The law of conservation of energy requires that the sum of the reflected and transmitted signal equals the source or incident signal. This example ignores any loss due to heat, which is usually negligible.

We can define a reflection coefficient (Γ), a vector quantity with both magnitude and phase, as the ratio of light being reflected to the total (incident) light. Similarly, the transmission coefficient (***T***) is the vector ratio of transmitted light to the incident light.

The ratio of reflected or transmitted light to the incident light is similar to the performance of the device under test (DUT). Thinking back on the light analogy, if the DUT were a mirror, you would want high reflectivity. If the DUT were a camera lens, you would want it to be highly transmissive. Similar practical measurements can be made in electrical networks. A network analyzer generates a sine wave signal, typically across a range of frequencies. The DUT response includes the incident signal being transmitted through the DUT and reflected back from it, as shown in the following figure.



In the preceding figure, ***T*** is defined by ***B***/***R*** and Γ is defined by ***A***/***R***.

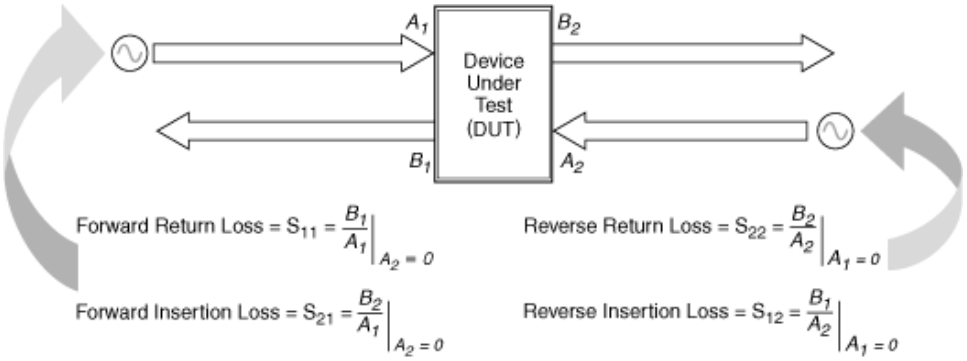
The response of the DUT to the incident signal is affected by the DUT properties as well as any discontinuities in the characteristic impedance of the system. For instance, a bandpass filter is highly reflective out of band, but it is highly transmissive in band. The amount of transmitted and reflected signal usually changes with the frequency. If the

DUT slightly differs from the characteristic impedance, resulting in an impedance mismatch, the DUT could generate additional unwanted responses. When using a VNA, the goal is to develop a measurement methodology that accurately measures the DUT response while minimizing or eliminating uncertainties.

Vector Network Analyzer Measurements

Reflection coefficient (Γ) is the ratio of reflected voltage divided by the incident voltage, and transmission coefficient (***T***) is the ratio of transmitted voltage divided by the incident voltage. Modern network analysis provides more information about these ratios with scattering parameters, or S-parameters.

S-parameters are complex vector quantities that represent the ratio of two RF voltages. S-parameters have a magnitude and phase or, in Cartesian form, real and imaginary terms. S-parameters are expressed as S_{xy}, where ***x*** represents the DUT output port being measured and ***y*** denotes the DUT input port stimulated by the incident RF signal. The following figure shows a simple two-port device, such as an RF filter, attenuator, or amplifier, and how its S-parameters would be derived.



Both quantities, S_{11} and S_{21}, are referred to as ***forward*** S-parameters because the incident signal originates from the RF source on port one. With the incident source on port two, S_{22} and S_{12} are called the ***reverse*** S-parameters. The following table defines the S-parameters that correspond with your NI vector network analyzer.

S-Parameter	Direction	Description
S_{11}	Reflected	The ratio of the voltage reflected at port one to the incident voltage placed on port one.

S_{21}	Transmitted	The ratio of the voltage transmitted through the DUT present at port two to the incident voltage placed on port one.
S_{22}	Reflected	The ratio of the voltage reflected at port two to the incident voltage placed on port two.
S_{12}	Transmitted	The ratio of the voltage transmitted through the DUT present at port one to the incident voltage placed on port two.

As shown in the table above, S_{11} and S_{22} (S-parameters with matching numbers) describe reflected signals. S_{12} and S_{21} (S-parameters with non-matching numbers) describe transmitted signals. Furthermore, the total number of S-parameters required to fully describe the RF characteristics of a device is given by the number of device ports squared.

S-parameters that describe transmission, such as S_{21}, are analogous to other familiar terms including gain, insertion loss, or attenuation. S-parameters that describe reflection, such as S_{11}, correspond to voltage standing wave ratio (VSWR), return loss, or reflection coefficient.



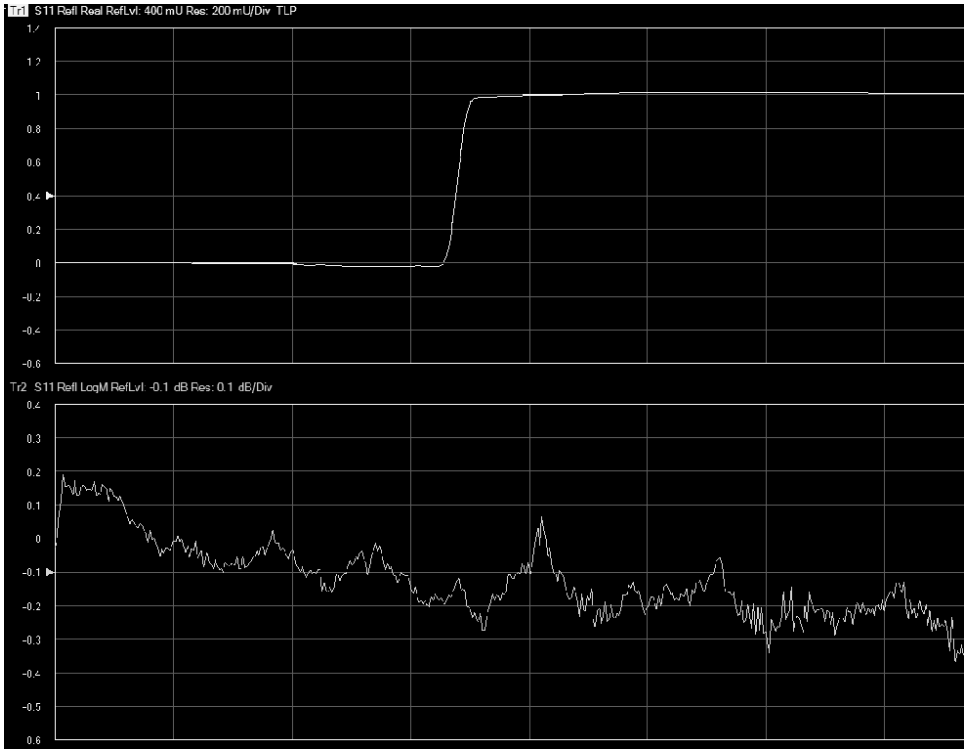
Note A spectrum analyzer combined with a tracking generator can perform similar measurements to a vector network analyzer, but the tracking generator system cannot perform phase measurements, and you cannot perform the same type of error correction and calibration that you can do with a vector network analyzer.

Time Domain Measurements

NI RF vector network analyzers that support time-domain measurements use an inverse FFT to transform data from the frequency domain to the time domain. Viewing data in the time domain allows you to discover information about device performance at specific locations. Time-domain measurements are primarily useful when analyzing broadband low loss devices such as transmission lines, connectors, test fixtures and similar components. Narrowband devices, such as filters, isolators, or matching networks, are less suited for time-domain measurements.

NI-VNA allows you to configure your time-domain measurement by selecting the type of time-domain measurement, choosing a window type, or selecting a gating shape.

The following figure shows the frequency data and the time domain data for a given measurement. In this figure, Trace 1 shows a time-domain lowpass step plot of the S11 data from Trace 2. The lowpass step time-domain data reveals information about the data that is difficult to interpret when viewing only the frequency-domain data.



Activating the Time-Domain Option on Your NI RF Vector Network Analyzer

Frequency measurements are supported on all NI RF vector network analyzers. Time-domain measurements, however, are a product option that is available for purchase from National Instruments. If you purchased the time domain option when you ordered your device, the device should ship from the factory with this option already activated. If you purchase the Time Domain option after you receive your module, use one of the following options to activate the Time Domain option on your device:

- Complete the following steps to use the NI-VNA SFP to activate the Time Domain option.
  1. Launch the NI-VNA SFP.
  2. Select **System»Diagnostics** from the Softkey Menu.
  3. Enter `NIVNA` as the password in the Diagnostics Access dialog box, and click **OK**.

4. Select **Enable Time Domain** from the Softkey Menu.
5. Enter the license key in the Enable Time Domain Option dialog box, and click **OK**.

- Complete the following steps to use Measurement & Automation Explorer (MAX) to activate the Time Domain option.
  1. Launch MAX.
  2. Select the VNA module from your chassis tree item.
  3. Verify that the Time Domain Status displays **Not Activated**.
  4. Right-click the VNA module, and select **Activate Feature** from the shortcut menu.
  5. In the Activate Feature dialog box, use the **Pick a feature to activate** control to select **Time Domain**.
  6. Enter the license key, and click **OK**.
- In LabVIEW, call the niVNA Activate Feature VI and set the **feature** parameter to **Time Domain**. Enter the key as the value of the **activation key** parameter.
- In C or LabWindows/CVI, call the `niVNA_ActivateFeature` function and set the **feature** parameter to `NIVNA_VAL_TIME_DOMAIN`. Enter the key as the value of the **activationKey** parameter.

Related Topics:

- Activating Time Domain in MAX
- Configuring Your Time-Domain Measurement
- Understanding How Resolution and Aliasing Impact Time-Domain Measurements

Understanding How Resolution and Aliasing Impact Time-Domain Measurements

Using time-domain analysis, you can determine the location of discontinuities or impedance mismatches. How closely you can pinpoint the location of the discontinuities is limited by the following main factors:

- **Resolution**—Resolution is controlled by the span of your measurement. The broader the frequency span of your measurement, or bandwidth, the more information presented during the time-domain analysis. This additional information increases the precision with which you can locate points.

Note If you have a DUT with limited bandwidth and the device passes the DC component, you will see a gradual roll-off of DUT performance at higher frequencies. If the limited-bandwidth DUT does not pass the DC component, you see the resolution degrade.

Resolution is also influenced by the processing method, window selection options, and the relative amplitude of the processed signals.

- **Aliasing**—In the frequency domain, measurements are made at a finite number of points, not continuously. The transformation of this data then causes aliasing in the time domain because the transform is a circular function that repeats itself outside of its inherent range of 1 divided by the step size. The more data points available, the smaller the step size and the longer the time frame before the data repeats. For example, with a 20 GHz frequency sweep and 401 data points, the alias-free range is 10 ns.

Configuring Your Time-Domain Measurement

When making a time-domain measurement using an NI vector network analyzer, there are a number of parameters that you can use to configure your measurement, such as the time domain type, frequency windowing type, or the time-domain gate.

Selecting a Time-Domain Measurement Type

NI RF vector network analyzers support the following types of time-domain measurements:

- **Bandpass Impulse**—Transforms the frequency domain data into the representation in the time domain using an inverse FFT.
- **Bandpass Phasor Impulse**—Extracts impedance information for a specific discontinuity from a bandpass display.

Tip For complete impedance information, position the start and stop position for your time domain measurement as close to the discontinuity as possible.

- **Lowpass Impulse**—Transforms the frequency domain data into the representation in the time domain using an inverse FFT. In contrast with Bandpass Impulse, this type requires you to use a harmonically related set of frequencies, starting from the lowest frequency possible.

- **Lowpass Step**—Lowpass step provides a time-domain reflectometer (TDR)-like functionality: the integral of the Lowpass Impulse response. This option allows you to analyze impedance versus time or distance. Similar to the Lowpass Impulse type, you must provide a harmonically related set of frequencies for the analysis.



Note During the time-domain analysis in either of the lowpass types, the NI RF vector network analyzer extrapolates a DC term, which is used to perform the integration and minimize DC error. However, if you know the error associated with your test environment, such as cabling losses, you can incorporate this information by manually specifying the DC term.



Tip For harmonic frequencies, configure your start and stop frequencies such that it satisfies the following equation:

$$\textit{stop frequency} = \textit{start frequency} \times \textit{number of points}$$

Time or Distance

In addition to configuring the type of time-domain measurement, you can also choose whether you want to perform the time-domain analysis in terms of distance or time.

Related Topics:

- Choosing Between Lowpass and Bandpass Measurements
- Windowing
- Gating

Choosing Between Lowpass and Bandpass Measurements

Lowpass processing offers the highest resolution for a given bandwidth and provides twice the resolution of the bandpass types. If the instrumentation, transmission line, and DUT permit lowpass processing, use one of the lowpass types. However, many situations, such as band-limited DUTs, preclude using a harmonically related set of frequencies, so you must use one of the bandpass types.

When using a bandpass type, time-domain analysis does not include a DC component,

so there is no phase reference and vector information is lost. Useful magnitude information is still available and can be displayed in either logarithmic magnitude or linear form. With bandpass types, you can also use an arbitrary number of points and an arbitrary frequency range. In addition, the bandpass phasor impulse type can extract impedance information.

Windowing

In practical applications, you obtain only a finite number of samples of the signal. The FFT assumes that this time record repeats. If you have an integral number of cycles in the time record, the repetition is smooth at the boundaries. However, in practical applications, you usually have an incomplete number of cycles, which results in discontinuities at the boundaries. These artificial discontinuities were not originally present in the signal and result in a smearing or leakage of energy from the actual frequency to all other frequencies. This phenomenon is known as spectral leakage. The amount of leakage depends on the amplitude of the discontinuity, with a larger amplitude causing more leakage.

Because the amount of leakage is dependent on the amplitude of the discontinuity at the boundaries, you can use windowing to reduce the size of the discontinuity and reduce spectral leakage. Windowing consists of multiplying the time-domain signal by another time-domain waveform, known as a window, whose amplitude tapers gradually and smoothly towards zero at edges. The result is a windowed signal with very small or no discontinuities and therefore reduced spectral leakage.

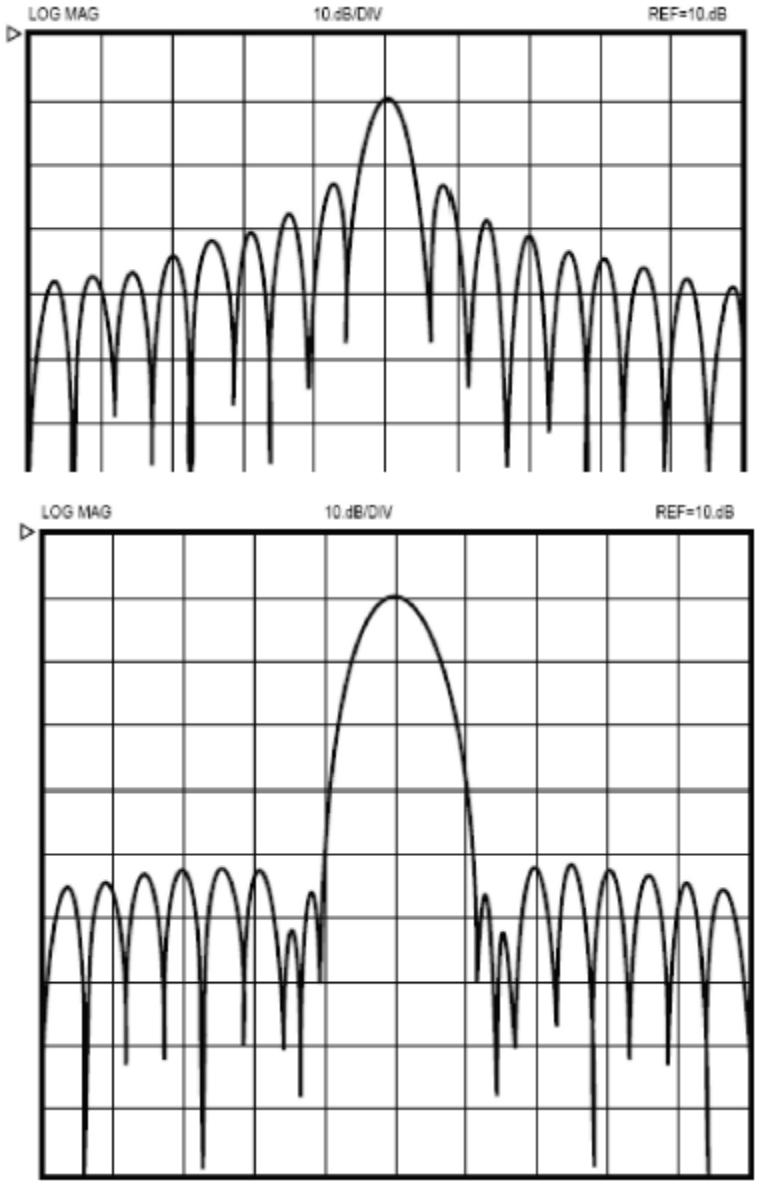
Each smoothing window has its own characteristics and is suitable for different applications. To choose a smoothing window, you must estimate the frequency content of the signal. If the signal contains strong interfering frequency components distant from the frequency of interest, choose a smoothing window with a high side lobe roll-off rate.

NI-VNA supports the following types of windows:

- **Hamming**—This window provides moderate side lobes and resolution.
- **Rectangular Window** —This window provides the maximum resolution but the highest side lobes.
- **Blackman Window**—This window provides low side lobes and low resolution.

- **Blackman-Harris Window**—This window provides the minimum side lobes, but low resolution.
- **Kaiser-Bessel Window**—When you select this value, you can adjust the window using the Window Kaiser-Bessel Beta property.
- **Dolph-Chebyshev Window**—When you select this value, you can adjust the window using the Window Dolph Chebyshev Gamma property.

The following figures show an example of how the Hamming window affects the time domain data. The figure on the left shows time domain data prior to windowing, and the figure on the right shows the time domain data after a Hamming window is applied.



Gating

A gate is essentially a filter that is applied in the time domain. You can use a gate to isolate or remove a given discontinuity or a response that can be attributed to a given DUT, such as a cable that has principal reflections at both ends. After you apply a gate to the measurement, you can then observe the performance of a circuit with the imperfection removed or without the influence of unwanted elements, such as connectors.

When you configure a gate for your measurement, you must also select a gate shape, which is similar to selecting a window. Since the gating process can eliminate some necessary data that can introduce errors or ripple in the frequency-domain response, you must select a gate that allows you to obtain the best tradeoff of resolution versus ripple.

NI-VNA supports the following types of gate shapes:

- **Hamming**—This gate shape provides moderate side lobes and resolution.
- **Rectangular Window** —This gate shape provides the maximum resolution but the highest side lobes.
- **Blackman Window**—This gate shape provides low side lobes and low resolution.
- **Blackman-Harris Window**—This gate shape provides the minimum side lobes, but low resolution.
- **Kaiser-Bessel Window**—When you select this value, you can adjust the gate shape using the Window Kaiser-Bessel Beta property.
- **Dolph-Chebyshev Window**—When you select this value, you can adjust the gate shape using the Window Dolph Chebyshev Gamma property.

Vector Network Analyzer Architectures

Network analyzers are available as both scalar (magnitude only) and vector (magnitude and phase) instruments. Scalar instruments were once widely used because of their simplicity and lower cost. Vector instruments offer better error correction and more complex measurement capability. With advances in technology, integration, computing power, and cost reduction, VNAs are increasingly common.

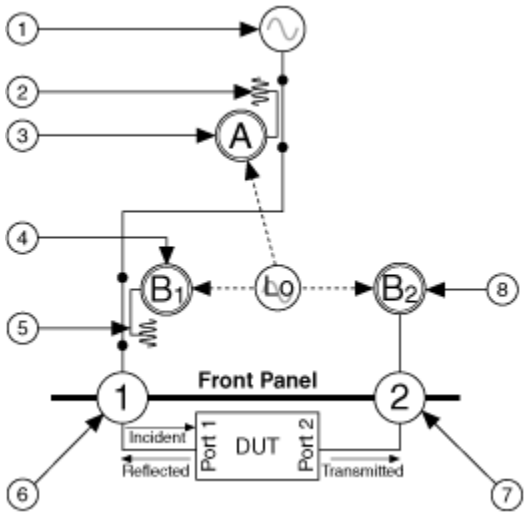
Modern VNAs have the same basic functional process. A signal source produces the

incident signal and feeds into the DUT input. The signal source enters the DUT through the signal separation section, also known as a ***test set***. At this stage, the reflected and transmitted signals are separated. For each frequency point, the device measures the individual voltages and computes the parameter value (for example S_{11} or VSWR).

Test sets are designed as either transmission/reflection (T/R) or full S-parameter.

T/R Test Set

The T/R test set is shown in the following figure.



1 Source 2 Directional Coupler 3 Reference Receiver 4 Receiver B_1	5 Directional Coupler 6 VNA Port 1 7 VNA Port 2 8 Receiver B_2

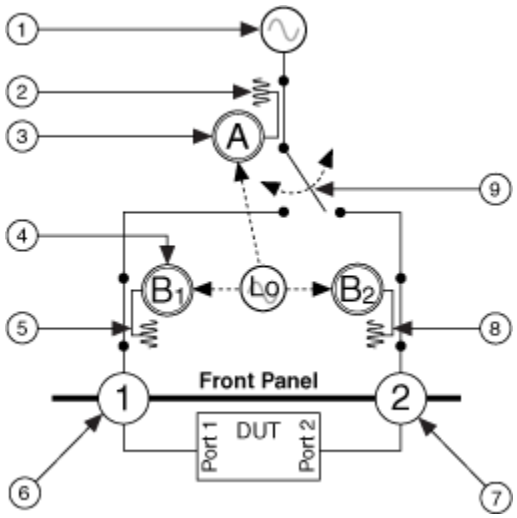
The T/R architecture includes a stable source that supplies a sinusoidal wave signal at a given frequency and power. A reference receiver, ***A***, connected with a power divider or directional coupler, measures the incident signal. The incident signal exits the VNA through port one and enters the DUT. The B_1 receiver measures signals reflected back to port one. The signal transmitted through the DUT enters port two, where the B_2 receiver measures the signal. A coherent LO shared among all receivers enables accurate, ratioed measurements.

T/R architectures are cost-effective, simple, and offer good performance. They measure only forward S-parameters. To measure reverse S-parameters, you need to

disconnect and reverse the DUT or rely on external switching.

Full S-Parameter Test Set

In a full S-parameter architecture, shown in the following figure, a switch is embedded in the signal path after the reference receiver.

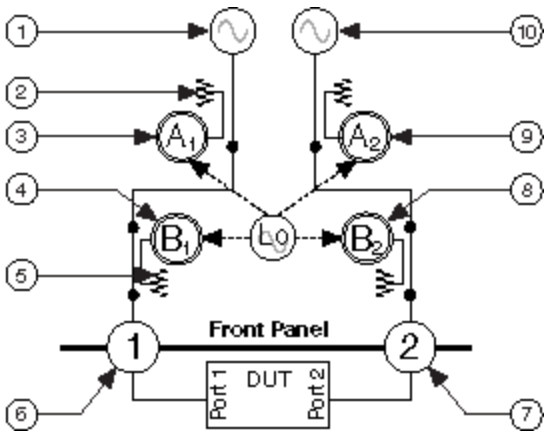


1 Source 2 Directional Coupler 3 Reference Receiver 4 Receiver B_1 5 Directional Coupler	6 VNA Port 1 7 VNA Port 2 8 Receiver B_2 9 Switch

With the switch in the port-one position, the VNA measures forward S-parameters. In the port-two position, the VNA measures the reverse S-parameters.

Full S-Parameter Test Set with Dual Sources

The following figure illustrates a full S-parameter test set architecture with dual sources.



1 Source 1	6 VNA Port 1
2 Directional Coupler	7 VNA Port 2
3 Reference Receiver A_1	8 Receiver B_2
4 Receiver B_1	9 Reference Receiver A_2
5 Directional Coupler	10 Source 2

This architecture includes four receivers and two sources rather than a single source and transfer switch. A dual source architecture eliminates the need for a transfer switch, which improves port-to-port isolation. The absence of a transfer switch also removes any errors during switching and allows for delivery of higher port power to the DUT.

VNA Measurement Best Practices

When making precise measurements with network analyzers, use the following practices to ensure the best results:

- Use quality components and sound measurement practices.
- Use high-quality connectors, adapters, and cables to achieve the highest measurement quality and precision from a VNA.
- Provide proper care for the components, connectors, adapters, and cables and clean them periodically.
- Use a calibrated torque wrench to ensure consistent connections between the VNA and DUT, or the VNA and calibration standards.

Making Measurements with the NI Vector Network Analyzer

When making precise measurements with network analyzers, you must correctly

implement each step in the process for best results. Complete the following steps to make a calibrated measurement with the NI vector network analyzer:

1. Prepare for the measurement.
    1. Warm up the NI vector network analyzer and the DUT as specified in the device specifications.
    2. Clean and inspect all connectors, and ensure all connectors are of the appropriate pin depth.
    3. Use phase-equal adapters to handle a non-insertable connection if you are using SOLT calibration.
    4. Connect cables and adapters to the VNA, following the guidelines in the Connector Care topic.
    5. Verify that the device is within the factory calibration interval.
2. Set up an initial, uncalibrated measurement.
    1. Preset the VNA.
    2. Set up source parameters, including reference frequency, output power, and IF bandwidth.



Tip Increasing the IF bandwidth improves the speed of the sweep measurement, but it introduces more noise. Decreasing the IF bandwidth yields more accurate measurements but at a lower speed. Higher power on the output port improves measurement accuracy, but amplifiers or sensitive devices may require lower output power.

    3. Connect the DUT to verify setup, cables, adapters, and operation.
    4. Select which S-parameter(s) to measure, and choose the display format.
3. Start the measurement.
4. Observe the uncalibrated response.
5. Remove the DUT.
6. Calibrate the device.
    1. Choose whether you want to perform a new calibration or load a previously performed calibration.
    2. To perform a new calibration, complete the following steps:
        1. Set the measurement parameters such as power, frequency, and number of points.
        2. Set IF bandwidth and averaging to minimize noise during calibration.
        3. Choose the proper calibration kit or input calibration standard definitions.

> **Tip** Keep the reference plane constant between your initial, uncalibrated measurement and your calibration setup. The reference plane is established at the location where the VNA connects to the AutoCal module or to the calibration standard. For example, your reference plane may be located at the end of an adapter connected to the end of a precision cable.

4. Calibrate manually or use automatic calibration.
5. Verify calibration quality using a known verification standard.

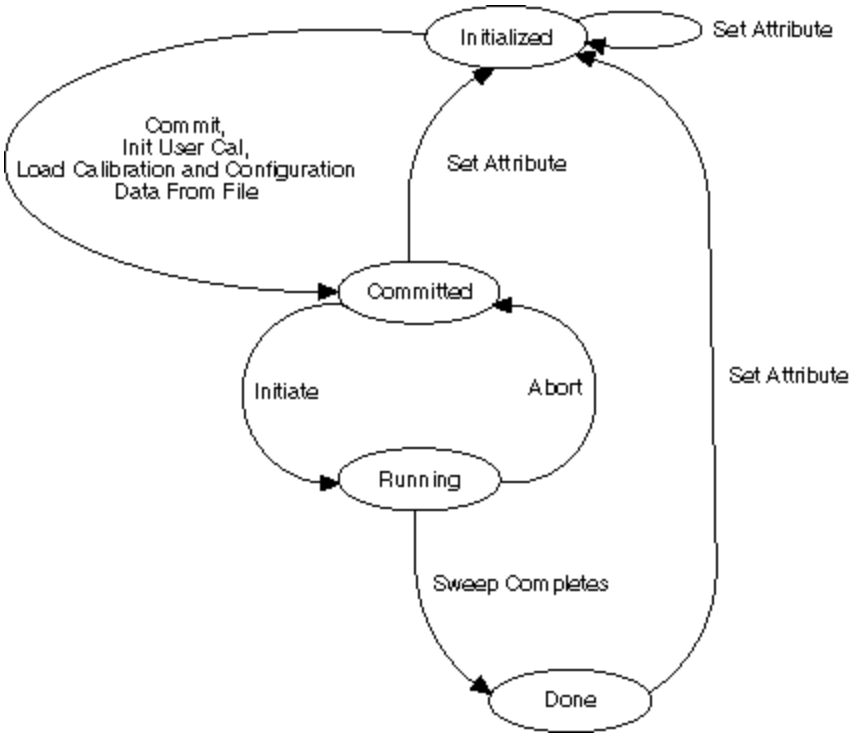
> **Note** Do not use the same calibration standards that you used for calibration to perform the verification.

6. Save the instrument state and calibration.

3. To load a saved calibration, complete the following steps:
   1. Select the calibration file.
   2. Ensure that the source parameters you specified in the step 2b match those that were used in the calibration. Perform a new calibration if the settings do not match.
7. Perform the calibrated measurement.
   1. Measure the DUT.
   2. Ensure the proper correction from the calibrate step is still being applied.
   3. Measure and save DUT parameters.
8. Verify that the calibrated measurement falls within your expected range.

NI-VNA State Diagram

The following figure illustrates the state diagram for an NI-VNA measurement.



In this diagram, after you initialize the device, the device transitions to the Initialized state. From this state, you can set properties and attributes to configure the device. Calling commit, initialize user calibration, or load calibration and configuration data from file commit the settings you have configured and transitions the device into the Committed state. If you set or reset any attributes in the Committed state, the device returns to the Initialized state.

From the committed state, you can initiate the device to enter the Running state, at which point the device waits for a Start Trigger and, after it receives it, starts the sweep operation. If you call abort, NI-VNA transitions to the Done state if the sweep is complete or to the Committed state if aborting stops a sweep in progress. After the device is in the Done state, you can set more attributes, if necessary, which returns the device to the Initialized state.

Calibration

Calibration is an important process for any RF instrument, particularly vector network analyzers. Calibration can be classified as either ***factory calibration*** (also known as ***external calibration***) or ***user calibration***. The user calibration process generally involves characterizing error terms using known standards so that the errors can be

removed from the actual measurements. Unlike factory calibration, user calibration must be performed frequently for network analyzer measurements because vector network analysis is specifically affected by temperature, test environment, cables, and test fixture differences.

Factory Calibration (External Calibration)

Factory calibration is performed yearly on a day specified by the product calibration certificate. Factory calibration is performed at a certified calibration laboratory to ensure the device is operating within manufacturer specifications. The laboratory also traces the instrument performance back to a standards body, such as the National Institute of Standards and Technology (NIST). To achieve higher accuracy, you can perform a user calibration more frequently.

User Calibration

A ***user calibration*** is accomplished with a set of calibration standards from a network analyzer calibration kit or user-fabricated, user-defined standards. By comparing the standard values, stored in the network analyzer, against the measured values of the calibration standards, a set of correction factors is created. These correction factors are applied to the data during the post-calibration measurements to compensate for the sources of error. User calibration removes uncertainty by applying the 12-term error model.

There are various types of user calibration to perform. ***RF Calibration*** refers to the standard calibration family types that calibrate for the accuracy of S-parameter measurements. ***Power Calibration*** refers to calibration methods that improve power accuracy delivered to and received from the device under test (DUT), specifically when you need accurate power levels at some point along the measurement path between the DUT and VNA test ports.

The following user calibration types are referenced in this help file:

RF Calibration:

- Automatic Calibration (AutoCal)
- Short-Open-Load-Through (SOLT) Calibration

- Short-Open-Load-Reciprocal (SOLR) Calibration
- Through-Reflect-Line (TRL) Calibration

Power Calibration:

- Source Power Calibration
- Test Receiver Calibration
- Reference Receiver Calibration

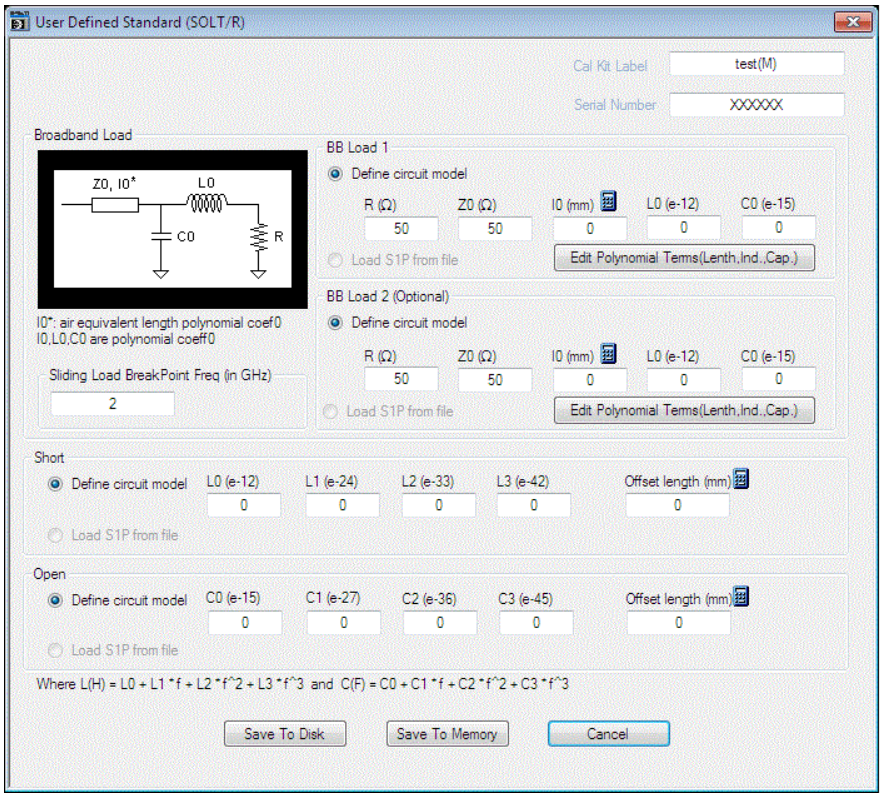
Many factors determine how often user calibration is performed, such as the required measurement accuracy, environmental conditions, and the repeatability of the DUT connection. For accurate results, your NI-VNA device must be calibrated at the ambient temperature after at least 30 minutes of warm-up time and before making any measurements. The NI-VNA should be recalibrated whenever the instrument setup changes or whenever a test port extension cable is added, removed, or replaced. Typically, network analyzers need user calibration every few hours to every few days. You should use verification standards, identification of the sources of measurement uncertainty, and personal experience to determine how often to calibrate. The remainder of this section uses the term ***calibration*** to describe user calibration, not the recommended yearly certified factory calibration.

Using a User-Defined Calibration Kit

NI-VNA allows you to use a third-party calibration kit with your VNA; however, you must specify all the calibration coefficients into the software for the equipment to be properly calibrated.

To specify the coefficients, complete the following steps:

1. Launch the NI-VNA Soft Front Panel.
2. Select **Calibration»Cal Kit/AutoCal Characterization»Create/Edit Kit** from the NI-VNA Settings Panel to launch the Cal Kit Info dialog box.
3. Select the **Cal Kit** option.
4. Enter a name for the new calibration kit in the Define New Cal Kit control, and specify the gender.
5. Click **OK** to launch the User Defined Standard dialog box, shown in the following image.



6. Enter the coefficients in the appropriate controls.
7. Click **Save To Disk** and specify the location to save the calibration file.

After you enter the information about your user-defined calibration kit, you can then save the information and load it again for later measurements. To load the data at a later time, use the niVNA Load User Defined Calibration Kit Data From File VI or the `niVNA_LoadUserDefinedManualCalibrationKitDataFromFile` function.

Calibration Tips

During user calibration, observe the following guidelines:

- To change the gender of a connector to make a through connection, use phase-equal adapters during all steps of calibration to keep the reference plane constant.
- Perform calibration under the lowest noise scenario (when you have a low IF bandwidth configured, high power, and averaging is enabled).
- Keep setup parameters (start frequency, stop frequency, and so on) constant for the most accurate calibration results.

RF Calibration

Each user calibration type has a variety of implementations. Which method you use depends on the DUT, test system, and measurement requirements. The following four methods are commonly used in network analyzer calibration.

- Automatic Calibration (AutoCal)
- Short-Open-Load-Through (SOLT) Calibration
- Short-Open-Load-Reciprocal (SOLR) Calibration
- Through-Reflect-Line (TRL) Calibration

Standard Calibrations

With user calibration, various standards are used to calibrate the number of discrete frequency points configured for the sweep. Refer to the NI-VNA LabVIEW and C/CVI Reference sections for information about how to configure the frequency range and number of frequencies for the sweep.

Changing some NI-VNA settings after running a calibration invalidates the calibration. These settings include the following NI-VNA properties and attributes, which are described in more detail in the NI-VNA Programming section of this help file.

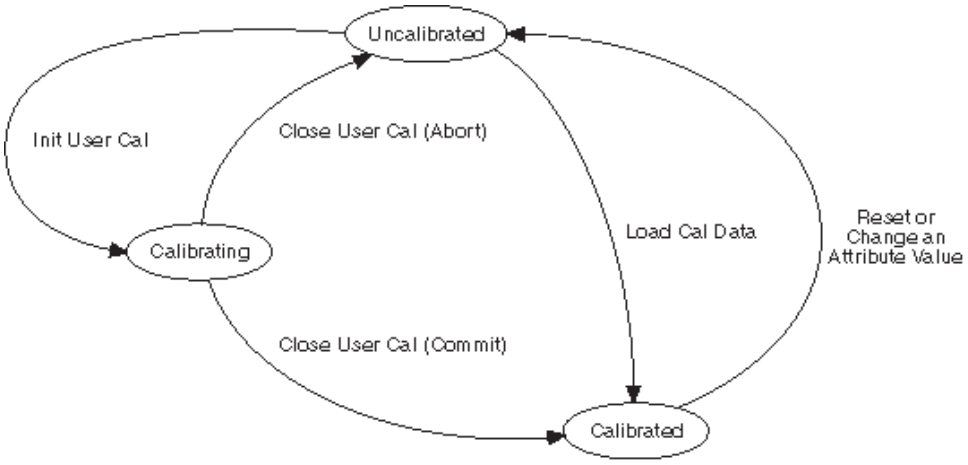
- IF bandwidth
- Source power
- Averaging type
- Frequency sweep characteristics
- Measurement enabled properties/attributes
- Calibration type
- Port connector gender and type
- Most AutoCal module properties/attributes

The one-port calibration is a short/open/load (SOL) calibration that removes source match, directivity, and frequency response errors. The one-path two-port calibration is a short/open/load/through (SOLT) calibration. Full, two-port calibrations can be completed with a through-reflect-line (TRL) calibration or a short/open/load/through (SOLT) calibration. Both TRL and SOLT calibration remove transmission response errors and transmission source match errors in addition to reflection error terms. You can also perform a reflection response calibration or a transmission response

calibration, which corrects the frequency response of the respective path. Reflection response calibration requires an open connected to PORT 1, and transmission response calibration requires a through standard connected between PORT 1 and PORT 2.

Note Reduced measurement accuracy may occur if you use a one-path two-port calibration instead of a full two-port calibration, because the one-path two-port calibration does not account for the load match of port two.

The following figure provides a state diagram for the NI-VNA calibration process.



For information about how to complete a calibration of your NI-VNA device using the NI-VNA SFP, refer to Performing an Automatic Calibration (AutoCal) Using the NI-VNA SFP, Performing a Manual Calibration Using the NI-VNA SFP, or Performing a User Calibration Using the NI-VNA API.

Calibration and Correction

When the device has a valid calibration, you can choose whether to "correct" the measurements by understanding the relationship between enabling correction and the calibration. This relationship is shown in the following table.

Correction Enabled	Calibration State		
	Uncalibrated	Calibrated	Calibrating
FALSE	NI-VNA does ***not*** have a valid set of calibration data for the current device	NI-VNA ***does*** have a valid set of calibration data for the current	N/A

	configuration. Measurement data is ***not*** corrected using calibration data.	device configuration; however, measurement data is ***not*** corrected using calibration data.	
TRUE	NI-VNA does ***not*** have a valid set of calibration data for the current device configuration; however, measurement data ***is*** modified using (potentially invalid) calibration data.	NI-VNA ***does*** have a valid set of calibration data for the current device configuration. Measurement data ***is*** corrected using calibration data.	N/A

Automatic Calibration (AutoCal)

Automatic calibration (AutoCal) units, or external automatic calibration modules, are popular because of their speed, repeatability, and ease of use. Moreover, they remove most human interaction, greatly reducing the chance for a mistake during calibration. These units typically contain an electronic component, such as a diode, termination, or other standard, with a corresponding detailed electrical description encoded in an AutoCal characterization file. When connected to the vector network analyzer, the automatic calibration unit is set to different states. The states measured during calibration are compared with the corresponding known states inside the characterization file to derive the correction values.

Automatic calibration offers a convenient calibration technique that avoids manual connection of standards, which can lead to errors (such as the wrong standard being connected), lost standards, damaged standards, and lost time. In an automatic calibration process, the standards are usually connected using a switch, so you only need to make one cable connection.

Note When you perform an automatic calibration, NI-VNA sets the power at each device port to -10 dBm during the calibration process, regardless of any settings you may have provided.

Note The automatic calibration unit uses an internal through connection to perform the calibration. However, you can also use an external through standard to perform the calibration. This configuration is known as ***true through auto-calibration***. When performing a true through auto-calibration, you must connect the test cables to complete the calibration. The benefit of this method is a higher accuracy calibration balanced against a longer calibration time, though it takes more user interaction to perform.

Automatic Calibration Validation

The automatic calibration unit performs an additional measurement after calibration that validates if the error correction has been correctly applied. The unit does so by measuring a standard within the unit and then comparing that measurement to the characterized measurement. If they match, the automatic calibration is said to be ***assured***.

Note If assurance fails, the calibration may still be valid. In cases where power is low or the number of sweep points is small and the frequency range is narrow, the automatic calibration assurance may not have enough data to assure the calibration. In these situations, always verify by measuring a DUT with a known response.

Short-Open-Load-Through (SOLT) Calibration

SOLT requires short, open, and load calibration standards in the characteristic system (and DUT) impedance. The exact standard values, as determined by their mechanical dimensions, are loaded into the network analyzer prior to calibration. The location where you attach the calibration standards (the network analyzer port, the end of a cable, or inside a test fixture) is where the measurement begins and ends. This location is known as the ***reference plane*** or ***measurement plane***.

In addition, you must make an insertable through connection. For instance, you would use a male-to-female cable connection, or another connection that does not require external adapters or devices, to complete the through connection during SOLT calibration. Inserting any component during calibration and not using it in the post calibration measurement results in measurement error. If you cannot make a through connection, the component is referred to as "non-insertable." There are several methods to handle non-insertable devices. The simplest method is to use a set of phase-equal adapters (included in most calibration kits) along with shorts, opens, and loads of each gender. Use one adapter to complete the through connection during calibration and swap it with an appropriate adapter for the DUT connection during post calibration measurements.

Other calibrations in the SOLT family include the response calibration. It is quick, but no more accurate than removing the path loss across frequency. It only accounts for the forward and reverse tracking terms in the 12-term error model. You can perform a one-port calibration by placing a short, open, and load on port one during calibration. This step saves some time if you require only a one-port measurement, such as when measuring the return loss of an antenna. An enhanced one-port calibration performs a full one-port calibration and uses the through connection to measure port two. This configuration is common in a T/R architecture, in which no source is present on port two. Finally, there is full two-port SOLT calibration, in which you place shorts, opens, and loads on both ports, as directed by the calibration routine. The full two-port calibration concludes with the through connection.

Short-Open-Load-Reciprocal (SOLR) Calibration

Short-Open-Load-Reciprocal (SOLR) calibration is a combination between the Short-Open-Load-Through (SOLT) and the Through-Reflect-Line (TRL) family of calibrations. SOLR uses the SOLT short, open, and load standards and requires the same characterization knowledge. The difference between SOLR and SOLT calibration is that SOLR does not require an electrical model of the through standard. However, SOLR calibration does require that the through standard be reciprocal, such that S_{21} equals S_{12}. As with TRL calibrations, the most accurate implementation of this calibration method requires a four-receiver hardware configuration, where SOLT calibration could be performed with a three- or four-receiver design.

The TRL calibration assumption of the ideal switch, which is inaccurate, is overcome using extra measurements made during the through portion of the calibration. These measurements, and the reciprocity of the through standard, allow the through standard to be characterized and corrected for during the calibration. You need to know the approximate electrical phase of the through to allow a root choice during the characterization and correction of the through standard.

The limited knowledge requirement on the through standard makes SOLR calibration ideally suited for situations where a non-ideal through standard is desired, such as with on-wafer probing where the probes are positioned 90 degrees from each other

(orthogonally) during the desired measurement. If this measurement configuration is necessary, there are two options for making calibrated measurements. One option is to move the measurement probes away from their desired calibration measurement positions using a straight through standard, which introduces errors in the final measurements. The errors are caused by the necessary cable movements when repositioning the ports for the final measurements. You also can calibrate using a through standard with a 90-degree bend. A bend makes the through behave non-ideal and therefore creates errors or degradation in those calibration models relying on the ***ideal*** characteristics. SOLR calibration does not have this limitation since it does not rely on an ***ideal*** through model and thus does not introduce this degradation during the measurement. The use of the SOLR algorithm with a 90-degree through standard simplifies both the calibration and the measurement, which provides the best calibration for these less ideal measurement configurations.



Note SOLR calibration is compatible with only the NI 5632.

Through-Reflect-Line (TRL) Calibration

TRL is one type in a family of two-port calibrations that better support noncoaxial environments such as on-wafer measurements and test fixturing. In this family, the name of the calibration is an acronym that identifies the following necessary calibration standards.

- Through-Reflect-Line (TRL)
- Line-Reflect-Match (LRM)
- Line-Reflect-Line (LRL)
- Through-Reflect-Match (TRM)

These calibrations derive error terms for the same 12-term error model as SOLT calibrations. Additional information about the port match of the hardware is required to correct perfect-match assumptions used to derive these calibration models. NI recommends you retrieve this information during the measurement of the calibration standards. Four receivers are required during calibration to provide correlated measurements of signals entering and exiting both ports, while only one port is sourcing a signal.

Fewer calibration standards, typically three, are required for this family of calibrations, compared to SOLT calibrations. In noncoaxial environments, standards are often harder to obtain but still easier to fabricate than performing a calibration and measurement with coaxial connections. The standards must be fabricated with care as the accuracy of the corrected measurements is dependent upon the quality and repeatability of the calibration standards. These calibrations generally use combinations of through, line, reflection, and match standards. The following table defines the standards that are used in these calibrations.

Standard	Description
Through	Ideally, a zero length through line where the measurement ports are directly connected. An alternative would be a line with electrical length, but this line must be electrically distinct from any other line standard. The intrinsic impedance of the through and line standards set the impedance of the calibration.
Reflection	A highly reflective device (most often a short or open). The absolute reflection coefficient is derived from the additional standard measurement information, but knowledge of the phase must be within one-fourth wavelength to ensure proper identification of the standard. It is important that you use the equivalent reflection standards to calibrate both ports.
Line	A nonzero length through line with the same intrinsic impedance as the through standard. The difference between the phase shifts of the line and through standards require a minimum of 10 degrees to 170 degrees or a preferred 20 degrees to 160 degrees. It is not necessary to know the propagation constant because you can derive it from the extra measurement information. The required phase relationship between the through and line standards creates restrictions on the frequency span that can be measured with a given though and line combination. If you need larger frequency spans, use multiple through and line pairs.
Match	If a match standard is used in place of a line standard, you can use its impedance to set the impedance of the calibration. The match standard should be identical for each port and should generally use characterized data or assume data to be a perfect match.

TRL is a calibration form that makes use of fewer standards, requiring less knowledge about the standards. TRL calibration is highly suited for noncoaxial environments where traditional standards may be hard to fabricate and fully characterize. It requires high-quality, repeatable standards to ensure accurate calibrations. Frequency span limitations exist that are enforced by the requirement to identify standards by their phase relationships during measurements, but this limitation can be overcome with

additional standards as required. Since much of the standard characterization is extracted during the measurements of the standards themselves, the general accuracy of the calibration is better than SOLT, which includes the use of mechanical standards electrical models.

	Note TRL calibration is compatible with only the NI 5632.

Power Calibration

The following three power calibration methods are commonly used in network analyzer calibration to improve power accuracy delivered to and received from the device under test (DUT).

- Source Power Calibration
- Test Receiver Calibration
- Reference Receiver Calibration

Source Power Calibration

The source power calibration feature improves the accuracy of the power delivered to the device under test (DUT) by moving the power reference plane to the DUT. If a network or cable, for example, is inserted between the DUT and the vector network analyzer, you can use source power calibration to ensure desired absolute power accuracy at the DUT connection. Absolute power accuracy can compensate for cable loss.

Power calibration is compatible in both frequency sweep and power sweep modes. In power sweep mode, the source power is adjusted for the specified frequency across the power sweep range to provide a linear level at the calibration plane. In frequency sweep mode, the source power is adjusted to provide a constant level across the frequency sweep range at the test port. You can connect a power sensor to the port to be calibrated, PORT 1 or PORT 2, and at each point in the sweep, the system adjusts to deliver the desired absolute power level within a tighter accuracy.

The following table lists supported power meters you can use for source power calibration.

RF Power Meter	Part Number	Frequency Range	Power Range	Connectivity
NI USB-5680	780167-01	50 MHz to 6 GHz	-40 dBm to +23 dBm	USB
NI USB-5681	781214-01	10 MHz to 18 GHz	-40 dBm to +20 dBm	USB

Note Install the appropriate driver software for the power meter before using the power meter with NI-VNA.

Note Source power calibration is compatible with only the NI 5632.

Test and Reference Receiver Calibration

The receiver calibration feature improves the accuracy of the individual receivers by compensating for the frequency response of each receiver. This feature is typically used when performing raw receiver measurements, or unratioed measurements, to achieve accurate absolute power levels. The ***test receiver*** designation refers to the B_1 and B_2 receivers, and the ***reference receiver*** designation refers to the A_1 and A_2 receivers, as shown in the full S-parameter test set with dual sources diagram.

Test receiver calibration can be viewed as moving the power reference plane of the B_1 or B_2 receiver from within the vector network analyzer to the calibrated power reference plane at the device under test (DUT) connection or at the end of the cable, where the absolute power is accurately known. Performing a reference receiver calibration results in an A_1 or A_2 receiver that reads back the known absolute power level obtained from an applied source power calibration. Therefore, the receiver displays the known input power to the DUT, which could be useful for amplifier compression measurements. If you have not completed a source power calibration, the power accuracy depends on the device factory calibration.

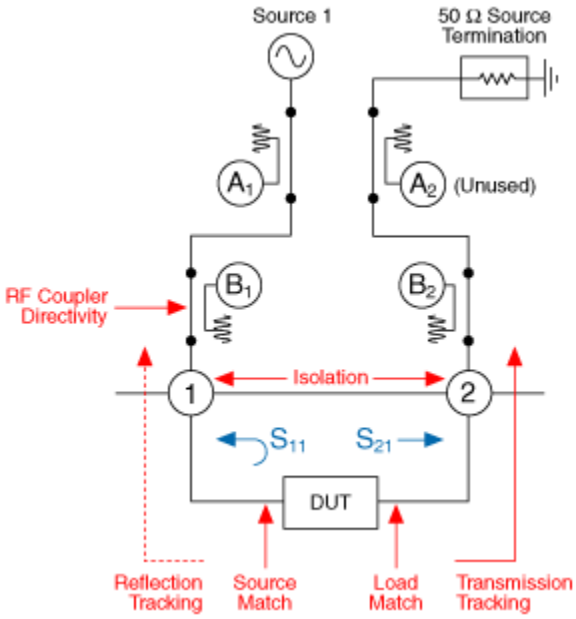
Note Test and reference receiver calibration is compatible with only the NI 5632.

Error and Uncertainty

Sources of error are minimized by two different types of calibration: factory (external)

calibration and user calibration. Sources of error can be characterized as either random or systematic. Random sources of error are not repeatable or predictable and can typically be reduced by narrowing IF bandwidth settings or through averaging. If you understand the systematic sources of error, however, and apply user calibration to reduce the errors, your VNA can make measurements with excellent accuracy.

The following figure shows the six systematic error terms associated with a full S-parameter dual source network analyzer measurement of forward S-parameters only.



The following systematic error terms are shown in the preceding figure:

- **Transmission and Reflection Tracking**—The first uncertainty is the transmitted and reflected signal loss across frequency, or ***transmission and reflection tracking***, respectively.
- **Load and Source Match**—The difference between the DUT input impedance and the network analyzer or system impedance are the ***source match*** and ***load match***, respectively. The same concept applies to the DUT output impedance.
- **Directivity**—An ideal directional coupler produces an output signal in the coupled arm that is proportional to the measured signal traveling in one direction of the main arm, while producing no output for a signal traveling in the opposite direction. The difference between the coupler output (coupled arm) and the measured input signal (through arm) is the ***coupling factor***. Values of 10 dB to 30 dB are common, which means that the coupler output RF power is 10 dB to

30 dB less than the input signal passing in the through arm in the proper direction. In theory, a directional coupler should produce no output for a signal traveling in the opposite direction; however, in reality, a signal traveling in the opposite direction through the coupler produces an unwanted response at the coupler's output, which is known as coupler leakage. The difference between the coupling factor and the coupler leakage is known as ***coupler directivity***.

- **Isolation**—A small amount of incident signal is radiated or conducted from port one and is detected at the port two receiver, which is known as ***isolation*** or crosstalk. An isolation calibration can correct for internal crosstalk for measurement of DUTs with high loss.

These forward terms combined with the six other terms from measuring the reverse S-parameters, total 12 error terms. User calibration needs to adequately account for these 12 error terms so that the VNA can apply the proper correction factors to the measured data. This correction is a primary contributor to the remarkable accuracy of vector network analyzers.

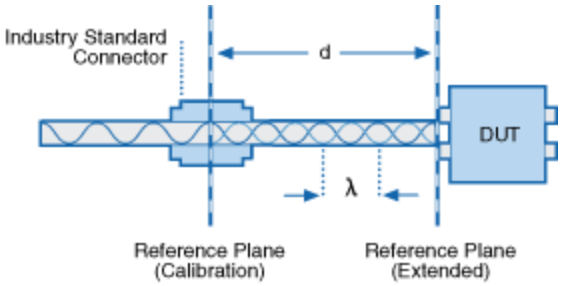
Reference Plane

In a VNA measurement, the ***reference plane*** is known as the location within the system where the user calibration is performed and is therefore the plane where the measurement is typically made.

VNA measurements are commonly made using cables connected from the device front panel to the device under test (DUT). When calibrating the VNA, you must account for the length and characteristics of those cables to remove any effects the cables could have on your measurement, such as time delay, phase shift, or attenuation. Establishing a reference plane is most commonly done by attaching calibration standards to the ends of the cables during calibration, which establishes the end of those cables as your calibration reference plane.

Reference Plane Extensions

The reference plane exists at the point where calibration was performed, but if needed, it is possible to extend the reference plane. A ***reference plane extension***, shown in the following figure, is one technique you can use when it may not be practical to use custom calibration standards that match the test setup.



In the preceding figure, distance (***d***) refers to the measured distance between the calibrated reference plane and the DUT input. This distance could represent an additional cable or test fixture that was not included in the calibration. If you are using a reference plane extension, ***d*** refers to the extended distance of the reference plane as you move it toward the DUT. The wavelength (λ) is inversely proportional to the frequency of the measurement.

To manually move or extend the reference plane after calibration, you must specify the properties of the cable or test fixture you want to account for, such as distance, time, phase offset, and loss. To specify these properties automatically, you can insert an open or short calibration standard at the location in the test setup where you want to relocate the reference plane. The figure above shows a full reflection condition when the DUT is replaced by an open or short, and the incident voltage is reflected back to the VNA. The VNA can then perform the calculations necessary to move the reference plane and adjust subsequent measurements to match the desired measurement setup.

You can apply reference plane extension to your system by navigating to the NI-VNA Soft Front Panel (SFP) and selecting **Measurement»Reference Plane** from the NI-VNA SFP Menu.

Configuring Hardware in MAX

Use Measurement & Automation Explorer (MAX) to configure your National Instruments hardware. MAX informs other programs about which devices reside in the system and how they are configured. MAX is automatically installed with NI-VNA.

1. Launch MAX by navigating to **Start»All Programs»National Instruments»Measurement & Automation** or by clicking the Measurement & Automation desktop icon.

2. In the Configuration pane, double-click **Devices and Interfaces** to see the list of installed devices. Installed devices appear under the name of their associated chassis.
3. Expand your **Chassis** tree item. MAX lists all devices installed in the chassis. Your default device names may vary.

> **Note** If you do not see your hardware listed, press to refresh the list of installed devices. If the device is still not listed, power off the system, ensure the device is correctly installed, and restart.

4. Record the device identifier MAX assigns to the hardware. Use this identifier when programming your NI vector network analyzer device.
5. The MAX self-test performs a basic verification of hardware resources. To self-test a module in MAX, select the module in the configuration tree and select **Self-Test** in the MAX toolbar.

Related Topic:

- Renaming a Hardware Module in MAX

Renaming a Hardware Module in MAX

Measurement & Automation Explorer (MAX) allows you to rename your NI vector network analyzer module. The MAX name is used in NI-VNA to operate the hardware resources. You are not required to change the module name from the default but doing so can make programming easier.

To rename a hardware module in MAX, complete the following steps:

1. Select the VNA module in the chassis tree item and select **Rename** in the MAX toolbar.
2. Enter the new name for the module.

> **Note** The device name must consist only of alphanumeric and underscore characters.

3. Press <Enter>.
4. Verify that the new name for the module is displayed.

Related Topic:

- Configuring Hardware in MAX

Activating Time Domain in MAX

Measurement & Automation Explorer (MAX) allows you to determine the activation status and activate the time domain feature of your VNA device.

Complete the following steps to activate the Time Domain option.

1. Launch MAX.
2. Select the VNA module from your chassis tree item.
3. Verify that the Time Domain Status displays **Not Activated**.
4. Select the VNA module, and select **Activate Feature** in the MAX toolbar.
5. In the Activate Feature dialog box, use the **Pick a feature to activate** control to select **Time Domain**.
6. Enter the license key, and click **OK**.

Related Topics:

- Configuring Hardware in MAX
- Renaming a Hardware Module in MAX

NI-VNA Examples

NI-VNA includes several example applications that demonstrate the functionality of your device and can serve as interactive tools, programming models, and building blocks for your own applications.

NI Example Finder

The NI Example Finder is a utility that organizes examples into categories and allows you to browse and search installed examples. For example, search for "`VNA`" to locate all NI RF vector network analyzer examples. You can see descriptions and compatible hardware models for each example or see all the examples compatible with one particular hardware model.

To locate examples using the NI Example Finder within LabVIEW or LabWindows/CVI, select **Help**»**Find Examples** and navigate to **Hardware Input and Output**»**Modular Instruments**»**NI-VNA**.

Installed Example Locations

Software Application	Installed Example Locations
LabVIEW	`<LabVIEW>\examples\instr\niVNA`, where `<LabVIEW>` is the LabVIEW directory for the specific LabVIEW version you installed on your system.
LabWindows/CVI	`<NIDocDir>\NI-VNA\examples`, where `<NIDocDir>` is one of the following directories: • Windows 8.1/8/7/Vista—`Users\Public\Documents\National Instruments` • Windows XP—`Documents and Settings\All Users\Shared Documents\National Instruments`
ANSI C or Microsoft Visual C/C++	

Devices

Expand this book to view the topics that describe your vector network analyzer hardware and functionality in more detail.

This section includes information about the following devices:

- NI 5630 vector network analyzer
- NI 5632 vector network analyzer

NI PXIe-5630

This section includes useful information about the NI PXIe-5630 (NI 5630) hardware, including module front panels, theory of operation, a block diagram, and calibration information.

The NI 5630 is a two-port/one-path vector network analyzer. This device implements a T/R test set architecture and has the following basic characteristics:

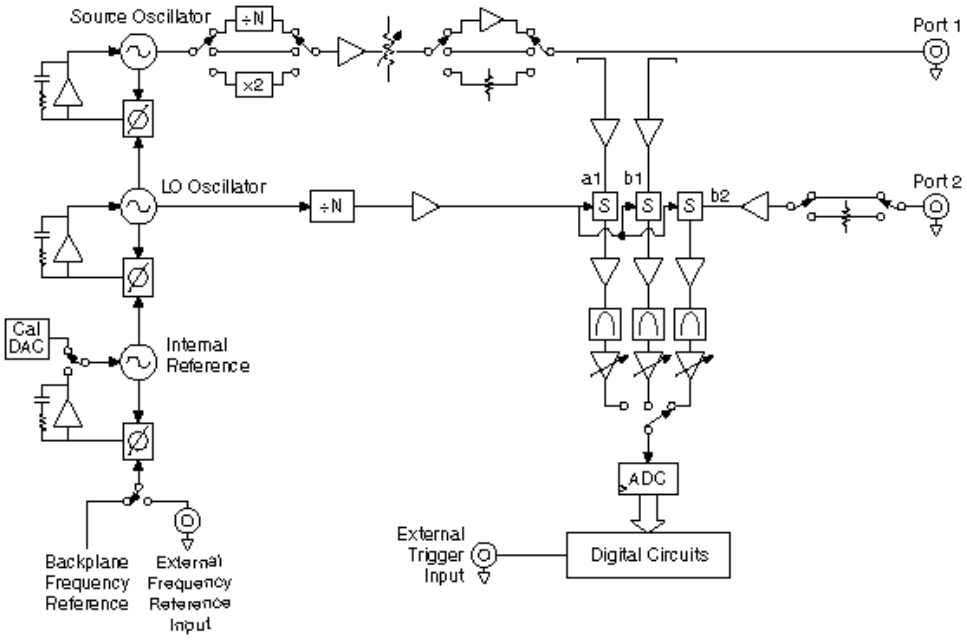
- Supports a frequency range from 10 MHz to 6 GHz
- Supports a power range from -30 dBm to +5 dBm in 0.5 dB steps
- Supports an IF bandwidth (IFBW) from 10 Hz to 30 kHz
- Can be programmed with the NI-VNA instrument driver or controlled interactively using the NI-VNA Soft Front Panel (SFP)
- Contained in a compact, two-slot-wide PXI Express 3U form factor

The NI 5630 follows industry-standard PlugPlay specifications for the PXI bus and seamless integration with compliant systems.

Note For detailed NI 5630 hardware specifications, refer to the ***NI PXIe-5630 Specifications*** document located at ni.com/manuals for updated documentation resources.

NI 5630 Block Diagram and Theory of Operation

The NI PXIe-5630 is a one-path two-port VNA. A simplified block diagram is shown in the following figure.



The source (SRC) voltage-controlled oscillator (VCO) is locked to the local oscillator (LO) VCO, which is in turn locked to a reference oscillator (VCTCXO). You can lock the reference VCTCXO oscillator to the 10 MHz system reference on the PXI Express backplane (if the backplane is supplied with a sufficiently good external clock

reference) or to an external frequency reference using the REF IN connector on the module front panel.

The SRC VCO tunes the frequency, and it is then converted into 10 MHz to 6.0 GHz through either a frequency doubler, fundamental through mode, or a series of frequency dividers. Power control is achieved through three switched gain states preceded by a programmable digital attenuator. The digital attenuator has 30 dB of control range, in 0.5 dB steps. The fixed gain states have approximate gains of +15 dB, 0 dB, and -20 dB. Together, the system control allows for 0.5 dB steps from +5 dBm to –30 dBm at PORT 1.

The SRC output goes through a dual-bridge structure that separates the forward waves from the reflected waves at PORT 1. The dual bridge is a four-port device. The through path of the bridge connects the source output to PORT 1. The coupled arms of the dual-bridge divert part of the forward signal from the SRC and the reflected signal from PORT 1. These are the incident (a1) and reflected (b1) signals, respectively. The RF source signal goes out PORT 1, through a device under test (DUT), and then back to PORT 2. The signal at PORT 2 is the transmitted signal (b2). The VNA uses these three signals (a1, b1, and b2) to calculate the S-parameters of the DUT.

The LO is also divided down. The receiver structure is sampler-based and is driven with a low drive frequency. This lower frequency is then put into a step-recovery diode (SRD), which creates extremely high harmonics of the LO. Phase-lock loops on the SRC VCO and LO VCO accurately control the source output frequency and set the VCO drive frequency such that mixing with the source signal gives the correct IF frequency.

The IF output signals from the samplers are approximately 195 kHz. Each of the three IF signals is filtered, amplified, and eventually digitized by a single ADC. The ADC output is sent to the digital circuits for detection processing. Receiver IF bandwidth control (specific frequencies from 10 Hz to 30 kHz) is accomplished using digital signal processing and optimizes receiver dynamic range versus sweep speed for a particular measurement. You can use NI-VNA to configure the IF bandwidth of the device, using the IF Bandwidth property or the `NIVNA_ATTR_IF_BANDWIDTH` attribute. Increasing the IF bandwidth improves the speed of the sweep measurement, but it introduces more noise. Decreasing the IF bandwidth yields more accurate measurements but at a lower speed.

The output power from PORT 1 is configurable using the Output Power property or the NIVNA_ATTR_OUTPUT_POWER attribute. You can set the output power to –30 dBm to 5 dBm, in 0.5 dB increments. Higher power on the output port improves measurement accuracy, but amplifiers or sensitive devices may require lower output power.

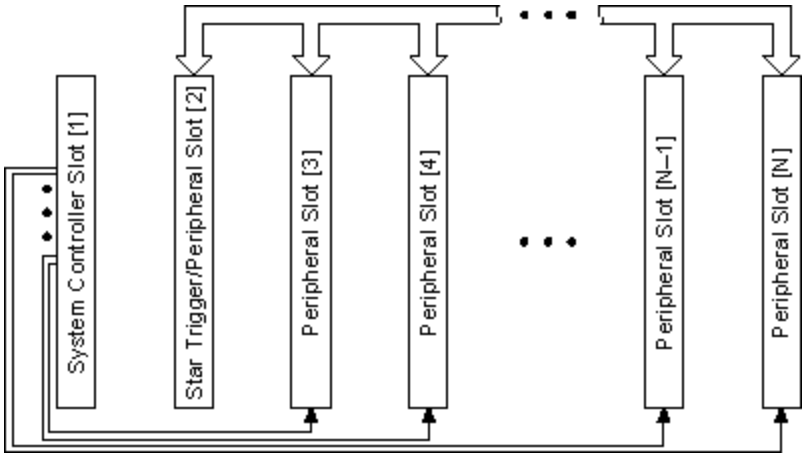
Note The default power of +5 dBm is always on, beginning when you power on the device. When the device is put on hold, it stops sweeping across frequencies but the output power continues to be generated.

In addition to the PORT 1 and PORT 2 connectors on the NI 5630 front panel, there is a connector for an external frequency reference input (REF IN) and one for an external trigger (PFI 0) that you can use to initiate a sweep.

The system is calibrated at production time for frequency accuracy, power level accuracy, and the gain changes in the RF and IF paths. User-defined calibrations are required to achieve specified measurement performance.

PXI Trigger Lines

Eight PXI bus trigger lines are highly flexible and can be used in a variety of ways. For example, triggers can be used to synchronize the operation of several different PXI peripheral devices. In other applications, one device can control carefully timed sequences of operations performed on other devices in the system. Triggers may be passed from one device to another, allowing precisely timed responses to asynchronous external events that are being monitored or controlled. The number of triggers that a particular application requires varies with the complexity and number of events involved.



The PXI Specification is implemented with the RTSI bus through the PXI trigger lines. PXI Specification requires eight lines, PXI_Trig<0..7>, on the P2/J2 connector of the PXI chassis for the trigger lines. The RTSI features of NI hardware is implemented on this sub-bus. The RTSI triggers <0..6> are implemented on PXI_Trig<0..6>, and the RTSI clock is routed on PXI_Trig7.

System Reference Clock

The PXI chassis supplies the PXI 10 MHz system Reference Clock signal (PXI_CLK10) independently to each peripheral slot. An independent buffer drives the clock signal to each peripheral slot. The buffer has a source impedance matched to the backplane and a skew ranging from less than 1 ns to better than 250 ps between slots. You can use this common Reference Clock signal to synchronize multiple devices in a measurement or control system. You can drive PXI_CLK10 from an external source through the PXI_CLK10_IN pin on the P2 connector of the PXI star trigger slot, which is Slot 2. Sourcing an external clock on this pin automatically disables the 10 MHz source on the backplane. You can synchronize multiple chassis that have connectors on the back panel for 10 MHz reference INPUT and 10 MHz reference OUTPUT. Refer to your PXI chassis documentation for more information.

Note NI-VNA devices do not work as PXI 10 MHz system clock signal sources.

PFI Lines

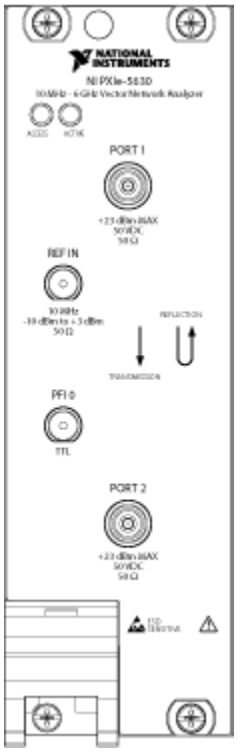
PFI lines are multipurpose programmable function input/output lines. These lines serve as connections to virtually all internal timing signals. NI vector network

analyzers have up to seven digital lines that can accept or generate a trigger, generate a marker, accept or generate a Reference Clock. The function of each PFI line is independent.

Caution If you enable a PFI line for output, do ***not*** connect any external signal source to it; doing so can damage the device, the computer, and the connected equipment.

NI 5630 Front Panel and Connectors

This section describes the connectors and LED indicators on the NI 5630 module front panel.



Connector	Type	Function
PORT 1	Type-K	Source terminal for NI 5630 measurements.
REF IN	SMA	Input terminal for an external reference signal for the NI 5630.
PFI 0	SMA	Input terminal for an external trigger signal.
PORT 2	Type-K	Receiver terminal for NI 5630 measurements.

The following table provides LED and indications information for the NI 5630 front panel LEDs.

LED	Indication
ACCESS	Indicates the basic hardware status of the NI 5630 module. • OFF—The module is not yet functional. • AMBER—The module is being accessed. Accessed means that the device is being communicated with over the PXI Express bus. • GREEN—The module is ready to be programmed by NI-VNA. • RED—The module has detected a hardware error, such as a hardware failure. The LED remains red until the error condition is removed.

	ACTIVE	Indicates the state of the NI 5630 module. • OFF—The module is not yet functional. • AMBER—The module is armed and waiting for a trigger. • GREEN—The module has received a Start Trigger. This state also indicates that the module is making a measurement. • RED—The module has detected a spurious error, such as when the PLL becomes unlocked. The LED remains red until the error condition is removed.

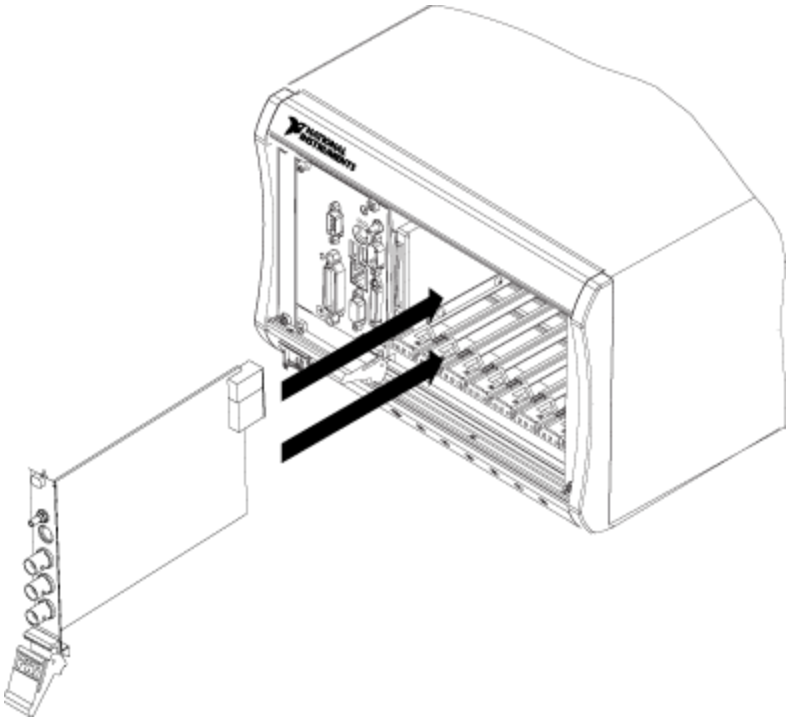
Hardware Integration and Maintenance

This section contains information about integrating and maintaining NI vector network analyzers in a PXI Express-based measurement system.

PXI Express Modules

The PXI Express Specification integrates PCI Express signaling into the PXI standard, which increases backplane bandwidth and enhances PXI timing and synchronization features by incorporating a 100 MHz differential Reference Clock and differential triggers. The PXI Express specification adds these features to PXI while maintaining backward compatibility.

All NI vector network analyzers exist in the PXI Express form factor.



Chassis Guidelines

NI PXI Express vector network analyzers can be installed in the following PXI Express chassis slots:

- **PXI hybrid slots**—Accepts either PXI modules that are hybrid slot-compatible or PXI Express modules
- **PXI Express slots**—Accepts PXI Express modules

Note Refer to the documentation for your PXI Express device and chassis for more information about installing and configuring PXI Express modules.

Maintaining PXI Express Hardware

Clean the fan filters on the chassis regularly to prevent fan blockage and to ensure efficient air circulation. Cleaning frequency depends on the amount of use and the operating environment. For specific information about cleaning procedures and other recommended maintenance, refer to the device specifications and the chassis user documentation.

Connector Care

Proper use and care of the connectors on your vector network analyzer is important. Improper connections can result in damage to your device and invalid measurements. Follow the guidelines in this section to prevent damage and to make the best possible measurements.

Caution NI is ***not*** liable for damage to NI VNA devices that results from improper connector care and signal connections.

Tighten Connectors Without Rotating the Center Pin

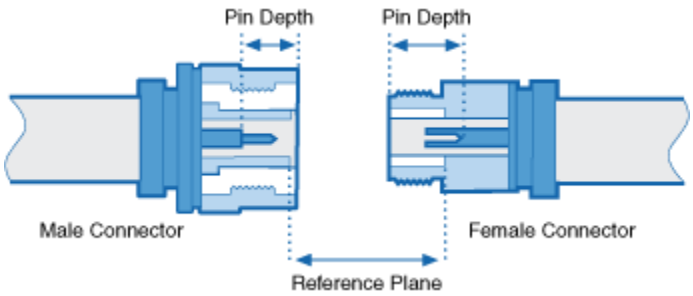
Never tighten the connectors, adapters, or cables such that the center pin rotates.

Avoid Touching Connectors with Bare Hands

Avoid touching Type-K connector mating planes with bare hands. Natural skin oils and microscopic dirt particles are difficult to remove.

Observe Pin Depth of Mating Connectors

Make sure to use the proper connectors to mate with the NI vector network analyzer. NI recommends that you use precision or instrument or metrology grade Type-K, 2.92 mm, 3.5 mm, or SMA connectors. Before mating the connector with another device or cable, measure the pin depth of the device that you want to mate with the RF component, using a pin depth gauge. Improper pin depth of mating connectors is a major cause of failure in the field. When an RF component is mated with a connector that has a center pin that is too long in respect to the reference plane of the connector, damage can occur to the RF component connector.



The center pin of RF component connectors has a precision tolerance measured in mils (1/1000 inch). Connectors on test devices that mate with RF components may not be precision types and may not have the proper depth. Therefore, measure these connectors before mating to ensure suitability. When gauging pin depth, if the test device connector measures outside the tolerances listed in the following table, the center pin is too long. These connections can damage the termination connector.

Port/Connector Type	Pin Depth (mils) (0.000 in)	Gauge Setting
N Male	0.207 in –0.000 in +0.001 in	0.207 +0.000 –0.001
N Female	0.207 in +0.000 in –0.001 in	Same as pin depth
3.5 mm Male	–0.0002 in –0.0007 in	Same as pin depth
3.5 mm Female		
K Male	–0.0002 in –0.0007 in	Same as pin depth
K Female		
V Male	–0.0002 in –0.0010 in	Same as pin depth
V Female		

On the other hand, if the test device connector measures out of tolerance in the "–" region, the center pin is too short. While this connection does not cause any damage, it results in a poor connection and a consequent degradation in performance.

Avoid Lateral Pressure

Never put lateral pressure on the center pin of the connector.

Always Torque Connectors to 0.9 N • m (8 in • lb) Using a Torque Wrench

Do not overtorque connectors; doing so may damage the connector center pin. Always use a 0.9 N · m (8 in· lb) torque wrench when tightening 2.92 mm, 3.5 mm, and SMA

connectors. Finger-tighten N-type connectors. Never use pliers.

Teflon Tuning Washers

The center conductor on most RF components contains a small teflon tuning washer. This washer compensates for minor impedance discontinuities at the interface. The washer location is critical to the RF component performance.

	**Caution** Do not disturb the teflon tuning washer in connectors.

Avoid Mechanical Shock

Precision connectors are designed to withstand years of normal bench handling. Handle the connectors carefully and avoid dropping them. Mechanical shock significantly reduces their service life.

Avoid Applying Excessive Power

The NI vector network analyzer is rated for a specific maximum continuous input power. Exceeding the maximum input power level permanently damages the internal components.

Connector Cleaning Instructions

Connector interfaces, especially the outer conductors on the SMA connectors, should be kept clean and free of dirt and other debris.

	**Note** NI does not recommend using cotton swabs to clean the NI vector network analyzer connectors. Cotton fibers can become lodged in the connector, causing damage or measurement errors.

The following information includes important tips for cleaning connectors:

- Use only pure isopropyl alcohol as a solvent.
- Always use an appropriate size foam swab.
- Gently move the foam swab around the center conductor.
- Never put lateral pressure on the connector's center pin.

- Verify that no foam or other foreign material remains in the connector after cleaning.
- Only dampen the foam swab. Do ***not*** saturate it.
- Pure, filtered, compressed air can be used to remove foreign particles and to dry the connector.
- Verify that the center pin has not been bent or damaged.

Store Connectors Properly

When not in use, keep the connectors covered with the dust cap included in the NI vector network analyzer shipping kit.

Uninstalling the Hardware

Complete the following steps to uninstall the hardware:

	**Hot Surface** If the module has been in use, it may exceed safe handling temperatures and cause burns. Allow the module to cool before removing it from the chassis.

	**Caution** Disconnect or disable any external RF, clock, or digital connections to the front panel. Applying external signals while the device is powered off may cause damage.

1. Disconnect any cables from the module front panel connectors.
2. Power off the chassis.
3. Ground yourself with a grounding strap or touch a grounded metal surface.
4. Unlatch the module by pushing down on the ejector handle.
5. Hold the module by the ejector handle and remove it from the slot.

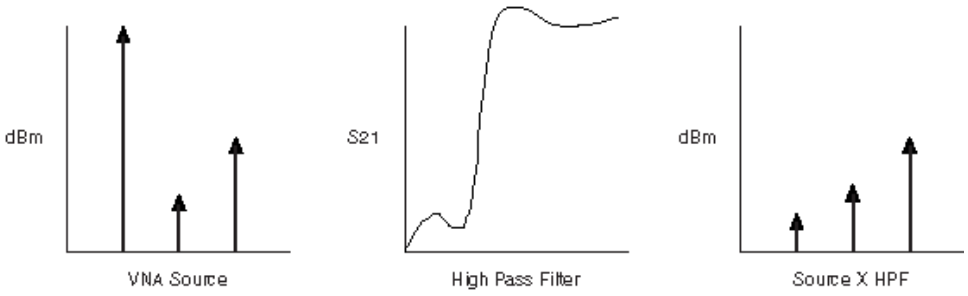
Store the module in the original antistatic packaging when not in use to avoid damage.

NI 5630 High-Sensitivity Mode

The NI PXIe-5630 VNA architecture is shown in the NI 5630 block diagram. The PORT 2 receiver can change the gain in the input RF path. This change allows the receiver to have a high-sensitivity mode, which extends the dynamic range of the VNA. In the high-sensitivity mode, the PORT 2 input path changes gain as a function of the PORT 2 receiver detected IF signal level. When the PORT 2 receiver detects that the

fundamental input signal level decreases below approximately –35 dBm, the gain is increased.

While this gain change increases the dynamic range of the receiver, the test conditions can also affect receiver operation. For example, the source or device under test (DUT) harmonics potentially impact the receiver path. A DUT can have high loss at the fundamental test frequency and less loss at other frequencies. For example, the following figure shows a VNA source with a fundamental frequency and harmonics as part of its spectrum. When the test signal goes through a highpass filter, the fundamental frequency is attenuated, and the higher harmonics are not impacted. The output signal from the filter (and thus the incident signal on the PORT 2 receiver) now has much higher harmonic content than the fundamental tone. This situation can result in a condition in which the PORT 2 broadband RF input path begins to compress due to the harmonic levels. In this situation, the result is distorted measurements over the frequency range that the compression exists. Increasing fundamental levels to reduce the input gain or decreasing harmonic levels would be required to reduce this condition.



Typical PORT 2 input levels that show compression are when any of the incident signal levels are in the –15 dBm range with the high-sensitivity mode enabled. Many potential DUTs do not exhibit the problem, but you may experience it in some situations. For example, if you are using a lowpass filter when the frequency source is above 3 GHz, there can be subharmonic levels from PORT 1. Also, the fundamental output frequencies below 1.5 GHz are achieved by dividing; therefore, they are richer in third harmonic content. To test for any potential non-linearity, adjust the output power level by a few dB and see if the measured response moves appreciably. If the system is in a linear (not limited) mode, changing the output power a few dB should not impact the measured response.

Accessories

National Instruments offers a variety of accessories to use with the NI 5630 vector network analyzer. The following table provides more information about these accessories.

Accessory	Part Number	Description
Precision Cables	781607-01 through 781614-01	High-quality, phase-stable coaxial cables that connect to the NI 5630 front panel K-type connector. These cables can be purchased in either 0.6 m or 1 m lengths with a variety of end connector types.
Phase-Equal Adapters	781597-01 to 781606-01	Precision 50 Ω insertable phase-equal adapters.
Automatic Calibration Kit	764134-01	Precision 50 Ω automatic calibration vector network analyzer calibration kit, 2.92 mm K-type connectors. This kit contains a precision automatic calibration unit, a USB Flash memory driver with calibration constants, insertable phase-equal adapters, a torque wrench, and a convenience wrench.
Manual Calibration Kit	764132-01 (K), 764133-01 (N)	A precision 50 Ω manual short, open, load, through (SOLT) vector network analyzer calibration kit, with 2.92 mm K-type or N-type connectors. This kit contains precision male and female shorts, opens, broadband loads, insertable phase-equal adapters, a torque wrench, and a convenience wrench.
Verification Kit	764135-01	Precision 50 Ω components with characteristics that are traceable to the National Institute of Standards and Technology (NIST). A memory device containing traceable factory measured test data for all components is supplied for comparison with customer-measured data.

For more information about these products, visit ni.com.

NI PXIe-5632

This section includes useful information about the NI PXIe-5632 (NI 5632) hardware, including module front panels, theory of operation, a block diagram, and calibration information.

The NI 5632 is a two-port vector network analyzer with a full S-Parameter test set based on a dual-source architecture. The NI 5632 has the following basic characteristics:

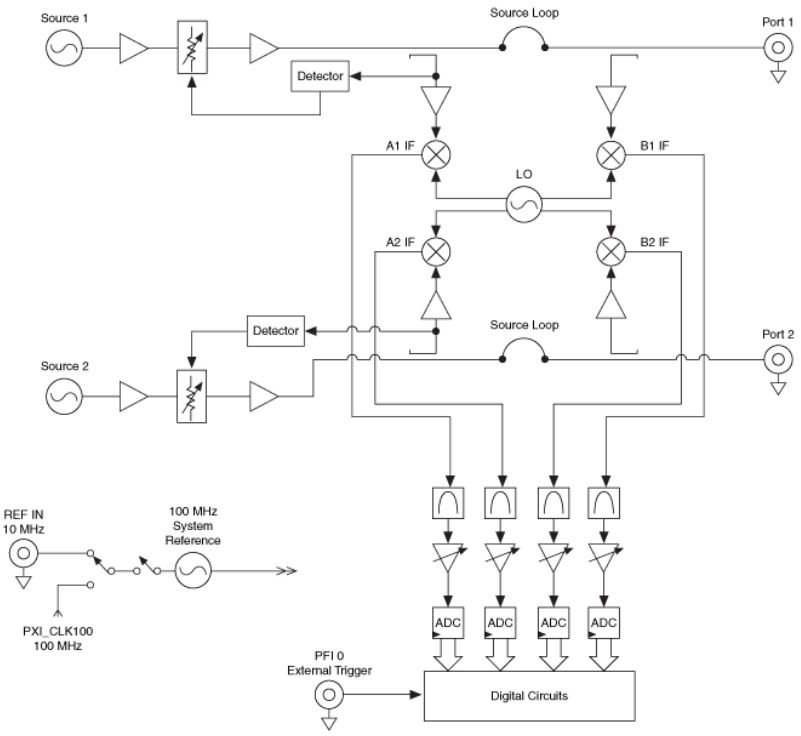
- Supports a frequency range from 300 kHz to 8.5 GHz
- Supports a power range from -30 dBm to +10 dBm in 0.01 dB steps
- Supports an IF bandwidth (IFBW) from 10 Hz to 500 kHz
- Can be programmed with the NI-VNA instrument driver or controlled interactively using the NI-VNA Soft Front Panel (SFP)
- Contained in a compact, three-slot-wide PXI Express 3U form factor

The NI 5632 follows industry-standard PlugPlay specifications for the PXI bus and seamless integration with compliant systems.

Note For detailed NI 5632 hardware specifications, refer to the ***NI PXIe-5632 Specifications*** document located at ni.com/manuals for updated documentation resources.

NI 5632 Block Diagram and Theory of Operation

The NI PXIe-5632 (NI 5632) is a two-port VNA capable of measuring forward and reverse S-parameters. A simplified block diagram is shown in the following figure.



The NI 5632 system architecture is based on a dual-source design. The PORT 1 output is generated by one source chain, while the PORT 2 output is supplied by a second source chain. The frequency synthesis of each source is based on voltage-controlled oscillators (VCOs) that are phase-locked to a direct digital synthesis (DDS) circuit. The DDS circuits enable the sources to achieve fine frequency tuning resolution. For frequencies above 2.5 GHz, the VCOs are used directly and are not multiplied. Frequencies below 2.5 GHz are created using heterodyne mixing.

Both sources are locked to a 100 MHz system reference oscillator. If an external 10 MHz input is injected, the system can be locked to that external reference. You also can use the PXI Express backplane 100 MHz reference directly as the VNA system reference.

The power output of each source is controllable through a variable amplitude modulator. An RF level detector and automatic level control (ALC) circuit work to maintain the user-defined power. The power level of each source can be controlled to deliver –30 dBm to +15 dBm in steps of 0.01 dB.

The PORT 1 source signal passes through a reference bridge. The through path of the bridge feeds the port output path, while the coupled arm of the bridge results in the A_1 reference signal. The VNA then uses a complimentary test bridge to measure the signal coming into PORT 1. The through path of this bridge connects the source output chain

to PORT 1, while the coupled arm measures the B_1 test signal. PORT 2 mirrors the PORT 1 chain. It has two bridges that transmit the second source output signal to PORT 2, while coupling off the A_2 reference signal and B_2 test signals. The VNA uses these four signals (A_1, A_2, B_1, and B_2) to calculate the S-parameters of the DUT.

Between each set of reference and test bridges, there is a source loop. The PORT 1 and PORT 2 output paths can each be accessed by the removal of their source loop. These loops allow a means of inserting an external device, such as an attenuator or modulator, into the incident signal path.

The receiver architecture is based upon broadband mixers. Four mixers (two reference and two test) are driven by a single local oscillator (LO). The LO synthesis occurs in the same manner as the source signals, using VCOs and DDS circuits. Unlike the sources, however, the LO does not have an ALC. You do not control its output power, instead, NI-VNA sets the output power to levels appropriate for driving the mixers based on the application.

The resulting intermediate frequency (IF) signals are filtered, amplified, and digitized by ADCs. Four separate IF chains exist for the four different nonratioed raw parameters. The multiple ADC outputs are sent to the digital circuits for detection processing. Receiver IF bandwidth control is accomplished using digital signal processing and optimizes receiver dynamic range versus sweep speed for a particular measurement. You can use NI-VNA to configure the IF bandwidth of the device, using the IF Bandwidth property or the `NIVNA_ATTR_IF_BANDWIDTH` attribute. Increasing the IF bandwidth improves the speed of the sweep measurement, but it introduces more noise. Decreasing the IF bandwidth yields more accurate measurements but at a lower speed.

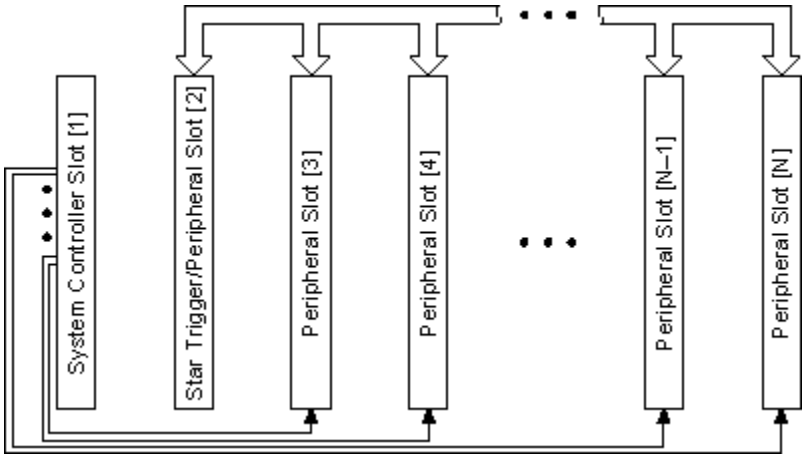
The output power is configurable using the Output Power property or the `NIVNA_ATTR_OUTPUT_POWER` attribute. You can set the output power to -30 dBm to +15 dBm, in 0.01 dB increments. Higher power on the output port increases dynamic range, but you should maintain active devices in a linear operating point and prevent those devices from compressing the VNA receiver. Reduce port power to account for the gain of such devices.

In addition to the PORT 1 and PORT 2 connectors on the NI 5632 front panel, there is a connector for an external frequency reference input (REF IN) and one for an external trigger (PFI 0) that you can use to initiate a sweep.

The system is calibrated at production time for frequency accuracy, power level accuracy, and the gain changes in the RF and IF paths. User-defined calibrations are required to achieve specified measurement performance.

PXI Trigger Lines

Eight PXI bus trigger lines are highly flexible and can be used in a variety of ways. For example, triggers can be used to synchronize the operation of several different PXI peripheral devices. In other applications, one device can control carefully timed sequences of operations performed on other devices in the system. Triggers may be passed from one device to another, allowing precisely timed responses to asynchronous external events that are being monitored or controlled. The number of triggers that a particular application requires varies with the complexity and number of events involved.



The PXI Specification is implemented with the RTSI bus through the PXI trigger lines. PXI Specification requires eight lines, PXI_Trig<0..7>, on the P2/J2 connector of the PXI chassis for the trigger lines. The RTSI features of NI hardware is implemented on this sub-bus. The RTSI triggers <0..6> are implemented on PXI_Trig<0..6>, and the RTSI clock is routed on PXI_Trig7.

System Reference Clock

The PXI chassis supplies the PXI 10 MHz system Reference Clock signal (PXI_CLK10) independently to each peripheral slot. An independent buffer drives the clock signal to each peripheral slot. The buffer has a source impedance matched to the backplane and a skew ranging from less than 1 ns to better than 250 ps between slots. You can

use this common Reference Clock signal to synchronize multiple devices in a measurement or control system. You can drive PXI_CLK10 from an external source through the PXI_CLK10_IN pin on the P2 connector of the PXI star trigger slot, which is Slot 2. Sourcing an external clock on this pin automatically disables the 10 MHz source on the backplane. You can synchronize multiple chassis that have connectors on the back panel for 10 MHz reference INPUT and 10 MHz reference OUTPUT. Refer to your PXI chassis documentation for more information.

Note NI-VNA devices do not work as PXI 10 MHz system clock signal sources.

Other Reference Clock Options

With the NI PXIe-5632 device, you can also use the PXI Express 100 MHz reference input (PXI_CLK100) on the NI 5632 PXI Express backplane directly as the VNA system Reference Clock.

To specify the PXI 100 MHz reference signal as the system Reference Clock, use the Reference Frequency Source property or the `NIVNA_ATTR_REFERENCE_FREQUENCY_SOURCE` attribute.

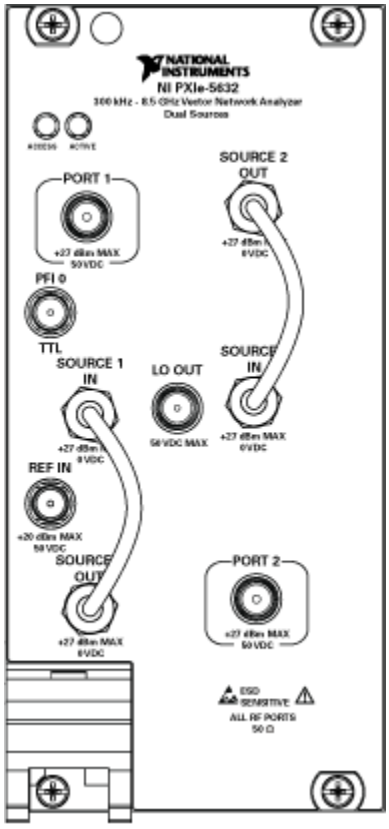
PFI Lines

PFI lines are multipurpose programmable function input/output lines. These lines serve as connections to virtually all internal timing signals. NI vector network analyzers have up to seven digital lines that can accept or generate a trigger, generate a marker, accept or generate a Reference Clock. The function of each PFI line is independent.

Caution If you enable a PFI line for output, do ***not*** connect any external signal source to it; doing so can damage the device, the computer, and the connected equipment.

NI 5632 Front Panel and Connectors

This section describes the connectors and LED indicators on the NI 5632 module front panel.



Connector	Type	Function
PORT 1	Type-K	Input/output terminal for NI 5632 measurements.
SOURCE 1 IN	SMA	Input terminal for the source 1 access loop.
SOURCE 2 IN	SMA	Input terminal for the source 2 access loop.
REF IN	SMA	Input terminal for a 10 MHz external reference signal for the NI 5632.
PFI 0	SMA	Input terminal for an external trigger signal.
LO OUT	SMA	Output terminal for the NI 5632 local oscillator.
SOURCE 1 OUT	SMA	Output terminal for the source 1 access loop.
SOURCE 2 OUT	SMA	Output terminal for the source 2 access loop.
PORT 2	Type-K	Input/output terminal for NI 5632 measurements.

The following table provides LED and indications information for the NI 5632 front panel LEDs.

LED	Indication
ACCESS	Indicates the basic hardware status of the NI 5632 module. • OFF—The module is not yet functional. • AMBER—The module is being accessed. Accessed means that the device is being communicated with over the PXI Express bus. • GREEN—The module is ready to be programmed by NI-VNA. • RED—The module has detected a hardware error, such as a hardware failure. The LED remains red until

	the error condition is removed.
ACTIVE	Indicates the state of the NI 5632 module. • OFF—The module is not yet functional. • AMBER—The module is armed and waiting for a trigger. • GREEN—The module has received a Start Trigger. This state also indicates that the module is making a measurement. • RED—The module has detected a spurious error, such as when the PLL becomes unlocked. The LED remains red until the error condition is removed.

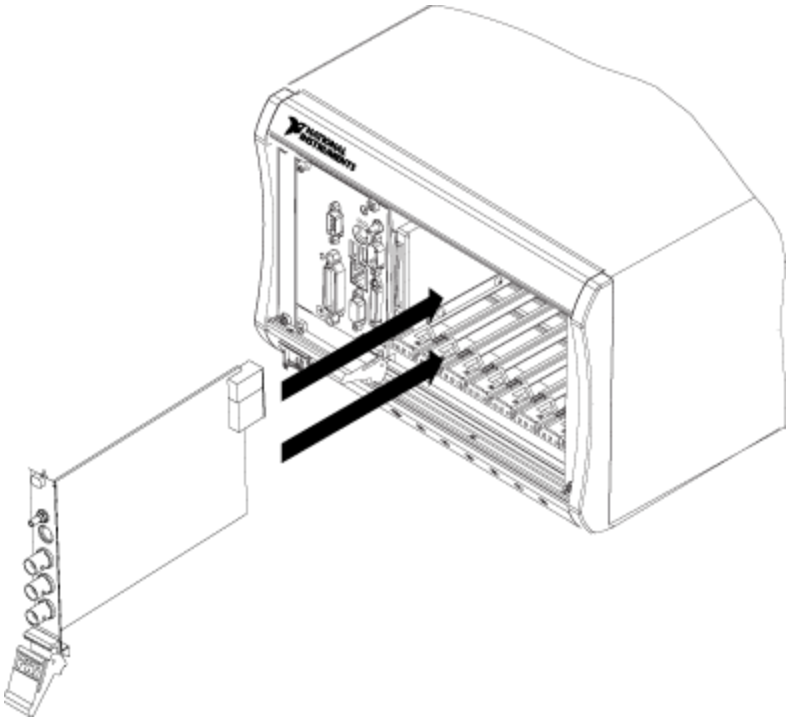
Hardware Integration and Maintenance

This section contains information about integrating and maintaining NI vector network analyzers in a PXI Express-based measurement system.

PXI Express Modules

The PXI Express Specification integrates PCI Express signaling into the PXI standard, which increases backplane bandwidth and enhances PXI timing and synchronization features by incorporating a 100 MHz differential Reference Clock and differential triggers. The PXI Express specification adds these features to PXI while maintaining backward compatibility.

All NI vector network analyzers exist in the PXI Express form factor.



Chassis Guidelines

NI PXI Express vector network analyzers can be installed in the following PXI Express chassis slots:

- **PXI hybrid slots**—Accepts either PXI modules that are hybrid slot-compatible or PXI Express modules
- **PXI Express slots**—Accepts PXI Express modules

Note Refer to the documentation for your PXI Express device and chassis for more information about installing and configuring PXI Express modules.

Maintaining PXI Express Hardware

Clean the fan filters on the chassis regularly to prevent fan blockage and to ensure efficient air circulation. Cleaning frequency depends on the amount of use and the operating environment. For specific information about cleaning procedures and other recommended maintenance, refer to the device specifications and the chassis user documentation.

Connector Care

Proper use and care of the connectors on your vector network analyzer is important. Improper connections can result in damage to your device and invalid measurements. Follow the guidelines in this section to prevent damage and to make the best possible measurements.

Caution NI is ***not*** liable for damage to NI VNA devices that results from improper connector care and signal connections.

Tighten Connectors Without Rotating the Center Pin

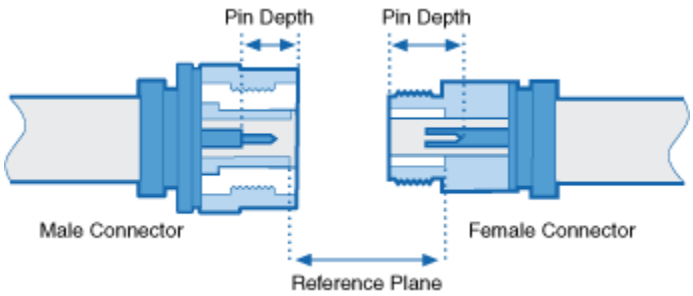
Never tighten the connectors, adapters, or cables such that the center pin rotates.

Avoid Touching Connectors with Bare Hands

Avoid touching Type-K connector mating planes with bare hands. Natural skin oils and microscopic dirt particles are difficult to remove.

Observe Pin Depth of Mating Connectors

Make sure to use the proper connectors to mate with the NI vector network analyzer. NI recommends that you use precision or instrument or metrology grade Type-K, 2.92 mm, 3.5 mm, or SMA connectors. Before mating the connector with another device or cable, measure the pin depth of the device that you want to mate with the RF component, using a pin depth gauge. Improper pin depth of mating connectors is a major cause of failure in the field. When an RF component is mated with a connector that has a center pin that is too long in respect to the reference plane of the connector, damage can occur to the RF component connector.



The center pin of RF component connectors has a precision tolerance measured in mils (1/1000 inch). Connectors on test devices that mate with RF components may not be precision types and may not have the proper depth. Therefore, measure these connectors before mating to ensure suitability. When gauging pin depth, if the test device connector measures outside the tolerances listed in the following table, the center pin is too long. These connections can damage the termination connector.

Port/Connector Type	Pin Depth (mils) (0.000 in)	Gauge Setting
N Male	0.207 in –0.000 in +0.001 in	0.207 +0.000 –0.001
N Female	0.207 in +0.000 in –0.001 in	Same as pin depth
3.5 mm Male	–0.0002 in –0.0007 in	Same as pin depth
3.5 mm Female		
K Male	–0.0002 in –0.0007 in	Same as pin depth
K Female		
V Male	–0.0002 in –0.0010 in	Same as pin depth
V Female		

On the other hand, if the test device connector measures out of tolerance in the "–" region, the center pin is too short. While this connection does not cause any damage, it results in a poor connection and a consequent degradation in performance.

Avoid Lateral Pressure

Never put lateral pressure on the center pin of the connector.

Always Torque Connectors to 0.9 N • m (8 in • lb) Using a Torque Wrench

Do not overtorque connectors; doing so may damage the connector center pin. Always use a 0.9 N · m (8 in· lb) torque wrench when tightening 2.92 mm, 3.5 mm, and SMA

connectors. Finger-tighten N-type connectors. Never use pliers.

Teflon Tuning Washers

The center conductor on most RF components contains a small teflon tuning washer. This washer compensates for minor impedance discontinuities at the interface. The washer location is critical to the RF component performance.



Caution Do not disturb the teflon tuning washer in connectors.

Avoid Mechanical Shock

Precision connectors are designed to withstand years of normal bench handling. Handle the connectors carefully and avoid dropping them. Mechanical shock significantly reduces their service life.

Avoid Applying Excessive Power

The NI vector network analyzer is rated for a specific maximum continuous input power. Exceeding the maximum input power level permanently damages the internal components.

Connector Cleaning Instructions

Connector interfaces, especially the outer conductors on the SMA connectors, should be kept clean and free of dirt and other debris.



Note NI does not recommend using cotton swabs to clean the NI vector network analyzer connectors. Cotton fibers can become lodged in the connector, causing damage or measurement errors.

The following information includes important tips for cleaning connectors:

- Use only pure isopropyl alcohol as a solvent.
- Always use an appropriate size foam swab.
- Gently move the foam swab around the center conductor.
- Never put lateral pressure on the connector's center pin.

- Verify that no foam or other foreign material remains in the connector after cleaning.
- Only dampen the foam swab. Do ***not*** saturate it.
- Pure, filtered, compressed air can be used to remove foreign particles and to dry the connector.
- Verify that the center pin has not been bent or damaged.

Store Connectors Properly

When not in use, keep the connectors covered with the dust cap included in the NI vector network analyzer shipping kit.

Uninstalling the Hardware

Complete the following steps to uninstall the hardware:



Hot Surface If the module has been in use, it may exceed safe handling temperatures and cause burns. Allow the module to cool before removing it from the chassis.



Caution Disconnect or disable any external RF, clock, or digital connections to the front panel. Applying external signals while the device is powered off may cause damage.

1. Disconnect any cables from the module front panel connectors.
2. Power off the chassis.
3. Ground yourself with a grounding strap or touch a grounded metal surface.
4. Unlatch the module by pushing down on the ejector handle.
5. Hold the module by the ejector handle and remove it from the slot.

Store the module in the original antistatic packaging when not in use to avoid damage.

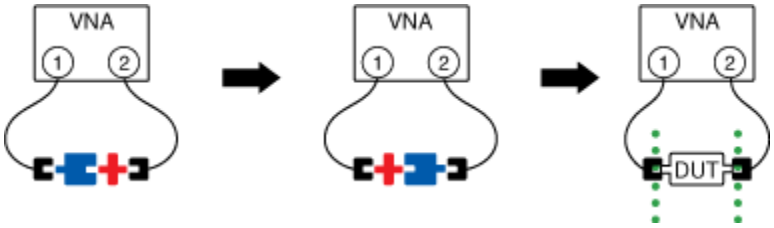
NI 5632 Adapter Removal Calibration

By performing adapter removal calibration, you can remove the effect of an adapter from the system when measuring a noninsertable device under test (DUT). A noninsertable DUT is a device with different connector types or connectors of the same gender that cannot mate together. To measure a noninsertable device, use cables with corresponding connectors that can mate to the device. Performing adapter removal

calibration is useful when you have mismatched in and out cable connectors, between the VNA and the device under test (DUT), and you must add an adapter with the same connectors as the DUT to proceed with your measurement.

Note When measuring a noninsertable device, you cannot calibrate to the end of the cables on the VNA because the connectors on the cables from the VNA cannot mate. Therefore, you cannot measure a through standard, which is common in SOLT calibrations.

To perform adapter removal calibration and remove the mismatch introduced by the adapter, you must perform two full two-port calibrations. As shown in the following figure, perform one calibration with the adapter on PORT 1 and a second calibration with the adapter on PORT 2 of the VNA. The two calibrations then combine, which produces a new set of 12-term error corrections. The error corrections apply one calibration that moves the reference plane to the ends of the cables where they connect to the DUT.



In the preceding figure, the blue device represents the AutoCal module, the red device represents the adapter, and the green dotted lines represent the location of reference plane after the calibration is complete.

To perform adapter removal calibration, the following items are required:

- An adapter that mates with both sides of the DUT. The electrical length of the adapter must be known to within 1/4 wavelength of the highest calibration frequency.
- Calibration standards that mate to each side of the adapter and allow a full two-port calibration.

Adapter removal calibration is available for both manual calibration and AutoCal. Using AutoCal, adapter removal calibration can only counter gender mismatch. If you perform manual calibration and the calibration standards are available in both connector types, you can correct connector mismatch and gender mismatch.

NI 5632 De-Embedding

The NI PXIe-5632 (NI 5632) includes a de-embedding feature, which is most commonly used to remove the effects of test fixtures between the VNA and the device under test (DUT). In general, the term ***de-embedding*** refers to the process of removing network or test fixture data that is introduced to your S-parameter measurements from final measured data. Although you can use certain calibration methods, including TRL calibration, to remove the effects of a test fixture, de-embedding is useful, specifically if you know the characterization model of the fixture.

You can set up de-embedding by specifying multiple networks on each port on the VNA. The networks you place on each of the ports are completely independent of each other, and the first network you add to either port should be the one immediately adjacent to the DUT.

The NI 5632 supports de-embedding of the various networks defined in the following table.

Network Name	NI-VNA Soft Front Panel (SFP) Control	Description
Inductive Circuit	L Circuit	The network to be de-embedded consists of a parallel or series inductance, defined by a manually entered inductance value, in nH.
Capacitive Circuit	C Circuit	The network to be de-embedded consists of a parallel or series capacitance, defined by a manually entered capacitance value, in pF.
Resistive Circuit	R Circuit	The network to be de-embedded consists of a parallel or series resistance, defined by a manually entered resistance value, in ohms.
Transmission Line	Trans. Line	The network to be de-embedded consists of a transmission line defined by an impedance, in ohms, a length, in mm, a loss at a specific frequency, in dB/mm at GHz, and dielectric properties.
S-Parameter File	S2P File	The network to be de-embedded is a generic black box defined entirely by a user-loaded S2P file, which contains the S-parameter data for that network. The S-parameter data fully defines the properties of the network and allows for de-embedding of more generic networks beyond the four

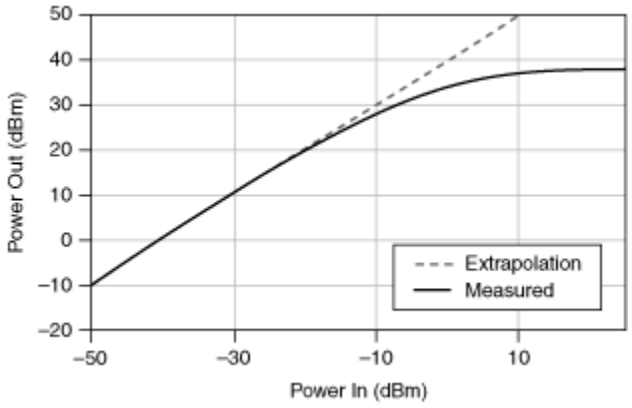
		types of networks listed above.

NI 5632 Power Sweeps

The NI PXIe-5632 (NI 5632) supports measurement sweep types including traditional frequency sweeps, segmented frequency sweeps, and power sweeps. ***Power sweeps*** describe a sweep of the port power defined by a start value, stop value, and number of power points. The frequency is held constant across the power sweep.

Gain compression is a common power sweep application. The 1 dB compression point is derived from the gain relationship between output power and input power. An amplifier, for example, maintains a constant gain for low-level input signals, but at higher input levels the amplifier goes into saturation, and its gain decreases. The 1 dB compression point indicates the power level that causes the gain to drop by 1 dB from its small signal value. You can determine the 1 dB compression point by measuring S_{21} or by measuring the raw receiver data.

A typical power sweep measurement, gain compression, is shown in the following figure.



Power sweeps help measure gain linearity of an amplifier or other DUT. Source power calibrations and receiver calibrations are required for high absolute accuracy compression measurements.

NI 5632 Segmented Sweep

Segmented sweep allows you to break the sweep into individual segments, with each

segment containing a different number of frequency points, power level, and intermediate frequency bandwidth (IFBW). You can independently control measurement settings for each segment, including IFBW, averaging, and port 1 and port 2 power settings. Controlling these settings can be useful when DUTs require different settings at various frequencies or when making measurements within specific frequency bands. Segmented sweeps can also provide customization of dynamic range and measurement speed based on the DUT.

The NI PXIe-5632 (NI 5632) supports both frequency-based and index-based segmented sweep types.

- ***Frequency-based*** segmented sweeps have individual frequencies defined that consistently increase within or between segments. The resulting plots can be based on the frequency or the index of the particular point. This mode is more common than index-based sweeps.
- Unlike frequency-based segmented sweeps, ***index-based*** segmented sweeps can have individual frequencies defined in any order within or between segments. The resulting plots refer to the point index rather than frequency because of the potential direction changes. This mode is often used when measuring reverse sweeps and when specific frequencies must be measured before others.

You can configure segmented sweep for the NI 5632 through the NI-VNA Soft Front Panel (SFP).

NI 5632 Individual Control of Source and Receiver Frequencies

You can individually control the NI PXIe-5632 receivers and two internal sources. You can perform the following specialized measurements using this feature because some restrictions on frequency associations can be removed.

- Mixers
- Frequency multipliers
- Dividers
- Harmonics/spurious
- Intermodulation Distortion (IMD)

Note You can perform the preceding specialized measurements by creating a custom

	application.

NI 5632 Raw Receiver Measurements

The NI PXIe-5632 (NI 5632) has four internal receivers (A_1, B_1, A_2, and B_2) that measure incident, reflected, and transmitted voltages between the VNA and DUT. The NI 5632 uses the four receiver measurements during calculation of the ratioed S-parameters. If you use the raw or actual receiver data instead of the S-parameters, the following measurements can be made:

- DUT raw output (unratioed)
- VNA raw performance
- Custom measurement configuration
- Application of error correction algorithms outside of the VNA driver

Raw receiver data can be useful for performing other specialized measurements such as gain compression and intermodulation distortion.

	**Note** Error corrections are not applied to raw receiver measurements, but source power correction or receiver power correction can be applied if necessary.

Accessories

National Instruments offers a variety of accessories to use with the NI 5632 vector network analyzer. The following table provides more information about these accessories.

Accessory	Part Number	Description
Precision Cables	781607-01 through 781614-01	High-quality, phase-stable coaxial cables that connect to the NI 5632 front panel K-type connector. These cables can be purchased in either 0.6 m or 1 m lengths with a variety of end connector types.
Phase-Equal Adapters	781597-01 to 781606-01	Precision 50 Ω insertable phase-equal adapters.
Automatic	782364-01	Precision 50 Ω automatic calibration vector network analyzer calibration

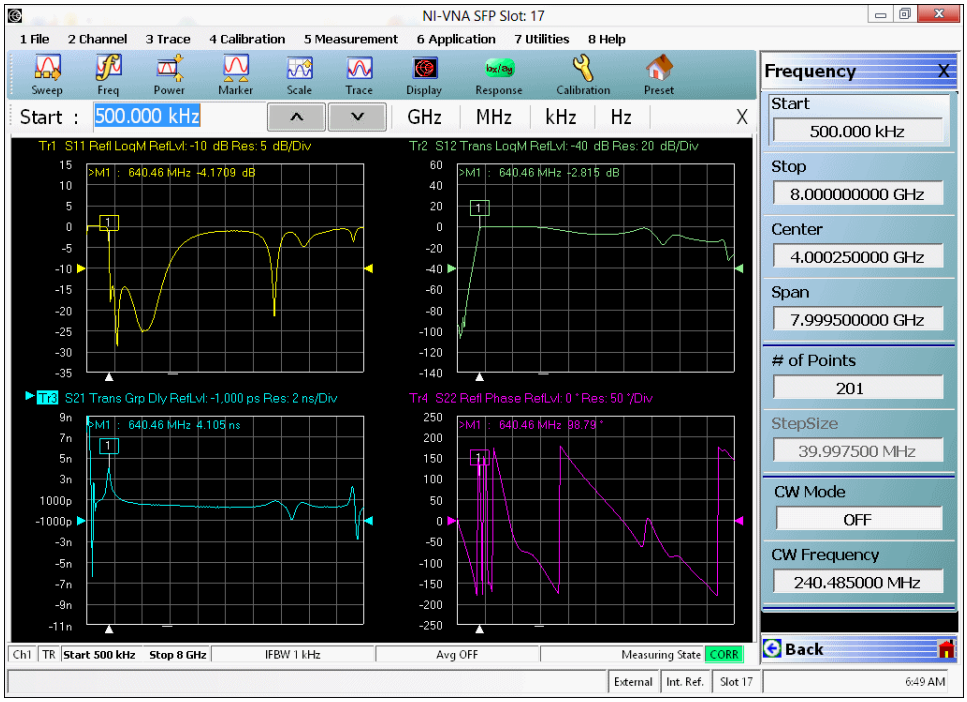
Calibration Kit	(M-F), 782364-02 (M-M), 782364-03 (F-F)	kit, 2.92 mm K-type connectors. This kit contains a precision automatic calibration unit, a USB Flash memory driver with calibration constants, insertable phase-equal adapters, a torque wrench, and a convenience wrench.
Manual Calibration Kit	781587-01 (K), 781588-01 (N)	A precision 50 Ω manual short, open, load, through (SOLT) vector network analyzer calibration kit, with 2.92 mm K-type or N-type connectors. This kit contains precision male and female shorts, opens, broadband loads, insertable phase-equal adapters, a torque wrench, and a convenience wrench.
Verification Kit	781590-01	Precision 50 Ω components with characteristics that are traceable to the National Institute of Standards and Technology (NIST). A memory device containing traceable factory measured test data for all components is supplied for comparison with customer-measured data.

For more information about these products, visit ni.com.

NI-VNA Soft Front Panel

The NI-VNA Soft Front Panel (SFP) controls the NI RF vector network analyzer and displays corresponding device measurements. You can access the NI-VNA SFP at **Start»All Programs»National Instruments»NI-VNA»NIVNA Soft Front Panel**.

The NI-VNA SFP is shown in the following figure, and the portions of the NI-VNA SFP are explained in more detail in the following topics.



The NI-VNA SFP includes the following main sections:

- Menu
- Toolbar
- Display Window
- Softkey Menu

NI-VNA SFP Menu

The NI-VNA SFP Menu spans the top of the SFP window. These menus allow you direct access to system menus and some button functions. The Menu is shown in the following figure.



File Menu

The File menu provides the following options:

- Recall Setup—Recalls previously saved setup and/or calibration files.
- Recall Data—Recalls previously saved active channel and/or active trace data files.
- Save Setup—Saves the currently applied system presets configuration file.

- Save Data—Saves the current channel data file.
- Print—Prints the current main display.
- Exit—Allows you to exit the NI-VNA SFP.

Channel Menu

The Channel menu provides the following options:

- Frequency—Displays frequency configuration options in the Settings Panel.
- Power—Displays power configuration options in the Settings Panel.
- Sweep—Displays sweep configuration options in the Settings Panel.
- Averaging—Displays averaging configuration options in the Settings Panel.

Trace Menu

The Trace menu provides the following options:

- Response—Displays response configuration options in the Settings Panel.
- Display—Displays display configuration options in the Settings Panel.
- Scale—Displays scale configuration options in the Settings Panel. The scale configuration options that are available are determined by the display type you select.
- Marker—Displays marker configuration options in the Settings Panel.
- Trace Menu—Displays trace configuration options in the Settings Panel.
- Trace Max—Maximizes the active trace for full-screen display.
- Trace Previous—Maximizes the previous trace for full-screen display.
- Trace Next—Maximizes the next trace for full-screen display.

Calibration Menu

The Calibration menu displays Calibration options in the Settings Panel.

Measurement Menu

The Measurement menu displays Measurement options in the Settings Panel.

Application Menu

The Application menu displays Application options in the Settings Panel.

Utilities Menu

The Utilities menu provides the following options:

- System—Displays system configuration options in the Settings Panel.
- Toolbar Off—Disables display of the NI-VNA SFP Toolbar. Select this option again to enable Toolbar display.
- Customize Toolbar—Launches the Customize Toolbar dialog box, which allows you to customize the NI-VNA SFP Toolbar.
- Preset—Restores the current active preset configuration, which is the instrument state at the last preset save.
- Clear—Disables display of the NI-VNA SFP Settings Panel. Select this option again to enable Settings Panel display.

Help Menu

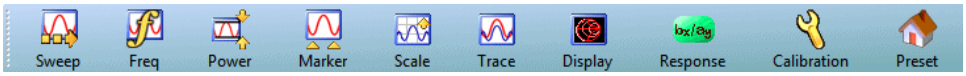
The Help menu provides the following options:

- Browse Help—Launches this help file.
- About NI-VNA—Displays basic information about the installed NI-VNA software version.

NI-VNA SFP Toolbar

The NI-VNA SFP Toolbar appears below the NI-VNA SFP Menu. These buttons allow you quick access to sets of configuration options that appear in the NI-VNA SFP Softkey Menu. The name of the toolbar button corresponds with the set of options that appear in the Softkey Menu.

The NI-VNA SFP Toolbar is shown in the following figure.



Note The **Preset** button does not load a set of configuration options in the Softkey Menu. This button restores the current active preset configuration, which is the instrument state at the last preset save.

Customizing the Toolbar

By selecting **Utilities»Customize Toolbar** from the NI-VNA SFP Menu, you can add additional buttons to the toolbar. These buttons perform common functions of the NI-VNA SFP or load sets of configuration options in the NI-VNA SFP Softkey Menu. You also can perform the following toolbar functions:

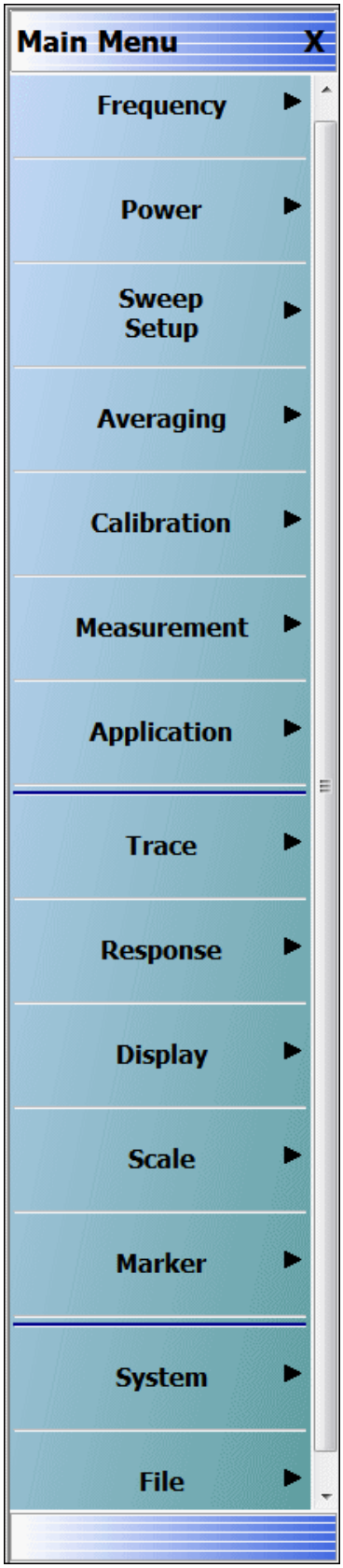
- Remove buttons from the toolbar
- Move icons on the toolbar
- Reset the toolbar to its default state

After you modify the toolbar, selecting **File»Save Setup** saves the current toolbar configuration.

NI-VNA SFP Softkey Menu

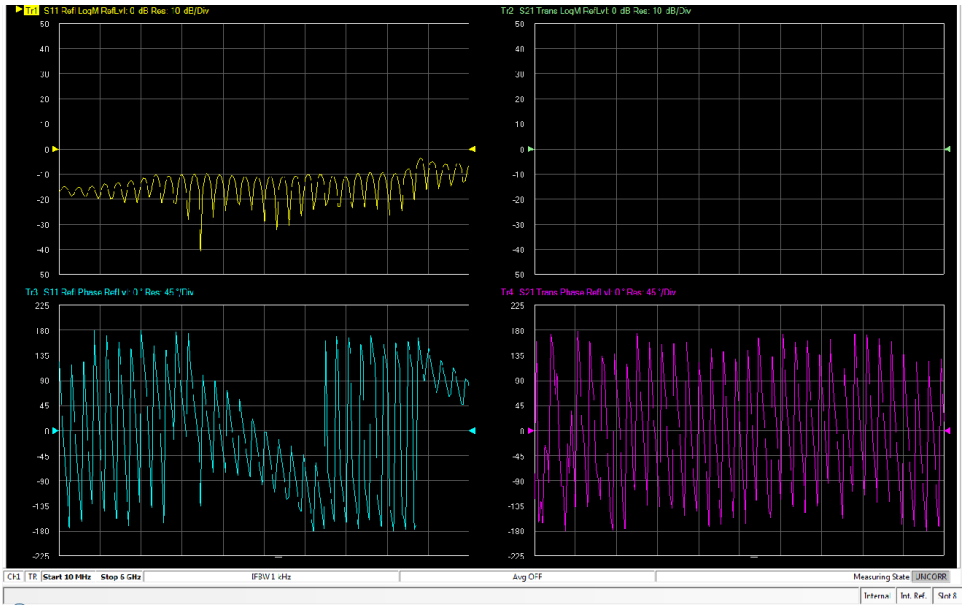
The NI-VNA SFP Softkey Menu spans the right side of the SFP window. This Softkey Menu displays lower-level submenus or configuration options that correspond to the menu options you select.

When you first launch the NI-VNA SFP, the Softkey Menu displays the Main Menu, which is shown in the following figure. This menu changes to show relevant controls according to which Menu or Toolbar option you select.


Main Menu
X
Frequency
Power
Sweep Setup
Averaging
Calibration
Measurement
Application
Trace
Response
Display
Scale
Marker
System
File

NI-VNA SFP Display Window

The NI-VNA Display Window is the largest portion of the NI-VNA SFP. The main area displays the plots that you configure for display, and the area below the plots displays important information regarding the system configuration.



Performing an Automatic SOLT Calibration (AutoCal) Using the NI-VNA SFP

This section describes how to complete an AutoCal of the NI vector network analyzer using an AutoCal module and the NI-VNA SFP. You can perform an AutoCal on the following devices:

- NI PXIe-5630
- NI PXIe-5632

Performing an Automatic SOLT Calibration (AutoCal) of the NI 5630 Using the NI-VNA SFP

⚠	**Caution** Use caution when attaching and detaching cables to and from the NI 5630, the AutoCal module, and manual calibration standards (open, short, load). A damaged connector on these items renders them unusable and necessitates repair or replacement. Always use an 0.9 N · m (8 in-lb) torque wrench for attaching cables to the NI 5630 module, the AutoCal

	module, and the manual calibration standards. Refer to Connector Care for more information about making proper signal connections.

This topic describes how to complete an AutoCal of the NI 5630 using an AutoCal module and the NI-VNA SFP. To perform an AutoCal, complete the following steps:

1. Connect the AutoCal module power supply to a power source. The green Power LED should be lit.
2. Wait for the internal heating element inside the AutoCal module to stabilize the module's internal temperature. When ready, the blue Operate LED on the AutoCal module illuminates.
3. Connect the serial cable to the AutoCal module and the PXI Express embedded controller serial port.
4. Attach flexible, phase-stable cables with K connectors to PORT 1 and PORT 2 on the NI 5630 module front panel. These cables should be of opposite gender to the DUT connector ends to make the DUT insertable.
5. Attach the DUT end of the cables to the AutoCal module.
6. Launch the NI-VNA SFP. When you launch the NI-VNA SFP, the Measuring State indicator at the bottom of the Display Window shows UNCORR. This status indicates that the NI 5630 is initially uncalibrated and is not applying correction factors.
7. Load the AutoCal module characterization file from the USB Flash memory drive. This file contains the information about the AutoCal module's internal standards. It has a filename of serialnumber . ACD, where serialnumber is the serial number of the AutoCal module, printed on the bottom of the module.

	**Note** Each AutoCal characterization file is specific to an AutoCal module. You cannot use one module's characterization file with a different AutoCal module.

8. Select **Utilities»System** from the NI-VNA SFP Menu.
9. Select **Utility»AutoCal Characterization»Load Charac. File** from the NI-VNA SFP Softkey Menu.
10. Select your AutoCal module characterization file (serialnumber . ACD) in the Open dialog box.
11. Select **Channel»Frequency** from the Menu.
12. Specify values for the **Start Frequency**, **Stop Frequency**, and **Number of Points** controls in the NI-VNA SFP Softkey Menu.

Note Wider sweep spans and larger numbers of points result in longer sweep and calibration times.

13. Select **Channel»Power** from the Menu.
14. Specify the Port 1 output power level in the Softkey Menu.
15. Select **Channel»Averaging** from the Menu.
16. Specify the IF bandwidth and Averaging in the Softkey Menu.

Note Enabling averaging and choosing narrower IF bandwidths result in longer sweep and calibration times.

17. Specify the AutoCal setup by completing the following steps:
    1. Select **Calibration** from the NI-VNA SFP Menu.
    2. Select **Calibrate»AutoCal»1 Path 2 Port»Modify Cal Setup** from the Softkey Menu.

       Note You can choose **1 Port Cal** instead of **1 Path 2 Port** as shown in the preceding menu path if you want to make only one-port measurements.

    3. Select the appropriate **Through Setup** from the Modify 1-Path 2-Port AutoCal Setup dialog box:
       1. **Internal thru** is inside the AutoCal module and does not require you to detach the AutoCal module from the cables to complete the AutoCal.
       2. **True thru** requires you to connect the two NI 5630 cables together for the final step of the AutoCal.
    4. Specify the **Orientation** from the Modify 1-Path 2-Port AutoCal Setup dialog box. Port 1 is always the transmission port of the NI 5630. Orientation tells the AutoCal module which terminal is connected to the signal source (PORT 1).
    5. Click **OK** to exit the dialog box.
18. Select **Begin Cal** from the Softkey Menu.

To verify the calibration completed in the preceding procedure, the SFP should now show a Measuring State of CORR, which indicates that the module is applying correction factors. After disconnecting the AutoCal module and connecting the cables together, the S_{21} amplitude and phase response should be flat.

Note The cables are connected together either after the AutoCal is completed if you use an

	Internal Thru or during the last phase of the AutoCal if you use True Thru.

To store the calibration data, select **File»Save Setup** to save the sweep settings and current calibration correction factors to disk for recall later. To recall the settings later, select **File»Recall Setup**.

Performing an Automatic SOLT Calibration (AutoCal) of the NI 5632 Using the NI-VNA SFP

⚠	**Caution** Use caution when attaching and detaching cables to and from the NI 5632, the AutoCal module, and manual calibration standards (open, short, load). A damaged connector on these items renders them unusable and necessitates repair or replacement. Always use an 0.9 N · m (8 in-lb) torque wrench for attaching cables to the NI 5632 module, the AutoCal module, and the manual calibration standards. Refer to Connector Care for more information about making proper signal connections.

This topic describes how to complete an AutoCal of the NI 5632 using an AutoCal module and the NI-VNA SFP. To perform an AutoCal, complete the following steps:

1. Connect the AutoCal module power supply to a power source. The green Power LED should be lit.
2. Wait for the internal heating element inside the AutoCal module to stabilize the module's internal temperature. When ready, the blue Operate LED on the AutoCal module illuminates.
3. Connect the serial cable to the AutoCal module and the PXI Express embedded controller serial port.
4. Attach flexible, phase-stable cables with K connectors to PORT 1 and PORT 2 on the NI 5632 module front panel. These cables should be of opposite gender to the DUT connector ends to make the DUT insertable.
5. Attach the DUT end of the cables to the AutoCal module.
6. Launch the NI-VNA SFP. When you launch the NI-VNA SFP, the Measuring State indicator at the bottom of the Display Window shows UNCORR. This status indicates that the NI 5632 is initially uncalibrated and is not applying correction factors.
7. Load the AutoCal module characterization file from the USB Flash memory drive. This file contains the information about the AutoCal module's internal standards. It has a filename of serialnumber . ACD, where serialnumber is the serial number of

the AutoCal module, printed on the bottom of the module.

> **Note** Each AutoCal characterization file is specific to an AutoCal module. You cannot use one module's characterization file with a different AutoCal module.

8. Select **Utilities»System** from the NI-VNA SFP Menu.
9. Select **Utility»AutoCal Characterization»Load Charac. File** from the NI-VNA SFP Softkey Menu.
10. Select your AutoCal module characterization file (serialnumber `.ACD`) in the Open dialog box.
11. Select **Channel»Frequency** from the Menu.
12. Specify values for the **Start Frequency**, **Stop Frequency**, and **Number of Points** controls in the NI-VNA SFP Softkey Menu.

> **Note** Wider sweep spans and larger numbers of points result in longer sweep and calibration times.

13. Select **Channel»Power** from the Menu.
14. Specify the output power levels for Port 1 and Port 2 in the Softkey Menu.
15. Select **Channel»Sweep** from the Menu.
16. Specify the type of sweep in the Softkey Menu.

> **Note** If you choose **Segmented Sweep (Freq-based)** or **Segmented Sweep (Index-based)** sweep types, you must configure the corresponding sweep setup in the Sweep Setup Softkey Menu.

17. Select **Channel»Averaging** from the Menu.
18. Specify the IF bandwidth and Averaging in the Softkey Menu.

> **Note** Enabling averaging and choosing narrower IF bandwidths result in longer sweep and calibration times.

19. Specify the AutoCal setup by completing the following steps:
    1. Select **Calibration** from the NI-VNA SFP Menu.
    2. Select **Calibrate»AutoCal»2-Port Cal»Modify Cal Setup** from the Softkey Menu.

> **Note** You can choose **1 Port Cal** instead of **2-Port Cal** as shown in the preceding menu path if you want to make only one-port measurements.

3. Select the appropriate **Through Setup** from the Modify 2-Port AutoCal Setup dialog box:
    1. **Internal thru** is inside the AutoCal module and does not require you to detach the AutoCal module from the cables to complete the AutoCal.
    2. **True thru** requires you to connect the two NI 5632 cables together for the final step of the AutoCal.
4. Specify the orientation from the Modify 2-Port AutoCal Setup dialog box by selecting **Auto Sense Module Orientation** or manually specifying the orientation using the configuration options at the bottom of the dialog box.
5. Click **OK** to exit the dialog box.

20. Select **Begin Cal** from the Softkey Menu.

To verify the calibration completed in the preceding procedure, the SFP should now show a Measuring State of CORR, which indicates that the module is applying correction factors. After disconnecting the AutoCal module and connecting the cables together, the S_{21} amplitude and phase response should be flat.



Note The cables are connected together either after the AutoCal is completed if you use an Internal Thru or during the last phase of the AutoCal if you use True Thru.

To store the calibration data, select **File»Save Setup** to save the sweep settings and current calibration correction factors to disk for recall later. To recall the settings later, select **File»Recall Setup**.

Performing a Manual SOLT Calibration Using the NI-VNA SFP

This section describes how to complete a manual calibration of the NI vector network analyzer using manual calibration standards and the NI-VNA SFP. You can perform a manual calibration on the following devices:

- NI PXIe-5630
- NI PXIe-5632

Performing a Manual SOLT Calibration of the NI 5630 Using the NI-VNA SFP

Caution Use caution when attaching and detaching cables to and from the NI 5630, the AutoCal module, and manual calibration standards (open, short, load). A damaged connector on these items renders them unusable and necessitates repair or replacement. Always use an 0.9 N · m (8 in-lb) torque wrench for attaching cables to the NI 5630 module, the AutoCal module, and the manual calibration standards. Refer to Connector Care for more information about making proper signal connections.

This topic describes how to complete a manual calibration of the NI 5630 using manual calibration standards and the NI-VNA SFP. To perform a manual calibration, complete the following steps:

1. Attach flexible, phase-stable cables with K connectors to PORT 1 and PORT 2 on the NI 5630 module front panel. These cables should be of opposite gender to the DUT connector ends to make the DUT insertable.

   Note You will attach the manual calibration standards to the end of the PORT 1 cable later in this procedure, so ensure the gender of the DUT end of the PORT 1 cable is the opposite gender of the manual calibration standard connector.

2. Launch the NI-VNA SFP. When you launch the NI-VNA SFP, the Measuring State indicator at the bottom of the Display Window shows UNCORR. This status indicates that the NI 5630 is initially uncalibrated and is not applying correction factors.
3. Select **Channel»Frequency** from the NI-VNA SFP Menu.
4. Specify values for the **Start Frequency**, **Stop Frequency**, and **Number of Points** controls in the NI-VNA SFP Softkey Menu.

   Note Wider sweep spans and larger numbers of points result in longer sweep and calibration times.

5. Select **Channel»Power** from the Menu.
6. Specify the Port 1 output power level in the Softkey Menu.
7. Select **Channel»Averaging** from the Menu.
8. Specify the IF bandwidth and Averaging in the Softkey Menu.

Note Enabling averaging and choosing narrower IF bandwidths result in longer sweep and calibration times.

9. Specify the manual calibration setup by completing the following steps:
   1. Select **Calibration** from the NI-VNA SFP Menu.
   2. Select **Calibrate»Manual Cal»1 Path 2 Port»Modify Cal Setup»Edit Cal Params** from the Softkey Menu.

      Note You can choose **1 Port Cal** instead of **1 Path 2 Port** as shown in the preceding menu path if you want to make only one-port measurements.

   3. Select **Broadband Load** as the Load Type in the 1 Path 2 Port Setup dialog box.
   4. Specify the connector type that is present on the manual calibration standards in the DUT Connector control. For example, the manual calibration standard from Anritsu Model OSLK50 has male K connectors for the short, open, and load, so you would select **K-Conn(M)** in this case.
   5. Click **OK** to close the dialog box.
10. Click the **Back** button at the bottom of the Softkey Menu.
11. Select **Port 1 Reflective Devices** from the Softkey Menu.
12. Attach the Open manual calibration standard to the end of the cable attached to PORT 1 on the NI 5630 module front panel.

    Caution After you start the manual calibration, take care to move the VNA cables as little as possible throughout the calibration process and during ensuing usage.

13. Select **Open** from the Softkey Menu. When the NI-VNA SFP completes the measurement, a checkbox appears next to the Open command in the Softkey Menu.
14. Attach the Short manual calibration standard to the end of the cable attached to PORT 1 on the NI 5630 module front panel.
15. Select **Short** from the Softkey Menu. When the NI-VNA SFP completes the measurement, a checkbox appears next to the Short command in the Softkey Menu.
16. Attach the Load manual calibration standard to the end of the cable attached to PORT 1 on the NI 5630 module front panel.
17. Select **Load** from the Softkey Menu. When the NI-VNA SFP completes the measurement, a checkbox appears next to the Load command in the Softkey

Menu.

18. Click the **Back** button at the bottom of the Softkey Menu. A checkmark should now appear next to the Port 1 Reflective Devices item in the Softkey Menu.
19. Select **Thru** from the Softkey Menu.
20. Connect the PORT 1 and PORT 2 VNA cables together.
21. Select **Thru 1-2** from the Softkey Menu. When the NI-VNA SFP completes the measurement, a checkbox appears next to the Thru 1-2 item in the Softkey Menu.
22. Click the **Back** button at the bottom of the Softkey Menu.
23. (Optional) Select **Isolation (Optional)** from the Softkey Menu.
24. Attach the Load manual calibration standards to the ends of the Port 1 and Port 2 VNA cables.
25. Select **Isolation 1-2** from the Softkey Menu. When the NI-VNA SFP completes the measurement, a checkbox appears next to the Isolation 1-2 item in the Softkey Menu.
26. Click the **Back** button at the bottom of the Softkey Menu.
27. Click the **Done** button in the Softkey Menu.

To verify the calibration completed in the preceding procedure, the SFP should now show a Measuring State of CORR, which indicates that the module is applying correction factors. After connecting the cables together, the S_{21} amplitude and phase response should be flat.

To store the calibration data, select **File»Save Setup** to save the sweep settings and current calibration correction factors to disk for recall later. To recall the settings later, select **File»Recall Setup**.

Performing a Manual SOLT Calibration of the NI 5632 Using the NI-VNA SFP

⚠	**Caution** Use caution when attaching and detaching cables to and from the NI 5632, the AutoCal module, and manual calibration standards (open, short, load). A damaged connector on these items renders them unusable and necessitates repair or replacement. Always use an 0.9 N · m (8 in-lb) torque wrench for attaching cables to the NI 5632 module, the AutoCal module, and the manual calibration standards. Refer to Connector Care for more information about making proper signal connections.

This topic describes how to complete a manual calibration of the NI 5632 using manual calibration standards and the NI-VNA SFP. To perform a manual calibration, complete

the following steps:

1. Attach flexible, phase-stable cables with K connectors to PORT 1 and PORT 2 on the NI 5632 module front panel. These cables should be of opposite gender to the DUT connector ends to make the DUT insertable.

> **Note** You will attach the manual calibration standards to the end of the PORT 1 cable later in this procedure, so ensure the gender of the DUT end of the PORT 1 cable is the opposite gender of the manual calibration standard connector.

2. Launch the NI-VNA SFP. When you launch the NI-VNA SFP, the Measuring State indicator at the bottom of the Display Window shows UNCORR. This status indicates that the NI 5632 is initially uncalibrated and is not applying correction factors.
3. Select **Channel»Frequency** from the NI-VNA SFP Menu.
4. Specify values for the **Start Frequency**, **Stop Frequency**, and **Number of Points** controls in the NI-VNA SFP Softkey Menu.

> **Note** Wider sweep spans and larger numbers of points result in longer sweep and calibration times.

5. Select **Channel»Power** from the Menu.
6. Specify the output power levels of Port 1 and Port 2 in the Softkey Menu.
7. Select **Channel»Sweep** from the Menu.
8. Specify the type of sweep in the Softkey Menu.

> **Note** If you choose **Segmented Sweep (Freq-based)** or **Segmented Sweep (Index-based)** sweep types, you must configure the corresponding sweep setup in the Sweep Setup Softkey Menu.

9. Select **Channel»Averaging** from the Menu.
10. Specify the IF bandwidth and Averaging in the Softkey Menu.

> **Note** Enabling averaging and choosing narrower IF bandwidths result in longer sweep and calibration times.

11. Specify the manual calibration setup by completing the following steps:
    1. Select **Calibration** from the NI-VNA SFP Menu.

2. Select **Calibrate»Manual Cal»2-Port Cal»Modify Cal Setup»Edit Cal Params** from the Softkey Menu.

> **Note** You can choose **1 Port Cal** instead of **2-Port Cal** as shown in the preceding menu path if you want to make only one-port measurements.

3. Select **Broadband Load** as the Load Type in the Two Port Cal Setup dialog box.
4. Specify the connector type that is present on the manual calibration standards in the DUT Connector control. For example, the manual calibration standard from Anritsu Model OSLK50 has male K connectors for the short, open, and load, so you would select **K-Conn(M)** in this case.
5. Click **OK** to close the dialog box.

12. Click the **Back** button at the bottom of the Softkey Menu.
13. Select **Port 1 Reflective Devices** from the Softkey Menu.
14. Attach the Open manual calibration standard to the end of the cable attached to PORT 1 on the NI 5632 module front panel.

> **Caution** After you start the manual calibration, take care to move the VNA cables as little as possible throughout the calibration process and during ensuing usage.

15. Select **Open** from the Softkey Menu. When the NI-VNA SFP completes the measurement, a checkbox appears next to the Open command in the Softkey Menu.
16. Attach the Short manual calibration standard to the end of the cable attached to PORT 1 on the NI 5632 module front panel.
17. Select **Short** from the Softkey Menu. When the NI-VNA SFP completes the measurement, a checkbox appears next to the Short command in the Softkey Menu.
18. Attach the Load manual calibration standard to the end of the cable attached to PORT 1 on the NI 5632 module front panel.
19. Select **Load** from the Softkey Menu. When the NI-VNA SFP completes the measurement, a checkbox appears next to the Load command in the Softkey Menu.
20. Click the **Back** button at the bottom of the Softkey Menu. A checkmark should now appear next to the Port 1 Reflective Devices item in the Softkey Menu.
21. Select **Port 2 Reflective Devices** from the Softkey Menu.
22. Repeat Steps 12 through 17 to complete calibration on the Port 2 reflective

devices.

23. Click the **Back** button at the bottom of the Softkey Menu. A checkmark should now appear next to the Port 2 Reflective Devices item in the Softkey Menu.
24. Select **Thru/Recip** from the Softkey Menu.
25. Connect the PORT 1 and PORT 2 VNA cables together.
26. Select **Thru 1-2** from the Softkey Menu. When the NI-VNA SFP completes the measurement, a checkbox appears next to the Thru 1-2 item in the Softkey Menu.
27. Click the **Back** button at the bottom of the Softkey Menu.
28. (Optional) Select **Isolation (Optional)** from the Softkey Menu.
29. Attach the Load manual calibration standards to the ends of the Port 1 and Port 2 VNA cables.
30. Select **Isolation 1-2** from the Softkey Menu. When the NI-VNA SFP completes the measurement, a checkbox appears next to the Isolation 1-2 item in the Softkey Menu.
31. Click the **Back** button at the bottom of the Softkey Menu.
32. Click the **Done** button in the Softkey Menu.

To verify the calibration completed in the preceding procedure, the SFP should now show a Measuring State of CORR, which indicates that the module is applying correction factors. After connecting the cables together, the S_{21} amplitude and phase response should be flat.

To store the calibration data, select **File»Save Setup** to save the sweep settings and current calibration correction factors to disk for recall later. To recall the settings later, select **File»Recall Setup**.

Programming Reference

This section provides reference and programming information regarding the NI-VNA API and its supported ADEs.

Using Properties and Attributes with NI-VNA

Attributes, which are typically called properties in LabVIEW, serve as a base for parameters.

NI-VNA contains high-level functions that set most of the instrument attributes. Some

attributes are not accessible through the high-level functions. The values for these attributes must be set using the attribute.

Accessing Attributes

In LabVIEW, you can find attributes in the NI-VNA property node. To access them, complete the following steps:

1. Open a VI.
2. Make sure that you are viewing the block diagram. Navigate to the NI-VNA palette at **All Functions»NI Measurements»NI-VNA**, and then drag the property node icon to the block diagram.
3. Left-click the property node, and select the attribute you want to use.
4. To add additional attributes, resize the property node.

In C, attributes are accessed with the NI-VNA Set and Get Attribute functions. These functions correspond to a particular data type. For example, to set the input impedance, which has a data type of ViReal64, use `niVNA_SetAttributeViReal64`.

Attributes Apply to a Device, Channel, Port, Standard, or Frequency Band

NI-VNA attributes can be applied to the device, channel, port, standard, or band. When an attribute is applied to the device, you can set only a single value for this attribute, and that value applies to the entire device. When an attribute is applied to the channel, port, standard, or band, there may be multiple values for the attribute (one for each separate port, channel, standard, or band).

When you set an attribute, you can specify which channel, port, standard, or band the value should apply to using the Active Channel property in LabVIEW or the **channel** parameter on the Set Attribute functions in C/CVI. When you pass NULL or the empty string ("") for this parameter, and the attribute applies to multiple channels or ports, NI-VNA attempts to set the value on all valid channels or ports. You must specifically pass a valid specifier for standard- and band-based attributes.

Device

Device attributes contain a single value that pertains to the entire device or driver session.

Valid Device Specifiers: NULL, ""

Example Attributes: Device Serial Number, Triggering and Synchronization attributes, Driver State

Channel

A channel represents a collection of ports. A measurement involves one or more ports, and channel attributes generally refer to settings that apply to the entire measurement, rather than a specific port.

Valid Channel Specifiers: "0"

Example Attributes: Sweep Number of Points, Sweep Start/Stop Frequencies, S-Parameter Enabled attributes

Port

Port attributes can contain different values, each of which applies to a specific physical port.

Valid Port Specifiers: "port1", "port2"

Example Attributes: Output Power, Reference Plane Extension attributes

Standard

Standard attributes can contain different values, each of which applies to a physical standard used for TRL calibration.

Valid Standard Specifiers: "standard1", "standard2", "standard3", "standard4", "standard5", "standard6"

Example Attributes: TRL Calibration Standard Length, TRL Calibration Standard Type

Band

Band attributes can contain different values, each of which applies to one frequency band of a TRL calibration.

Valid Band Specifiers: "band1", "band2"

Example Attributes: TRL Calibration List of Standards, Band Break Point attributes

Refer to the Programming book in this help file for a complete listing of available attributes and properties.

Setting Properties and Attributes Before Reading Them

Properties and attributes are modified when you set them or when you call a configuration VI or function that sets them, respectively. It is important to set the properties or attributes or call any configuration VIs or functions before reading back any property or attribute values for the following reasons:

- Values read are coerced depending on the current configuration of the session. If you read a property or attribute value and then set other properties or attributes, the value read may no longer be valid.
- The driver verifies that the configuration of the device is valid at the time the property or attribute is read. It is possible to get an error when reading a property or attribute if the configuration is not valid at that point, even when a setting later could make it valid.
- Reading properties or attributes causes the driver to verify the current configuration. If you change some of the settings later, those settings need to be validated again.

Using the Standard Functionality for error in Parameters

Many LabVIEW nodes such as VIs contain **error in** parameters you can use to manage errors. These parameters typically provide the same, standard functionality. When a node exhibits different parameter functionality, the exceptions are documented in the reference material for that node. Standard **error in** behavior is as follows

	Note Some nodes, such as error handling VIs, contain an **error in** parameter that does not provide standard error in functionality, but that contains an **error in** cluster that is standard.

	error in describes error conditions that occur before this node runs. The default is `no error`. If an error occurred before this node runs, the node passes the **error in** value to **error out**. This node runs normally only if no error occurred before this node runs. If an error occurs while this node runs, it runs normally and sets its own error status in **error out**. Use **error in** and **error out** to check errors and to specify execution order by wiring **error out** from one node to **error in** of the next node.

The **error in** cluster contains the following cluster elements:

	status is TRUE (X) if an error occurred before this node ran or FALSE (checkmark) to indicate a warning or that no error occurred before this node ran. The default is FALSE.
	code is the error or warning code. The default is 0. If **status** is TRUE, **code** is an error code. If **status** is FALSE, **code** is 0 or a warning code.
	source specifies the origin of the error or warning and is, in most cases, the name of the node that produced the error or warning. The default is an empty string.

Using the Standard Functionality for error out Parameters

Many LabVIEW nodes such as VIs contain an **error out** parameter you can use to manage errors. These parameters typically provide the same, standard functionality. When a node exhibits different parameter functionality, the exceptions are documented in the reference material for that node.

Standard **error out** functionality is as follows:

	error out contains error information. If **error in** indicates that an error occurred before this VI ran, **error out** contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the **error out** front panel indicator and select **Explain Error** from the shortcut menu for more information about the error.

The **error out** contains the following cluster elements:

	**status** is TRUE (X) if an error occurred before this node ran or during the running of this node or FALSE (checkmark) to indicate a warning or that no error occurred before this node ran or during the running of this node.
	**code** is the error or warning code. If **status** is TRUE, **code** is an error code. If **status** is FALSE, **code** is 0 or a warning code.
	**source** specifies the origin of the error or warning and is, in most cases, the name of the node that produced the error or warning.

Using the Sweep Settings Configuration File

NI-VNA allows you to create a custom configuration file to save settings for sweep measurements. You can use the niVNA Load Sweep Settings Data From File VI or the `niVNA_LoadSweepSettingsDataFromFile` function to load the configuration file in NI-VNA.

The format of the configuration file is very specific. If you do not use the correct format, NI-VNA cannot read the file and returns the error NIVNA_ERROR_INVALID_CFG_FORMAT.



Note The file extension of the configuration file must be `.cfg`.



Tip NI-VNA installs an example configuration file, `exampleSweepConfiguration.cfg`, to `<NIDocDir>\NI-VNA\`. You can use and modify this example configuration file as necessary. Refer to the ***NI-VNA Readme*** for the location of `<NIDocDir>` on your computer.

The configuration file is made up of settings, and each setting contains two top-level parameters: `numPorts` and `segments`.

A configuration file can contain as many settings as possible, but you can only load one in NI-VNA at a time.

Related Information

Specifying Sweep Settings

Specifying Setting Parameters

Segment Types

Possible Errors

Specifying Sweep Settings

You can use the **Setting Name** input of the niVNA Load Sweep Settings Data From File VI or the **setting** input of the `niVNA_LoadSweepSettingsDataFromFile` function to specify the name of the sweep setting in the configuration file.

You must use the text `SweepSettings` followed by the custom setting name contained within quotes and nested inside brackets to specify a sweep setting. The folllwing example shows the format for specifying a sweep setting called `setting1`:

```
SweepSettings["setting1"] =
{
...
};
```

where `setting1` is the name of the setting.

Related Information

Using the Sweep Settings Configuration File

Specifying Setting Parameters

Segment Types

Possible Errors

Specifying Setting Parameters

You can specify the following parameters for the sweep setting inside of the setting brackets: `AveragingFactor`, `freq`, `freqStart`,`freqStop`, `IFBW`, `numPoints`, `numPorts`, `portPower`, `segments`, `stepSize`, and `type`.

Refer to the following table for more information about the sweep setting parameters.

Parameter Name	Description	Data Type
AveragingFactor	Specifies the number of times the data in the sweep is averaged before being calculated in all segment types. This parameter is required for all segment types.	Number
freq	Specifies the frequency in a zeroSpan segment type. This parameter is required for the zeroSpan segment type and is invalid for the startStop and the startStep segment types.	Number
freqStart	Specifies the beginning of the frequency range. This parameter is required for the startStop and the startStep segment types and is invalid for the zeroSpan segment type.	Number
freqStop	Specifies the end of the frequency range. This parameter is required for the startStop segment type and is invalid for the startStep and the zeroSpan segment types.	Number
IFBW	Specifies the intermediate frequency bandwidth in all segment types. This parameter is required for all segment types.	Number
numPoints	Specifies the number of points in all segment types. This parameter is required for all segment types.	Number
numPorts	Specifies how many ports are swept.  **Note** The numPorts parameter must match the number of entries in the portPower parameter in each segment.	Number
portPower	Specifies the power at each respective port, depending on the numPorts parameter specified in all segment types. This parameter is composed of a table that is specified by an opening and closing brace, for example, {}. This parameter is required for all segment types.  **Note** The number of entries in the portPower parameter must match the value of the numPorts parameter.	Number(s)
segments	Specifies the start of the list of segments in the configuration file.	String
stepSize	Specifies the step size of the frequency range. You can use this parameter along with the numPoints and freqStart parameters to calculate the end of the frequency range. This parameter is required for the startSte	Number

	p segment type and is invalid for the startStop and the zeroSpan segment types.	
type	Specifies the type of segment. The following values are valid for the type parameter: • startStop • startStep • zeroSpan Refer to the ***Segment Types*** topic for more information about the segment types and their parameters.	String

Formatting Parameters

You must nest all the previous parameters except for numPorts and segments underneath the segments parameter. The following example shows the format of a startStop segment of setting1 and its parameters:

```
SweepSettings["setting1"] =
{

["numPorts"] = 2,

["segments"] =

{

{

["type"] = "startStop",

["freqStart"] = 10*Mega,

["freqStop"] = 1*Giga,

["IFBW"] = 100*kilo,

["numPoints"] = 3,
```

```
["portPower"] =
{
1,
1
},
["AveragingFactor"] = 1
},
}
};
```

You must place a comma after the closing brace of each segment and after you specify a value for a parameter, unless the parameter is the final parameter in the segment.

Note Placing a comma incorrectly results in a NIVNA_ERROR_INVALID_CFG_FORMAT error.

All parameters are case-sensitive and must be specified using the correct case. For example, `NumPoints` is not valid, whereas `numPoints` is valid. All parameters must also be surrounded by open and closed brackets as well as opening and closing quotation marks, for example, `["parameterName"]`, where parameterName is the name of a parameter.

Formatting the portPower Parameter

The `portPower` parameter is the only parameter that deviates from the format of the other parameters specified in the `segments` parameter. The `portPower` parameter can be described as either one or two parameters, so it has a table format.

After specifying the `portPower` parameter, you must create open and close braces which end in a comma depending on if `portPower` is the last parameter defined in the segment. Inside of the `portPower` braces, you must specify the port powers at

each port and the last port power should not have a comma after it. If you specify only one port power then the table will contain only one number as shown in the following example:

```
["portPower"] =

{

-5

}
```

Parameter Data Ranges

Most of the setting parameters have valid data ranges that you can specify. If you set the value of a parameter to a value outside the data range, NI-VNA returns the error NIVNA_ERROR_DATA_OUT_OF_RANGE.

Refer to the following table for the valid data ranges of the setting parameters.

Parameter Name	Lower Bound	Upper Bound
`AveragingFactor`	1	1024
`freq`	50 Hz	8.5 GHz
`freqStart`	50 Hz	8.5 GHz
`freqStop`	50 Hz	8.5 GHz
`IFBW`	10 Hz	500 kHz
`numPoints`	1	20001
`numPorts`	1	2
`portPower`	-30 dB	30 dB

The value that you specify for each of the previous parameters must be greater than or equal to the value in the ***Lower Bound*** column and less than or equal to the value in the ***Upper Bound*** column to be valid.

Implicitly Defined Parameters

You can implicitly use previously defined values for the `AveragingFactor`, `IFBW`, and `portPower` parameters in any subsequent segment of the first segment in the `segments` list. If you do not define a value for the `AveragingFactor`, `IFBW`, and/or `portPower` parameter in any subsequent segment of the first segment, NI-VNA uses the most recently defined value of that parameter.



Note You must define a value for all required parameters for the first segment in the `segments` list, otherwise NI-VNA returns the error NIVNA_ERROR_REQUIRED_PARAMETER_NOT_FOUND.

The following example shows the `AveragingFactor`, `IFBW`, and `portPower` parameters using previously defined values:

```
SweepSettings["setting1"] =
{

["numPorts"] = 2,
["segments"] =
{

{

["type"] = "startStop",
["freqStart"] = 10*Mega,
["freqStop"] = 1*Giga,
["IFBW"] = 100*kilo,
["numPoints"] = 3,
["portPower"] =
{

1,
1

},
["AveragingFactor"] = 1
```

```
},
{

["type"] = "startStep",
["freqStart"] = 2*Giga,
["stepSize"] = 10*kilo,
["IFBW"] = 1*kilo,
["numPoints"] = 5,
["portPower"] =
{

0,
-5

}

}, --Note: the AveragingFactor parameter uses the valu
e in the startStop segment.
{

["type"] = "zeroSpan",
["freq"] = 1*Giga,
["numPoints"] = 5

}, --Note: the IFBW and portPower parameters use the v
alues in the startStep segment and the AveragingFactor
parameter uses the value in the startStop segment.

}

};
```

In the previous example, the first segment, `startStop`, defines all required parameters, including `AveragingFactor`, `IFBW`, and `portPower`. The `AveragingFactor` parameter is missing in the second segment, `startStep`, because it uses the value of `1` in the previous segment, `startStop`. The `IFBW` and `portPower` parameters are missing in the third segment, `zeroSpan`, because they use the values of `1*kilo` and `{0,-5}` respectively, in the second segment, `startStep`.

Related Information

Using the Sweep Settings Configuration File

Specifying Sweep Settings

Segment Types

Possible Errors

Segment Types

You can specify the following types of segments in the configuration file: `startStop`, `startStep`, and `zeroSpan`.

Each segment type corresponds to a possible frequency definition in the NI-VNA Soft Front Panel (SFP). Refer to the following table for information about which segment type corresponds to which NI-VNA SFP frequency definition.

Segment Type	NI-VNA SFP Frequency Definition
startStop	Start & Stop
startStep	Start & Step Size
zeroSpan	CW

You must specify all required parameters for the first segment in the `segments` list. However, you can implicitly use previously defined values for the `AveragingFactor`, `IFBW`, and `portPower` parameters in any subsequent segment of the first segment in the `segments` list. Refer to the ***Specifying Setting Parameters*** topic for more information about implicitly using previously defined values.

startStop Segment Type

The `startStop` segment type allows you to specify the beginning and end of a frequency range as well as the number of ports.

Refer to the following table for information about which parameters are required or invalid for the `startStop` segment type.

Parameter Name	Required or Invalid?
`AveragingFactor`	Required
`freq`	Invalid
`freqStart`	Required
`freqStop`	Required
`IFBW`	Required
`numPoints`	Required
`portPower`	Required
`stepSize`	Invalid
`type`	Required

startStep Segment Type

The `startStep` segment type has similar functionality as the `startStop` type; however, instead of specifying the end frequency, you specify the `stepSize` along with the start of the frequency range. This segment type is useful if, for example, you're interested in something that occurs every 10 MHz starting after 4 GHz.

Refer to the following table for information about which parameters are required or invalid for the `startStep` segment type.

Parameter Name	Required or Invalid?
`AveragingFactor`	Required
`freq`	Invalid
`freqStart`	Required
`freqStop`	Invalid
`IFBW`	Required
`numPoints`	Required

portPower	Required
stepSize	Required
type	Required

zeroSpan Segment type

The zeroSpan segment type sweeps a single frequency, specified with the freq parameter.

Refer to the following table for information about which parameters are required or invalid for the zeroSpan segment type.

Parameter Name	Required or Invalid?
AveragingFactor	Required
freq	Required
freqStart	Invalid
freqStop	Invalid
IFBW	Required
numPoints	Required
portPower	Required
stepSize	Invalid
type	Required

Related Information

Using the Sweep Settings Configuration File

Specifying Sweep Settings

Specifying Setting Parameters

Possible Errors

Possible Errors

Refer to the following table for information about the possible errors NI-VNA can return when reading the configuration file.

Error Name	Error Description	Example	IVI Error Code
NIVNA_ERROR_BAD_SWEEP_PARAMETER	Indicates when unsupported parameters are attempting to be read.	"Nummports" was specified in the configuration file instead of "`numPorts`".	`(IVI_SPECIFIC_ERROR_BASE+128)`
NIVNA_ERROR_DATA_OUT_OF_RANGE	Indicates when data is outside the specified parameter data range.	The `freqStop` parameter is set to 9.6 GHz.	`(IVI_SPECIFIC_ERROR_BASE+129)`
NIVNA_ERROR_INCONSISTENT_PORT_NUMBER	Indicates when the port number and number of port powers defined do not match.	The `numPorts` parameter is set to `2`, but `portPower` has only one number defined in a segment.	`(IVI_SPECIFIC_ERROR_BASE+126)`
NIVNA_ERROR_INVALID_CFG_FORMAT	Indicates when the format of the configuration file is incorrect.	The configuration file is missing a required bracket or comma.	`(IVI_SPECIFIC_ERROR_BASE+1`

			25)
NIVNA_ERROR_INVALID_DATA_TYPE	Indicates the data type associated with a sweep settings parameter is invalid.	A string was entered as the data for `freqStart`.	(IVI_SPECIFIC_ERROR_BASE+127)
NIVNA_ERROR_REQUIRED_PARAMETER_NOT_FOUND	Indicates when a parameter required for the segment type was not found.	A required parameter, such as `numPorts`, was not specified.	(IVI_SPECIFIC_ERROR_BASE+120)

Note The NI-VNA `IVI_SPECIFIC_ERROR_BASE` is `0xBFFA4000`.

Related Information

Using the Sweep Settings Configuration File

Specifying Sweep Settings

Specifying Setting Parameters

Segment Types

Performing a User Calibration Using the NI-VNA API

You can perform the following types of manual calibration using the NI-VNA instrument driver:

- Manual SOL (1-Port, 1-Path)
- Manual SOLT (2-Port, 1-Path)

- Manual TRL (2-Port, 2-Path)
- Reflection Response
- Transmission Response

You can also perform an auto calibration of your device. Refer to the following topics for more information about programming this operation using NI-VNA:

- Completing an Auto Calibration Using the NI-VNA API
- Completing an Auto Calibration with a True Through Using the NI-VNA API

Note Refer to the NI-VNA C Examples at **Start»All Programs»National Instruments»NI-VNA»Examples»C** for complete programming steps for performing a user calibration using the NI-VNA API.

Performing a Manual SOL (1-Port, 1-Path) Calibration

1. Configure the device for the final measurement you will be taking.
2. Configure the device to calibrate for a PORT 1, 1-port calibration.
   1. Call `niVNA_SetAttributeViBoolean(vi, "0", NIVNA_ATTR_S11_ENABLED, VI_TRUE)`.
   2. Call `niVNA_SetAttributeViBoolean(vi, "0", NIVNA_ATTR_S21_ENABLED, VI_FALSE)`.
   3. Call `niVNA_SetAttributeViBoolean(vi, "0", NIVNA_ATTR_S12_ENABLED, VI_FALSE)`.
   4. Call `niVNA_SetAttributeViBoolean(vi, "0", NIVNA_ATTR_S22_ENABLED, VI_FALSE)`.
3. (Optional) If desired, enable High-Sensitivity mode during calibration by calling `niVNA_SetAttributeViBoolean(vi, "0", NIVNA_ATTR_CALIBRATION_HIGH_SENSITIVITY_ENABLED, VI_TRUE)`.
4. Call `niVNA_InitUserCal(vi, "0", NIVNA_VAL_CALIBRATION_TYPE_1PORT_SOL)`.
5. Connect the open calibration standard to PORT 1.
6. Call `niVNA_PerformCalibrationAction(vi, "port1", NIVNA_VAL_CALIBRATION_ACTION_OPEN)`.
7. Call `niVNA_WaitForCalibrationActionDone(vi, "0", <timeout>)`.
8. Connect the short calibration standard to PORT 1.

9. Call `niVNA_PerformCalibrationAction(vi, "port1", NIVNA_VAL_CALIBRATION_ACTION_SHORT)`.
10. Call `niVNA_WaitForCalibrationActionDone(vi, "0", <timeout>)`.
11. Connect the load calibration standard to PORT 1.
12. Call n`iVNA_PerformCalibrationAction(vi, "port1", NIVNA_VAL_CALIBRATION_ACTION_LOAD)`.
13. Call `niVNA_WaitForCalibrationActionDone(vi, "0", <timeout>)`.
14. If the calibration has been successful to this point and you wish to apply the new calibration, call `niVNA_CloseUserCal(vi, "0", NIVNA_VAL_USER_CALIBRATION_COMMIT)`. Otherwise, call `niVNA_CloseUserCal(vi, "0", NIVNA_VAL_USER_CALIBRATION_ABORT)`
15. To store the current configuration and calibration settings to a file, call `niVNA_SaveCalibrationAndConfigurationDataToFile(vi, "0", <filePath>, VI_TRUE)`.

Performing a Manual SOLT (2-Port, 1-Path) Calibration

1. Configure the device for the final measurement you will be taking.
2. Configure the device to calibrate for a PORT 1, 2-port calibration.
    1. Call `niVNA_SetAttributeViBoolean(vi, "0", NIVNA_ATTR_S11_ENABLED, VI_TRUE)`.
    2. Call `niVNA_SetAttributeViBoolean(vi, "0", NIVNA_ATTR_S21_ENABLED, VI_TRUE)`.
    3. Call `niVNA_SetAttributeViBoolean(vi, "0", NIVNA_ATTR_S12_ENABLED, VI_FALSE)`.
    4. Call `niVNA_SetAttributeViBoolean(vi, "0", NIVNA_ATTR_S22_ENABLED, VI_FALSE)`.
3. (Optional) If desired, enable High-Sensitivity mode during calibration by calling `niVNA_SetAttributeViBoolean(vi, "0", NIVNA_ATTR_CALIBRATION_HIGH_SENSITIVITY_ENABLED, VI_TRUE)`.
4. Call `niVNA_InitUserCal(vi, "0", NIVNA_VAL_CALIBRATION_TYPE_1PORT_SOL)`.
5. Connect the open calibration standard to PORT 1.
6. Call `niVNA_PerformCalibrationAction(vi, "port1", NIVNA_VAL_CALIBRATION_ACTION_OPEN)`.

7. Call `niVNA_WaitForCalibrationActionDone(vi, "0", <timeout>)`.
8. Connect the short calibration standard to PORT 1.
9. Call `niVNA_PerformCalibrationAction(vi, "port1", NIVNA_VAL_CALIBRATION_ACTION_SHORT)`.
10. Call `niVNA_WaitForCalibrationActionDone(vi, "0", <timeout>)`.
11. Connect the load calibration standard to PORT 1.
12. Call n`iVNA_PerformCalibrationAction(vi, "port1", NIVNA_VAL_CALIBRATION_ACTION_LOAD)`.
13. Call `niVNA_WaitForCalibrationActionDone(vi, "0", <timeout>)`.
14. Connect the through calibration standard between PORT 1 and PORT 2.
15. Call `niVNA_PerformCalibrationAction(vi, "port1,port2", NIVNA_VAL_CALIBRATION_ACTION_THRU)`.
16. Call `niVNA_WaitForCalibrationActionDone(vi, "0", <timeout>)`.
17. Connect a termination or load calibration standard to PORT 1.
18. Call `niVNA_PerformCalibrationAction(vi, "port1,port2", NIVNA_VAL_CALIBRATION_ACTION_ISOLATION)`.
19. Call `niVNA_WaitForCalibrationActionDone(vi, "0", <timeout>)`.
20. If the calibration has been successful to this point and you wish to apply the new calibration, call `niVNA_CloseUserCal(vi, "0", NIVNA_VAL_USER_CALIBRATION_COMMIT)`. Otherwise, call `niVNA_CloseUserCal(vi, "0", NIVNA_VAL_USER_CALIBRATION_ABORT)`.
21. To store the current configuration and calibration settings to a file, call `niVNA_SaveCalibrationAndConfigurationDataToFile(vi, "0", <filePath>, VI_TRUE)`.

Performing a Manual TRL (2-Port, 2-Path) Calibration

1. Configure the device for the final measurement you will be taking.
2. Configure the reference plane location and number of bands (one or two).
3. Configure the attributes of each standard that will you will use.
4. Configure the list of standards used for each band and the band breakpoint.
5. Call `niVNA_InitUserCal(vi, "0", NIVNA_VAL_CALIBRATION_TYP`

E_TRL).

6. Connect standard1 to PORT 1 and PORT 2.
7. Call `niVNA_PerformTRLCalibrationAction(vi, "port1, port2", "standard1")`.
8. Call `niVNA_WaitForCalibrationActionDone(vi, "0", <timeout>)`.
9. Connect standard4 to PORT 1 and PORT 2.
10. Call `niVNA_PerformTRLCalibrationAction(vi, "port1, port2", "standard4")`.
11. Call `niVNA_WaitForCalibrationActionDone(vi, "0", <timeout>)`.
12. Connect standard2 to PORT 1.
13. Call `niVNA_PerformTRLCalibrationAction(vi, "port1", "standard2")`.
14. Call `niVNA_WaitForCalibrationActionDone(vi, "0", <timeout>)`.
15. Connect standard2 to PORT 2.
16. Call `niVNA_PerformTRLCalibrationAction(vi, "port2", "standard2")`.
17. Call `niVNA_WaitForCalibrationActionDone(vi, "0", <timeout>)`.
18. If using two bands, continue with the calibration by performing the following steps, otherwise continue to Step 20.
    1. Connect standard3 to PORT 1.
    2. Call `niVNA_PerformTRLCalibrationAction(vi, "port1", "standard3")`.
    3. Call `niVNA_WaitForCalibrationActionDone(vi, "0", <timeout>)`.
    4. Connect standard3 to PORT 2.
    5. Call `niVNA_PerformTRLCalibrationAction(vi, "port2", "standard3")`.
    6. Call `niVNA_WaitForCalibrationActionDone(vi, "0", <timeout>)`.
19. If the calibration has been successful to this point and you wish to apply the new calibration, call `niVNA_CloseUserCal(vi, "0", NIVNA_VAL_USER_CALIBRATION_COMMIT)`. Otherwise, call `niVNA_CloseUserCal(vi, "0", NIVNA_VAL_USER_CALIBRATION_ABORT)`.

20. To store the current configuration and calibration settings to a file, call `niVNA_SaveCalibrationAndConfigurationDataToFile(vi, "0", <filepath>, VI_TRUE)`.

Performing a Reflection Response Calibration

1. Configure the device for the final measurement you will be taking.
2. (Optional) If desired, enable High-Sensitivity mode during calibration by calling `niVNA_SetAttributeViBoolean(vi, "0", NIVNA_ATTR_CALIBRATION_HIGH_SENSITIVITY_ENABLED, VI_TRUE)`.
3. Call `niVNA_InitUserCal(vi, "0", NIVNA_VAL_CALIBRATION_TYPE_REFLECTION_RESPONSE)`.
4. Connect the open calibration standard to PORT 1.
5. Call `niVNA_PerformCalibrationAction(vi, "port1", NIVNA_VAL_CALIBRATION_ACTION_OPEN)`.
6. Call `niVNA_WaitForCalibrationActionDone(vi, "0", <timeout>)`.
7. (Optional) Connect the LOAD cal standard to PORT 1.
8. (Optional) Call `niVNA_PerformCalibrationAction(vi, "port1", NIVNA_VAL_CALIBRATION_ACTION_LOAD)`.
9. (Optional) Call `niVNA_WaitForCalibrationActionDone(vi, "0", <timeout>)`.
10. If the calibration has been successful to this point and you wish to apply the new calibration, call `niVNA_CloseUserCal(vi, "0", NIVNA_VAL_USER_CALIBRATION_COMMIT)`. Otherwise, call `niVNA_CloseUserCal(vi, "0", NIVNA_VAL_USER_CALIBRATION_ABORT)`.
11. To store the current configuration and calibration settings to a file, call `niVNA_SaveCalibrationAndConfigurationDataToFile(vi, "0", <filePath>, VI_TRUE)`.

Performing a Transmission Response Calibration

1. Configure the device for the final measurement you will be taking.
2. (Optional) If desired, enable High-Sensitivity mode during calibration by calling `niVNA_SetAttributeViBoolean(vi, "0", NIVNA_ATTR_CALIBRATION_HIGH_SENSITIVITY_ENABLED, VI_TRUE)`.
3. Call `niVNA_InitUserCal(vi, "0", NIVNA_VAL_CALIBRATION_TYP`

E_TRANSMISSION_RESPONSE).

4. Connect the through calibration standard between PORT 1 and PORT 2.
5. Call niVNA_PerformCalibrationAction(vi, "port1", NIVNA_VAL_CALIBRATION_ACTION_OPEN).
6. Call niVNA_WaitForCalibrationActionDone(vi, "0", <timeout>).
7. Connect the load calibration standard to PORT 1.
8. (Optional) Call niVNA_PerformCalibrationAction(vi, "port1", NIVNA_VAL_CALIBRATION_ACTION_ISOLATION).
9. (Optional) Call niVNA_WaitForCalibrationActionDone(vi, "0", <timeout>).
10. If the calibration has been successful to this point and you wish to apply the new calibration, call niVNA_CloseUserCal(vi, "0", NIVNA_VAL_USER_CALIBRATION_COMMIT). Otherwise, call niVNA_CloseUserCal(vi, "0", NIVNA_VAL_USER_CALIBRATION_ABORT).
11. To store the current configuration and calibration settings to a file, call niVNA_SaveCalibrationAndConfigurationDataToFile(vi, "0", <filePath>, VI_TRUE).

Performing an Auto Calibration Using the NI-VNA API

1. Configure the device for the final measurement you will be taking.
2. (Optional) If desired, enable High-Sensitivity mode during calibration by calling niVNA_SetAttributeViBoolean(vi, "0", NIVNA_ATTR_CALIBRATION_HIGH_SENSITIVITY_ENABLED, VI_TRUE).
3. Connect the AutoCal module to the desired serial port, and inform the driver of the port used by calling niVNA_SetAttributeViString(vi, "0", NIVNA_ATTR_AUTOCAL_BOX_COMMUNICATION_PORT, <port>), where <port> could be "COM1", "COM2", and so on.
4. Load characterization data for the AutoCal module by calling niVNA_LoadAutoCalCharacterizationDataFromFile(vi, "0", <filepath>), where <filepath> should be a path to a .acd file included with the AutoCal kit.
5. Connect the AutoCal module to PORT 1 and PORT 2. Which ports connect to which AutoCal module terminals can be configured by setting niVNA_SetAttributeViInt32(vi, "0", NIVNA_ATTR_AUTOCAL_BOX_ORIENTATION, <orientation>), where <orientation> could be NIVNA_VAL_AUTOCAL_BOX_ORIENTATION_PORT1_IS_LEFT or NIVNA_VAL_AUTOCAL_BOX_ORIENTATIO

N_PORT1_IS_RIGHT.

6. Call `niVNA_InitUserCal(vi, "0", <calibrationType>)`, where <calibrationType> can be `NIVNA_VAL_CALIBRATION_TYPE_2PORT_SOLT` or `NIVNA_VAL_CALIBRATION_TYPE_1PORT_SOL`.
7. Call `niVNA_PerformCalibrationAction(vi, "port1,port2", NIVNA_VAL_CALIBRATION_ACTION_AUTOCAL_STANDARD)`.

Note If `NIVNA_VAL_CALIBRATION_TYPE_1PORT_SOL` was chosen as the calibration type, only "port1" is required for the **portList** parameter.

8. Call `niVNA_WaitForCalibrationActionDone(vi, "0", <timeout>)`.
9. If the calibration has been successful to this point and you wish to apply the new calibration, call `niVNA_CloseUserCal(vi, "0", NIVNA_VAL_USER_CALIBRATION_COMMIT)`. Otherwise, call `niVNA_CloseUserCal(vi, "0", NIVNA_VAL_USER_CALIBRATION_ABORT)`.
10. To store the current configuration and calibration settings to a file, call `niVNA_SaveCalibrationAndConfigurationDataToFile(vi, "0", <filePath>, VI_TRUE)`.

Performing an Auto Calibration with a True Through Using the NI-VNA API

1. Configure the device for the final measurement you will be taking.
2. (Optional) If desired, enable High-Sensitivity mode during calibration by calling `niVNA_SetAttributeViBoolean(vi, "0", NIVNA_ATTR_CALIBRATION_HIGH_SENSITIVITY_ENABLED, VI_TRUE)`.
3. Connect the AutoCal module to the desired serial port, and inform the driver of the port used by calling `niVNA_SetAttributeViString(vi, "0", NIVNA_ATTR_AUTOCAL_BOX_COMMUNICATION_PORT, <port>)`, where <port> could be "COM1", "COM2", and so on.
4. Load characterization data for the AutoCal module by calling `niVNA_LoadAutoCalCharacterizationDataFromFile(vi, "0", <filepath>)`, where <filepath> should be a path to a `.acd` file included with the AutoCal kit.
5. Connect the AutoCal module to PORT 1 and PORT 2. Which ports connect to which AutoCal module terminals can be configured by setting `niVNA_SetAttributeV`

iInt32(vi, "0", NIVNA_ATTR_AUTOCAL_BOX_ORIENTATION, <orientation>), where <orientation> could be NIVNA_VAL_AUTOCAL_BOX_ORIENTATION_PORT1_IS_LEFT or NIVNA_VAL_AUTOCAL_BOX_ORIENTATION_PORT1_IS_RIGHT.

6. Call niVNA_InitUserCal(vi, "0", <calibrationType>), where <calibrationType> can be NIVNA_VAL_CALIBRATION_TYPE_2PORT_SOLT.
7. Call niVNA_PerformCalibrationAction(vi, "port1,port2", NIVNA_VAL_CALIBRATION_ACTION_AUTOCAL_TRUE_THRU).
8. Call niVNA_WaitForCalibrationActionDone(vi, "0", <timeout>).
9. Connect a termination or load calibration standard to PORT 1.
10. Call niVNA_PerformCalibrationAction(vi, "port1,port2", NIVNA_VAL_CALIBRATION_ACTION_ISOLATION).
11. Call niVNA_WaitForCalibrationActionDone(vi, "0", <timeout>).
12. If the calibration has been successful to this point and you wish to apply the new calibration, call niVNA_CloseUserCal(vi, "0", NIVNA_VAL_USER_CALIBRATION_COMMIT). Otherwise, call niVNA_CloseUserCal(vi, "0", NIVNA_VAL_USER_CALIBRATION_ABORT).
13. To store the current configuration and calibration settings to a file, call niVNA_SaveCalibrationAndConfigurationDataToFile(vi, "0", <filePath>, VI_TRUE).

Reference

This book contains reference information about the NI-VNA VIs, properties, functions, and attributes.

NI-VNA LabVIEW Reference

This book contains information about NI-VNA LabVIEW VIs and properties.

NI-VNA VIs

Installed With: NI-VNA

September 2014, 373154D-01

This help file contains technical and programming support for using the NI-VNA LabVIEW API. This help file provides reference material for the NI-VNA VIs.

Use the NI-VNA VIs to develop applications for your RF vector network analyzer.

Palette Object	Description
niVNA Initialize	Creates a new session to the device.
niVNA Close	Closes the session to the device.
niVNA Initialize With Options	Creates a new session to the device and specifies initialization options, such as setting default values or enabling simulation.
niVNA Property Node	Gets (reads) or sets (writes) NI-VNA properties. When you read a property, NI-VNA analyzes the current configuration in order to return the coerced value for that property. Setting a property may transition the session to the Initialized state.

Subpalette	Description
Configuration	Use the NI-VNA Configuration VIs to configure operations with your RF vector network analyzer.
Measurement	Use the NI-VNA Measurement VIs to control measurement operations with your RF vector network analyzer.
Calibration	Controls calibration operations for the NI-VNA device.
Utility	Use the NI-VNA Utility VIs to access utility features of NI-VNA.

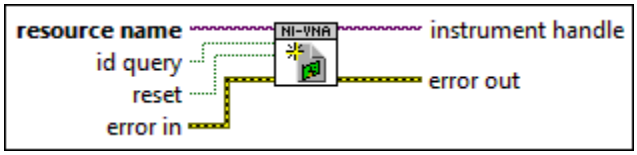
© 2010–2014 National Instruments. All Rights Reserved.

niVNA Initialize VI

Installed With: NI-VNA

Creates a new session to the device.

Supported Devices: NI 5630/5632



I/O	**resource name** specifies the resource name of the device to initialize.
TF	**id query** specifies whether NI-VNA performs an ID query. Regardless of how you set this parameter, NI-VNA always checks the device type.
TF	**reset** specifies whether to reset the device to a known initialization state.
	The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

TF	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
I32	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or

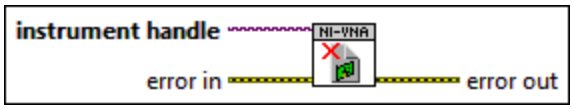
	Explain Warning) gives more information about the error displayed.
	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	instrument handle identifies your instrument session.
	The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.
	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

niVNA Close VI

Installed With: NI-VNA

Closes the session to the device.

Supported Devices: NI 5630/5632



I/O	**instrument handle** identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.
TF	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
I32	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
abc	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.
TF	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

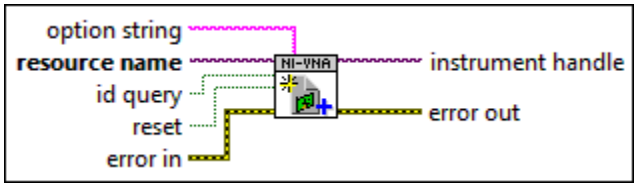
I32	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
abc	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

niVNA Initialize With Options VI

Installed With: NI-VNA

Creates a new session to the device and specifies initialization options, such as setting default values or enabling simulation.

Supported Devices: NI 5630/5632



I/O	**resource name** specifies the resource name of the device to initialize.
TF	**id query** specifies whether NI-VNA performs an ID query. Regardless of how you set this parameter, NI-VNA always checks the device type.
TF	**reset** specifies whether to reset the device to a known initialization state.
abc	**option string** ("") sets the initial value of certain properties for the session.

	The following properties are used in this parameter: • Range Check • Query Instrument Status • Cache • Record Value Coercions • Driver Setup • Simulate The format of this string is "***PropertyName=Value***", where ***PropertyName*** is the name of the property and ***Value*** is the value to which the property is set. To set multiple properties, separate their assignments with a comma.
	The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.  The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.  The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.  The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	**instrument handle** identifies your instrument session.
	The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error

	displayed.
	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

niVNA Property Node VI

Installed With: NI-VNA

Gets (reads) or sets (writes) NI-VNA properties. When you read a property, NI-VNA analyzes the current configuration in order to return the coerced value for that property. Setting a property may transition the session to the Initialized state.

Supported Devices: NI 5630/5632



	**instrument handle** identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	The **error in** cluster can accept error information wired from VIs previously called. Use this

	information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed. **status** The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed. **code** The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed. **source** The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	**instrument handle out** passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed. **status** The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed. **code** The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

Configuration

Owning Palette: NI-VNA VIs**Installed With:** NI-VNA

Use the NI-VNA Configuration VIs to configure operations with your RF vector network analyzer.

Palette Object	Description
niVNA Configure Connector	Configures the connector on the manual calibration standard. Selecting correct values for the parameters in this VI ensures that calibration quality is optimized because the calibration characterization (resistance, capacitance, or inductance) is as accurate as possible.
niVNA Perform Auto Reference Plane Extension	Automatically determines the reference plane extension length and the reference plane extension time by measuring the connected manual short/open standards. Ensure that the manual short or open standard is connected to the RF cable connected to your device, and perform at least one frequency sweep operation before you call this VI.

Subpalette	Description
Sweep	Configures the NI-VNA device sweep measurement.
Trigger	Use the NI-VNA Trigger VIs to configure the triggers for an RF vector network analyzer measurement.

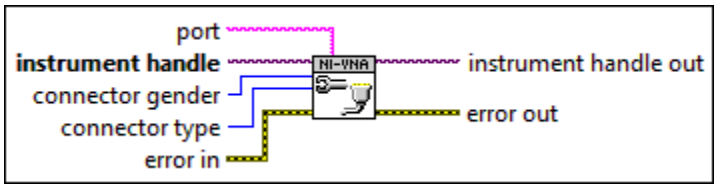
Time Domain	Use the NI-VNA Time Domain VIs to configure the time-domain measurement on the NI RF vector network analyzer.

niVNA Configure Connector VI

Installed With: NI-VNA

Configures the connector on the manual calibration standard. Selecting correct values for the parameters in this VI ensures that calibration quality is optimized because the calibration characterization (resistance, capacitance, or inductance) is as accurate as possible.

Supported Devices: NI 5630/5632



I/O	**instrument handle** identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
I32	**connector gender** specifies the connector gender. **Male** — The port uses a male connector. **Female** — The port uses a female connector.
I32	**connector type** specifies the connector type. **N** — The port uses an N-type connector.

K	The port uses a K-type connector.
GPC 3.5	The port uses a GPC 3.5 connector.
SMA	The port uses an SMA connector.

port specifies the port to which to apply the settings. Valid port specifiers are "`port1`" or "`port2`".

The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

instrument handle out passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.

The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

TF	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
I32	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
abc	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

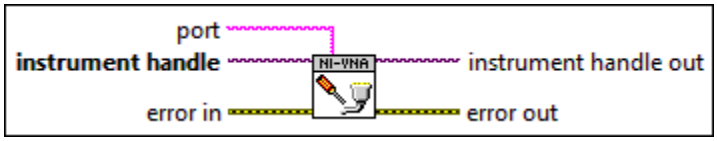
niVNA Perform Auto Reference Plane Extension VI

Installed With: NI-VNA

Automatically determines the reference plane extension length and the reference plane extension time by measuring the connected manual short/open standards. Ensure that the manual short or open standard is connected to the RF cable connected to your device, and perform at least one frequency sweep operation before you call this VI.

The corresponding properties for these parameters are set to the value that is measured by this VI.

Supported Devices: NI 5630/5632



	**instrument handle** identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	**port** specifies the port to which to apply the settings. Valid port specifiers are "`port1`" or "`port2`".
	The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.  The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.  The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.  The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	**instrument handle out** passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

Sweep

Owning Palette: Configuration**Installed With:** NI-VNA

Configures the NI-VNA device sweep measurement.

Palette Object	Description
niVNA Configure Sweep	Configures the number of points for the sweep measurement and the sweep mode.
niVNA Configure Frequency	Configures the frequency range of the frequency sweep.
niVNA Configure Power Start Stop	Configures the start and stop power of the power sweep.
niVNA Configure Frequency Offset Equation	Configures the frequency offset equation that helps control the source and receiver.

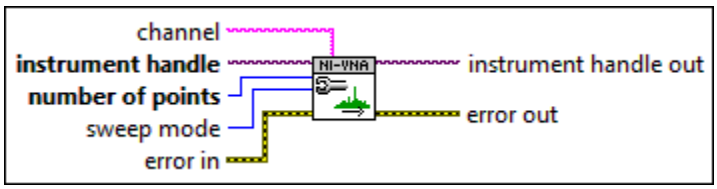
niVNA Load Sweep Settings Data From File	Loads sweep settings data from a configuration file that you specify with the **file path** parameter.

niVNA Configure Sweep VI

Installed With: NI-VNA

Configures the number of points for the sweep measurement and the sweep mode.

Supported Devices: NI 5630/5632



I/O	**instrument handle** identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
I32	**number of points** specifies the number of points to acquire as part of the sweep measurement. **Valid Ranges**: • NI 5630—1 to 3,201 • NI 5632—1 to 20,001 **Default Value**: 201
I32	**sweep mode** specifies the mode for the sweep measurement. For the NI 5630, **Frequency Sweep** is the only supported value.

Frequency Sweep	The device performs a frequency sweep.
Power Sweep	The device performs a power sweep.

channel specifies the channel for which this operation is targeted.

The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

instrument handle out passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.

The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

niVNA Configure Frequency VI

Installed With: NI-VNA

Configures the frequency range of the frequency sweep.

You can configure a frequency range in one of two ways:

- As a span of data surrounding the center frequency.
- As a span of data between a start frequency and a stop frequency.



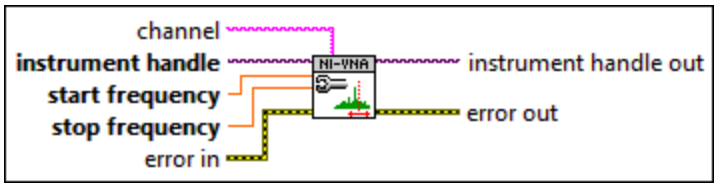
Tip You can configure a zero span sweep (CW mode) by setting the **start frequency** and **stop frequency** to the same value or by configuring a **center frequency** with a **span** of 0.

Supported Devices: NI 5630/5632

niVNA Configure Frequency Start Stop

Configures the start and stop frequency for the frequency sweep.

Supported Devices: NI 5630/5632



I/O	**instrument handle** identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
DBL	**start frequency** specifies the lower limit of a span of frequencies. This value is expressed in hertz (Hz). **Valid Values**: • NI 5630—≥10 MHz • NI 5632—≥300 kHz
DBL	**stop frequency** specifies the upper limit of a span of frequencies. This value is expressed in hertz (Hz). **Valid Values**: • NI 5630—≤6 GHz • NI 5632—≤8.5 GHz
abc	**channel** specifies the channel for which this operation is targeted.
	The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed. TF — The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error

	or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
I32	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
abc	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

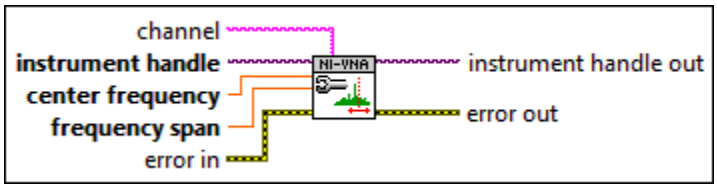
I/O	**instrument handle out** passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.

	The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.
TF	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
I32	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
abc	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

niVNA Configure Frequency Center Span

Configures the start and stop frequencies for a frequency sweep, based on the **center frequency** and **frequency span**.

Supported Devices: NI 5630/5632



I/O	**instrument handle** identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
DBL	**center frequency** specifies the center frequency in a frequency sweep. This value is expressed in hertz (Hz).
DBL	**frequency span** specifies the frequency span for the frequency sweep. This frequency span is centered at the value you select for **center frequency**. The value is expressed in hertz (Hz).
abc	**channel** specifies the channel for which this operation is targeted.
	The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

TF	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

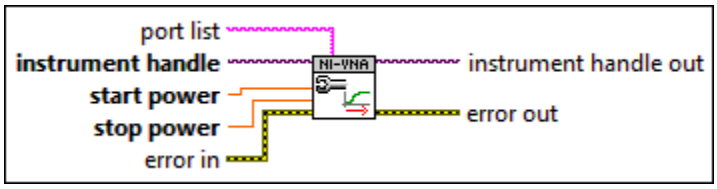
	<table><tr><td>  </td><td>The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.</td></tr><tr><td>  </td><td>The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.</td></tr></table>
	**instrument handle out** passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.<table><tr><td>  </td><td>The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.</td></tr><tr><td>  </td><td>The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.</td></tr><tr><td>  </td><td>The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.</td></tr></table>

niVNA Configure Power Start Stop VI

Installed With: NI-VNA

Configures the start and stop power of the power sweep.

Supported Devices: NI 5632



I/O	**instrument handle** identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
DBL	**start power** specifies the start power for power sweep. The default value is -30. This value is expressed in dBm. **Valid Range**: ≥-30 to ≤10
DBL	**stop power** specifies the stop power for power sweep. The default value is 10 dBm. **Valid Range**: ≥-30 to ≤10
abc	**port list** specifies the port(s) for which the start and stop power of the power sweep are applied. Valid port list specifiers are "`port1`", "`port2`", or "`port1,port2`".
	The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.
TF	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	instrument handle out passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.
	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

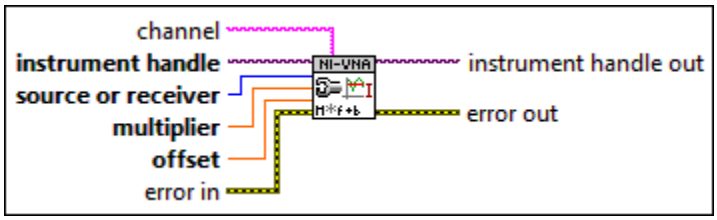
niVNA Configure Frequency Offset Equation VI

Installed With: NI-VNA

Configures the frequency offset equation that helps control the source and receiver.

The frequency offset equation is (***M*** * ***f***) + ***b***, where ***M*** is the **multiplier**, ***f*** is the frequency list defined by the **start frequency** and **stop frequency** parameters of the niVNA Configure Frequency VI, and ***b*** is the **offset**.

Supported Devices: NI 5632



I/O	**instrument handle** identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
I32	**source or receiver** specifies whether the frequency offset equation applies to either the source or receiver. **Source** — The frequency offset equation is applied to the source frequency. **Receiver** — The frequency offset equation is applied to the receiver frequency.
DBL	**multiplier** specifies the ***M*** value in the frequency offset equation.

	**offset** specifies the ***b*** value in the frequency offset equation.
	**channel** specifies the channel for which this operation is targeted. If you set **source or receiver** to **Source**, the valid channel specifier is "`port1`" or "`port2`". If you set **source or receiver** to **Receiver**, the valid channel specifier is "".
	The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.  The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.  The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.  The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	**instrument handle out** passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

TF	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
I32	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
abc	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

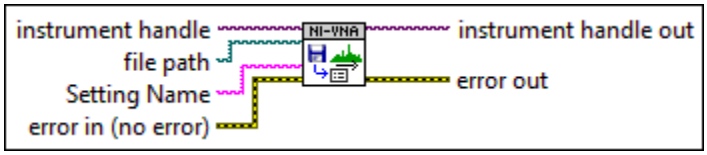
niVNA Load Sweep Settings Data From File VI

Installed With: NI-VNA

Loads sweep settings data from a configuration file that you specify with the **file path** parameter.

Refer to Using the Sweep Settings Configuration File for information about using the configuration file.

Supported Devices: NI 5632



I/O	**instrument handle** identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular

	instrument session.
	**file path** specifies the absolute file path to the configuration file that contains the sweep settings data.
	**Setting Name** specifies the name of the sweep setting in the configuration file.
	The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed. **  ** The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed. **  ** The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed. **  ** The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	**instrument handle out** passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

Trigger

Owning Palette: Configuration**Installed With:** NI-VNA

Use the NI-VNA Trigger VIs to configure the triggers for an RF vector network analyzer measurement.

Palette Object	Description
niVNA Configure Trigger	Configures the Start and Step Triggers for your NI-VNA device.
niVNA Send Software Edge Trigger	Asserts a software trigger. You must first use the niVNA Configure Software Edge Start Trigger VI to configure the device to use a software trigger.
niVNA Export Signal	Routes signals to the specified output terminal.

niVNA Configure Trigger VI

Installed With: NI-VNA

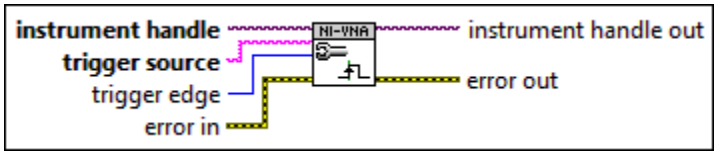
Configures the Start and Step Triggers for your NI-VNA device.

Supported Devices: NI 5630/5632

niVNA Configure Digital Edge Start Trigger

Configures the device to wait for a digital edge Start Trigger before starting a sweep.

Supported Devices: NI 5630/5632



I/O	**instrument handle** identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
abc	**trigger source** specifies the source of the digital edge for the Start Trigger.

PFI 0	The trigger is received on PFI 0.
PXI_Trig0	The trigger is received on PXI trigger line 0.
PXI_Trig1	The trigger is received on PXI trigger line 1.
PXI_Trig2	The trigger is received on PXI trigger line 2.
PXI_Trig3	The trigger is received on PXI trigger line 3.
PXI_Trig4	The trigger is received on PXI trigger line 4.
PXI_Trig5	The trigger is received on PXI trigger line 5.
PXI_Trig6	The trigger is received on PXI trigger line 6.

PXI_Trig7	The trigger is received on PXI trigger line 7.

trigger edge specifies the trigger edge on which the sweep starts.

Rising	The sweep starts on the rising edge of the trigger.
Falling	The sweep starts on the falling edge of the trigger.

The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

instrument handle out passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.

The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error

displayed.

TF	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
I32	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
abc	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

niVNA Disable Start Trigger

Configures the device to start measuring without waiting for a Start Trigger. This VI is necessary only if you want to disable a configured Start Trigger.

Supported Devices: NI 5630/5632



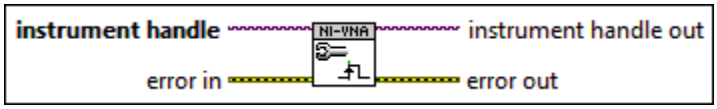
I/O	**instrument handle** identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	instrument handle out passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.
	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

niVNA Configure Software Edge Start Trigger

Configures the device to wait for a software Start Trigger before starting measurement. The device waits until you call the niVNA Send Software Edge Trigger VI to assert the trigger.

Supported Devices: NI 5630/5632



I/O	**instrument handle** identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.
TF	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
I32	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
abc	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
I/O	**instrument handle out** passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With

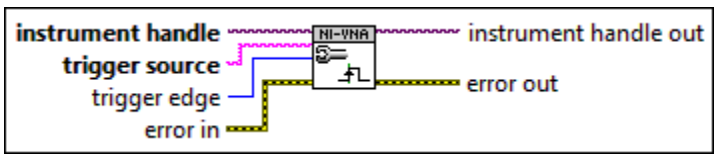
	Options VI and identifies a particular instrument session.
	The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

TF	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
I32	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
abc	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

niVNA Configure Digital Edge Step Trigger

Configures the device to wait for a digital edge Step Trigger before every individual point measurement within a sweep. The device expects ***N*** Step Triggers per sweep, where ***N*** is the number of points configured for the sweep.

Supported Devices: NI 5632



I/O	**instrument handle** identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular

instrument session.

trigger source specifies the source of the digital edge for the Step Trigger.

PFI 0	The trigger is received on PFI 0.
PXI_Trig0	The trigger is received on PXI trigger line 0.
PXI_Trig1	The trigger is received on PXI trigger line 1.
PXI_Trig2	The trigger is received on PXI trigger line 2.
PXI_Trig3	The trigger is received on PXI trigger line 3.
PXI_Trig4	The trigger is received on PXI trigger line 4.
PXI_Trig5	The trigger is received on PXI trigger line 5.
PXI_Trig6	The trigger is received on PXI trigger line 6.
PXI_Trig7	The trigger is received on PXI trigger line 7.

trigger edge specifies the trigger edge on which the sweep starts.

Rising	The sweep starts on the rising edge of the trigger.
Falling	The sweep starts on the falling edge of the trigger.

The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	instrument handle out passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.
	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

niVNA Disable Step Trigger

Configures the device to sweep without waiting for a Step Trigger to take an individual point measurement. This VI is necessary only if you want to disable a configured Step

Trigger.

Supported Devices: NI 5632



I/O	**instrument handle** identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.
TF	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
I32	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
abc	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
I/O	**instrument handle out** passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

	The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

TF	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
I32	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
abc	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

niVNA Send Software Edge Trigger VI

Installed With: NI-VNA

Asserts a software trigger. You must first use the niVNA Configure Software Edge Start Trigger VI to configure the device to use a software trigger.

Supported Devices: NI 5630/5632



I/O	**instrument handle** identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	The **error in** cluster can accept error information wired from VIs previously called. Use this

information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



instrument handle out passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.



The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

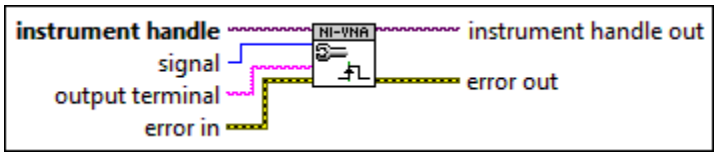
	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

niVNA Export Signal VI

Installed With: NI-VNA

Routes signals to the specified output terminal.

Supported Devices: NI 5632



	instrument handle identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	signal specifies the type of signal to route. **Ready for Start Event** — NI-VNA routes a Ready for Start Event. **Ready for Step Event** — NI-VNA routes a Ready for Step Event. **End of Step Event** — NI-VNA routes an End of Step Event. **End of Sweep Event** — NI-VNA routes an End of Sweep Event.
	output terminal specifies the terminal where the signal is exported. You can also choose not to export a signal.

Do not export signal	The signal is not exported.
PFI0	The signal is exported to PFI 0.
PXI_Trig0	The signal is exported to the PXI trigger line 0.
PXI_Trig1	The signal is exported to the PXI trigger line 1.
PXI_Trig2	The signal is exported to the PXI trigger line 2.
PXI_Trig3	The signal is exported to the PXI trigger line 3.
PXI_Trig4	The signal is exported to the PXI trigger line 4.
PXI_Trig5	The signal is exported to the PXI trigger line 5.
PXI_Trig6	The signal is exported to the PXI trigger line 6.
PXI_Trig7	The signal is exported to the PXI trigger line 7.



The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



instrument handle out passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With

	Options VI and identifies a particular instrument session.
	The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

Time Domain

Owning Palette: Configuration**Installed With:** NI-VNA

Use the NI-VNA Time Domain VIs to configure the time-domain measurement on the NI RF vector network analyzer.

Palette Object	Description
niVNA Configure Time Domain	Configures the VNA for time-domain analysis.
niVNA Configure Time Domain Gate	Configures the gate for the time-domain analysis.

niVNA Configure Time Domain VI

Installed With: NI-VNA

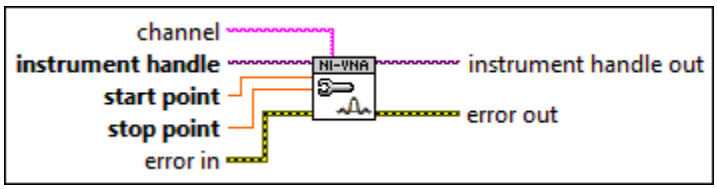
Configures the VNA for time-domain analysis.

Supported Devices: NI 5630/5632

niVNA Configure Time Domain Start Stop Distance

Configures the start and stop points for the time-domain analysis. This VI configures the start and stop points in terms of distance. If you want to configure the start and stop points in terms of time, use the niVNA Configure Time Domain Start Stop Time instance of this VI.

Supported Devices: NI 5630/5632



I/O	**instrument handle** identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
DBL	**start point** specifies the starting point for the time-domain analysis. This value is expressed in meters.
DBL	**stop point** specifies the stopping point for the time-domain analysis. This value is expressed in meters.
abc	**channel** specifies the channel for which this operation is targeted.

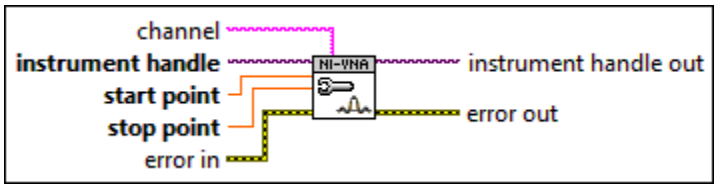
	The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed. **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed. **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed. **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	**instrument handle out** passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed. **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed. **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

niVNA Configure Time Domain Start Stop Time

Configures the start and stop points for the time-domain analysis. This VI configures the start and stop points in terms of time. If you want to configure the start and stop points in terms of distance, use the niVNA Configure Time Domain Start Stop Distance instance of this VI.

Supported Devices: NI 5630/5632



	instrument handle identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	start point specifies the starting point for the time-domain analysis. This value is expressed in seconds.
	stop point specifies the stopping point for the time-domain analysis.This value is expressed in seconds.
	channel specifies the channel for which this operation is targeted.

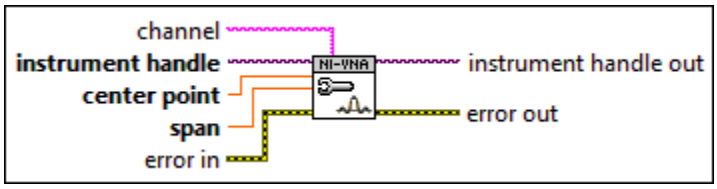
	The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.
	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	instrument handle out passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.
	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

niVNA Configure Time Domain Center Span Distance

Configures the start and stop points for the time-domain analysis based on the the **center point** and **span** you specify. This VI configures the span in terms of distance. If you want to configure a time-domain analysis span in terms of time, use the niVNA Configure Time Domain Center Span Time instance of this VI.

Supported Devices: NI 5630/5632



	instrument handle identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	center point specifies the center point of the span used for the time-domain analysis.This value is expressed in meters.
	span specifies the span for the time-domain analysis. This value is expressed in meters.
	channel specifies the channel for which this operation is targeted.
	The **error in** cluster can accept error information wired from VIs previously called. Use this

information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



instrument handle out passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.



The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

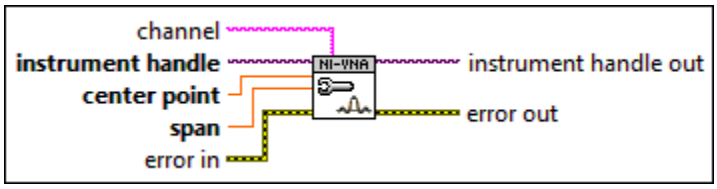
	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

niVNA Configure Time Domain Center Span Time

Configures the start and stop times for the time-domain analysis based on the the **center point** and **span** you specify. This VI configures the span in terms of time. If you want to configure a time-domain analysis span in terms of distance, use the niVNA Configure Time Domain Center Span Distance instance of this VI.

Supported Devices: NI 5630/5632



	instrument handle identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	center point specifies the center time of the span used for the time-domain analysis. This
	span specifies the time span for the time-domain analysis. This value is expressed in seconds.
	channel specifies the channel for which this operation is targeted.
	The **error in** cluster can accept error information wired from VIs previously called. Use this

information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



instrument handle out passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.



The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

niVNA Configure Time Domain Gate VI

Installed With: NI-VNA

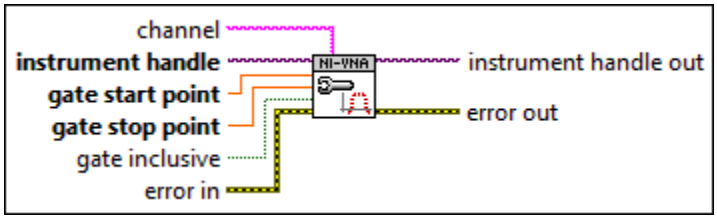
Configures the gate for the time-domain analysis.

Supported Devices: NI 5630/5632

niVNA Configure Time Domain Gate Start Stop Distance

Configures the start and stop points for the gate used in time-domain analysis. This VI configures the start and stop points in terms of distance. If you want to configure the start and stop points in terms of time, use the niVNA Configure Time Domain Gate Start Stop Time instance of this VI.

Supported Devices: NI 5630/5632



	instrument handle identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	gate start point specifies the start point for the gate used in the time-domain analysis. This value is expressed in meters.

	**gate stop point** specifies the stop point for the gate used in the time-domain analysis.This value is expressed in meters.
	**gate inclusive** specifies whether the gate function is inclusive or exclusive. If the gate function is inclusive, the time-domain operation includes only the data between the **gate start point** and **gate stop point**, but if it is exclusive, the time-domain operation excludes the data between the **gate start point** and **gate stop point**.
	**channel** specifies the channel for which this operation is targeted.
	The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed. **  ** The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed. **  ** The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed. **  ** The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	**instrument handle out** passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.

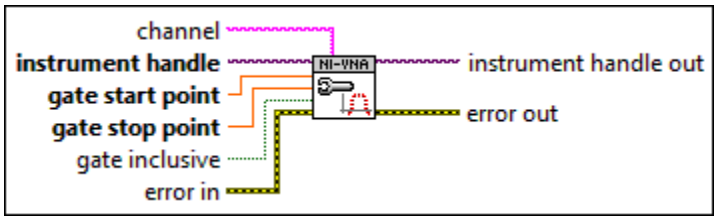
The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

TF	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
I32	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
abc	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

niVNA Configure Time Domain Gate Start Stop Time

Configures the start and stop points for the gate used in time-domain analysis. This VI configures the start and stop points in terms of time. If you want to configure the start and stop points for the gate span in terms of distance, use the niVNA Configure Time Domain Gate Start Stop Distance instance of this VI.

Supported Devices: NI 5630/5632



I/O	**instrument handle** identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.

	**gate start point** specifies the start point for the gate used in the time-domain analysis. This value is expressed in seconds.
	**gate stop point** specifies the stop point for the gate used in the time-domain analysis. This value is expressed in seconds.
	**gate inclusive** specifies whether the gate function is inclusive or exclusive. If the gate function is inclusive, the time-domain operation includes only the data between the **gate start point** and **gate stop point**, but if it is exclusive, the time-domain operation excludes the data between the **gate start point** and **gate stop point**.
	**channel** specifies the channel for which this operation is targeted.
	The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed. **status**  The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed. **code**  The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed. **source**  The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	**instrument handle out** passes a reference to your instrument session to the next VI.

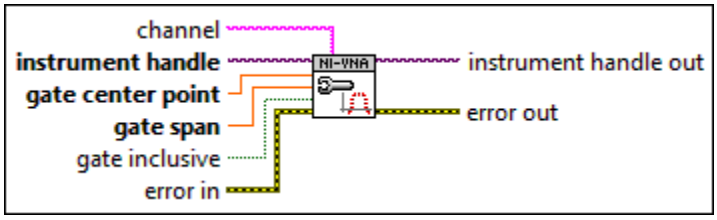
	instrument handle is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

TF	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
I32	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
abc	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

niVNA Configure Time Domain Gate Center Span Distance

Configures the start and stop points for the gate used in time-domain analysis based on the the **gate center point** and **gate span** you specify. This VI configures the span in terms of distance. If you want to configure a time-domain analysis gate span in terms of time, use the niVNA Configure Time Domain Gate Center Span Time instance of this VI.

Supported Devices: NI 5630/5632



	**instrument handle** identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	**gate center point** specifies the center point of the gate used in the time-domain analysis. This value is expressed in meters.
	**gate span** specifies the span of the gate used in the time-domain analysis. This value is expressed in meters.
	**gate inclusive** specifies whether the gate function is inclusive or exclusive. If the gate function is inclusive, the time-domain operation includes only the data inside the **gate span**, but if it is exclusive, the time-domain operation excludes the data inside the **gate span**.
	**channel** specifies the channel for which this operation is targeted.
	The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

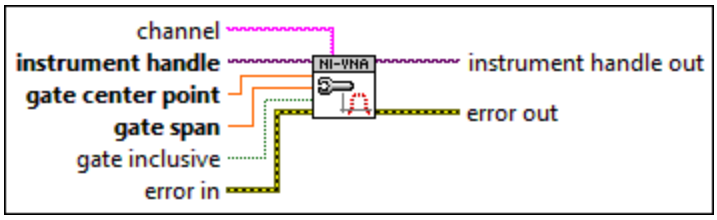
	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **source** string describes the origin of the error or warning. The pop-up option

	Explain Error (or Explain Warning) gives more information about the error displayed.
	**instrument handle out** passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.
	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

niVNA Configure Time Domain Gate Center Span Time

Configures the start and stop times for the gate used in time-domain analysis based on the the **gate center point** and **gate span** you specify. This VI configures the span in terms of time. If you want to configure a gate span in terms of distance, use the niVNA Configure Time Domain Gate Center Span Distance instance of this VI.

Supported Devices: NI 5630/5632



I/O	**instrument handle** identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
DBL	**gate center point** specifies the center point of the gate used in the time-domain analysis. This value is expressed in seconds.
DBL	**gate span** specifies the span of the gate used in the time-domain analysis. This value is e
TF	**gate inclusive** specifies whether the gate function is inclusive or exclusive. If the gate function is inclusive, the time-domain operation includes only the data inside the **gate span**, but if it is exclusive, the time-domain operation excludes the data inside the **gate span**.
abc	**channel** specifies the channel for which this operation is targeted.
	The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

TF	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

	<table><tr><td>  </td><td>The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.</td></tr><tr><td>  </td><td>The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.</td></tr></table>
	**instrument handle out** passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.<table><tr><td>  </td><td>The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.</td></tr><tr><td>  </td><td>The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.</td></tr><tr><td>  </td><td>The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.</td></tr></table>

Measurement

Owning Palette: NI-VNA VIs**Installed With:** NI-VNA

Use the NI-VNA Measurement VIs to control measurement operations with your RF vector network analyzer.

Palette Object	Description
niVNA Initiate	Starts the measurement. Use this VI in conjunction with the niVNA Wait For Sweep Done VI to wait for the sweep measurement to complete and the niVNA Fetch VI to retrieve acquired data. If this VI runs successfully, NI-VNA transitions to the Running state.
niVNA Fetch	Fetches data from the VNA after the measurement completes. Use this VI in conjunction with the niVNA Initiate VI to start the measurement.
niVNA Fetch Time Domain	Fetches time-domain data from the NI-VNA device after the measurement completes.
niVNA Abort	Stops a measurement previously started with the niVNA Initiate VI. Calling this VI is optional, unless you want to stop a measurement before it is complete. If this VI runs successfully, NI-VNA transitions to the Done state (if the sweep is complete) or the Committed state (if this call stops a sweep in progress).
niVNA Commit	Completes any remaining property validation and commits the settings to hardware. This VI commits all routes, and this action may cause glitching PXI or PFI lines. This VI also begins locking to a reference frequency if you have configured one using the Reference Frequency Source property. If the commit action is successful, NI-VNA transitions to the Committed state.
niVNA Wait For Sweep Done	Waits for the sweep measurement to complete. If the measurement does not complete within the specified **timeout**, this VI returns a timeout error. If this VI runs successfully, NI-VNA transitions to the Done state.
niVNA Is Sweep Done	Checks the status of the measurement. Use this VI to check for any errors that may

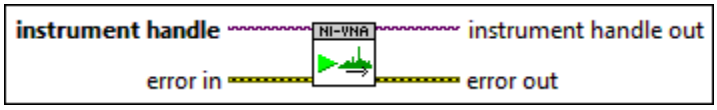
	occur during acquisition or to check whether the device is done measuring the sweep. If the sweep is complete, NI-VNA transitions to the Done state.
niVNA Fetch Receiver Measurement	Fetches the raw receiver data from the NI-VNA device after the measurement completes.

niVNA Initiate VI

Installed With: NI-VNA

Starts the measurement. Use this VI in conjunction with the niVNA Wait For Sweep Done VI to wait for the sweep measurement to complete and the niVNA Fetch VI to retrieve acquired data. If this VI runs successfully, NI-VNA transitions to the Running state.

Supported Devices: NI 5630/5632



I/O	**instrument handle** identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.
TF	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more

	information about the error displayed.
I32	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
abc	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

I/O	**instrument handle out** passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.

	The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.
TF	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
I32	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
abc	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

niVNA Fetch VI

Installed With: NI-VNA

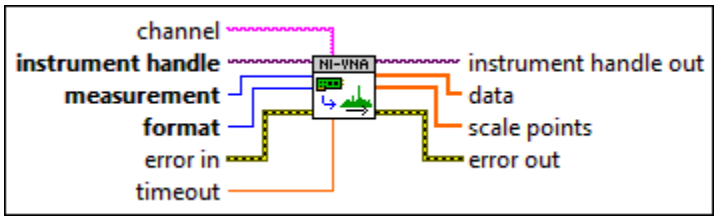
Fetches data from the VNA after the measurement completes. Use this VI in conjunction with the niVNA Initiate VI to start the measurement.

Supported Devices: NI 5630/5632

niVNA Fetch Sweep Measurement (DBL)

Fetches floating 64-bit data and fetches scale points from the last measurement.

Supported Devices: NI 5630/5632



I/O	**instrument handle** identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
I32	**measurement** specifies the measurement for which to fetch data. The NI 5630 supports only the **S11** and **S21** values. For more information about S-parameters, refer to the Vector Network Analyzer Measurements topic.

S11	NI-VNA fetches S11 measurement data.
S21	NI-VNA fetches S21 measurement data.
S12	NI-VNA fetches S12 measurement data.
S22	NI-VNA fetches S22 measurement data.

format specifies the format in which to fetch the data.

Log Magnitude (dB)	NI-VNA fetches the data in log magnitude format. NI-VNA log magnitude data is presented in decibels (20 log (data)).
Lin Magnitude	NI-VNA fetches the data in linear magnitude format. NI-VNA linear magnitude data is presented as a linear ratio.
Phase (deg)	NI-VNA fetches the data in phase format. NI-VNA phase data is presented in degrees of phase shift of the response (port 1 or 2) with respect to the stimulus (port 1). Phase data is presented in a wrapped format (+/-180 degree segments).
VSWR	NI-VNA fetches the data in voltage standing wave ratio (VSWR) format. NI-VNA magnitude data is presented as a linear ratio of (maximum RF voltage/ minimum RF voltage).
Group Delay	NI-VNA fetches the data in group delay format. Group delay expresses the rate of change of phase with respect to frequency (dp/df) over a specified range of frequency points, known as a window, expressed as time delay. The units are seconds.

channel specifies the channel for which this operation is targeted.

timeout specifies the time to wait for the VI to complete operation before returning an error. A negative value specifies that NI-VNA waits indefinitely. This value is expressed in seconds (s).

The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error

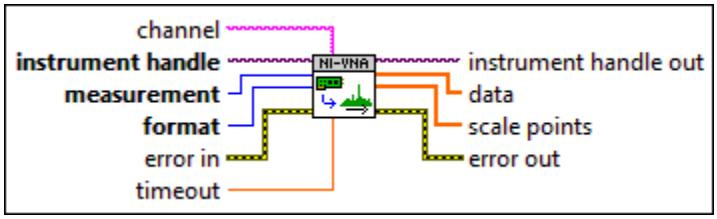
	or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	**instrument handle out** passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	**data** returns the measurement data.
	**scale points** returns the sweep points that were included in the measurement. For a frequency sweep operation, this parameter returns the frequencies at which **data** was taken. For a time-domain operation, this parameter returns the time or distance. For a power sweep operation, this parameter returns the power values at which the **data** was taken.
	The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.
	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

I32	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
abc	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

niVNA Fetch Sweep Measurement Complex (CDB)

Fetches complex data and scale points from the last measurement.

Supported Devices: NI 5630/5632



I/O	**instrument handle** identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
I32	**measurement** specifies the measurement for which to fetch data. The NI 5630 supports only the **S11** and **S21** values. For more information about S-parameters, refer to the Vector Network Analyzer Measurements topic.

S11	NI-VNA fetches S11 measurement data.
S21	NI-VNA fetches S21 measurement data.
S12	NI-VNA fetches S12 measurement data.
S22	NI-VNA fetches S22 measurement data.

format specifies the format in which to fetch the data.

Impedance	NI-VNA fetches the data in impedance format. The impedance is expressed in ohms. NI-VNA magnitude and phase data are presented as a complex real and imaginary Cartesian number pair in ohms.
Coefficient	NI-VNA fetches the data as the reflection (for S11) or transmission (for S22) coefficient (gamma). The coefficient is related to impedance according to the formula ***Impedance = Zo * (1 + gamma)/(1 - gamma)***, where ***Zo*** is assumed to be 50.

channel specifies the channel for which this operation is targeted.

timeout specifies the time to wait for the VI to complete operation before returning an error. A negative value specifies that NI-VNA waits indefinitely. This value is expressed in seconds (s).

The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

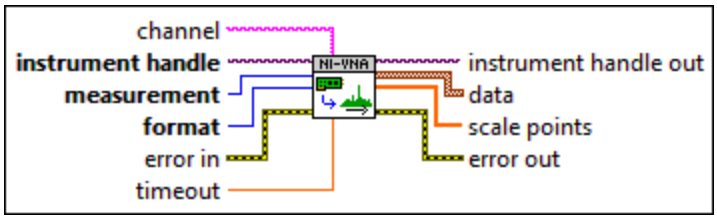
	**instrument handle out** passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	**data** returns the measurement data.
	**scale points** returns the sweep points that were included in the measurement. For a frequency sweep operation, this parameter returns the frequencies at which **data** was taken. For a time-domain operation, this parameter returns the time or distance. For a power sweep operation, this parameter returns the power values at which the **data** was taken.
	The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

niVNA Fetch Sweep Measurement Complex (Cluster)

Fetches complex data and scale points from the last measurement.

Supported Devices: NI 5630/5632



instrument handle identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.

measurement specifies the measurement for which to fetch data. The NI 5630 supports only the **S11** and **S21** values. For more information about S-parameters, refer to the Vector Network Analyzer Measurements topic.

S11	NI-VNA fetches S11 measurement data.
S21	NI-VNA fetches S21 measurement data.
S12	NI-VNA fetches S12 measurement data.
S22	NI-VNA fetches S22 measurement data.

format specifies the format in which to fetch the data.

Impedance	NI-VNA fetches the data in impedance format. The impedance is expressed in ohms. NI-VNA magnitude and phase data are presented as a complex real and imaginary Cartesian number pair in ohms.
Coefficient	NI-VNA fetches the data as the reflection (for S11) or transmission (for S22) coefficient (gamma). The coefficient is related to impedance according to the formula ***Impedance = Zo * (1 + gamma)/(1 - gamma)***, where ***Zo*** is assumed to be 50.

channel specifies the channel for which this operation is targeted.

timeout specifies the time to wait for the VI to complete operation before returning an error.

	A negative value specifies that NI-VNA waits indefinitely. This value is expressed in seconds (s).
	The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

TF	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
I32	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
abc	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

	instrument handle out passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	data returns the measurement data.

DBL	**real** contains the real part of the measurement data.
DBL	**imaginary** contains the imaginary part of the measurement data.

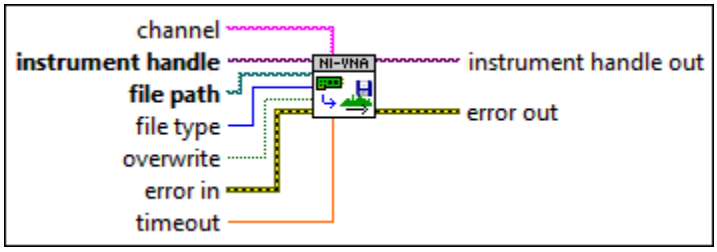
DBL	**scale points** returns the sweep points that were included in the measurement. For a frequency sweep operation, this parameter returns the frequencies at which **data** was taken. For a time-domain operation, this parameter returns the time or distance. For a power sweep operation, this parameter returns the power values at which the **data** was taken.
	The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

TF	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
I32	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
abc	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

niVNA Fetch Sweep Measurement Into File

Fetches the data from the device directly into a one- or two-port S-parameter file from the latest measurement.

Supported Devices: NI 5630/5632



	**instrument handle** identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	**file path** specifies the absolute file path from which to save the file.
	**file type** specifies the type of file into which to save the data. **S1P** — The file is saved in a one-port S-parameter file. **S2P** — The file is saved in a two-port S-parameter file.
	**overwrite** specifies whether to overwrite an existing data file, if one exists.
	**channel** specifies the channel for which this operation is targeted.
	**timeout** specifies the time to wait for the VI to complete operation before returning an error. A negative value specifies that NI-VNA waits indefinitely. This value is expressed in seconds (s).
	The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.  The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

	<table><tr><td>  </td><td>The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.</td></tr><tr><td>  </td><td>The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.</td></tr></table>
	**instrument handle out** passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.<table><tr><td>  </td><td>The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.</td></tr><tr><td>  </td><td>The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.</td></tr><tr><td>  </td><td>The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.</td></tr></table>

niVNA Fetch Time Domain VI

Installed With: NI-VNA

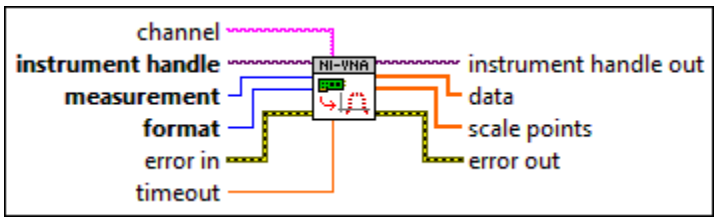
Fetches time-domain data from the NI-VNA device after the measurement completes.

Supported Devices: NI 5630/5632

niVNA Fetch Time Domain Gate Shape (DBL)

Fetches floating 64-bit data and scale points from the time-domain analysis gate shape.

Supported Devices: NI 5630/5632



I/O	**instrument handle** identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
I32	**measurement** specifies the measurement for which to fetch data. The NI 5630 supports only the **S11** and **S21** values. For more information about S-parameters, refer to the Vector Network Analyzer Measurements topic.

S11	NI-VNA fetches S11 measurement data.
S21	NI-VNA fetches S21 measurement data.
S12	NI-VNA fetches S12 measurement data.
S22	NI-VNA fetches S22 measurement data.

I32 **format** specifies the format in which to fetch the data.

Log Magnitude	NI-VNA fetches the data in log magnitude format. NI-VNA log magnitude data is presented in decibels (20 log (data)).

(dB)	
Lin Magnitude	NI-VNA fetches the data in linear magnitude format. NI-VNA linear magnitude data is presented as a linear ratio.
Phase (deg)	NI-VNA fetches the data in phase format. NI-VNA phase data is presented in degrees of phase shift of the response (port 1 or 2) with respect to the stimulus (port 1). Phase data is presented in a wrapped format (+/-180 degree segments).
VSWR	NI-VNA fetches the data in voltage standing wave ratio (VSWR) format. NI-VNA magnitude data is presented as a linear ratio of (maximum RF voltage/ minimum RF voltage).
Real	NI-VNA fetches the real component of the complex data.



channel specifies the channel for which this operation is targeted.



timeout specifies the time to wait for the VI to complete operation before returning an error. A negative value specifies that NI-VNA waits indefinitely. This value is expressed in seconds (s).



The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

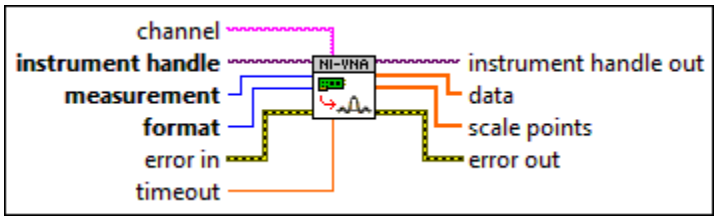
	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	**instrument handle out** passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	**scale points** returns the sweep points that were included in the measurement. For a frequency sweep operation, this parameter returns the frequencies at which **data** was taken. For a time-domain operation, this parameter returns the time or distance. For a power sweep operation, this parameter returns the power values at which the **data** was taken.
	**data** returns the gate shape data.
	The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.
	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

niVNA Fetch Time Domain Transform (DBL)

Fetches floating 64-bit data and scale points from the time-domain analysis transform.

Supported Devices: NI 5630/5632



I/O	**instrument handle** identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
abc	**channel** specifies the channel for which this operation is targeted.
I32	**measurement** specifies the measurement for which to fetch data. The NI 5630 supports only the **S11** and **S21** values. For more information about S-parameters, refer to the Vector Network Analyzer Measurements topic.

S11	NI-VNA fetches S11 measurement data.
S21	NI-VNA fetches S21 measurement data.
S12	NI-VNA fetches S12 measurement data.
S22	NI-VNA fetches S22 measurement data.

I32	**format** specifies the format in which to fetch the data.

Log Magnitude (dB)	NI-VNA fetches the data in log magnitude format. NI-VNA log magnitude data is presented in decibels (20 log (data)).

Lin Magnitude	NI-VNA fetches the data in linear magnitude format. NI-VNA linear magnitude data is presented as a linear ratio.
Phase (deg)	NI-VNA fetches the data in phase format. NI-VNA phase data is presented in degrees of phase shift of the response (port 1 or 2) with respect to the stimulus (port 1). Phase data is presented in a wrapped format (+/-180 degree segments).
VSWR	NI-VNA fetches the data in voltage standing wave ratio (VSWR) format. NI-VNA magnitude data is presented as a linear ratio of (maximum RF voltage/ minimum RF voltage).
Real	NI-VNA fetches the real component of the complex data.



timeout specifies the time to wait for the VI to complete operation before returning an error. A negative value specifies that NI-VNA waits indefinitely. This value is expressed in seconds (s).



The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

	**instrument handle out** passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	**scale points** returns the sweep points that were included in the measurement. For a frequency sweep operation, this parameter returns the frequencies at which **data** was taken. For a time-domain operation, this parameter returns the time or distance. For a power sweep operation, this parameter returns the power values at which the **data** was taken.
	**data** returns the measurement data.
	The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

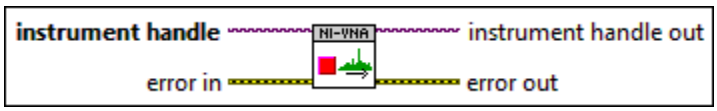
niVNA Abort VI

Installed With: NI-VNA

Stops a measurement previously started with the niVNA Initiate VI. Calling this VI is

optional, unless you want to stop a measurement before it is complete. If this VI runs successfully, NI-VNA transitions to the Done state (if the sweep is complete) or the Committed state (if this call stops a sweep in progress).

Supported Devices: NI 5630/5632



I/O	**instrument handle** identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.
TF	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
I32	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
abc	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
I/O	**instrument handle out** passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.

	The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

niVNA Commit VI

Installed With: NI-VNA

Completes any remaining property validation and commits the settings to hardware. This VI commits all routes, and this action may cause glitching PXI or PFI lines. This VI also begins locking to a reference frequency if you have configured one using the Reference Frequency Source property. If the commit action is successful, NI-VNA transitions to the Committed state.

Calling this VI before you call the niVNA Initiate VI is optional, but it has the following benefits:

- Routes are committed, so signals are imported.
- Any Reference Clock and external clock circuits are phase-locked.
- Any subsequent niVNA Initiate VI calls can run faster because the device is already configured.

Supported Devices: NI 5630/5632



I/O	**instrument handle** identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.
TF	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
I32	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
abc	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
I/O	**instrument handle out** passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

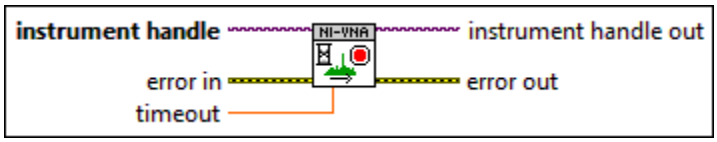
TF	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
I32	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
abc	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

niVNA Wait For Sweep Done VI

Installed With: NI-VNA

Waits for the sweep measurement to complete. If the measurement does not complete within the specified **timeout**, this VI returns a timeout error. If this VI runs successfully, NI-VNA transitions to the Done state.

Supported Devices: NI 5630/5632



I/O	**instrument handle** identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
DBL	**timeout** specifies the time to wait before returning an error if the calibration is not complete. A negative value specifies that NI-VNA waits indefinitely. This value is expressed in seconds

	(s).
	The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed. TF — The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed. I32 — The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed. abc — The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
I/O	**instrument handle out** passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed. TF — The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

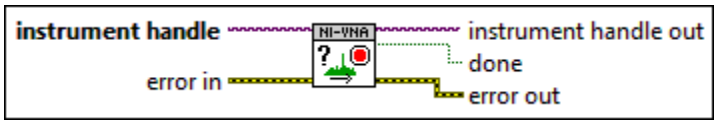
	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

niVNA Is Sweep Done VI

Installed With: NI-VNA

Checks the status of the measurement. Use this VI to check for any errors that may occur during acquisition or to check whether the device is done measuring the sweep. If the sweep is complete, NI-VNA transitions to the Done state.

Supported Devices: NI 5630/5632



	instrument handle identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

	<table><tr><td>  </td><td>The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.</td></tr><tr><td>  </td><td>The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.</td></tr><tr><td>  </td><td>The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.</td></tr></table>
	**instrument handle out** passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	**done** returns whether the operation is complete.
	The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.<table><tr><td>  </td><td>The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.</td></tr><tr><td>  </td><td>The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.</td></tr></table>

	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

niVNA Fetch Receiver Measurement VI

Installed With: NI-VNA

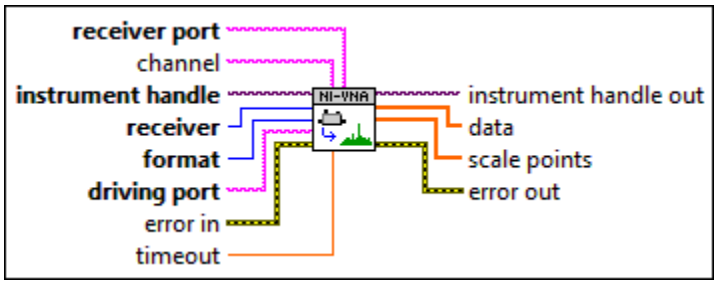
Fetches the raw receiver data from the NI-VNA device after the measurement completes.

Supported Devices: NI 5632

niVNA Fetch Receiver Measurement (DBL)

Fetches the receiver measurement and scale points in floating 64-bit data from the last measurement.

Supported Devices: NI 5632



	instrument handle identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	receiver specifies the receiver on which the measurement is performed. For more

information about test and reference receivers, refer to the Test and Reference Receiver Calibration topic.

Reference Receiver	NI-VNA fetches the raw receiver measurements from the reference receiver.
Test Receiver	NI-VNA fetches the raw receiver measurements from the test receiver.



format specifies the format in which to fetch the data.

Log Magnitude (dB)	NI-VNA fetches the data in log magnitude format. NI-VNA log magnitude data is presented in decibels (20 log (data)).
Lin Magnitude	NI-VNA fetches the data in linear magnitude format. NI-VNA linear magnitude data is presented as a linear ratio.
Phase (deg)	NI-VNA fetches the data in phase format. NI-VNA phase data is presented in degrees of phase shift of the response (port 1 or 2) with respect to the stimulus (port 1). Phase data is presented in a wrapped format (+/-180 degree segments).
VSWR	NI-VNA fetches the data in voltage standing wave ratio (VSWR) format. NI-VNA magnitude data is presented as a linear ratio of (maximum RF voltage/ minimum RF voltage).
Group Delay	NI-VNA fetches the data in group delay format. Group delay expresses the rate of change of phase with respect to frequency (dp/df) over a specified range of frequency points, known as a window, expressed as time delay. The units are seconds.



driving port specifies the port from which the power is sourced. Valid driving port specifiers are "`port1`" or "`port2`".



receiver port specifies the port from which the measurement is performed. Valid receiver port specifiers are "`port1`" or "`port2`".

	channel specifies the channel for which this operation is targeted.
	timeout specifies the time to wait for the VI to complete operation before returning an error. A negative value specifies that NI-VNA waits indefinitely. This value is expressed in seconds (s).
	The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.
	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	instrument handle out passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	data returns the measurement data.
	scale points returns the sweep points that were included in the measurement. For a frequency sweep operation, this parameter returns the frequencies at which **data** was taken.

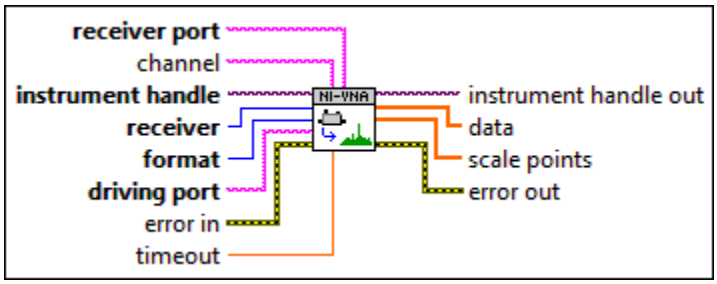
	For a time-domain operation, this parameter returns the time or distance. For a power sweep operation, this parameter returns the power values at which the **data** was taken.
	The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

TF	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
I32	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
abc	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

niVNA Fetch Receiver Measurement Complex (CDB)

Fetches the receiver measurement in complex data and fetches scale points from the last sweep measurement.

Supported Devices: NI 5632



	**instrument handle** identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	**receiver** specifies the receiver on which the measurement is performed. For more information about test and reference receivers, refer to the Test and Reference Receiver Calibration topic. **Reference Receiver** — NI-VNA fetches the raw receiver measurements from the reference receiver. **Test Receiver** — NI-VNA fetches the raw receiver measurements from the test receiver.
	**format** specifies the format in which to fetch the data. **Impedance** — NI-VNA fetches the data in impedance format. The impedance is expressed in ohms. NI-VNA magnitude and phase data is presented as a complex real and imaginary Cartesian number pair in ohms. **Coefficient** — NI-VNA fetches the data as the reflection (for S11 and S22) or transmission (for S21 and S12) coefficient (gamma). The coefficient is related to impedance according to the formula ***Impedance = Zo * (1 + gamma)/(1 - gamma)***, where ***Zo*** is assumed to be 50 ohms.
	**driving port** specifies the port from which the power is sourced. Valid driving port specifiers are "`port1`" or "`port2`".
	**receiver port** specifies the port from which the measurement is performed. Valid receiver port specifiers are "`port1`" or "`port2`".
	**channel** specifies the channel for which this operation is targeted.

DBL	**timeout** specifies the time to wait for the VI to complete operation before returning an error. A negative value specifies that NI-VNA waits indefinitely. This value is expressed in seconds (s).
	The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.
TF	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
I32	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
abc	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
I/O	**instrument handle out** passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
CDB	**data** returns the measurement data.
DBL	**scale points** returns the sweep points that were included in the measurement. For a frequency sweep operation, this parameter returns the frequencies at which **data** was taken. For a time-domain operation, this parameter returns the time or distance. For a power sweep operation, this parameter returns the power values at which the **data** was taken.

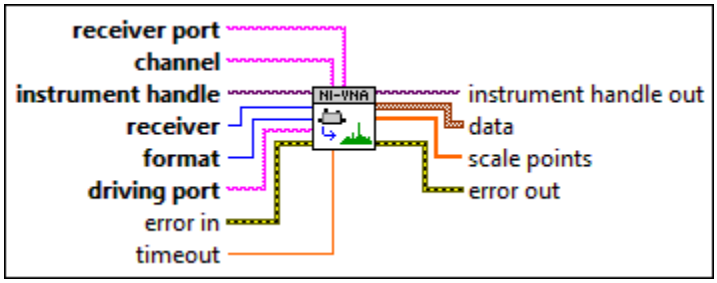
The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

TF	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
I32	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
abc	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

niVNA Fetch Receiver Measurement Complex (Cluster)

Fetches the receiver measurement in complex data and fetches scale points from the last sweep measurement.

Supported Devices: NI 5632



I/O	**instrument handle** identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.



receiver specifies the receiver on which the measurement is performed. For more information about test and reference receivers, refer to the Test and Reference Receiver Calibration topic.

Reference Receiver	NI-VNA fetches the raw receiver measurements from the reference receiver.
Test Receiver	NI-VNA fetches the raw receiver measurements from the test receiver.



format specifies the format in which to fetch the data.

Impedance	NI-VNA fetches the data in impedance format. The impedance is expressed in ohms. NI-VNA magnitude and phase data is presented as a complex real and imaginary Cartesian number pair in ohms.
Coefficient	NI-VNA fetches the data as the reflection (for S11 and S22) or transmission (for S21 and S12) coefficient (gamma). The coefficient is related to impedance according to the formula ***Impedance = Zo * (1 + gamma)/(1 - gamma)***, where ***Zo*** is assumed to be 50 ohms.



driving port specifies the port from which the power is sourced. Valid driving port specifiers are "`port1`" or "`port2`".



channel specifies the channel for which this operation is targeted.



receiver port specifies the port from which the measurement is performed. Valid receiver port specifiers are "`port1`" or "`port2`".



timeout specifies the time to wait for the VI to complete operation before returning an error. A negative value specifies that NI-VNA waits indefinitely. This value is expressed in seconds (s).

The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



instrument handle out passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.

data returns the measurement data.

	**real** contains the real part of the measurement data.
	**imaginary** contains the imaginary part of the measurement data.





scale points returns the sweep points that were included in the measurement. For a frequency sweep operation, this parameter returns the frequencies at which **data** was taken. For a time-domain operation, this parameter returns the time or distance. For a power sweep

	operation, this parameter returns the power values at which the **data** was taken.
	The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

Calibration

Owning Palette: NI-VNA VIs**Installed With:** NI-VNA

Controls calibration operations for the NI-VNA device.

Palette Object	Description
niVNA Init User Cal	Initializes user calibration.
niVNA Perform Calibration Action	Performs one calibration action on the specified port list or standard.

niVNA Abort Calibration Action	Aborts the user calibration action that is in progress.
niVNA Close User Cal	Closes the user calibration session.
niVNA Is Calibration Action Done	Checks the status of the user calibration action. Use this VI to check for any errors that may occur during the calibration action or to check whether the last calibration action is complete.
niVNA Wait For Calibration Action Done	Waits for the last user calibration action to complete. If the calibration action does not complete before the specified timeout, this VI returns a timeout error.
niVNA Get Calibration State	Returns the state of the user calibration.
niVNA Apply Adapter Removal Calibration	Loads the calibration files, specifies the adapter length, and prepares the device for performing adapter removal calibration.
niVNA Load Calibration	Loads calibration and configuration data from a specified file.
niVNA Save Calibration	Saves calibration and configuration data to a file in a specified location.
niVNA Load User Defined Manual Calibration Kit Data From File	Loads the user-defined manual calibration characterization data.

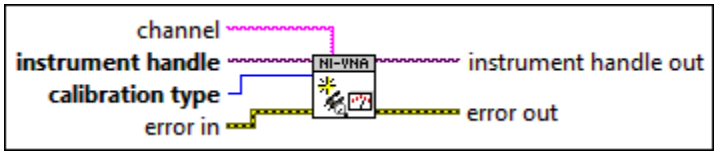
Subpalette	Description
AutoCal	Configures the calibration operation when you are using a supported AutoCal box.

niVNA Init User Cal VI

Installed With: NI-VNA

Initializes user calibration.

Supported Devices: NI 5630/5632



I/O	**instrument handle** identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
I32	**calibration type** specifies the type of calibration that is performed.

2 Port SOLT	NI-VNA is configured to perform a two-port, one-path short/open/load/through (SOLT) calibration.
1 Port SOL	NI-VNA is configured to perform a one-port short/open/load (SOL) calibration.
Reflection Response	NI-VNA is configured to perform a reflection-response calibration.
Transmission Response	NI-VNA is configured to perform a transmission-response calibration.
TRL	NI-VNA is configured to perform a full two-port through-reflect-line (TRL) calibration.
Power	NI-VNA is configured to perform a power calibration, including source and receiver calibration.
Adapter Removal	NI-VNA is configured to apply a calibration that de-embeds an adapter.

	channel specifies the channel for which this operation is targeted.
	The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed. The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed. The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed. The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	instrument handle out passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed. The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

I32	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
abc	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

niVNA Perform Calibration Action VI

Installed With: NI-VNA

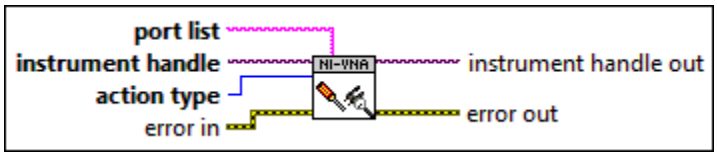
Performs one calibration action on the specified port list or standard.

Supported Devices: NI 5630/5632

niVNA Perform Calibration Action (Power)

Performs one power calibration action on the specified **port list**.

Supported Devices: NI 5632



I/O	**instrument handle** identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
I32	**action type** specifies the type of calibration action to perform.

Source Power Cal	NI-VNA performs a source power calibration using a power meter.
Test Receiver Cal	NI-VNA performs a test receiver power calibration using a through standard.
Reference Receiver Cal	NI-VNA performs a reference receiver power calibration using a match load standard or a match connection.



port list specifies the ports on which to perform the specified calibration action. If you set **action type** to **Source Power Cal** or **Reference Receiver Cal**, the valid port list specifier is "`port1`" or "`port2`". If you set **action type** to **Test Receiver Cal**, the valid port list specifier is "`port1,port2`" to use the source on PORT 1 and calibrate the receiver on PORT 2 or "`port2,port1`" to use the source on PORT 2 and calibrate the receiver on PORT 1.



The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



instrument handle out passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With

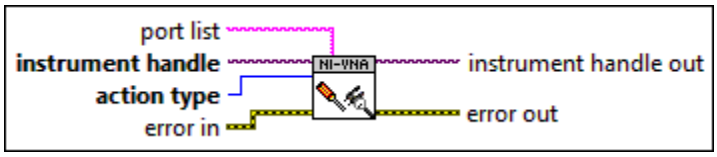
	Options VI and identifies a particular instrument session.
	The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

TF	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
I32	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
abc	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

niVNA Perform Calibration Action (SOLT)

Performs one short/open/load/through (SOLT) calibration action on the specified **port list**.

Supported Devices: NI 5630/5632



I/O	**instrument handle** identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.

action type specifies the type of calibration action to perform.

Open	NI-VNA performs a calibration using the manual open standard.
Short	NI-VNA performs a calibration using the manual short standard.
Load	NI-VNA performs a calibration using the manual load standard.
Thru	NI-VNA performs a calibration using the manual through standard.
Isolation	NI-VNA performs a calibration using the manual load standard to measure leakage from two terminated ports.
Standard AutoCal	NI-VNA performs a standard automatic calibration using an AutoCal module.
AutoCal with True Thru	NI-VNA performs a standard automatic calibration using an AutoCal module and an external through connection between the port cables.

port list specifies the ports on which to perform the specified calibration action.

The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

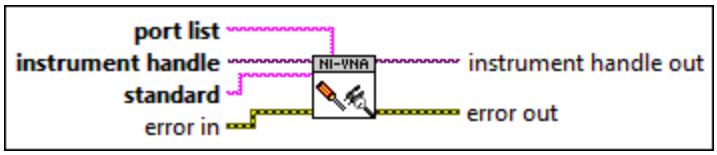
	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **source** string describes the origin of the error or warning. The pop-up option

	Explain Error (or Explain Warning) gives more information about the error displayed.
I/O	**instrument handle out** passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.
TF	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
I32	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
abc	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

niVNA Perform Calibration Action (TRL)

Performs one through-reflect-line (TRL) calibration action on the specified **port list** and **standard**.

Supported Devices: NI 5632



	instrument handle identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	standard specifies the standard on which to perform the calibration action. Valid standard specifiers are "standard1", "standard2", "standard3", "standard4", "standard5", or "standard6".
	port list specifies the ports on which to perform the specified calibration action. Valid port list specifiers are "port1", "port2", or "port1,port2". For more information about valid specifiers, refer to the TRL Calibration (niVNA_TRL_Cal.vi) Example VI.
	The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **source** string describes the origin of the error or warning. The pop-up option

	Explain Error (or Explain Warning) gives more information about the error displayed.

I/O	**instrument handle out** passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.

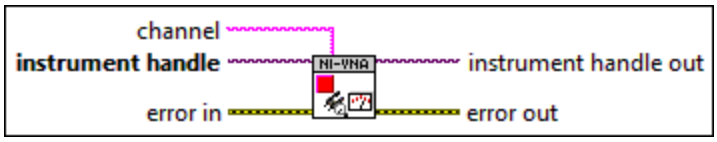
	The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.
TF	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
I32	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
abc	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

niVNA Abort Calibration Action VI

Installed With: NI-VNA

Aborts the user calibration action that is in progress.

Supported Devices: NI 5630/5632



	instrument handle identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	channel specifies the channel for which this operation is targeted.
	The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

	instrument handle out passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

	The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

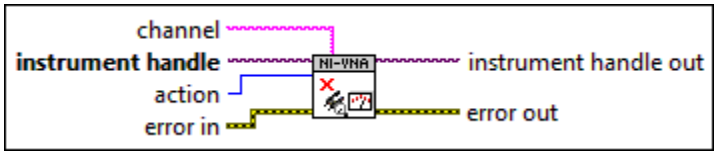
TF	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
I32	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
abc	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

niVNA Close User Cal VI

Installed With: NI-VNA

Closes the user calibration session.

Supported Devices: NI 5630/5632



I/O	**instrument handle** identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
I32	**action** specifies the action to perform when closing the user calibration.

	Abort — NI-VNA discards the calibration constants calculated during the user calibration. **Commit** — NI-VNA commits the calibration constants to the device and transitions the device to the Calibrated calibration state. After you commit the values, you should not change some device properties for this session.
	**channel** specifies the channel for which this operation is targeted.
	The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.  The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.  The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.  The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	**instrument handle out** passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

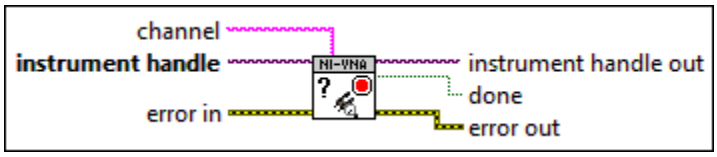
TF	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
I32	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
abc	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

niVNA Is Calibration Action Done VI

Installed With: NI-VNA

Checks the status of the user calibration action. Use this VI to check for any errors that may occur during the calibration action or to check whether the last calibration action is complete.

Supported Devices: NI 5630/5632



I/O	**instrument handle** identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
abc	**channel** specifies the channel for which this operation is targeted.

	The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed. **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed. **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed. **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	instrument handle out passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	done returns whether the operation is complete.
	The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed. **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

I32	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
abc	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

niVNA Wait For Calibration Action Done VI

Installed With: NI-VNA

Waits for the last user calibration action to complete. If the calibration action does not complete before the specified timeout, this VI returns a timeout error.

Supported Devices: NI 5630/5632

niVNA Wait For Calibration Action Done

I/O	**instrument handle** identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
abc	**channel** specifies the channel for which this operation is targeted.
DBL	**timeout** specifies the time to wait before returning an error if the calibration is not complete. A negative value specifies that NI-VNA waits indefinitely. This value is expressed in seconds (s).

	The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed. **TF** The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed. **I32** The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed. **abc** The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	**instrument handle out** passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed. **TF** The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed. **I32** The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

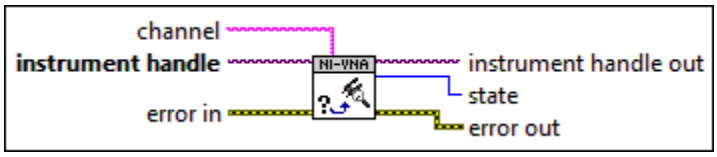
	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

niVNA Get Calibration State VI

Installed With: NI-VNA

Returns the state of the user calibration.

Supported Devices: NI 5630/5632



	instrument handle identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	channel specifies the channel for which this operation is targeted.
	The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.
	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more

	information about the error displayed.
I32	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
abc	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.

I32 **state** returns the state of the calibration.

The relationship between the Correction Enabled property and the **calibration state** parameter is shown in the following table.

Correction Enabled	Calibration State		
	Uncalibrated	**Calibrated**	**Calibrating**
FALSE	NI-VNA does ***not*** have a valid set of calibration data for the current device configuration. Measurement data is ***not*** corrected using calibration data.	NI-VNA ***does*** have a valid set of calibration data for the current device configuration; however, measurement data is ***not*** corrected using calibration data.	N/A
TRUE	NI-VNA does ***not*** have a valid set of calibration data for the current device configuration; however, measurement data ***is*** modified using (potentially invalid)	NI-VNA ***does*** have a valid set of calibration data for the current device configuration. Measurement data ***is*** corrected using calibration	N/A

	calibration data.	data.	

The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

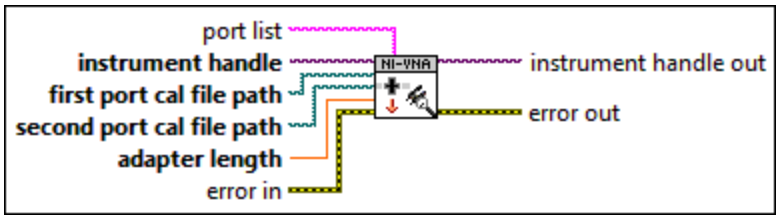
TF	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
I32	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
abc	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

niVNA Apply Adapter Removal Calibration VI

Installed With: NI-VNA

Loads the calibration files, specifies the adapter length, and prepares the device for performing adapter removal calibration.

Supported Devices: NI 5632



	instrument handle identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	first port cal file path specifies the saved calibration file path with the adapter connected to the first port in the **port list**.
	second port cal file path specifies the saved calibration file path with the adapter connected to the second port in the **port list**.
	adapter length specifies the electrical length of the adapter, in seconds.
	port list specifies the ports on which to perform the specified calibration action.
	The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or

	Explain Warning) gives more information about the error displayed.
	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

	instrument handle out passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.

	The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.
	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

niVNA Load Calibration VI

Installed With: NI-VNA

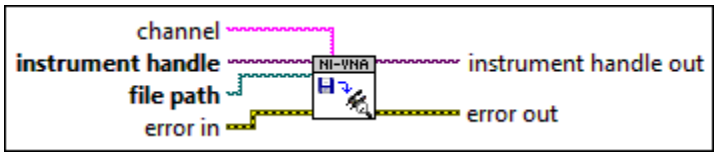
Loads calibration and configuration data from a specified file.

Supported Devices: NI 5630/5632

niVNA Load Calibration And Configuration Data From File

Loads user calibration and configuration data from a file.

Supported Devices: NI 5630/5632



I/O	**instrument handle** identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
path	**file path** specifies the absolute file path from which to load the user calibration data.
abc	**channel** specifies the channel for which this operation is targeted.
cluster	The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

TF	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
I32	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
I/O	**instrument handle out** passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.
TF	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
I32	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
abc	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

niVNA Save Calibration VI

Installed With: NI-VNA

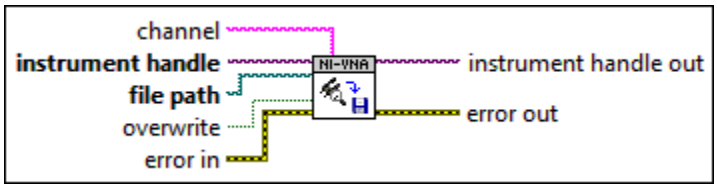
Saves calibration and configuration data to a file in a specified location.

Supported Devices: NI 5630/5632

niVNA Save Calibration And Configuration Data To File

Saves the current user calibration and configuration data into a file.

Supported Devices: NI 5630/5632



I/O	**instrument handle** identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
path	**file path** specifies the absolute file path from which to save the user calibration data.
TF	**overwrite** specifies whether to overwrite an existing calibration data file, if one exists.
abc	**channel** specifies the channel for which this operation is targeted.
error	The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

TF	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
I32	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or

	Explain Warning) gives more information about the error displayed.
	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	instrument handle out passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.
	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

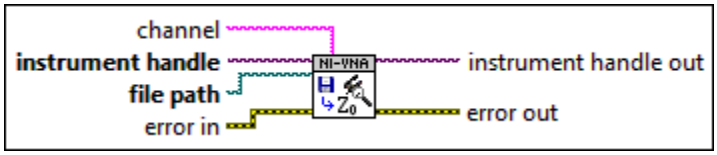
niVNA Load User Defined Manual Calibration Kit Data From File VI

Installed With: NI-VNA

Loads the user-defined manual calibration characterization data.

Specify the user-defined manual calibration kit data using the NI-VNA SFP. Refer to the User Calibration topic for more information about specifying manual calibration kit data.

Supported Devices: NI 5630/5632



I/O	**instrument handle** identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
path	**file path** specifies the absolute file path from which to load the user-defined manual calibration kit data.
abc	**channel** specifies the channel for which this operation is targeted.
cluster	The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

TF	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
I32	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
I/O	**instrument handle out** passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.
TF	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
I32	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
abc	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

AutoCal

Owning Palette: Calibration**Installed With:** NI-VNA

Configures the calibration operation when you are using a supported AutoCal box.

Palette Object	Description

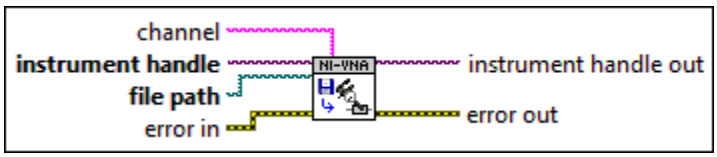
niVNA Load AutoCal Characterization Data From File	Loads the AutoCal calibration characterization data.
niVNA Query AutoCal Box Info	Queries the AutoCal module information, including the **serial number** and **type**.

niVNA Load AutoCal Characterization Data From File VI

Installed With: NI-VNA

Loads the AutoCal calibration characterization data.

Supported Devices: NI 5630/5632



	instrument handle identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	file path specifies the absolute file path from which to load the AutoCal characterization data.
	channel specifies the channel for which this operation is targeted.
	The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information

about the error displayed.

TF	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
I32	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
abc	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.

The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

TF	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
I32	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
abc	The **source** string describes the origin of the error or warning. The pop-up option

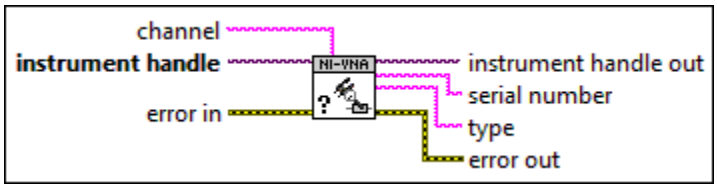
	Explain Error (or Explain Warning) gives more information about the error displayed.

niVNA Query AutoCal Box Info VI

Installed With: NI-VNA

Queries the AutoCal module information, including the **serial number** and **type**.

Supported Devices: NI 5630/5632



I/O	**instrument handle** identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
abc	**channel** specifies the channel for which this operation is targeted.
	The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

TF	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	instrument handle out passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	serial number returns the serial number of the AutoCal module.
	type returns the type of the AutoCal module.
	The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.
	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

Utility

Owning Palette: NI-VNA VIs**Installed With:** NI-VNA

Use the NI-VNA Utility VIs to access utility features of NI-VNA.

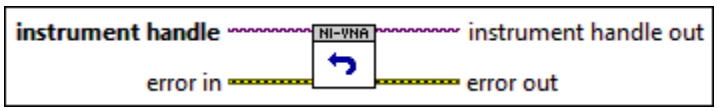
Palette Object	Description
niVNA Reset	Resets the device to a known initialization state.
niVNA Reset With Defaults	Resets the device to a known initialization state with defaults as specified for the device in the **option string** parameter in the niVNA Initialize With Options VI.
niVNA Disable	Places the instrument in a quiescent state where it has minimal or no impact on the system to which it is connected.  **Note** This VI is currently not supported.
niVNA Self Test	Performs a self-test on the NI-VNA device and returns the test result. This VI performs a simple series of tests verifying that the NI-VNA device is powered on and responding.
niVNA Revision Query	Queries the revision of the device.
niVNA Error Message	Converts an **error code** returned by an NI-VNA VI into a user-readable string.
niVNA Activate Feature	Activates a feature for the specified device.
niVNA Query Feature Activation Status	Queries the activation status of a certain feature.

niVNA Reset VI

Installed With: NI-VNA

Resets the device to a known initialization state.

Supported Devices: NI 5630/5632



I/O	**instrument handle** identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.
TF	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
I32	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
abc	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
I/O	**instrument handle out** passes a reference to your instrument session to the next VI.

	instrument handle is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

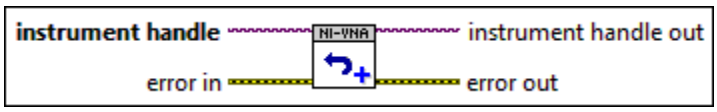
TF	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
I32	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
abc	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

niVNA Reset With Defaults VI

Installed With: NI-VNA

Resets the device to a known initialization state with defaults as specified for the device in the **option string** parameter in the niVNA Initialize With Options VI.

Supported Devices: NI 5630/5632



I/O	**instrument handle** identifies your instrument session. **instrument handle** is obtained from

	the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.
	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	instrument handle out passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.
	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

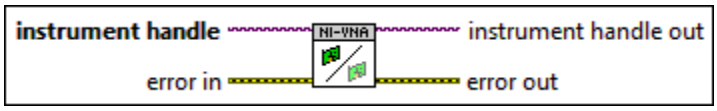
niVNA Disable VI

Installed With: NI-VNA

Places the instrument in a quiescent state where it has minimal or no impact on the system to which it is connected.

	Note This VI is currently not supported.

Supported Devices: NI 5630/5632



	instrument handle identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

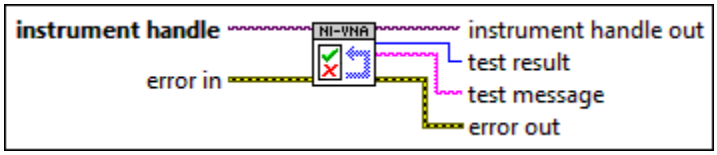
	<table><tr><td>  </td><td>The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.</td></tr><tr><td>  </td><td>The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.</td></tr><tr><td>  </td><td>The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.</td></tr></table>
	**instrument handle out** passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.<table><tr><td>  </td><td>The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.</td></tr><tr><td>  </td><td>The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.</td></tr><tr><td>  </td><td>The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.</td></tr></table>

niVNA Self Test VI

Installed With: NI-VNA

Performs a self-test on the NI-VNA device and returns the test result. This VI performs a simple series of tests verifying that the NI-VNA device is powered on and responding.

Supported Devices: NI 5630/5632



I/O	**instrument handle** identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

TF	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
I32	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
abc	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

	**instrument handle out** passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	**test result** returns the value from the device self-test. A value of 0 indicates that the device is powered on and responding. A value of 1 indicates that the device failed the self test.
	**test message** returns the self-test response string from the NI-VNA device.
	The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

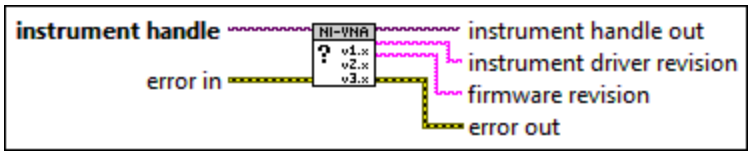
	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

niVNA Revision Query VI

Installed With: NI-VNA

Queries the revision of the device.

Supported Devices: NI 5630/5632



I/O	**instrument handle** identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

TF	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
I32	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
abc	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

I/O	**instrument handle out** passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
abc	**instrument driver revision** returns the instrument driver revision information.
abc	**firmware revision** returns firmware revision information.

	The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

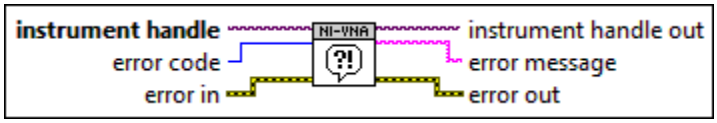
TF	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
I32	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
abc	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

niVNA Error Message VI

Installed With: NI-VNA

Converts an **error code** returned by an NI-VNA VI into a user-readable string.

Supported Devices: NI 5630/5632



I/O	**instrument handle** identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.

	**error code** passes the status code that is returned from any NI-VNA VIs. The default value is 0.
	The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed. **status** —  The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed. **code** —  The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed. **source** —  The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	**instrument handle out** passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	**error message** returns the user-readable message string that corresponds to the status code specified in **error code**.
	The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

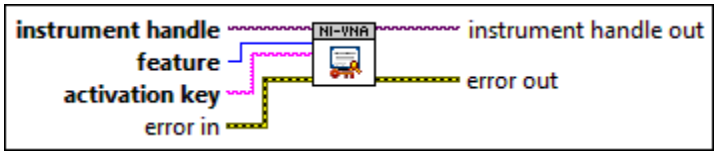
TF	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
I32	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
abc	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

niVNA Activate Feature VI

Installed With: NI-VNA

Activates a feature for the specified device.

Supported Devices: NI 5630/5632



I/O	**instrument handle** identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
I32	**feature** specifies the feature for which NI-VNA queries the activation status. **Time Domain** — Activates time-domain analysis on the device.

	activation key specifies the key you use to activate certain features. NI provides this key after you purchase the option.
	The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

	instrument handle out passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
	The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

	The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more

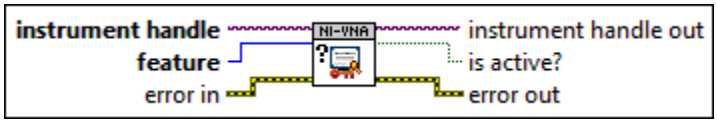
	information about the error displayed.
I32	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
abc	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

niVNA Query Feature Activation Status VI

Installed With: NI-VNA

Queries the activation status of a certain feature.

Supported Devices: NI 5630/5632



I/O	**instrument handle** identifies your instrument session. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
I32	**feature** specifies the feature for which NI-VNA queries the activation status. **Time Domain** — Activates time-domain analysis on the device.

	The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed. TF — The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed. I32 — The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed. abc — The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
I/O	**instrument handle out** passes a reference to your instrument session to the next VI. **instrument handle** is obtained from the niVNA Initialize VI or the niVNA Initialize With Options VI and identifies a particular instrument session.
TF	**is active?** returns the activation status of the feature.
	The **error out** cluster passes error or warning information out of a VI to be used by other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed. TF — The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

	The **code** input identifies the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.
	The **source** string describes the origin of the error or warning. The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

niVNA Properties

September 2014, 373155D-01

Use the NI-VNA properties to access advanced configuration options for VNA applications.

Property	Description
Active Channel	Specifies the channel name used to access all subsequent properties in this instance of the property node. If the property you want to use is channel- or port-based, you must first select this property and then pass the name of the specific channel or port. If the property you specify is not channel- or port-based, pass an empty string, or omit setting this property. The default value is " ". Details
Trigger:Start Trigger Type	Specifies the Start Trigger type. Details
Trigger:Start Trigger Source	Specifies the Start Trigger source. Details
Trigger:Start Trigger Edge	Specifies the active edge for the Start Trigger. This property is used only when the Start Trigger Type property is set to **Digital Edge**. Details

Trigger:Step Trigger Type	Specifies the Step Trigger type. Details
Trigger:Step Trigger Source	Specifies the Step Trigger source. Details
Trigger:Step Trigger Edge	Specifies the active edge for the Step Trigger. This property is used only when the Step Trigger Type property is set to **Digital Edge**. Details
Test Signal:IF Bandwidth	Specifies the intermediate frequency (IF) bandwidth for the NI-VNA device. Details
Test Signal:Output Power	Specifies the output power for the active port. Details
Sweep:Power Sweep Frequency	Specifies the output frequency for power sweep. Details
Test Signal:Reference Frequency:Source	Specifies the source for the reference frequency. Details
Test Signal:Reference Frequency:Reference Frequency	Returns the reference frequency. Details
Test Signal:Ref Plane Extension:Distance	Specifies the length of the reference plane extension. Port extension parameters can aid with adapter removal, thus providing a more accurate measurement. Details
Test Signal:Ref Plane Extension:Time	Specifies the time it takes for the signal to travel across the reference plane extension. Port extension parameters can aid with adapter removal, thus providing a more accurate measurement. Details

Test Signal:Ref Plane Extension:Dielectric Type	Specifies the reference plane extension dielectric type. Port extension parameters can aid with adapter removal, thus providing a more accurate measurement. Details
Test Signal:Ref Plane Extension:Dielectric User Value	Specifies the reference plane extension dielectric user value. Details
Test Signal:Ref Plane Extension:Insertion Loss	Specifies the reference plane extension dielectric insertion loss. Port extension parameters can aid with adapter removal and fixture de-embedding, thus providing a more accurate measurement. Details
Sweep:Sweep Mode	Specifies the sweep mode for the NI vector network analyzers. Details
Sweep:Sweep Number of Points	Specifies the number of points for the sweep measurement. Details
Sweep:Sweep Step Delay	Specifies the delay for each sweep point. The sweep step delay allows, for example, synchronization with external devices, such as antenna rotors. Details
Sweep:Format Phase Offset	Specifies the phase offset that is applied to the formatted data. This property only shifts the phase of the returned data by the specified amount. The reference plane configuration is not affected. Details
Averaging And Smoothing:Averaging Count	Specifies the averaging count for sweep averaging. Sweep averaging can be enabled by setting this property to a value greater than 1. Specify the type of sweep averaging using the Averaging Type property. Details

Averaging And Smoothing:Averaging Type	Specifies the type of sweep averaging. Details
Averaging And Smoothing:Smoothing Aperture	Specifies the smoothing aperture for the formatted data. Smoothing is applied to the formatted data after sweep averaging and error correction (calibration term application). Details
Sweep:Sweep Frequency Start	Specifies the start frequency for the frequency sweep. Details
Sweep:Sweep Frequency Stop	Specifies the stop frequency for the frequency sweep. Details
Individual Sources/ Receiver:Source Driving State	Specifies whether the port source is active or inactive. Details
Individual Sources/ Receiver:Source Frequency Multiplier	Specifies the multiplier value in the source frequency offset equation. Details
Individual Sources/ Receiver:Source Frequency Offset	Specifies the offset value in the source frequency offset equation. Details
Individual Sources/ Receiver:Receiver Frequency Multiplier	Specifies the multiplier value in the receiver frequency offset equation. Details
Individual Sources/ Receiver:Receiver Frequency Offset	Specifies the offset value in the receiver frequency offset equation. Details
Sweep:Sweep Power Start	Specifies the start power for the power sweep. Details
Sweep:Sweep Power Stop	Specifies the stop power for the power sweep. Details

Calibration Settings:RF Calibration Status	Indicates the RF calibration status of the channel. If RF calibration is available, use the Correction Enabled property to apply an RF calibration. Details
Calibration Settings:Source Power Calibration Status	Indicates the source power calibration status of the specified port. Details
Calibration Settings:Test Receiver Calibration Status	Indicates the test receiver power calibration status of the receiver you specify in the Active Channel property. Details
Calibration Settings:Reference Receiver Calibration Status	Indicates the reference receiver power calibration status of the receiver you specify in the Active Channel property. Details
Calibration Settings:Source Calibration Power Offset	Specifies the power offset to be applied during source power calibration. The offset applies to all sweep types. Details
Calibration Settings:De-embedding Enabled	Returns whether de-embedding is enabled. Details
Sweep:S11 Enabled	Specifies whether the S11 measurement is enabled. If you disable the S11 measurement, you cannot read the S11 parameter. Details
Sweep:S21 Enabled	Specifies whether the S21 measurement is enabled. If you disable the S21 measurement, you cannot read the S21 parameter. Details
Sweep:Driver State	Returns the state of NI-VNA. Details
Sweep:S12 Enabled	Specifies whether the S12 measurement is enabled. If you disable the S12 measurement, you cannot read the S12 parameter. Details

Sweep:S22 Enabled	Specifies whether the S22 measurement is enabled. If you disable the S22 measurement, you cannot read the S22 parameter. Details
Calibration Settings:Correction Enabled	Specifies whether NI-VNA applies calibration data to the raw measurements from the hardware. Changing property values after calibration can result in inappropriate correction and inaccurate measurements. For highest accuracy, calibrate after the device is fully configured and take measurements without changing any configuration values. Details
Calibration Settings:Calibration State	Returns the calibration state. The calibration state indicates whether NI-VNA has calibration data for the current configuration. Details
Calibration Settings:RF Calibration Type	Returns the type of RF calibration performed on the device. Details
Calibration Settings:Allow Changes While Calibrated	Specifies whether changes are allowed after the device is calibrated. Details
Calibration Settings:High Sensitivity Enabled	Specifies whether high-sensitivity mode is enabled while calibrating. Details
Calibration Settings:Connectors:Connector Gender	Specifies the connector gender. Details
Calibration Settings:Connectors:Connector Type	Specifies the connector type. Details
Calibration Settings:Calibration Thru Line Type	Specifies the through line type for manual calibration. A ***through line*** is a direct cable connection between the test ports, with no intervening connectors. Details

Calibration Settings:Calibration Thru Line Length	Specifies the through line length for manual calibration. A ***through line*** is a direct cable connection between the test ports, with no intervening connectors. Details
Calibration Settings:Calibration Thru Line Impedance	Specifies the through line impedance for manual calibration. A ***through line*** is a direct cable connection between the test ports, with no intervening connectors. Details
Calibration Settings:Calibration Thru Line Loss	Specifies the line loss of the through line for manual calibration. A ***through line*** is a direct cable connection between the test ports, with no intervening connectors. Details
Calibration Settings:Calibration Thru Reference Frequency	Specifies the through line reference frequency for manual calibration. A ***through line*** is a direct cable connection between the test ports, with no intervening connectors. Details
Calibration Settings:AutoCal:AutoCal True Thru Length	Specifies the true through length for the AutoCal. A ***true through*** is a direct cable connection between the test ports, with no intervening connectors. If desired, you can remove the AutoCal module for the through calibration step, and you can supply the parameters of the true through connection using this property. Details
Calibration Settings:AutoCal:AutoCal True Thru Impedance	Specifies a true through impedance for the AutoCal. A ***true through*** is a direct cable connection between the test ports, with no intervening connectors. If desired, you can remove the AutoCal module for the through calibration step, and you can specify the parameters of the true through connection using this property. Details
Calibration Settings:AutoCal:AutoCal True Thru Line Loss	Specifies a line loss for the true through for AutoCal. A ***true through*** is a direct cable connection between the test ports, with

	no intervening connectors. If desired, you can remove the AutoCal module for the through calibration step, and you can specify the parameters of the true through connection using this property. Details
Calibration Settings:AutoCal:AutoCal True Thru Reference Frequency	Specifies a true through reference frequency for the AutoCal. A ***true through*** is a direct cable connection between the test ports, with no intervening connectors. If desired, you can remove the AutoCal module for the through calibration step, and you can specify the parameters of the true through connection using this property. Details
Calibration Settings:AutoCal:AutoCal Box Orientation	Specifies the orientation of the AutoCal module. If you set this property incorrectly, the AutoCal assurance fails. Enable AutoCal assurance using the AutoCal Assurance Enabled property. Details
Calibration Settings:AutoCal:AutoCal Box Communication Port	Specifies the communication port to which the AutoCal module is connected. Details
Calibration Settings:AutoCal:AutoCal Assurance Enabled	Specifies whether an assurance step is automatically executed for the AutoCal. The assurance step verifies whether AutoCal ran successfully. If enabled, the assurance step is performed at the time that the niVNA Close User Cal VI is called. Details
Calibration Settings:TRL Calibration:Reference Plane Location	Specifies the location of the reference plane for TRL calibration. Details
Calibration Settings:TRL Calibration:Number of Bands	Specifies the number of frequency bands to use for TRL calibration. Details
Calibration Settings:TRL Calibration:Standard Type	Specifies the type of standard used for TRL calibration. Details

Calibration Settings:TRL Calibration:Standard Length	Specifies the length of the TRL calibration standard. Details
Calibration Settings:TRL Calibration:Standard Line Loss	Specifies the line loss of the active standard. Details
Calibration Settings:TRL Calibration:Standard Line Loss Frequency	Specifies the frequency at which the TRL Calibration Standard Line Loss property value is valid. Details
Calibration Settings:TRL Calibration:List of Standards	Specifies the list of TRL calibration standards to include in a TRL calibration band. Details
Calibration Settings:TRL Calibration:Band Breakpoint Type	Specifies whether the band breakpoint for TRL calibration is calculated automatically or specified manually using the TRL Calibration Band Breakpoint Frequency property. Details
Calibration Settings:TRL Calibration:Band Breakpoint Frequency	Specifies the manual frequency band breakpoint at which the TRL calibration stops using the standards associated with this frequency band and uses the subsequent band standards instead. Details
Calibration Settings:Power Meter Name	Specifies the resource name of the external power meter used for power calibration. Details
Calibration Settings:Calibration File Intended For SFP	Specifies whether the created calibration file is optimized for use with the NI-VNA soft front panel (SFP). Details
External Calibration Information:External Calibration Year	Returns the year of the last external calibration. Details
External Calibration Information:External Calibration Month	Returns the month of the last external calibration. The return value is expressed as an integer. For example, December is

	represented as 12. Details
External Calibration Information:External Calibration Day	Returns the day of the last external calibration. Details
External Calibration Information:External Calibration Hour	Returns the hour of the last external calibration. The time returned is 24-hour (military) local time; for example, if the device was calibrated at 2:00 PM, this function returns 14 as the value of this property. Details
External Calibration Information:External Calibration Minute	Returns the minute of the last external calibration. Details
External Calibration Information:External Calibration Temperature	Returns the temperature of the last external calibration. Details
External Calibration Information:External Calibration Interval	Returns the interval since the last external calibration. Complete the steps in the external calibration procedure for the device if the specified interval has elapsed since the last external calibration date. Details
User Calibration Information:User Calibration Year	Returns the year of the last user calibration. Details
User Calibration Information:User Calibration Month	Returns the month of the last user calibration. The returned value is expressed as an integer. For example, December is represented as 12. Details
User Calibration Information:User Calibration Day	Returns the day of the last user calibration. Details
User Calibration Information:User Calibration Hour	Returns the hour of the last user calibration. Details

User Calibration Information:User Calibration Minute	Returns the minute of the last user calibration. Details
User Calibration Information:User Calibration Temperature	Returns the temperature at which the last user calibration was performed. Details
Time Domain:Transform Type	Specifies the type of time-domain analysis. Details
Time Domain:Trip Type	Specifies the trip type for time-domain analysis. Trip type refers to the path the signal takes through the DUT. Setting this property determines how the frequency domain data is interpreted as time-domain data. Details
Time Domain:Impulse Width at 3 dB	Returns the calculated impulse width from time domain analysis. This width implies the resolution of the time-domain analysis calculations. Details
Time Domain:Alias Free Range	Returns the calculated alias-free range from the time-domain analysis. This value is calculated after each change to the frequency points. Details
Time Domain:Range:Units	Specifies the units for time-domain analysis. Details
Time Domain:Range:Start Point	Specifies the start point for the time-domain analysis. Details
Time Domain:Range:Stop Point	Specifies the stop point for the time-domain analysis. Details
Time Domain:Window:Shape	Specifies the window shape for time-domain analysis. Details

Time Domain:Window:Kaiser-Bessel Beta	Specifies the Kaiser-Bessel window beta value for time-domain analysis. Details
Time Domain:Window:Dolph-Chebyshev Gamma	Specifies the Dolph-Chebyshev window gamma value for time-domain analysis. Details
Time Domain:Gate:Shape	Specifies the gate shape for the time domain analysis. Details
Time Domain:Gate:Enable	Specifies whether the gate is enabled for the time-domain analysis. Details
Time Domain:Gate:Inclusive	Specifies the scope of the gated time/distance span. If you set this property to TRUE, the data outside the gated span is suppressed. If you set this property to FALSE, the data inside the gated span is suppressed. Details
Time Domain:Gate:Range:Start Point	Specifies the starting point for the time domain gate. Details
Time Domain:Gate:Range:Stop Point	Specifies the stopping point for the time domain gate. Details
Time Domain:Gate:Kaiser-Bessel Beta	Specifies the Kaiser-Bessel beta value for the time-domain analysis gate. Details
Time Domain:Gate:Dolph-Chebyshev Gamma	Specifies the Dolph-Chebyshev gamma value for the time-domain analysis gate. Details
Time Domain:DC Term Extrapolation:Type	Specifies the DC term extrapolation type for lowpass time-domain calculations. Details

Time Domain:DC Term Extrapolation:Method	Specifies the DC term extrapolation method for lowpass time-domain calculations. Details
Time Domain:DC Term Extrapolation:Coefficient	Specifies the coefficient that is used for the DC term extrapolation calculations. This value corresponds with the DC impedance of the DUT. Details
Time Domain:DC Term Extrapolation:User Value	Specifies the DC term value for lowpass time-domain calculations. Details
Device Information:PXI Chassis	Returns the identifier for the PXI chassis in which the NI vector network analyzer is installed. Details
Device Information:PXI Slot	Returns the number for the PXI slot in which the NI vector network analyzer is installed. Details
Device Information:Device Temperature	Returns the current temperature of the NI-VNA device. Details
Device Information:Device Serial Number	Returns the serial number of the NI-VNA device. Details
Device Information:Number of Ports	Returns the number of ports for the NI vector network analyzer device. Details
Inherent IVI Properties:User Options:Cache	Specifies whether to cache the value of properties. When caching is enabled, NI-VNA tracks the current NI-VNA device settings and avoids sending redundant commands to the device. NI-VNA can always cache or never cache particular properties, regardless of the setting of this property. Use the niVNA Initialize With Options VI to override the default

	value. Details
Inherent IVI Properties:User Options:Interchange Check	Specifies whether to perform interchangeability checking and retrieve interchangeability warnings. **Note** Interchangeability checking is currently not supported.
Inherent IVI Properties:User Options:Driver Setup	The DriverSetup string is used to set the initial values for properties that are specific to NI-VNA. The format of the DriverSetup string is ***Tag***:***Value***. ***Tag*** is the name of the ***DriverSetup*** string property. ***Value*** is the value set to the property. To set multiple properties, separate their assignments with a semicolon. Details
Inherent IVI Properties:User Options:Query Instrument Status	Specifies whether NI-VNA queries the NI-VNA device status after each operation. Querying the device status is useful for debugging. After you validate your program, you can set this property to FALSE to disable status checking and maximize performance. NI-VNA can choose to ignore status checking for particular properties, regardless of the setting of this property. **Note** Use the niVNA Initialize With Options VI to override this value.
Inherent IVI Properties:User Options:Range Check	Specifies whether to validate property values and VI parameters. If enabled, NI-VNA validates the parameter values that you pass to NI-VNA VIs. Range checking parameters is very useful for debugging. After you validate your program, you can set this property to FALSE to disable range checking and maximize performance.

	Note Use the niVNA Initialize With Options VI to override this value.
Inherent IVI Properties:User Options:Record Value Coercions	Specifies whether the IVI engine keeps a list of the value coercions it makes for integer and real type properties. **Note** This property is currently not supported.
Inherent IVI Properties:User Options:Simulate	Specifies whether to simulate NI-VNA I/O operations. This property is useful for debugging applications without using hardware. After a session is opened, you cannot change the simulation state. Use the niVNA Initialize With Options VI to enable simulation. **Note** NI-VNA does not support simulation.
Inherent IVI Properties:Driver Identification:Description	Returns a string that contains a brief description of NI-VNA. Details
Inherent IVI Properties:Driver Identification:Driver Prefix	Returns a string that contains the prefix for NI-VNA. The name of each user-callable VI in NI-VNA starts with this prefix. Details
Inherent IVI Properties:Driver Identification:Driver Vendor	Returns a string that contains the name of the vendor that supplies NI-VNA. Details
Inherent IVI Properties:Driver Identification:Revision	Returns a string that contains additional version information about NI-VNA. Details
Inherent IVI Properties:Driver Identification:Class Specification Major Version	Returns the major version number of the class specification with which NI-VNA is compliant. Details

Inherent IVI Properties:Driver Identification:Class Specification Minor Version	Returns the minor version number of the class specification with which NI-VNA is compliant. Details
Inherent IVI Properties:Driver Capabilities:Class Group Capabilities	Returns a string that contains a comma-separated list of class-extension groups that NI-VNA implements. Details
Inherent IVI Properties:Driver Capabilities:Supported Instrument Models	Returns a comma-separated list of supported devices. Details
Inherent IVI Properties:Instrument Identification:Manufacturer	Returns a string that contains the name of the manufacturer for the NI-VNA device you are currently using. Details
Inherent IVI Properties:Instrument Identification:Model	Returns a string that contains the model number or name of the NI-VNA device that you are currently using. Details
Inherent IVI Properties:Instrument Identification:Firmware Revision	Returns a string that contains the firmware revision information for the NI-VNA device you are currently using. Details
Inherent IVI Properties:Advanced Session Information:Logical Name	Contains the logical name you specified when opening the current IVI session. You may pass a logical name to the niVNA Initialize VI or the niVNA Initialize With Options VI. The IVI Configuration Utility must contain an entry for the logical name. The logical name entry refers to a driver session section in the IVI Configuration file. The driver session section specifies a physical device and initial user options. Details
Inherent IVI Properties:Advanced Session Information:I/O Resource Descriptor	Indicates the resource name NI-VNA uses to identify the physical device. If you initialize NI-VNA with a logical name, this property contains the resource name that corresponds to the entry in the IVI Configuration Utility. If you initialize NI-VNA with the resource name, this property contains that value. Details

Events:Ready For Start Event:Output Terminal	Specifies the terminal where the signal will be exported. You can also choose to not export a signal. Details
Events:Ready For Start Event:Active Level	Specifies the active level for the Ready For Start Event. Details
Events:Ready For Step Event:Output Terminal	Specifies the terminal where the signal will be exported. You can also choose to not export a signal. Details
Events:Ready For Step Event:Active Level	Specifies the active level for the Ready For Step Event. Details
Events:End Of Step Event:Output Terminal	Specifies the terminal where the signal will be exported. You can also choose not to export any signal. Details
Events:End Of Step Event:Edge	Specifies the active edge for the End of Step Event. Details
Events:End Of Sweep Event:Output Terminal	Specifies the terminal where the signal will be exported. You can also choose to not export a signal. Details
Events:End Of Sweep Event:Edge	Specifies the active edge for the End Of Sweep Event. Details

Active Channel Property

Short Name: Active Channel

Property of niVNA

Specifies the channel name used to access all subsequent properties in this instance of the property node. If the property you want to use is channel- or port-based, you must first select this property and then pass the name of the specific channel or port. If the

property you specify is not channel- or port-based, pass an empty string, or omit setting this property. The default value is " ".

For more information about device-, channel-, port-, standard-, and band-based properties, refer to Using Properties and Attributes with NI-VNA.

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Write Only
High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Trigger:Start Trigger Type Property

Short Name: Start Trigger Type

Property of niVNA

Specifies the Start Trigger type.

Default Value: **None**

Supported Devices: NI 5630/5632

None (0)	A Start Trigger is not configured. The sweep measurement starts immediately after you call the niVNA Initiate VI.
Digital Edge (1)	The sweep measurement starts after the device receives the configured digital edge from the selected source. The source of the digital edge is specified with the Start Trigger Digital Edge Source property.
Software Edge (2)	The sweep measurement starts after you call the niVNA Send Software Edge Trigger VI. You can assert the software trigger by calling the niVNA Send Software Edge Trigger VI and selecting **Start Trigger** as the value of the **trigger** parameter.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	niVNA Configure Trigger
Channel-based	No
Port-based	No
Device-based	Yes
Standard-based	No
Band-based	No
Resettable	No

Trigger:Start Trigger Source Property

Short Name: Start Trigger Source

Property of niVNA

Specifies the Start Trigger source.

Default Value: `PFI0`

Supported Devices: NI 5630/5632

PFI0	The trigger is received from PFI 0, which is the external trigger input terminal on the module front panel.
PXI_TRIG0	The trigger is received on PXI trigger line 0.
PXI_TRIG1	The trigger is received on PXI trigger line 1.
PXI_TRIG2	The trigger is received on PXI trigger line 2.
PXI_TRIG3	The trigger is received on PXI trigger line 3.
PXI_TRIG4	The trigger is received on PXI trigger line 4.
PXI_TRIG5	The trigger is received on PXI trigger line 5.
PXI_TRIG6	The trigger is received on PXI trigger line 6.
PXI_TRIG7	The trigger is received on PXI trigger line 7.

PFI0 ("PFI0")	The Trigger is received from the PFI 0 connector.
PXI_Trig0 ("PXI_Trig0")	The Trigger is received from the PXI trigger line 0.
PXI_Trig1 ("PXI_Trig1")	The Trigger is received from the PXI trigger line 1.
PXI_Trig2 ("PXI_Trig2")	The Trigger is received from the PXI trigger line 2.
PXI_Trig3 ("PXI_Trig3")	The Trigger is received from the PXI trigger line 3.
PXI_Trig4 ("PXI_Trig4")	The Trigger is received from the PXI trigger line 4.

PXI_Trig5 ("PXI_Trig5")	The Trigger is received from the PXI trigger line 5.
PXI_Trig6 ("PXI_Trig6")	The Trigger is received from the PXI trigger line 6.
PXI_Trig7 ("PXI_Trig7")	The Trigger is received from the PXI trigger line 7.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	niVNA Configure Trigger
Channel-based	No
Port-based	No
Device-based	Yes
Standard-based	No
Band-based	No
Resettable	No

Trigger:Start Trigger Edge Property

Short Name: Start Trigger Edge

Property of niVNA

Specifies the active edge for the Start Trigger. This property is used only when the Start Trigger Type property is set to **Digital Edge**.

Default Value: **Rising**

Supported Devices: NI 5630/5632

Rising (1)	The sweep measurement starts on the rising edge of the signal.
Falling (0)	The sweep measurement starts on the falling edge of the signal.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	niVNA Configure Trigger
Channel-based	No
Port-based	No
Device-based	Yes
Standard-based	No
Band-based	No
Resettable	No

Trigger:Step Trigger Type Property

Short Name: Step Trigger Type

Property of niVNA

Specifies the Step Trigger type.

Default Value: **None**

Supported Devices: NI 5632

None (0)	A Step Trigger is not configured. The sweep continues after you call the niVNA Initiate VI.
Digital Edge (1)	The sweep measurement advances to new points after the device receives the configured digital edge from the selected source. The source of the digital edge is specified with the Step Trigger Digital Edge Source property.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	Yes
Standard-based	No
Band-based	No
Resettable	No

Trigger:Step Trigger Source Property

Short Name: Step Trigger Source

Property of niVNA

Specifies the Step Trigger source.

Default Value: `PFI0`

Supported Devices: NI 5632

PFI0	The trigger is received on PFI 0, which is the external trigger input terminal on the module front panel.
PXI_TRIG0	The trigger is received on PXI trigger line 0.
PXI_TRIG1	The trigger is received on PXI trigger line 1.
PXI_TRIG2	The trigger is received on PXI trigger line 2.
PXI_TRIG3	The trigger is received on PXI trigger line 3.
PXI_TRIG4	The trigger is received on PXI trigger line 4.
PXI_TRIG5	The trigger is received on PXI trigger line 5.
PXI_TRIG6	The trigger is received on PXI trigger line 6.
PXI_TRIG7	The trigger is received on PXI trigger line 7.

PFI0 ("PFI0")	The Trigger is received from the PFI 0 connector.
PXI_Trig0 ("PXI_Trig0")	The Trigger is received from the PXI trigger line 0.
PXI_Trig1 ("PXI_Trig1")	The Trigger is received from the PXI trigger line 1.
PXI_Trig2 ("PXI_Trig2")	The Trigger is received from the PXI trigger line 2.
PXI_Trig3 ("PXI_Trig3")	The Trigger is received from the PXI trigger line 3.
PXI_Trig4 ("PXI_Trig4")	The Trigger is received from the PXI trigger line 4.
PXI_Trig5 ("PXI_Trig5")	The Trigger is received from the PXI trigger line 5.
PXI_Trig6 ("PXI_Trig6")	The Trigger is received from the PXI trigger line 6.

PXI_Trig7 ("PXI_Trig7")	The Trigger is received from the PXI trigger line 7.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	Yes
Standard-based	No
Band-based	No
Resettable	No

Trigger:Step Trigger Edge Property

Short Name: Step Trigger Edge

Property of niVNA

Specifies the active edge for the Step Trigger. This property is used only when the Step Trigger Type property is set to **Digital Edge**.

Default Value: **Rising**

Supported Devices: NI 5632

Rising (1)	The sweep measurement starts on the rising edge of the signal.

Falling (0)	The sweep measurement starts on the falling edge of the signal.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	Yes
Standard-based	No
Band-based	No
Resettable	No

Test Signal:IF Bandwidth Property

Short Name: IF Bandwidth

Property of niVNA

Specifies the intermediate frequency (IF) bandwidth for the NI-VNA device.

Valid Values:

NI 5630—10 Hz, 30 Hz, 100 Hz, 300 Hz, 1 kHz, 3 kHz, 10 kHz, or 30 kHz

NI 5632—10 Hz, 30 Hz, 50 Hz, 70 Hz, 100 Hz, 300 Hz, 500 Hz, 700 Hz, 1 kHz, 3 kHz, 5 kHz, 7 kHz, 10 kHz, 30 kHz, 50 kHz, 70 kHz, 100 kHz, 300 kHz, or 500 kHz

Default Value: **1 kHz**

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Port-based	No
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Test Signal:Output Power Property

Short Name: Output Power

Property of niVNA

Specifies the output power for the active port.

Units: dBm

Valid Ranges:

NI 5630—(–25) to 5, in 0.5 dbm increments

NI 5632—(–30) to 10, in 0.01 dBm increments

Default Value: 5

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	DBL
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Port-based	Yes
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Test Signal:Reference Frequency:Source Property

Short Name: Reference Frequency Source

Property of niVNA

Specifies the source for the reference frequency.

Note The NI 5630 does not support the **PXI_CLK100** value as a frequency source.

Default Value: `OnboardReference`

Supported Devices: NI 5630/5632

OnboardReference ("OnboardReference")	The reference signal received from the onboard Reference Clock is used as the source for the frequency reference signal.
PXI_CLK100 ("PXI_CLK100")	The reference signal received from the PXI_CLK100 terminal on the NI 5632 PXI Express backplane is used as the source for the frequency reference signal.
RefIn ("RefIn")	The reference signal received from the REF IN connector on the module front panel is used as the source for the frequency reference signal.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	Yes
Standard-based	No
Band-based	No
Resettable	No

Test Signal:Reference Frequency:Reference Frequency Property

Short Name: Reference Frequency

Property of niVNA

Returns the reference frequency.

Valid Values:

NI 5630—10 MHz

NI 5632—10 MHz when you set the Reference Frequency Source property to **RefIn**, 100 MHz when you set the Reference Frequency Source property to **PXI_CLK100**

Units: Hz

Default Value: 10 MHz

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	Yes
Standard-based	No
Band-based	No

Resettable	No

Test Signal:Ref Plane Extension:Distance Property

Short Name: Ref Plane Ext Distance

Property of niVNA

Specifies the length of the reference plane extension. Port extension parameters can aid with adapter removal, thus providing a more accurate measurement.

Units: m

Default Value: 0

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Port-based	Yes
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Test Signal:Ref Plane Extension:Time Property

Short Name: Ref Plane Ext Time

Property of niVNA

Specifies the time it takes for the signal to travel across the reference plane extension. Port extension parameters can aid with adapter removal, thus providing a more accurate measurement.

Units: seconds (s)

Default Value: 0

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Port-based	Yes
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Test Signal:Ref Plane Extension:Dielectric Type Property

Short Name: Ref Plane Ext Dielectric Type

Property of niVNA

Specifies the reference plane extension dielectric type. Port extension parameters can aid with adapter removal, thus providing a more accurate measurement.

The dielectric type (***k***) is related to velocity factor (***VF***) as ***VF*** = 1/√***k***. In free space, electromagnetic waves, such as RF signals, travel at the speed of light. However, traveling in a medium other than free space, such as a coaxial cable, they are slowed to less than the speed of light. This reduction in velocity is described as a decimal percentage known as ***velocity factor***, and it ranges from 0 to 1. Therefore, the speed of electromagnetic waves in any medium can be calculated as ***Speed*** = (velocity factor) x (speed of light). In free space, the velocity factor is 1. In coaxial cables, values typically range from 0.6 to 0.8. Since speed, time, and distance are directly related, specify the dielectric type when making vector network analyzer measurements that are time or distance sensitive, such as group delay, to ensure the correct measurement values are displayed.

Default Value: **Air**

Supported Devices: NI 5630/5632

Air (0)	The reference plane extension uses air as the dielectric, which is associated with a dielectric constant of 1.000649.
Polyethylene (1)	The reference plane extension uses a polyethylene dielectric, which is associated with a dielectric constant of 2.26.

Teflon (2)	The reference plane extension uses a teflon dielectric, which is associated with a dielectric constant of 2.1.
Micro Teflon (3)	The reference plane extension uses a microteflon dielectric, which is associated with a dielectric constant of 1.69.
User Value (4)	The reference plane extension uses the value you specify for the Dielectric User Value property as the dielectric value.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Port-based	Yes
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Test Signal:Ref Plane Extension:Dielectric User Value Property

Short Name: Ref Plane Ext Dielectric User Value

Property of niVNA

Specifies the reference plane extension dielectric user value.

Default Value: 0

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Port-based	Yes
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Test Signal:Ref Plane Extension:Insertion Loss Property

Short Name: Ref Plane Ext Insertion Loss

Property of niVNA

Specifies the reference plane extension dielectric insertion loss. Port extension parameters can aid with adapter removal and fixture de-embedding, thus providing a more accurate measurement.

Units: dB

Default Value: 0

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	DBL
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Port-based	Yes
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Sweep:Power Sweep Frequency Property

Short Name: Power Sweep Frequency

Property of niVNA

Specifies the output frequency for power sweep.

Units: Hz

Valid Range: 300 kHz to 8.5 GHz, in 0.1 Hz increments

Default Value: 300 kHz

Supported Devices: NI 5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Port-based	No
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Sweep:Sweep Mode Property

Short Name: Sweep Mode

Property of niVNA

Specifies the sweep mode for the NI vector network analyzers.

	**Note** The NI 5630 supports only the **Frequency Sweep** mode. For more information about sweep modes, refer to the NI 5632 Segmented Sweep topic.

	**Note** You can configure the Frequency-Based Segmented Sweep and Index-Based Segmented Sweep modes in the NI-VNA Soft Front Panel.

Default Value: **Frequency Sweep**

Supported Devices: NI 5630/5632

Frequency Sweep (0)	The device sweeps across a range of frequencies with constant power.
Power Sweep (1)	The device sweeps across a range of powers with constant frequency.
Frequency-Based Segmented Sweep (2)	The device sweeps across a list of monotonically increasing frequencies.
Index-Based Segmented Sweep (3)	The device sweeps across a list of frequencies.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	niVNA Configure Sweep
Channel-based	Yes
Port-based	No
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Sweep:Sweep Number of Points Property

Short Name: Sweep Number of Points

Property of niVNA

Specifies the number of points for the sweep measurement.

Valid Ranges:

NI 5630—1 to 3,201

NI 5632—1 to 20,001

Default Value: 201

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	niVNA Configure Sweep
Channel-based	Yes
Port-based	No
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Sweep:Sweep Step Delay Property

Short Name: Sweep Step Delay

Property of niVNA

Specifies the delay for each sweep point. The sweep step delay allows, for example, synchronization with external devices, such as antenna rotors.

Units: seconds (s)

Default Value: 0 (no delay)

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Port-based	No
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Sweep:Format Phase Offset Property

Short Name: Format Phase Offset

Property of niVNA

Specifies the phase offset that is applied to the formatted data. This property only shifts the phase of the returned data by the specified amount. The reference plane configuration is not affected.

Units: degrees

Default Value: 0

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	DBL
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	Yes
Standard-based	No
Band-based	No
Resettable	No

Sweep:Sweep Frequency Start Property

Short Name: Sweep Frequency Start

Property of niVNA

Specifies the start frequency for the frequency sweep.

Valid Values:

NI 5630—≥10 MHz

NI 5632—≥300 kHz

Units: Hz

Default Value: 10 MHz

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	niVNA Configure Frequency Start Stop
Channel-based	Yes
Port-based	No
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Sweep:Sweep Frequency Stop Property

Short Name: Sweep Frequency Stop

Property of niVNA

Specifies the stop frequency for the frequency sweep.

Valid Values:

NI 5630—≤6 GHz

NI 5632—≤8.5 GHz

Units: Hz

Default Value: 6 GHz

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	niVNA Configure Frequency Start Stop
Channel-based	Yes
Port-based	No
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Sweep:Sweep Power Start Property

Short Name: Sweep Power Start

Property of niVNA

Specifies the start power for the power sweep.

Units: dBm

Default Value: -30.0

Valid Values: ≥-30.0

Supported Devices: NI 5632

Remarks

The following table lists the characteristics of this property.

Datatype	DBL
Permissions	Read/Write
High-level VIs	niVNA Configure Power Start Stop
Channel-based	No
Port-based	Yes
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Sweep:Sweep Power Stop Property

Short Name: Sweep Power Stop

Property of niVNA

Specifies the stop power for the power sweep.

Units: dBm

Default Value: 10

Valid Values: ≤10

Supported Devices: NI 5632

Remarks

The following table lists the characteristics of this property.

Datatype	DBL
Permissions	Read/Write
High-level VIs	niVNA Configure Power Start Stop
Channel-based	No
Port-based	Yes
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Sweep:S11 Enabled Property

Short Name: S11 Enabled

Property of niVNA

Specifies whether the S11 measurement is enabled. If you disable the S11 measurement, you cannot read the S11 parameter.

Note If you enable the appropriate S-parameters for your measurement, the VNA is capable of optimizing measurement time.

TRUE	S11 measurement is enabled.
FALSE	S11 measurement is disabled.

Default Value: TRUE

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	TF
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Port-based	No
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Sweep:S21 Enabled Property

Short Name: S21 Enabled

Property of niVNA

Specifies whether the S21 measurement is enabled. If you disable the S21 measurement, you cannot read the S21 parameter.

Note If you enable the appropriate S-parameters for your measurement, the VNA is capable of optimizing measurement time.

TRUE	S21 measurement is enabled.
FALSE	S21 measurement is disabled.

Default Value: TRUE

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	TF
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Port-based	No
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

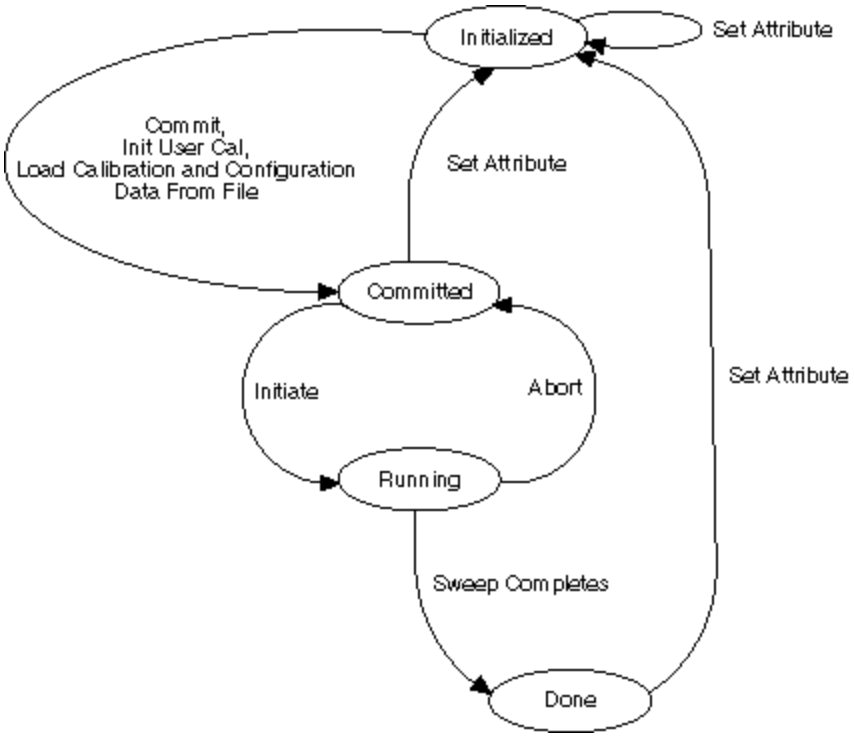
Sweep:Driver State Property

Short Name: Driver State

Property of niVNA

Returns the state of NI-VNA.

The possible states are shown in the following diagram.



Supported Devices: NI 5630/5632

Initialized (1)	The driver has been initialized.
Committed (4)	The driver settings have been committed.
Running (5)	The driver is running.
Done (6)	The driver has finished acquiring data.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read Only

High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	Yes
Standard-based	No
Band-based	No
Resettable	No

Sweep:S12 Enabled Property

Short Name: S12 Enabled

Property of niVNA

Specifies whether the S12 measurement is enabled. If you disable the S12 measurement, you cannot read the S12 parameter.



Note If you enable the appropriate S-parameters for your measurement, the VNA is capable of optimizing measurement time.

TRUE	S12 measurement is enabled.
FALSE	S12 measurement is disabled.

Default Value: TRUE

Supported Devices: NI 5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write

High-level VIs	N/A
Channel-based	Yes
Port-based	No
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Sweep:S22 Enabled Property

Short Name: S22 Enabled

Property of niVNA

Specifies whether the S22 measurement is enabled. If you disable the S22 measurement, you cannot read the S22 parameter.



Note If you enable the appropriate S-parameters for your measurement, the VNA is capable of optimizing measurement time.

TRUE	S22 measurement is enabled.
FALSE	S22 measurement is disabled.

Default Value: TRUE

Supported Devices: NI 5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write

High-level VIs	N/A
Channel-based	Yes
Port-based	No
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Averaging And Smoothing:Averaging Count Property

Short Name: Averaging Count

Property of niVNA

Specifies the averaging count for sweep averaging. Sweep averaging can be enabled by setting this property to a value greater than 1. Specify the type of sweep averaging using the Averaging Type property.

Default Value: 1

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Port-based	No

Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Averaging And Smoothing:Averaging Type Property

Short Name: Averaging Type

Property of niVNA

Specifies the type of sweep averaging.

Default Value: **Point by Point**

Supported Devices: NI 5630/5632

Point by Point (0)	The NI vector network analyzer uses point-by-point averaging. Point-by-point averaging takes the specified number of measurements at each point of the sweep.
Sweep by Sweep (1)	The NI vector network analyzer uses sweep-by-sweep averaging. Sweep-by-sweep averaging performs the given number of sweeps and then averages each point.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A

Channel-based	Yes
Port-based	No
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Averaging And Smoothing:Smoothing Aperture Property

Short Name: Smoothing Aperture

Property of niVNA

Specifies the smoothing aperture for the formatted data. Smoothing is applied to the formatted data after sweep averaging and error correction (calibration term application).

Units: percentage

Valid Range: 0 to 20

Default Value: 0

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A

Channel-based	Yes
Port-based	No
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Individual Sources/Receiver:Source Driving State Property

Short Name: Source Driving State

Property of niVNA

Specifies whether the port source is active or inactive.

Default Value: 0

Supported Devices: NI 5632

Automatic (0)	The port source is automatically activated or deactivated during a sweep, depending on the S-parameters measured during the sweep.
Active (1)	The port source is active during a sweep.
Inactive (2)	The port source is inactive during a sweep.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Port-based	Yes
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Individual Sources/Receiver:Source Frequency Multiplier Property

Short Name: Source Frequency Multiplier

Property of niVNA

Specifies the multiplier value in the source frequency offset equation.

The frequency offset equation is (***M*** * ***f***) + ***b***, where ***M*** is the multiplier, ***f*** is the frequency list defined by the start and stop frequency properties, and ***b*** is the offset defined by the Source Frequency Offset property.

Note If you specify values for ***M*** and ***b*** that, after calculation, result in a value that is out of the frequency range, NI-VNA returns an error.

Default Value: 1

Supported Devices: NI 5632

Remarks

The following table lists the characteristics of this property.

Datatype	DBL
Permissions	Read/Write
High-level VIs	niVNA Configure Frequency Offset Equation
Channel-based	No
Port-based	Yes
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Individual Sources/Receiver:Source Frequency Offset Property

Short Name: Source Frequency Offset

Property of niVNA

Specifies the offset value in the source frequency offset equation.

The frequency offset equation is (***M*** * ***f***) + ***b***, where ***M*** is the multiplier, ***f*** is the frequency list defined by the start and stop frequency properties, and ***b*** is the offset.

Note If you specify values for ***M*** and ***b*** that, after calculation, result in a value that is out of the frequency range, NI-VNA returns an error.

Default Value: 0

Supported Devices: NI 5632

Remarks

The following table lists the characteristics of this property.

Datatype	DBL
Permissions	Read/Write
High-level VIs	niVNA Configure Frequency Offset Equation
Channel-based	No
Port-based	Yes
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Individual Sources/Receiver:Receiver Frequency Multiplier Property

Short Name: Receiver Frequency Multiplier

Property of niVNA

Specifies the multiplier value in the receiver frequency offset equation.

The frequency offset equation is (***M*** * ***f***) + ***b***, where ***M*** is the multiplier, ***f*** is the frequency list defined by the start and stop frequency properties, and ***b*** is the offset defined by the Receiver Frequency Offset property.

Note If you specify values for ***M*** and ***b*** that, after calculation, result in a value that is out of the frequency range, NI-VNA returns an error.

Default Value: 1

Supported Devices: NI 5632

Remarks

The following table lists the characteristics of this property.

Datatype	DBL
Permissions	Read/Write
High-level VIs	niVNA Configure Frequency Offset Equation
Channel-based	Yes
Port-based	No
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Individual Sources/Receiver:Receiver Frequency Offset Property

Short Name: Receiver Frequency Offset

Property of niVNA

Specifies the offset value in the receiver frequency offset equation.

The frequency offset equation is (***M*** * ***f***) + ***b***, where ***M*** is the multiplier, ***f*** is the frequency list defined by the start and stop frequency properties, and ***b*** is the offset.

Note If you specify values for ***M*** and ***b*** that, after calculation, result in a value that is out of the frequency range, NI-VNA returns an error.

Default Value: 0

Supported Devices: NI 5632

Remarks

The following table lists the characteristics of this property.

Datatype	DBL
Permissions	Read/Write
High-level VIs	niVNA Configure Frequency Offset Equation
Channel-based	Yes
Port-based	No
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Calibration Settings:RF Calibration Status Property

Short Name: RF Calibration Status

Property of niVNA

Indicates the RF calibration status of the channel. If RF calibration is available, use the Correction Enabled property to apply an RF calibration.

Default Value: FALSE

Supported Devices: NI 5632

TRUE	RF calibration is available.
FALSE	RF calibration is not available.

Remarks

The following table lists the characteristics of this property.

Datatype	TF
Permissions	Read Only
High-level VIs	N/A
Channel-based	Yes
Port-based	No
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Calibration Settings:Source Power Calibration Status Property

Short Name: Source Power Calibration Status

Property of niVNA

Indicates the source power calibration status of the specified port.

Default Value: FALSE

Supported Devices: NI 5632

TRUE	Calibration is applied.
FALSE	Calibration is not applied.

Remarks

The following table lists the characteristics of this property.

Datatype	TF
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Port-based	Yes
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Calibration Settings:Test Receiver Calibration Status Property

Short Name: Test Receiver Calibration Status

Property of niVNA

Indicates the test receiver power calibration status of the receiver you specify in the Active Channel property.

Supported Devices: NI 5632

TRUE	Calibration is applied.
FALSE	Calibration is not applied.

Remarks

The following table lists the characteristics of this property.

Datatype	TF
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Port-based	Yes
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Calibration Settings:Reference Receiver Calibration Status Property

Short Name: Reference Receiver Calibration Status

Property of niVNA

Indicates the reference receiver power calibration status of the receiver you specify in the Active Channel property.

Supported Devices: NI 5632

TRUE	Calibration is applied.
FALSE	Calibration is not applied.

Remarks

The following table lists the characteristics of this property.

Datatype	TF
Permissions	Read Only

High-level VIs	N/A
Channel-based	No
Port-based	Yes
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Calibration Settings:Source Calibration Power Offset Property

Short Name: Source Calibration Power Offset

Property of niVNA

Specifies the power offset to be applied during source power calibration. The offset applies to all sweep types.

Power offset occurs when you place a device with gain or loss between the source port and the device under test (DUT). You can then place the power meter at the reference plane, and the gain or loss is removed out of the system during calibration so that the actual power applied to the DUT is the port power plus the offset. This property affects only the output of the VNA after a source power calibration.

Units: dB

Default Value: 0

Valid Range: -100 to 100

Supported Devices: NI 5632

Remarks

The following table lists the characteristics of this property.

Datatype	DBL
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Calibration Settings:De-embedding Enabled Property

Short Name: De-embedding Enabled

Property of niVNA

Returns whether de-embedding is enabled.

For more information about de-embedding a test fixture, refer to the NI 5632 De-Embedding topic.

Supported Devices: NI 5632

De-embedding disabled (0)	De-embedding is disabled.

De-embedding enabled (1)	De-embedding is enabled.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	Yes
Port-based	No
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Calibration Settings:Correction Enabled Property

Short Name: Correction Enabled

Property of niVNA

Specifies whether NI-VNA applies calibration data to the raw measurements from the hardware. Changing property values after calibration can result in inappropriate correction and inaccurate measurements. For highest accuracy, calibrate after the device is fully configured and take measurements without changing any configuration values.

Note This property is valid only for RF calibrations and does not apply to power calibrations

	including, source power calibration, test receiver calibration, and reference receiver calibration.

The relationship between the Correction Enabled property and the Calibration State property is shown in the following table.

Correction Enabled	Calibration State		
	Uncalibrated	**Calibrated**	**Calibrating**
FALSE	NI-VNA does ***not*** have a valid set of calibration data for the current device configuration. Measurement data is ***not*** corrected using calibration data.	NI-VNA ***does*** have a valid set of calibration data for the current device configuration; however, measurement data is ***not*** corrected using calibration data.	N/A
TRUE	NI-VNA does ***not*** have a valid set of calibration data for the current device configuration; however, measurement data ***is*** modified using (potentially invalid) calibration data.	NI-VNA ***does*** have a valid set of calibration data for the current device configuration. Measurement data ***is*** corrected using calibration data.	N/A

Note This property is automatically set to TRUE when calibration is successfully loaded from a file, performed manually, or performed using the AutoCal module.

Default Value: FALSE

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	TF
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes

Port-based	No
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Calibration Settings:Calibration State Property

Short Name: Calibration State

Property of niVNA

Returns the calibration state. The calibration state indicates whether NI-VNA has calibration data for the current configuration.

The relationship between the Correction Enabled property and the Calibration State property is shown in the following table.

Correction Enabled	Calibration State		
	Uncalibrated	**Calibrated**	**Calibrating**
FALSE	NI-VNA does ***not*** have a valid set of calibration data for the current device configuration. Measurement data is ***not*** corrected using calibration data.	NI-VNA ***does*** have a valid set of calibration data for the current device configuration; however, measurement data is ***not*** corrected using calibration data.	N/A
TRUE	NI-VNA does ***not*** have a valid set of calibration data for the current device configuration; however, measurement data ***is*** modified using (potentially invalid) calibration data.	NI-VNA ***does*** have a valid set of calibration data for the current device configuration. Measurement data ***is*** corrected using calibration data.	N/A

Default Value: **Uncalibrated**

Supported Devices: NI 5630/5632

Uncalibrated (0)	Indicates that the NI vector network analyzer is not calibrated.
Calibrating (1)	Indicates that the NI vector network analyzer is performing calibration.
Fully Calibrated (2)	Indicates that the NI vector network analyzer is calibrated.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	Yes
Port-based	No
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Calibration Settings:RF Calibration Type Property

Short Name: RF Calibration Type

Property of niVNA

Returns the type of RF calibration performed on the device.

You must set the following S-parameter properties so that the RF Calibration Type property can determine the appropriate calibration port.

- S11 Enabled
- S12 Enabled
- S21 Enabled
- S22 Enabled

The following table lists the RF calibration types and corresponding S-parameter property settings.

RF Calibration Type	Port Source	S11 Enabled	S21 Enabled	S12 Enabled	S22 Enabled
Two-Port	PORT 1/PORT 2	TRUE	TRUE	TRUE	TRUE
Two-Port, One-Path	PORT 1	TRUE	TRUE	FALSE	FALSE
Two-Port, One-Path	PORT 2	FALSE	FALSE	TRUE	TRUE
One-Port	PORT 1	TRUE	FALSE	FALSE	FALSE
One-Port	PORT 2	FALSE	FALSE	FALSE	TRUE
Reflection	PORT 1	TRUE	FALSE	FALSE	FALSE
Reflection	PORT 2	FALSE	FALSE	FALSE	TRUE
Reflection	PORT 1/PORT 2	TRUE	FALSE	FALSE	TRUE
Transmission	PORT 1	FALSE	TRUE	FALSE	FALSE
Transmission	PORT 2	FALSE	FALSE	TRUE	FALSE
Transmission	PORT 1/PORT 2	FALSE	TRUE	TRUE	FALSE

Note **TRL** is a valid value for only the NI 5632.

Default Value: **2-port SOLT**

Supported Devices: NI 5630/5632

2-port SOLT (0)	The NI vector network analyzer performs two-port, one-path short/open/load/thru (SOLT) calibration.
1-port SOL (1)	The NI vector network analyzer performs one-port short/open/load (SOL) calibration.
Reflection Response (2)	The NI vector network analyzer performs one-port reflection response calibration.
Transmission Response (3)	The NI vector network analyzer performs transmission response calibration.
TRL (4)	The NI vector network analyzer performs a full two-port through-reflect-line (TRL) calibration.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	Yes
Port-based	No
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Calibration Settings:Allow Changes While Calibrated Property

Short Name: Allow Changes While Calibrated

Property of niVNA

Specifies whether changes are allowed after the device is calibrated.

Default Value: FALSE

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Port-based	No
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Calibration Settings:High Sensitivity Enabled Property

Short Name: Calibration High Sensitivity Enabled

Property of niVNA

Specifies whether high-sensitivity mode is enabled while calibrating.

Default Value: FALSE

Supported Devices: NI 5630

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Port-based	No
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Calibration Settings:Connectors:Connector Gender Property

Short Name: Connector Gender

Property of niVNA

Specifies the connector gender.

Default Value: **Male**

Supported Devices: NI 5630/5632

Male (0)	The port has a male connector.
Female (1)	The port has a female connector.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	niVNA Configure Connector
Channel-based	No
Port-based	Yes
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Calibration Settings:Connectors:Connector Type Property

Short Name: Connector Type

Property of niVNA

Specifies the connector type.

Default Value: **K**

Supported Devices: NI 5630/5632

N (0)	The port uses an N-type connector.
K (1)	The port uses a K-type connector.
GPC 3.5 (2)	The port uses a GPC 3.5 connector.
SMA (3)	The port uses an SMA connector.
User-Defined (100)	The port uses a user-defined connector.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	niVNA Configure Connector
Channel-based	No
Port-based	Yes
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Calibration Settings:Calibration Thru Line Type Property

Short Name: Calibration Thru Line Type

Property of niVNA

Specifies the through line type for manual calibration. A ***through line*** is a direct cable connection between the test ports, with no intervening connectors.

	**Note** The NI 5630 supports only the **Thru** mode.

Default Value: **Thru**

Supported Devices: NI 5630/5632

Thru (0)	The VNA uses a through line for manual calibration.
Reciprocal (1)	The VNA uses a reciprocal line for manual calibration.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Port-based	No
Device-based	No

Standard-based	No
Band-based	No
Resettable	No

Calibration Settings:Calibration Thru Line Length Property

Short Name: Calibration Thru Line Length

Property of niVNA

Specifies the through line length for manual calibration. A ***through line*** is a direct cable connection between the test ports, with no intervening connectors.

Units: mm

Default Value: 0

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Port-based	No
Device-based	No
Standard-based	No
Band-based	No

Resettable	No

Calibration Settings:Calibration Thru Line Impedance Property

Short Name: Calibration Thru Line Impedance

Property of niVNA

Specifies the through line impedance for manual calibration. A ***through line*** is a direct cable connection between the test ports, with no intervening connectors.

Units: ohms

Default Value: 50

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Port-based	No
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Calibration Settings:Calibration Thru Line Loss Property

Short Name: Calibration Thru Line Loss

Property of niVNA

Specifies the line loss of the through line for manual calibration. A ***through line*** is a direct cable connection between the test ports, with no intervening connectors.

Use the Calibration Thru Reference Frequency property to specify the frequency at which line loss occurs.

Units: dB/mm

Default Value: 0

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Port-based	No
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Calibration Settings:Calibration Thru Reference Frequency Property

Short Name: Calibration Thru Reference Frequency

Property of niVNA

Specifies the through line reference frequency for manual calibration. A ***through line*** is a direct cable connection between the test ports, with no intervening connectors.

Use the Calibration Thru Line Loss property to specify line loss at the frequency specified for this property.

Units: Hz

Default Value: 0

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Port-based	No
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Calibration Settings:AutoCal:AutoCal True Thru Length Property

Short Name: AutoCal True Thru Length

Property of niVNA

Specifies the true through length for the AutoCal. A ***true through*** is a direct cable connection between the test ports, with no intervening connectors. If desired, you can remove the AutoCal module for the through calibration step, and you can supply the parameters of the true through connection using this property.

Units: mm

Default Value: 0

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Port-based	No
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Calibration Settings:AutoCal:AutoCal True Thru Impedance Property

Short Name: AutoCal True Thru Impedance

Property of niVNA

Specifies a true through impedance for the AutoCal. A ***true through*** is a direct cable connection between the test ports, with no intervening connectors. If desired, you can remove the AutoCal module for the through calibration step, and you can specify the parameters of the true through connection using this property.

Units: ohms

Default Value: 50

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Port-based	No
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Calibration Settings:AutoCal:AutoCal True Thru Line Loss Property

Short Name: AutoCal True Thru Line Loss

Property of niVNA

Specifies a line loss for the true through for AutoCal. A ***true through*** is a direct cable connection between the test ports, with no intervening connectors. If desired, you can remove the AutoCal module for the through calibration step, and you can specify the parameters of the true through connection using this property.

Units: dB/mm

Default Value: 0

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Port-based	No
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Calibration Settings:AutoCal:AutoCal True Thru Reference Frequency Property

Short Name: AutoCal True Thru Reference Frequency

Property of niVNA

Specifies a true through reference frequency for the AutoCal. A ***true through*** is a direct cable connection between the test ports, with no intervening connectors. If desired, you can remove the AutoCal module for the through calibration step, and you can specify the parameters of the true through connection using this property.

Units: Hz

Default Value: 0

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Port-based	No
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Calibration Settings:AutoCal:AutoCal Box Orientation Property

Short Name: AutoCal Box Orientation

Property of niVNA

Specifies the orientation of the AutoCal module. If you set this property incorrectly, the AutoCal assurance fails. Enable AutoCal assurance using the AutoCal Assurance Enabled property.

Default Value: **Orientation A**

Supported Devices: NI 5630/5632

<table>
<tr><td>Auto Sense (2)</td><td>Specifies that the VNA determines the orientation of the AutoCal module. The supported calibration is two-port SOLT.</td></tr>
<tr><td>Orientation A (1)</td><td>Specifies that either the VNA PORT 1 connects to the left port on the AutoCal module or PORT 2 connects to the right port on the AutoCal module, as indicated in the following table. An X indicates that the specified port is not applicable.
<table>
<tr><th>Calibration</th><th>Port 1</th><th>Port 2</th></tr>
<tr><td>1 Port (S11)</td><td>Left</td><td>X</td></tr>
<tr><td>1 Port (S22)</td><td>X</td><td>Right</td></tr>
<tr><td>2 Port, 2 Port 1 Path</td><td>Left</td><td>Right</td></tr>
</table></td></tr>
<tr><td>Orientation B (0)</td><td>Specifies that either the VNA PORT 1 connects to the right port on the AutoCal module or PORT 2 connects to the left port on the AutoCal module, as indicated in the following table. An X indicates that the specified port is not applicable.</td></tr>
</table>

Calibration	Port 1	Port 2
1 Port (S11)	Right	X
1 Port (S22)	X	Left
2 Port, 2 Port 1 Path	Right	Left

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Port-based	No
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Calibration Settings:AutoCal:AutoCal Box Communication Port Property

Short Name: AutoCal Box Communication Port

Property of niVNA

Specifies the communication port to which the AutoCal module is connected.

Default Value: "COM1"

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Port-based	No
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Calibration Settings:AutoCal:AutoCal Assurance Enabled Property

Short Name: AutoCal Assurance Enabled

Property of niVNA

Specifies whether an assurance step is automatically executed for the AutoCal. The assurance step verifies whether AutoCal ran successfully. If enabled, the assurance step is performed at the time that the niVNA Close User Cal VI is called.

Default Value: TRUE

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	TF
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Port-based	No
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Calibration Settings:TRL Calibration:Reference Plane Location Property

Short Name: TRL Calibration Reference Plane Location

Property of niVNA

Specifies the location of the reference plane for TRL calibration.

Default Value: **Middle of standard1**

Supported Devices: NI 5632

Ends of standard1 (1)	The reference plane is located at the ends of the "standard1" line standard.

Middle of standard1 (0)	The reference plane is located at the middle of the "standard1" line standard.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Port-based	No
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Calibration Settings:TRL Calibration:Number of Bands Property

Short Name: TRL Calibration Number of Bands

Property of niVNA

Specifies the number of frequency bands to use for TRL calibration.

Default Value: 1

Valid Values: 1 or 2

Supported Devices: NI 5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Port-based	No
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Calibration Settings:TRL Calibration:Standard Type Property

Short Name: TRL Calibration Standard Type

Property of niVNA

Specifies the type of standard used for TRL calibration.

Default Value: **Line**

Supported Devices: NI 5632

Short (0)	The standard is a short-like device.
Open (1)	The standard is an open-like device.

Line (2)	The standard is a line device.
Match (3)	The standard is a matched load.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	No
Standard-based	Yes
Band-based	No
Resettable	No

Calibration Settings:TRL Calibration:Standard Length Property

Short Name: TRL Calibration Standard Length

Property of niVNA

Specifies the length of the TRL calibration standard.

If you set the TRL Calibration Standard Type property to **Line**, the TRL Calibration Standard Length property corresponds to line length in meters. If you set the TRL Calibration Standard Type property to **Open** or **Short**, the TRL Calibration Standard

Length property corresponds to the offset.

	Note NI-VNA ignores this property if you set the TRL Calibration Standard Type property to **Match**.

Units: meters

Default Value: 0

Supported Devices: NI 5632

Remarks

The following table lists the characteristics of this property.

Datatype	DBL
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	No
Standard-based	Yes
Band-based	No
Resettable	No

Calibration Settings:TRL Calibration:Standard Line Loss Property

Short Name: TRL Calibration Standard Line Loss

Property of niVNA

Specifies the line loss of the active standard.

The line loss is proportional to the square root of (***f***/***f0***), where ***f0*** is the TRL Calibration Standard Line Loss Frequency property value and ***f*** is the frequency list defined by the start frequency and stop frequency parameters of the niVNA Configure Frequency VI. If you set the line loss frequency to 0, NI-VNA assumes that the line loss is constant across frequencies. This property is valid only if you set the TRL Calibration Standard Type property to **Line**.

Units: dB/mm

Default Value: 0

Supported Devices: NI 5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	No
Standard-based	Yes
Band-based	No
Resettable	No

Calibration Settings:TRL Calibration:Standard Line Loss Frequency Property

Short Name: TRL Calibration Standard Line Loss Frequency

Property of niVNA

Specifies the frequency at which the TRL Calibration Standard Line Loss property value is valid.

The line loss is proportional to the square root of (***f***/***f0***), where ***f0*** is the TRL Calibration Standard Line Loss Frequency property value and ***f*** is the frequency list defined by the start frequency and stop frequency parameters of the niVNA Configure Frequency VI. If you set the line loss frequency to 0, NI-VNA assumes that the line loss is constant across frequencies. This property is valid only if you set the TRL Calibration Standard Type property to **Line**.

Units: Hz

Default Value: 0

Supported Devices: NI 5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Port-based	No

Device-based	No
Standard-based	Yes
Band-based	No
Resettable	No

Calibration Settings:TRL Calibration:List of Standards Property

Short Name: TRL Calibration List of Standards

Property of niVNA

Specifies the list of TRL calibration standards to include in a TRL calibration band.

Default Value: " "

Valid Values: "standard1", "standard2", "standard3", "standard4", "standard5", or "standard6"

Supported Devices: NI 5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	No
Standard-based	No

Band-based	Yes
Resettable	No

Calibration Settings:TRL Calibration:Band Breakpoint Type Property

Short Name: TRL Calibration Band Breakpoint Type

Property of niVNA

Specifies whether the band breakpoint for TRL calibration is calculated automatically or specified manually using the TRL Calibration Band Breakpoint Frequency property.

The band breakpoint is the frequency at which the TRL calibration stops using the standards associated with one frequency band and begins using the standards associated with the next band.

Default Value: **Automatic**

Supported Devices: NI 5632

Automatic (1)	The band breakpoint is calculated automatically.
Manual (0)	The band breakpoint is manually specified.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write

High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	No
Standard-based	No
Band-based	Yes
Resettable	No

Calibration Settings:TRL Calibration:Band Breakpoint Frequency Property

Short Name: TRL Calibration Band Breakpoint Frequency

Property of niVNA

Specifies the manual frequency band breakpoint at which the TRL calibration stops using the standards associated with this frequency band and uses the subsequent band standards instead.

Writing to this value specifies the manual band breakpoint frequency. Reading from this value returns the automatically calculated value when you set the TRL Calibration Band Breakpoint Type property to **Automatic** and returns the manually specified value when you set the TRL Calibration Band Breakpoint Type property to **Manual**.

Units: Hz

Default Value: 2 GHz

Supported Devices: NI 5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	No
Standard-based	No
Band-based	Yes
Resettable	No

Calibration Settings:Power Meter Name Property

Short Name: Power Meter Name

Property of niVNA

Specifies the resource name of the external power meter used for power calibration.

To determine the power meter name, use the NI-VNA Soft Front Panel (SFP). Within the NI-VNA SFP, select **Power**»**Power Cal**»**Perform Cal** from the Softkey Menu. The power meter name, or COM port name, is listed in the drop-down menu under the **Model_SerialNumber_ComPort** control.

Supported Devices: NI 5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write

High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	Yes
Standard-based	No
Band-based	No
Resettable	No

Calibration Settings:Calibration File Intended For SFP Property

Short Name: Calibration File Intended For SFP

Property of niVNA

Specifies whether the created calibration file is optimized for use with the NI-VNA soft front panel (SFP).

Default Value: FALSE

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Port-based	No

Device-based	No
Standard-based	No
Band-based	No
Resettable	No

External Calibration Information:External Calibration Year Property

Short Name: External Calibration Year

Property of niVNA

Returns the year of the last external calibration.

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	Yes
Standard-based	No
Band-based	No
Resettable	No

External Calibration Information:External Calibration Month Property

Short Name: External Calibration Month

Property of niVNA

Returns the month of the last external calibration. The return value is expressed as an integer. For example, December is represented as 12.

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	Yes
Standard-based	No
Band-based	No
Resettable	No

External Calibration Information:External Calibration Day Property

Short Name: External Calibration Day

Property of niVNA

Returns the day of the last external calibration.

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	Yes
Standard-based	No
Band-based	No
Resettable	No

External Calibration Information:External Calibration Hour Property

Short Name: External Calibration Hour

Property of niVNA

Returns the hour of the last external calibration. The time returned is 24-hour (military) local time; for example, if the device was calibrated at 2:00 PM, this function returns 14 as the value of this property.

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	Yes
Standard-based	No
Band-based	No
Resettable	No

External Calibration Information:External Calibration Minute Property

Short Name: External Calibration Minute

Property of niVNA

Returns the minute of the last external calibration.

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only

High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	Yes
Standard-based	No
Band-based	No
Resettable	No

External Calibration Information:External Calibration Temperature Property

Short Name: External Calibration Temperature

Property of niVNA

Returns the temperature of the last external calibration.

Units: degrees Celsius

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	Yes

Standard-based	No
Band-based	No
Resettable	No

External Calibration Information:External Calibration Interval Property

Short Name: External Calibration Interval

Property of niVNA

Returns the interval since the last external calibration. Complete the steps in the external calibration procedure for the device if the specified interval has elapsed since the last external calibration date.

Units: months

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	Yes
Standard-based	No
Band-based	No
Resettable	No

User Calibration Information:User Calibration Year Property

Short Name: User Calibration Year

Property of niVNA

Returns the year of the last user calibration.

	**Note** This property is currently not supported.

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	Yes
Standard-based	No
Band-based	No
Resettable	No

User Calibration Information:User Calibration Month Property

Short Name: User Calibration Month

Property of niVNA

Returns the month of the last user calibration. The returned value is expressed as an integer. For example, December is represented as 12.

	**Note** This property is currently not supported.

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	Yes
Standard-based	No
Band-based	No
Resettable	No

User Calibration Information:User Calibration Day Property

Short Name: User Calibration Day

Property of niVNA

Returns the day of the last user calibration.

	**Note** This property is currently not supported.

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	Yes
Standard-based	No
Band-based	No
Resettable	No

User Calibration Information:User Calibration Hour Property

Short Name: User Calibration Hour

Property of niVNA

Returns the hour of the last user calibration.

	**Note** This property is currently not supported.

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	Yes
Standard-based	No
Band-based	No
Resettable	No

User Calibration Information:User Calibration Minute Property

Short Name: User Calibration Minute

Property of niVNA

Returns the minute of the last user calibration.

	**Note** This property is currently not supported.

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	Yes
Standard-based	No
Band-based	No
Resettable	No

User Calibration Information:User Calibration Temperature Property

Short Name: User Calibration Temperature

Property of niVNA

Returns the temperature at which the last user calibration was performed.

	Note This property is currently not supported.

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	DBL
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	Yes
Standard-based	No
Band-based	No
Resettable	No

Time Domain:Transform Type Property

Short Name: Time Domain Transform Type

Property of niVNA

Specifies the type of time-domain analysis.

Both Lowpass types require the frequencies to be linearly spaced (harmonic frequencies). Lowpass types provide higher time or distance resolution and also extra information as to the type of impedance mismatch (inductive, resistive, or capacitive) encountered in the DUT. When either of the lowpass modes is selected, NI-VNA does not automatically recalculate and change the frequency to be harmonic.

Supported Devices: NI 5630/5632

Bandpass Impulse (0)	The device performs a time-domain bandpass analysis with simulated stimulus-impulse response. Select this option for general-purpose operation when making time-domain measurements. The Bandpass type allows you to use any frequency range and is a good fit for measuring band-limited devices, such as bandpass filters.
Lowpass Impulse (1)	The device performs a time-domain lowpass analysis with simulated stimulus-impulse response.
Lowpass Step (2)	The device performs a time-domain lowpass analysis with simulated stimulus step response.
Bandpass Phasor Impulse (3)	The device performs a time-domain bandpass analysis with simulated phasor stimulus-impulse response.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A

Channel-based	Yes
Port-based	No
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Time Domain:Trip Type Property

Short Name: Time Domain Trip Type

Property of niVNA

Specifies the trip type for time-domain analysis. Trip type refers to the path the signal takes through the DUT. Setting this property determines how the frequency domain data is interpreted as time-domain data.

This property is valid only if you have activated the Time Domain option on your NI RF vector network analyzer.

Default Value: **Auto**

Supported Devices: NI 5630/5632

Auto (0)	The device determines the trip type based on the S-parameter being measured, and it generates one-way times for both reflection and transmission.
Round-Trip (1)	The device interprets the frequency domain data as traveling on a round-trip path, which means the signal is transmitted to and received from the DUT. Select **Round-Trip** for reflective measurements.
One-Way (2)	The device interprets the frequency domain data as traveling on a one-way path, which

	means the signal as it is transmitted through the DUT. Select **One-Way** for transmission measurements.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Port-based	No
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Time Domain:Impulse Width at 3 dB Property

Short Name: Time Domain Impulse Width at 3 dB

Property of niVNA

Returns the calculated impulse width from time domain analysis. This width implies the resolution of the time-domain analysis calculations.

This property is valid only if you have activated the Time Domain option on your NI RF vector network analyzer and the Time Domain Type property is set to **Lowpass Impulse** or **Bandpass Impulse**.

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	Yes
Port-based	No
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Time Domain:Alias Free Range Property

Short Name: Time Domain Alias Free Range

Property of niVNA

Returns the calculated alias-free range from the time-domain analysis. This value is calculated after each change to the frequency points.

This property is valid only if you have activated the Time Domain option on your NI RF vector network analyzer.

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	DBL
Permissions	Read Only
High-level VIs	N/A
Channel-based	Yes
Port-based	No
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Time Domain:Range:Units Property

Short Name: Time Domain Units

Property of niVNA

Specifies the units for time-domain analysis.

This property is valid only if you have activated the Time Domain option on your NI RF vector network analyzer.

Supported Devices: NI 5630/5632

Time (0)	The time-domain analysis is interpreted in terms of time.
Distance (1)	The time-domain analysis is interpreted in terms of distance.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Port-based	No
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Time Domain:Range:Start Point Property

Short Name: Time Domain Start Point

Property of niVNA

Specifies the start point for the time-domain analysis.

This property is valid only if you have activated the Time Domain option on your NI RF vector network analyzer.

Units: Seconds (s) or meters (m) depending on the Time Domain Units property value.

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	DBL
Permissions	Read/Write
High-level VIs	niVNA Configure Time Domain

Channel-based	Yes
Port-based	No
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Time Domain:Range:Stop Point Property

Short Name: Time Domain Stop Point

Property of niVNA

Specifies the stop point for the time-domain analysis.

This property is valid only if you have activated the Time Domain option on your NI RF vector network analyzer.

Units: Seconds (s) or meters (m) depending on the Time Domain Units property value.

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	niVNA Configure Time Domain
Channel-based	Yes
Port-based	No
Device-based	No

Standard-based	No
Band-based	No
Resettable	No

Time Domain:Window:Shape Property

Short Name: Time Domain Window Shape

Property of niVNA

Specifies the window shape for time-domain analysis.

This property is valid only if you have activated the Time Domain option on your NI RF vector network analyzer and set the Time Domain Type property to either of the lowpass types.

Supported Devices: NI 5630/5632

Rectangular Window (0)	NI-VNA uses a rectangular window for time-domain analysis. This window provides the maximum resolution but the highest side lobes.
Hamming Window (1)	NI-VNA uses a Hamming window for time-domain analysis. This window provides moderate side lobes and resolution. This value is the default.
Blackman Window (2)	NI-VNA uses a Blackman window for time-domain analysis. This window provides low side lobes and low resolution.
Blackman-Harris Window (3)	NI-VNA uses a Blackman-Harris window for time-domain analysis. This window provides the minimum side lobes, but low resolution.
Kaiser-Bessel Window (4)	NI-VNA uses a Kaiser-Bessel window for time-domain analysis. When you select this

	value, you can adjust the window using the Window Kaiser-Bessel Beta property.
Dolph-Chebyshev Window (5)	NI-VNA uses a Dolph-Chebyshev window for time-domain analysis. When you select this value, you can adjust the window using the Window Dolph Chebyshev Gamma property.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Port-based	No
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Time Domain:Window:Kaiser-Bessel Beta Property

Short Name: Time Domain Window Kaiser-Bessel Beta

Property of niVNA

Specifies the Kaiser-Bessel window beta value for time-domain analysis.

Note If you want a window with a lower side lobe, use a wider gate, and vice versa.

This property is valid only if you have activated the Time Domain option on your NI RF vector network analyzer, set the Time Domain Type property to either of the lowpass types, and set the Time Domain Window Shape property value to **Kaiser-Bessel Window**.

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	DBL
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Port-based	No
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Time Domain:Window:Dolph-Chebyshev Gamma Property

Short Name: Time Domain Window Dolph-Chebyshev Gamma

Property of niVNA

Specifies the Dolph-Chebyshev window gamma value for time-domain analysis.

	Note If you want a window with a lower side lobe, use a wider gate, and vice versa.

This property is valid only if you have activated the Time Domain option on your NI RF vector network analyzer, set the Time Domain Type property to either of the lowpass types, and set the Time Domain Window Shape property value to **Dolph-Chebyshev Window**.

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	DBL
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Port-based	No
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Time Domain:Gate:Shape Property

Short Name: Time Domain Gate Shape

Property of niVNA

Specifies the gate shape for the time domain analysis.

This property is valid only if you have activated the Time Domain option on your NI RF vector network analyzer.

Supported Devices: NI 5630/5632

Rectangular (0)	NI-VNA uses a rectangular shape for the time-domain analysis gate. This gate shape provides the maximum resolution but the highest side lobes.
Hamming (1)	NI-VNA uses a Hamming shape for the time-domain analysis gate. This gate shape provides moderate side lobes and resolution. This value is the default.
Blackman (2)	NI-VNA uses a Blackman shape for the time-domain analysis gate. This gate shape provides low side lobes and low resolution.
Blackman-Harris (3)	NI-VNA uses a Blackman-Harris shape for the time-domain analysis gate. This gate shape provides the minimum side lobes, but low resolution.
Kaiser-Bessel (4)	NI-VNA uses a Kaiser-Bessel shape for the time-domain analysis gate. When you select this value, you can adjust the window using the Gate Kaiser-Bessel Beta property.
Dolph-Chebyshev (5)	NI-VNA uses a Dolph-Chebyshev shape for the time-domain analysis gate. When you select this value, you can adjust the window using the Gate Dolph Chebyshev Gamma property.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Port-based	No

Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Time Domain:Gate:Enable Property

Short Name: Time Domain Gate Enable

Property of niVNA

Specifies whether the gate is enabled for the time-domain analysis.

This property is valid only if you have activated the Time Domain option on your NI RF vector network analyzer.

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Port-based	No
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Time Domain:Gate:Inclusive Property

Short Name: Time Domain Gate Inclusive

Property of niVNA

Specifies the scope of the gated time/distance span. If you set this property to TRUE, the data outside the gated span is suppressed. If you set this property to FALSE, the data inside the gated span is suppressed.

This property is valid only if you have activated the Time Domain option on your NI RF vector network analyzer.

Default Value: TRUE

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	niVNA Configure Time Domain
Channel-based	Yes
Port-based	No
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Start Point

Time Domain:Gate:Range:Start Point Property

Short Name: Time Domain Gate Start Point

Property of niVNA

Specifies the starting point for the time domain gate.

This property is valid only if you have activated the Time Domain option on your NI RF vector network analyzer.

Units: Seconds (s) or meters (m)

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	niVNA Configure Time Domain
Channel-based	Yes
Port-based	No
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Stop Point

Time Domain:Gate:Range:Stop Point Property

Short Name: Time Domain Gate Stop Point

Property of niVNA

Specifies the stopping point for the time domain gate.

This property is valid only if you have activated the Time Domain option on your NI RF vector network analyzer.

Units: Seconds (s) or meters (m)

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	DBL
Permissions	Read/Write
High-level VIs	niVNA Configure Time Domain
Channel-based	Yes
Port-based	No
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Time Domain:Gate:Kaiser-Bessel Beta Property

Short Name: Time Domain Gate Kaiser-Bessel Beta

Property of niVNA

Specifies the Kaiser-Bessel beta value for the time-domain analysis gate.

	Note If you use a window with a lower side lobe, use a wider gate, and vice versa.

This property is valid only if you have activated the Time Domain option on your NI RF vector network analyzer, set the Time Domain Type property to either of the lowpass types, and set the Time Domain Gate Shape property value to **Kaiser-Bessel**.

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Port-based	No
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Time Domain:Gate:Dolph-Chebyshev Gamma Property

Short Name: Time Domain Gate Dolph-Chebyshev Gamma

Property of niVNA

Specifies the Dolph-Chebyshev gamma value for the time-domain analysis gate.

Note If you use a window with a lower side lobe, use a wider gate, and vice versa.

This property is valid only if you have activated the Time Domain option on your NI RF vector network analyzer, set the Time Domain Type property to either of the lowpass types, and set the Time Domain Gate Shape property value to **Dolph-Chebyshev**.

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	DBL
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Port-based	No
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Time Domain:DC Term Extrapolation:Type Property

Short Name: Time Domain DC Term Extrapolation Type

Property of niVNA

Specifies the DC term extrapolation type for lowpass time-domain calculations.

This property is valid only if you have activated the Time Domain option on your NI RF vector network analyzer and set the Time Domain Type property to either of the lowpass types.

Supported Devices: NI 5630/5632

Automatic (0)	NI-VNA automatically extrapolates the DC term from existing frequency data used in the time-domain analysis.
User defined (1)	If you select this value, you must supply the DC term used in the time-domain analysis with the Time Domain DC Term Extrapolation User Value property.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Port-based	No
Device-based	No

Standard-based	No
Band-based	No
Resettable	No

Time Domain:DC Term Extrapolation:Method Property

Short Name: Time Domain DC Term Extrapolation Method

Property of niVNA

Specifies the DC term extrapolation method for lowpass time-domain calculations.

This property is valid only if you have activated the Time Domain option on your NI RF vector network analyzer and set the Time Domain Type property to either of the lowpass types.

Supported Devices: NI 5630/5632

Phase Only (0)	Assumes the magnitude remains constant, so only phase information is extrapolated. This option is commonly used when using long cables to connect the VNA to your DUT.
Log Mag & Phase (1)	Extrapolates both magnitude and phase information.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A

Channel-based	Yes
Port-based	No
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Time Domain:DC Term Extrapolation:Coefficient Property

Short Name: Time Domain DC Term Extrapolation Coefficient

Property of niVNA

Specifies the coefficient that is used for the DC term extrapolation calculations. This value corresponds with the DC impedance of the DUT.

This property is valid only if you have activated the Time Domain option on your NI RF vector network analyzer and set the Time Domain Type property to either of the lowpass types.

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	Yes
Port-based	No

Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Time Domain:DC Term Extrapolation:User Value Property

Short Name: Time Domain DC Term Extrapolation User Value

Property of niVNA

Specifies the DC term value for lowpass time-domain calculations.

This property is valid only if you have activated the Time Domain option on your NI RF vector network analyzer and set the Time Domain Type property to either of the lowpass types.

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Port-based	No
Device-based	No
Standard-based	No
Band-based	No

Resettable	No

Device Information:PXI Chassis Property

Short Name: PXI Chassis

Property of niVNA

Returns the identifier for the PXI chassis in which the NI vector network analyzer is installed.

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	Yes
Standard-based	No
Band-based	No
Resettable	No

Device Information:PXI Slot Property

Short Name: PXI Slot

Property of niVNA

Returns the number for the PXI slot in which the NI vector network analyzer is installed.

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	Yes
Standard-based	No
Band-based	No
Resettable	No

Device Information:Device Temperature Property

Short Name: Device Temperature

Property of niVNA

Returns the current temperature of the NI-VNA device.

Units: degrees Celsius

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	Yes
Standard-based	No
Band-based	No
Resettable	No

Device Information:Device Serial Number Property

Short Name: Device Serial Number

Property of niVNA

Returns the serial number of the NI-VNA device.

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only

High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	No
Standard-based	No
Band-based	No
Resettable	No

Device Information:Number of Ports Property

Short Name: Number of Ports

Property of niVNA

Returns the number of ports for the NI vector network analyzer device.

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	Yes
Standard-based	No
Band-based	No

Resettable	No

Inherent IVI Properties:User Options:Cache Property

Short Name: Cache

Property of niVNA

Specifies whether to cache the value of properties. When caching is enabled, NI-VNA tracks the current NI-VNA device settings and avoids sending redundant commands to the device.

NI-VNA can always cache or never cache particular properties, regardless of the setting of this property.

Use the niVNA Initialize With Options VI to override the default value.

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	Yes
Standard-based	No
Band-based	No

Resettable	No

Inherent IVI Properties:User Options:Interchange Check Property

Short Name: Interchange Check

Property of niVNA

Specifies whether to perform interchangeability checking and retrieve interchangeability warnings.

	**Note** Interchangeability checking is currently not supported.

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	Yes
Standard-based	No
Band-based	No
Resettable	No

Inherent IVI Properties:User Options:Driver Setup Property

Short Name: Driver Setup

Property of niVNA

The DriverSetup string is used to set the initial values for properties that are specific to NI-VNA. The format of the DriverSetup string is ***Tag***:***Value***. ***Tag*** is the name of the ***DriverSetup*** string property. ***Value*** is the value set to the property. To set multiple properties, separate their assignments with a semicolon.

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	Yes
Standard-based	No
Band-based	No
Resettable	No

Inherent IVI Properties:User Options:Query Instrument Status Property

Short Name: Query Instrument Status

Property of niVNA

Specifies whether NI-VNA queries the NI-VNA device status after each operation. Querying the device status is useful for debugging. After you validate your program, you can set this property to FALSE to disable status checking and maximize performance.

NI-VNA can choose to ignore status checking for particular properties, regardless of the setting of this property.



Note Use the niVNA Initialize With Options VI to override this value.

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	Yes
Standard-based	No
Band-based	No
Resettable	No

Inherent IVI Properties:User Options:Range Check Property

Short Name: Range Check

Property of niVNA

Specifies whether to validate property values and VI parameters. If enabled, NI-VNA validates the parameter values that you pass to NI-VNA VIs. Range checking parameters is very useful for debugging. After you validate your program, you can set this property to FALSE to disable range checking and maximize performance.

	**Note** Use the niVNA Initialize With Options VI to override this value.

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	Yes
Standard-based	No
Band-based	No
Resettable	No

Inherent IVI Properties:User Options:Record Value Coercions Property

Short Name: Record Value Coercions

Property of niVNA

Specifies whether the IVI engine keeps a list of the value coercions it makes for integer and real type properties.

	**Note** This property is currently not supported.

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	Yes
Standard-based	No
Band-based	No
Resettable	No

Inherent IVI Properties:User Options:Simulate Property

Short Name: Simulate

Property of niVNA

Specifies whether to simulate NI-VNA I/O operations. This property is useful for debugging applications without using hardware. After a session is opened, you cannot change the simulation state. Use the niVNA Initialize With Options VI to enable simulation.

	**Note** NI-VNA does not support simulation.

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	Yes
Standard-based	No
Band-based	No
Resettable	No

Inherent IVI Properties:Driver Identification:Description Property

Short Name: Description

Property of niVNA

Returns a string that contains a brief description of NI-VNA.

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	Yes
Standard-based	No
Band-based	No
Resettable	No

Inherent IVI Properties:Driver Identification:Driver Prefix Property

Short Name: Driver Prefix

Property of niVNA

Returns a string that contains the prefix for NI-VNA. The name of each user-callable VI in NI-VNA starts with this prefix.

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	Yes
Standard-based	No
Band-based	No
Resettable	No

Inherent IVI Properties:Driver Identification:Driver Vendor Property

Short Name: Driver Vendor

Property of niVNA

Returns a string that contains the name of the vendor that supplies NI-VNA.

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	Yes
Standard-based	No
Band-based	No
Resettable	No

Inherent IVI Properties:Driver Identification:Revision Property

Short Name: Revision

Property of niVNA

Returns a string that contains additional version information about NI-VNA.

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	Yes

Standard-based	No
Band-based	No
Resettable	No

Inherent IVI Properties:Driver Identification:Class Specification Major Version Property

Short Name: Class Specification Major Version

Property of niVNA

Returns the major version number of the class specification with which NI-VNA is compliant.

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	Yes
Standard-based	No
Band-based	No
Resettable	No

Inherent IVI Properties:Driver Identification:Class Specification Minor Version Property

Short Name: Class Specification Minor Version

Property of niVNA

Returns the minor version number of the class specification with which NI-VNA is compliant.

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	Yes
Standard-based	No
Band-based	No
Resettable	No

Inherent IVI Properties:Driver Capabilities:Class Group Capabilities Property

Short Name: Class Group Capabilities

Property of niVNA

Returns a string that contains a comma-separated list of class-extension groups that NI-VNA implements.

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	Yes
Standard-based	No
Band-based	No
Resettable	No

Inherent IVI Properties:Driver Capabilities:Supported Instrument Models Property

Short Name: Supported Instrument Models

Property of niVNA

Returns a comma-separated list of supported devices.

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	Yes
Standard-based	No
Band-based	No
Resettable	No

Inherent IVI Properties:Instrument Identification:Manufacturer Property

Short Name: Manufacturer

Property of niVNA

Returns a string that contains the name of the manufacturer for the NI-VNA device you are currently using.

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	Yes
Standard-based	No
Band-based	No
Resettable	No

Inherent IVI Properties:Instrument Identification:Model Property

Short Name: Model

Property of niVNA

Returns a string that contains the model number or name of the NI-VNA device that you are currently using.

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	Yes
Standard-based	No
Band-based	No
Resettable	No

Inherent IVI Properties:Instrument Identification:Firmware Revision Property

Short Name: Firmware Revision

Property of niVNA

Returns a string that contains the firmware revision information for the NI-VNA device you are currently using.

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	Yes
Standard-based	No
Band-based	No
Resettable	No

Inherent IVI Properties:Advanced Session Information:Logical Name Property

Short Name: Logical Name

Property of niVNA

Contains the logical name you specified when opening the current IVI session. You may pass a logical name to the niVNA Initialize VI or the niVNA Initialize With Options VI. The IVI Configuration Utility must contain an entry for the logical name. The logical name entry refers to a driver session section in the IVI Configuration file. The driver session section specifies a physical device and initial user options.

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	Yes
Standard-based	No
Band-based	No
Resettable	No

Inherent IVI Properties:Advanced Session Information:I/O Resource Descriptor Property

Short Name: I/O Resource Descriptor

Property of niVNA

Indicates the resource name NI-VNA uses to identify the physical device. If you initialize NI-VNA with a logical name, this property contains the resource name that corresponds to the entry in the IVI Configuration Utility.

If you initialize NI-VNA with the resource name, this property contains that value.

Supported Devices: NI 5630/5632

Remarks

The following table lists the characteristics of this property.

Datatype	

Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	Yes
Standard-based	No
Band-based	No
Resettable	No

Events:Ready For Start Event:Output Terminal Property

Short Name: Exported Ready for Start Event Output Terminal

Property of niVNA

Specifies the terminal where the signal will be exported. You can also choose to not export a signal.

Supported Devices: NI 5632

Do not export signal ("")	The signal is not exported.
PFI0 ("PFI0")	The signal is exported to the PFI 0 connector, which is the external signal input connector on the module front panel.
PXI_Trig0 ("PXI_Trig0")	The signal is exported to the PXI trigger line 0.
PXI_Trig1 ("PXI_Trig1")	The signal is exported to the PXI trigger line 1.

PXI_Trig2 ("PXI_Trig2")	The signal is exported to the PXI trigger line 2.
PXI_Trig3 ("PXI_Trig3")	The signal is exported to the PXI trigger line 3.
PXI_Trig4 ("PXI_Trig4")	The signal is exported to the PXI trigger line 4.
PXI_Trig5 ("PXI_Trig5")	The signal is exported to the PXI trigger line 5.
PXI_Trig6 ("PXI_Trig6")	The signal is exported to the PXI trigger line 6.
PXI_Trig7 ("PXI_Trig7")	The signal is exported to the PXI trigger line 7.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	niVNA Export Signal
Channel-based	No
Port-based	No
Device-based	Yes
Standard-based	No
Band-based	No
Resettable	No

Events:Ready For Start Event:Active Level Property

Short Name: Ready for Start Event Level

Property of niVNA

Specifies the active level for the Ready For Start Event.

Supported Devices: NI 5632

Low Digital Level (0)	Specifies a low digital level.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	Yes
Standard-based	No
Band-based	No
Resettable	No

Events:Ready For Step Event:Output Terminal Property

Short Name: Exported Ready For Step Event Output Terminal

Property of niVNA

Specifies the terminal where the signal will be exported. You can also choose to not export a signal.

Supported Devices: NI 5632

Do not export signal ("")	The signal is not exported.
PFI0 ("PFI0")	The signal is exported to the PFI 0 connector, which is the external signal input connector on the module front panel.
PXI_Trig0 ("PXI_Trig0")	The signal is exported to the PXI trigger line 0.
PXI_Trig1 ("PXI_Trig1")	The signal is exported to the PXI trigger line 1.
PXI_Trig2 ("PXI_Trig2")	The signal is exported to the PXI trigger line 2.
PXI_Trig3 ("PXI_Trig3")	The signal is exported to the PXI trigger line 3.
PXI_Trig4 ("PXI_Trig4")	The signal is exported to the PXI trigger line 4.
PXI_Trig5 ("PXI_Trig5")	The signal is exported to the PXI trigger line 5.

PXI_Trig6 ("PXI_Trig6")	The signal is exported to the PXI trigger line 6.
PXI_Trig7 ("PXI_Trig7")	The signal is exported to the PXI trigger line 7.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	niVNA Export Signal
Channel-based	No
Port-based	No
Device-based	Yes
Standard-based	No
Band-based	No
Resettable	No

Events:Ready For Step Event:Active Level Property

Short Name: Ready for Step Event Level

Property of niVNA

Specifies the active level for the Ready For Step Event.

Supported Devices: NI 5632

Low Digital Level (0)	Specifies a low digital level.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	Yes
Standard-based	No
Band-based	No
Resettable	No

Events:End Of Step Event:Output Terminal Property

Short Name: Exported End of Step Event Output Terminal

Property of niVNA

Specifies the terminal where the signal will be exported. You can also choose not to export any signal.

Supported Devices: NI 5632

Do not export signal ("")	The signal is not exported.

PFI0 ("PFI0")	The signal is exported to the PFI 0 connector, which is the external signal input connector on the module front panel.
PXI_Trig0 ("PXI_Trig0")	The signal is exported to the PXI trigger line 0.
PXI_Trig1 ("PXI_Trig1")	The signal is exported to the PXI trigger line 1.
PXI_Trig2 ("PXI_Trig2")	The signal is exported to the PXI trigger line 2.
PXI_Trig3 ("PXI_Trig3")	The signal is exported to the PXI trigger line 3.
PXI_Trig4 ("PXI_Trig4")	The signal is exported to the PXI trigger line 4.
PXI_Trig5 ("PXI_Trig5")	The signal is exported to the PXI trigger line 5.
PXI_Trig6 ("PXI_Trig6")	The signal is exported to the PXI trigger line 6.
PXI_Trig7 ("PXI_Trig7")	The signal is exported to the PXI trigger line 7.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	niVNA Export Signal
Channel-based	No

Port-based	No
Device-based	Yes
Standard-based	No
Band-based	No
Resettable	No

Events:End Of Step Event:Edge Property

Short Name: End of Step Event Edge

Property of niVNA

Specifies the active edge for the End of Step Event.

Supported Devices: NI 5632

Falling Edge (0)	The end of the step is signaled by a falling edge.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	Yes
Standard-based	No
Band-based	No

Resettable	No

Events:End Of Sweep Event:Output Terminal Property

Short Name: Exported End of Sweep Event Output Terminal

Property of niVNA

Specifies the terminal where the signal will be exported. You can also choose to not export a signal.

Supported Devices: NI 5632

Do not export signal ("")	The signal is not exported.
PFI0 ("PFI0")	The signal is exported to the PFI 0 connector, which is the external signal input connector on the module front panel.
PXI_Trig0 ("PXI_Trig0")	The signal is exported to the PXI trigger line 0.
PXI_Trig1 ("PXI_Trig1")	The signal is exported to the PXI trigger line 1.
PXI_Trig2 ("PXI_Trig2")	The signal is exported to the PXI trigger line 2.
PXI_Trig3 ("PXI_Trig3")	The signal is exported to the PXI trigger line 3.
PXI_Trig4 ("PXI_Trig4")	The signal is exported to the PXI trigger line 4.

PXI_Trig5 ("PXI_Trig5")	The signal is exported to the PXI trigger line 5.
PXI_Trig6 ("PXI_Trig6")	The signal is exported to the PXI trigger line 6.
PXI_Trig7 ("PXI_Trig7")	The signal is exported to the PXI trigger line 7.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	niVNA Export Signal
Channel-based	No
Port-based	No
Device-based	Yes
Standard-based	No
Band-based	No
Resettable	No

Events:End Of Sweep Event:Edge Property

Short Name: End of Sweep Event Edge

Property of niVNA

Specifies the active edge for the End Of Sweep Event.

Supported Devices: NI 5632

Falling Edge (0)	The end of the sweep is signaled by a falling edge.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Port-based	No
Device-based	Yes
Standard-based	No
Band-based	No
Resettable	No

NI-VNA C Reference Help

September 2014, 373156E-01

This help file contains information about the NI-VNA functions, attributes, and values that you can use when programming your application in C or LabWindows™/CVI™.

To comment on National Instruments documentation, refer to the National Instruments website.

© 2010–2014 National Instruments. All rights reserved.

NI-VNA Functions

Class/Panel Name	Function Name
Initialize	niVNA_init

Initialize With Options	niVNA_InitWithOptions
Configure	
Sweep	
Configure Frequency Center Span	niVNA_ConfigureFrequencyCenterSpan
Configure Frequency Start Stop	niVNA_ConfigureFrequencyStartStop
Configure Frequency Offset Equation	niVNA_ConfigureFrequencyOffsetEquation
Configure Power Start Stop	niVNA_ConfigurePowerStartStop
Configure Sweep	niVNA_ConfigureSweep
Load Sweep Settings Data From File	niVNA_LoadSweepSettingsDataFromFile
Time Domain	
Configure Time Domain Center Span Time	niVNA_ConfigureTimeDomainCenterSpanTime
Configure Time Domain Center Span Distance	niVNA_ConfigureTimeDomainCenterSpanDistance
Configure Time Domain Start Stop Time	niVNA_ConfigureTimeDomainStartStopTime
Configure Time Domain Start Stop Distance	niVNA_ConfigureTimeDomainStartStopDistance
Gate	
Configure Time Domain Gate Center Span Time	niVNA_ConfigureTimeDomainGateCenterSpanTime
Configure Time Domain Gate Center Span Distance	niVNA_ConfigureTimeDomainGateCenterSpanDistanc
Configure Time Domain Gate Start Stop Time	niVNA_ConfigureTimeDomainGateStartStopTime
Configure Time Domain Gate Start Stop Distance	niVNA_ConfigureTimeDomainGateStartStopDistance
Trigger	
Start Trigger	
Configure Digital Edge Start Trigger	niVNA_ConfigureDigitalEdgeStartTrigger
Configure Software Edge Start Trigger	niVNA_ConfigureSoftwareEdgeStartTrigger
Disable Start Trigger	niVNA_DisableStartTrigger
Step Trigger	
Configure Digital Edge Step Trigger	niVNA_ConfigureDigitalEdgeStepTrigger
Disable Step Trigger	niVNA_DisableStepTrigger
Send Software Edge Trigger	niVNA_SendSoftwareEdgeTrigger

Export Signal	niVNA_ExportSignal
Set/Get Attribute	
Set Attribute	
Set Attribute ViInt32	niVNA_SetAttributeViInt32
Set Attribute ViInt64	niVNA_SetAttributeViInt64
Set Attribute ViReal64	niVNA_SetAttributeViReal64
Set Attribute ViString	niVNA_SetAttributeViString
Set Attribute ViBoolean	niVNA_SetAttributeViBoolean
Set Attribute ViSession	niVNA_SetAttributeViSession
Get Attribute	
Get Attribute ViInt32	niVNA_GetAttributeViInt32
Get Attribute ViInt64	niVNA_GetAttributeViInt64
Get Attribute ViReal64	niVNA_GetAttributeViReal64
Get Attribute ViString	niVNA_GetAttributeViString
Get Attribute ViBoolean	niVNA_GetAttributeViBoolean
Get Attribute ViSession	niVNA_GetAttributeViSession
Invalidate All Attributes	niVNA_InvalidateAllAttributes
Connector Configuration	
Configure Connector	niVNA_ConfigureConnector
Perform Auto Reference Plane Extension	niVNA_PerformAutoReferencePlaneExtension
Measurement	
Sweep	
Initiate	niVNA_Initiate
Commit	niVNA_Commit
Abort	niVNA_Abort
Is Sweep Done	niVNA_IsSweepDone
Wait For Sweep Done	niVNA_WaitForSweepDone
Fetch Sweep Measurement F64	niVNA_FetchSweepMeasurementF64

Fetch Sweep Measurement Complex	niVNA_FetchSweepMeasurementComplex
Fetch Sweep Measurement Into File	niVNA_FetchSweepMeasurementIntoFile
Fetch Time Domain Transform F64	niVNA_FetchTimeDomainTransformF64
Fetch Time Domain Gate Shape F64	niVNA_FetchTimeDomainGateShapeF64
Fetch Receiver Measurement F64	niVNA_FetchReceiverMeasurementF64
Fetch Receiver Measurement Complex	niVNA_FetchReceiverMeasurementComplex
Calibration	
Init User Calibration	niVNA_InitUserCal
Perform Calibration Action	niVNA_PerformCalibrationAction
Perform TRL Calibration Action	niVNA_PerformTRLCalibrationAction
Apply Adapter Removal Calibration	niVNA_ApplyAdapterRemovalCalibration
Get Calibration State	niVNA_GetCalibrationState
Is Calibration Action Done	niVNA_IsCalibrationActionDone
Wait For Calibration Action Done	niVNA_WaitForCalibrationActionDone
Abort Calibration Action	niVNA_AbortCalibrationAction
Close User Calibration	niVNA_CloseUserCal
Load Calibration And Configuration Data From File	niVNA_LoadCalibrationAndConfigurationDataFromFil
Save Calibration And Configuration Data To File	niVNA_SaveCalibrationAndConfigurationDataToFile
Load User Defined Manual Calibration Kit Data From File	niVNA_LoadUserDefinedManualCalibrationKitDataFro
AutoCal	
Query AutoCal Box Info	niVNA_QueryAutoCalBoxInfo
Load AutoCal Characterization Data From File	niVNA_LoadAutoCalCharacterizationDataFromFile
Utility	
Self test	niVNA_self_test
Reset	niVNA_reset
Reset With Defaults	niVNA_ResetWithDefaults
Disable	niVNA_Disable
Revision query	niVNA_revision_query

Feature Activation	
Activate Feature	niVNA_ActivateFeature
Query Feature Activation Status	niVNA_QueryFeatureActivationStatus
Error Info	
Error query	niVNA_error_query
Error message	niVNA_error_message
Get Error	niVNA_GetError
Clear Error	niVNA_ClearError
Get Next Coercion Record	niVNA_GetNextCoercionRecord
Interchangeability Info	
Get Next Interchange Warning	niVNA_GetNextInterchangeWarning
Reset Interchange Check	niVNA_ResetInterchangeCheck
Clear Interchange Warnings	niVNA_ClearInterchangeWarnings
Locking	
Lock Session	niVNA_LockSession
Unlock Session	niVNA_UnlockSession
Close	niVNA_close

niVNA_init

niVNA_init(ViRsrc resourceName, ViBoolean IDQuery, ViBoolean reset, ViSession *instrumentHandle);

Purpose

Creates a new session for the device.

Supported Devices: NI 5630/5632

Parameters

<table>
<tr><th colspan="3">Input</th></tr>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td>resourceName</td><td>ViRsrc</td><td>Specifies the resource name of the device to initialize.

<table>
<tr><th>Example #</th><th>Device Type</th><th>Syntax</th></tr>
<tr><td>1</td><td><code>myDAQmxDevice</code></td><td>NI-DAQmx device, device name = "myDAQmxDevice"</td></tr>
<tr><td>2</td><td>myLogicalName</td><td>IVI logical name, name = "myLogicalName"</td></tr>
</table>

For NI-DAQmx devices, the syntax is the device name specified in MAX, as shown in Example 1. Typical default names for NI-DAQmx devices in MAX are <code>Dev1</code> or <code>PXI1Slot2</code>. You can rename an NI-DAQmx device by right-clicking on the name in MAX and entering a new name. You also can pass in the name of an IVI logical name configured with the IVI Configuration utility. For additional information, refer to the IVI topic of the Measurement & Automation Explorer Help.

<table>
<tr><td>  </td><td>Caution NI-DAQmx device names are not case-sensitive. However, IVI logical names are case-sensitive. If you use an IVI logical name, verify the name is identical to the name shown in the IVI Configuration Utility.</td></tr>
</table>
</td></tr>
<tr><td>IDQuery</td><td>ViBoolean</td><td>Specifies whether NI-VNA performs an ID query. When you perform an ID query, NI-VNA verifies that the device you initialize is supported.

<table>
<tr><td>VI_TRUE (Yes)</td><td>Perform an ID query. This value is the default.</td></tr>
<tr><td>VI_FALSE (No)</td><td>Do not perform an ID query.</td></tr>
</table>
</td></tr>
<tr><td>reset</td><td>ViBoolean</td><td>Specifies whether the NI-VNA device is reset during the initialization</td></tr>
</table>

		procedure. VI_TRUE (Yes) — The device is reset. VI_FALSE (No) — The device is not reset. This value is the default.
Output		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function. The general meaning of the status code is as follows: **Value** — **Meaning** 0 — Success Positive Values — Warnings Negative Values — Errors

niVNA_InitWithOptions

niVNA_InitWithOptions(ViRsrc resourceName, ViBoolean idQuery, ViBoolean reset, ViConstString optionString, ViSession *instrumentHandle);

Purpose

Creates a new session to the device and specifies initialization options, such as setting default values or enabling simulation.

Supported Devices: NI 5630/5632

Parameters

<table>
<tr><td colspan="3">Input</td></tr>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td>resourceName</td><td>ViRsrc</td><td>Specifies the resource name of the device to initialize.

<table>
<tr><th>Example #</th><th>Device Type</th><th>Syntax</th></tr>
<tr><td>1</td><td><code>myDAQmxDevice</code></td><td>NI-DAQmx device, device name = "myDAQmxDevice"</td></tr>
<tr><td>2</td><td>myLogicalName</td><td>IVI logical name, name = "myLogicalName"</td></tr>
</table>

For NI-DAQmx devices, the syntax is the device name specified in MAX, as shown in Example 1. Typical default names for NI-DAQmx devices in MAX are <code>Dev1</code> or <code>PXI1Slot2</code>. You can rename an NI-DAQmx device by right-clicking on the name in MAX and entering a new name. You also can pass in the name of an IVI logical name configured with the IVI Configuration utility. For additional information, refer to the IVI topic of the Measurement & Automation Explorer Help.

<table>
<tr><td>  </td><td>Caution NI-DAQmx device names are not case-sensitive. However, IVI logical names are case-sensitive. If you use an IVI logical name, verify the name is identical to the name shown in the IVI Configuration Utility.</td></tr>
</table>
</td></tr>
<tr><td>IDQuery</td><td>ViBoolean</td><td>Specifies whether NI-VNA performs an ID query. When you perform an ID query, NI-VNA verifies that the device you initialize is</td></tr>
</table>

		supported. VI_TRUE (Yes) — Perform an ID query. This value is the default. VI_FALSE (No) — Do not perform an ID query.
reset	ViBoolean	Specifies whether the NI-VNA device is reset during the initialization procedure. VI_TRUE (Yes) — The device is reset. VI_FALSE (No) — The device is not reset. This value is the default.
optionString	ViBoolean	Sets the initial value of certain properties for the session. The following attributes are used in this parameter: • NIVNA_ATTR_RANGE_CHECK • NIVNA_ATTR_QUERY_INSTRUMENT_STATUS • NIVNA_ATTR_CACHE • NIVNA_ATTR_RECORD_COERCIONS • NIVNA_ATTR_DRIVER_SETUP • NIVNA_ATTR_SIMULATE The format of this string is "AttributeName=Value," where ***AttributeName*** is the name of the attribute and ***Value*** is the value to which the attribute is set. To set multiple attributes, separate their assignments with a comma.
Output		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.

Return Value

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>returnValue</td><td>ViStatus</td><td>Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred.

To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function.

The general meaning of the status code is as follows:

<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>0</td><td>Success</td></tr>
<tr><td>Positive Values</td><td>Warnings</td></tr>
<tr><td>Negative Values</td><td>Errors</td></tr>
</table>
</td></tr>
</table>

Configure

Class/Panel Name	Function Name
Configure	
Sweep	
Configure Frequency Center Span	niVNA_ConfigureFrequencyCenterSpan
Configure Frequency Start Stop	niVNA_ConfigureFrequencyStartStop
Configure Frequency Offset Equation	niVNA_ConfigureFrequencyOffsetEquation
Configure Power Start Stop	niVNA_ConfigurePowerStartStop
Configure Sweep	niVNA_ConfigureSweep
Load Sweep Settings Data From File	niVNA_LoadSweepSettingsDataFromFile
Time Domain	

Configure Time Domain Center Span Time	niVNA_ConfigureTimeDomainCenterSpanTime
Configure Time Domain Center Span Distance	niVNA_ConfigureTimeDomainCenterSpanDistance
Configure Time Domain Start Stop Time	niVNA_ConfigureTimeDomainStartStopTime
Configure Time Domain Start Stop Distance	niVNA_ConfigureTimeDomainStartStopDistance
Gate	
Configure Time Domain Gate Center Span Time	niVNA_ConfigureTimeDomainGateCenterSpanTime
Configure Time Domain Gate Center Span Distance	niVNA_ConfigureTimeDomainGateCenterSpanDistance
Configure Time Domain Gate Start Stop Time	niVNA_ConfigureTimeDomainGateStartStopTime
Configure Time Domain Gate Start Stop Distance	niVNA_ConfigureTimeDomainGateStartStopDistance
Trigger	
Start Trigger	
Configure Digital Edge Start Trigger	niVNA_ConfigureDigitalEdgeStartTrigger
Configure Software Edge Start Trigger	niVNA_ConfigureSoftwareEdgeStartTrigger
Disable Start Trigger	niVNA_DisableStartTrigger
Step Trigger	
Configure Digital Edge Step Trigger	niVNA_ConfigureDigitalEdgeStepTrigger
Disable Step Trigger	niVNA_DisableStepTrigger
Send Software Edge Trigger	niVNA_SendSoftwareEdgeTrigger
Export Signal	niVNA_ExportSignal
Set/Get Attribute	
Set Attribute	
Set Attribute ViInt32	niVNA_SetAttributeViInt32
Set Attribute ViInt64	niVNA_SetAttributeViInt64
Set Attribute ViReal64	niVNA_SetAttributeViReal64
Set Attribute ViString	niVNA_SetAttributeViString
Set Attribute ViBoolean	niVNA_SetAttributeViBoolean
Set Attribute ViSession	niVNA_SetAttributeViSession
Get Attribute	

Get Attribute ViInt32	niVNA_GetAttributeViInt32
Get Attribute ViInt64	niVNA_GetAttributeViInt64
Get Attribute ViReal64	niVNA_GetAttributeViReal64
Get Attribute ViString	niVNA_GetAttributeViString
Get Attribute ViBoolean	niVNA_GetAttributeViBoolean
Get Attribute ViSession	niVNA_GetAttributeViSession
Invalidate All Attributes	niVNA_InvalidateAllAttributes
Connector Configuration	
Configure Connector	niVNA_ConfigureConnector
Perform Auto Reference Plane Extension	niVNA_PerformAutoReferencePlaneExtension

Sweep

Class/Panel Name	Function Name
Sweep	
Configure Frequency Center Span	niVNA_ConfigureFrequencyCenterSpan
Configure Frequency Start Stop	niVNA_ConfigureFrequencyStartStop
Configure Frequency Offset Equation	niVNA_ConfigureFrequencyOffsetEquation
Configure Power Start Stop	niVNA_ConfigurePowerStartStop
Configure Sweep	niVNA_ConfigureSweep
Load Sweep Settings Data From File	niVNA_LoadSweepSettingsDataFromFile

niVNA_ConfigureFrequencyCenterSpan

niVNA_ConfigureFrequencyCenterSpan(ViSession instrumentHandle, ViConstString channel, ViReal64 centerFrequency, ViReal64 frequencySpan);

Purpose

Configures the start and stop frequencies for a frequency sweep, based on the center frequency and frequency span.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.
channel	ViConstString	Specifies the channel for which this operation is targeted.
centerFrequency	ViReal64	Specifies the center frequency in a frequency sweep. The value is expressed in hertz (Hz).
frequencySpan	ViReal64	Specifies the frequency span for the frequency sweep. This frequency span is centered at the value you select for **centerFrequency**. The value is expressed in hertz (Hz).

Return Value

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>returnValue</td><td>ViStatus</td><td>Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred.

To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function.

The general meaning of the status code is as follows:

<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>0</td><td>Success</td></tr>
</table>
</td></tr>
</table>

		Positive Values: Warnings Negative Values: Errors

niVNA_ConfigureFrequencyStartStop

niVNA_ConfigureFrequencyStartStop(ViSession instrumentHandle, ViConstString channel, ViReal64 startFrequency, ViReal64 stopFrequency);

Purpose

Configures the start and stop frequency for the frequency sweep.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.
channel	ViConstString	Specifies the channel for which this operation is targeted.
startFrequency	ViReal64	Specifies the lower limit of a span of frequencies. This value is expressed in hertz (Hz). **Valid Values**: • NI 5630—≥10 MHz • NI 5632—≥300 kHz
stopFrequency	ViReal64	Specifies the upper limit of a span of frequencies. This value is expressed in hertz (Hz). **Valid Values**:

		• NI 5630—≤6 GHz • NI 5632—≤8.5 GHz

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function. The general meaning of the status code is as follows: **Value** — **Meaning** 0 — Success Positive Values — Warnings Negative Values — Errors

niVNA_ConfigureFrequencyOffsetEquation

niVNA_ConfigureFrequencyOffsetEquation(ViSession instrumentHandle, ViConstString channel, ViInt32 srcReceiver, ViReal64 multiplier, ViReal64 offset);

Purpose

Configures the frequency offset equation that helps control the source and receiver.

The frequency offset equation is (***M*** * ***f***) + ***b***, where ***M*** is the **multiplier**, ***f*** is the

frequency list defined by the **startFrequency** and **stopFrequency** parameters of the niVNA_ConfigureFrequencyStartStop function, and ***b*** is the **offset**.

Supported Devices: NI 5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.
channel	ViConstString	Specifies the channel for which this operation is targeted. If you set **srcReceiver** to NIVNA_VAL_SOURCE, the valid channel specifier is "port1" or "port2". If you set **srcReceiver** to NIVNA_VAL_RECEIVER, the valid channel specifier is "".
srcReceiver	ViInt32	Specifies whether the frequency offset equation applies to either the source or receiver. NIVNA_VAL_SOURCE: The frequency offset equation is applied to the source frequency. NIVNA_VAL_RECEIVER: The frequency offset equation is applied to the receiver frequency.
multiplier	ViInt32	Specifies the ***M*** value in the frequency offset equation.
offset	ViReal64	Specifies the ***b*** value in the frequency offset equation.

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the

		error information from NI-VNA, call the niVNA_ClearError function. The general meaning of the status code is as follows: **Value** / **Meaning** 0 / Success Positive Values / Warnings Negative Values / Errors

niVNA_ConfigurePowerStartStop

niVNA_ConfigurePowerStartStop(ViSession instrumentHandle, ViConstString portList, ViReal64 startPower, ViReal64 stopPower);

Purpose

Configures the start and stop power of the power sweep.

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.
portList	ViConstString	Specifies the port(s) for which the start and stop power of the power sweep are applied. Valid port list specifiers are "`port1`", "`port2`", or "`port1,port2`".
startPower	ViReal64	Specifies the start power for power sweep. The defualt value is -30. This value is expressed in dBm. **Valid Range**: ≥-30 to ≤10
stopPower	ViReal64	Specifies the stop power for power sweep. The default value is 10. This value is expressed in dBm.

		Valid Range: ≥-30 to ≤10

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function. The general meaning of the status code is as follows: **Value** / **Meaning** 0 / Success Positive Values / Warnings Negative Values / Errors

niVNA_ConfigureSweep

niVNA_ConfigureSweep(ViSession instrumentHandle, ViConstString channel, ViInt32 numberOfPoints, ViInt32 sweepMode);

Purpose

Configures the number of points for the sweep measurement and the sweep mode.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.
channel	ViConstString	Specifies the channel for which this operation is targeted.
numberOfPoints	ViInt32	Specifies the number of points to acquire as part of the sweep measurement. **Valid Ranges:** • NI 5630—1 to 3,201 • NI 5632—1 to 20,001 **Default Value**: 201
sweepMode	ViInt32	Specifies the mode for the sweep measurement. For the NI 5630, NIVNA_VAL_SWEEP_FREQUENCY is the only supported value. NIVNA_VAL_SWEEP_FREQUENCY — The device performs a frequency sweep. NIVNA_VAL_SWEEP_POWER — The device performs a power sweep.

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information

		about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function. The general meaning of the status code is as follows: **Value** / **Meaning** 0 / Success Positive Values / Warnings Negative Values / Errors

niVNA_LoadSweepSettingsDataFromFile

niVNA_LoadSweepSettingsDataFromFile(ViSession instrumentHandle, ViConstString filePath, ViConstString setting);

Purpose

Loads sweep settings data from a configuration file that you specify with the **filePath** parameter.

Refer to Using the Sweep Settings Configuration File for information about using the configuration file.

Supported Devices: NI 5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.
filePath	ViConstString	Specifies the absolute file path to the configuration file that contains the sweep settings.

setting	ViConstString	Specifies the name of the sweep setting in the configuration file.

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function. The general meaning of the status code is as follows: **Value** / **Meaning** 0 / Success Positive Values / Warnings Negative Values / Errors

Time Domain

Class/Panel Name	Function Name
Time Domain	
Configure Time Domain Center Span Time	niVNA_ConfigureTimeDomainCenterSpanTime
Configure Time Domain Center Span Distance	niVNA_ConfigureTimeDomainCenterSpanDistance
Configure Time Domain Start Stop Time	niVNA_ConfigureTimeDomainStartStopTime
Configure Time Domain Start Stop Distance	niVNA_ConfigureTimeDomainStartStopDistance
Gate	
Configure Time Domain Gate Center Span Time	niVNA_ConfigureTimeDomainGateCenterSpanTime

Configure Time Domain Gate Center Span Distance	niVNA_ConfigureTimeDomainGateCenterSpanDistance
Configure Time Domain Gate Start Stop Time	niVNA_ConfigureTimeDomainGateStartStopTime
Configure Time Domain Gate Start Stop Distance	niVNA_ConfigureTimeDomainGateStartStopDistance

niVNA_ConfigureTimeDomainCenterSpanTime

niVNA_ConfigureTimeDomainCenterSpanTime(ViSession instrumentHandle, ViConstString channel, ViReal64 centerPoint, ViReal64 span);

Purpose

Configures the start and stop times for the time-domain analysis based on the the **centerPoint** and **span** you specify. This function calculates the span in terms of time. If you want to configure a time-domain analysis span in terms of distance, use the `niVNA_ConfigureTimeDomainCenterSpanDistance` function.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.
channel	ViConstString	Specifies the channel for which this operation is targeted.
centerPoint	ViReal64	Specifies the center time of the span used for the time-domain analysis. This value is expressed in seconds.
span	ViReal64	Specifies the time span for the time-domain analysis. This value is expressed in seconds.

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates

		success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function. The general meaning of the status code is as follows: **Value** — **Meaning** 0 — Success Positive Values — Warnings Negative Values — Errors

niVNA_ConfigureTimeDomainCenterSpanDistance

niVNA_ConfigureTimeDomainCenterSpanDistance(ViSession instrumentHandle, ViConstString channel, ViReal64 centerPoint, ViReal64 span);

Purpose

Configures the start and stop points for the time-domain analysis based on the the **centerPoint** and **span** you specify. This function configures the span in terms of distance. If you want to configure a time-domain analysis span in terms of time, use the `niVNA_ConfigureTimeDomainCenterSpanTime` function.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.

channel	ViConstString	Specifies the channel for which this operation is targeted.
centerPoint	ViReal64	Specifies the center point, in meters, of the span used for the time-domain analysis.
span	ViReal64	Specifies the span, in meters, for the time-domain analysis.

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function. The general meaning of the status code is as follows: **Value** / **Meaning** 0 / Success Positive Values / Warnings Negative Values / Errors

niVNA_ConfigureTimeDomainStartStopTime

niVNA_ConfigureTimeDomainStartStopTime(ViSession instrumentHandle, ViConstString channel, ViReal64 startPoint, ViReal64 stopPoint);

Purpose

Configures the start and stop points for the time-domain analysis. This function configures the start and stop points in terms of time. If you want to configure the start

and stop points in terms of distance, use the niVNA_ConfigureTimeDomainStartStopDistance function.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.
channel	ViConstString	Specifies the channel for which this operation is targeted.
startPoint	ViReal64	Specifies the starting point for the time domain analysis. This value is expressed in seconds.
stopPoint	ViReal64	Specifies the stopping point for the time domain analysis. This value is expressed in seconds.

Return Value

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>returnValue</td><td>ViStatus</td><td>Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred.

To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function.

The general meaning of the status code is as follows:

<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>0</td><td>Success</td></tr>
<tr><td>Positive Values</td><td>Warnings</td></tr>
<tr><td>Negative Values</td><td>Errors</td></tr>
</table>
</td></tr>
</table>

niVNA_ConfigureTimeDomainStartStopDistance

niVNA_ConfigureTimeDomainStartStopDistance(ViSession instrumentHandle, ViConstString channel, ViReal64 startPoint, ViReal64 stopPoint);

Purpose

Configures the start and stop points for the time-domain analysis. This function configures the start and stop points in terms of distance. If you want to configure the start and stop points in terms of time, use the niVNA_ConfigureTimeDomainStartStopTime function.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.
channel	ViConstString	Specifies the channel for which this operation is targeted.
startPoint	ViReal64	Specifies the starting point for the time domain analysis. This value is expressed in meters.
stopPoint	ViReal64	Specifies the stopping point for the time domain analysis. This value is expressed in meters.

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function.

		The general meaning of the status code is as follows: **Value** / **Meaning** 0 — Success Positive Values — Warnings Negative Values — Errors

Gate

Class/Panel Name	Function Name
Gate	
Configure Time Domain Gate Center Span Time	niVNA_ConfigureTimeDomainGateCenterSpanTime
Configure Time Domain Gate Center Span Distance	niVNA_ConfigureTimeDomainGateCenterSpanDistance
Configure Time Domain Gate Start Stop Time	niVNA_ConfigureTimeDomainGateStartStopTime
Configure Time Domain Gate Start Stop Distance	niVNA_ConfigureTimeDomainGateStartStopDistance

niVNA_ConfigureTimeDomainGateCenterSpan

niVNA_ConfigureTimeDomainGateCenterSpanTim

niVNA_ConfigureTimeDomainGateCenterSpanTime(ViSession instrumentHandle, ViConstString channel, ViReal64 gateCenterPoint, ViReal64 gateSpan, ViBoolean gateInclusive);

Purpose

Configures the start and stop times for the gate used in time-domain analysis based on the the **gateCenterPoint** and **gateSpan** you specify. This function calculates the span

in terms of time. If you want to configure a gate span in terms of distance, use the niVNA_ConfigureTimeDomainGateCenterSpanDistance function.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.
channel	ViConstString	Specifies the channel for which this operation is targeted.
gateCenterPoint	ViReal64	Specifies the center point of the gate used in the time-domain analysis. This value is expressed in seconds.
gateSpan	ViReal64	Specifies the span, in seconds, of the gate used in the time-domain analysis.
gateInclusive	ViBoolean	Specifies whether the gate function is inclusive or exclusive. If the gate function is inclusive, the time-domain operation includes only the data inside the **gateSpan**, but if it is exclusive, the time-domain operation excludes the data inside the **gateSpan**.

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function. The general meaning of the status code is as follows: **Value** | **Meaning**

		0 Success Positive Values Warnings Negative Values Errors

niVNA_ConfigureTimeDomainGateCenterSpa

niVNA_ConfigureTimeDomainGateCenterSpanDis

niVNA_ConfigureTimeDomainGateCenterSpanDistance(ViSession instrumentHandle, ViConstString channel, ViReal64 gateCenterPoint, ViReal64 gateSpan, ViBoolean gateInclusive);

Purpose

Configures the start and stop points for the gate used in time-domain analysis based on the the **gateCenterPoint** and **gateSpan** you specify. This function configures the span in terms of distance. If you want to configure a time-domain analysis gate span in terms of time, use the `niVNA_ConfigureTimeDomainGateCenterSpanTime` function.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.
channel	ViConstString	Specifies the channel for which this operation is targeted.
gateCenterPoint	ViReal64	Specifies the center point of the gate used in the time-domain analysis. This value is expressed in meters.

gateSpan	ViReal64	Specifies the span of the gate used in the time-domain analysis. This value is expressed in meters.
gateInclusive	ViBoolean	Specifies whether the gate function is inclusive or exclusive. If the gate function is inclusive, the time-domain operation includes only the data inside the **gateSpan**, but if it is exclusive, the time-domain operation excludes the data inside the **gateSpan**.

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function. The general meaning of the status code is as follows: **Value** / **Meaning** 0 / Success Positive Values / Warnings Negative Values / Errors

niVNA_ConfigureTimeDomainGateStartStopTi

niVNA_ConfigureTimeDomainGateStartStopTime

niVNA_ConfigureTimeDomainGateStartStopTime(ViSession instrumentHandle, ViConstString channel, ViReal64 gateStartPoint, ViReal64 gateStopPoint, ViBoolean gateInclusive);

Purpose

Configures the start and stop points for the gate used in time-domain analysis. This function configures the start and stop points in terms of time. If you want to configure the start and stop points for the gate span in terms of distance, use the `niVNA_ConfigureTimeDomainGateStartStopDistance` function.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.
channel	ViConstString	Specifies the channel for which this operation is targeted.
gateStartPoint	ViReal64	Specifies the start point for the gate used in the time-domain analysis. This value is expressed in seconds.
gateStopPoint	ViReal64	Specifies the stop point for the gate used in the time-domain analysis. This value is expressed in seconds.
gateInclusive	ViBoolean	Specifies whether the gate function is inclusive or exclusive. If the gate function is inclusive, the time-domain operation includes only the data between the **gateStartPoint** and **gateStopPoint**, but if it is exclusive, the time-domain operation excludes the data between the **gateStartPoint** and **gateStopPoint**.

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function.

		The general meaning of the status code is as follows: **Value** / **Meaning** 0 — Success Positive Values — Warnings Negative Values — Errors

niVNA_ConfigureTimeDomainGateStartStopD

niVNA_ConfigureTimeDomainGateStartStopDistar

niVNA_ConfigureTimeDomainGateStartStopDistance(ViSession instrumentHandle, ViConstString channel, ViReal64 gateStartPoint, ViReal64 gateStopPoint, ViBoolean gateInclusive);

Purpose

Configures the start and stop points for the gate used in time-domain analysis. This function configures the start and stop points in terms of distance. If you want to configure the start and stop points in terms of time, use the niVNA_ConfigureTimeDomainGateStartStopTime function.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.
channel	ViConstString	Specifies the channel for which this operation is targeted.

gateStartPoint	ViReal64	Specifies the start point for the gate used in the time-domain analysis. This value is expressed in meters.
gateStopPoint	ViReal64	Specifies the stop point for the gate used in the time-domain analysis. This value is expressed in meters.
gateInclusive	ViBoolean	Specifies whether the gate function is inclusive or exclusive. If the gate function is inclusive, the time-domain operation includes only the data between the **gateStartPoint** and **gateStopPoint**, but if it is exclusive, the time-domain operation excludes the data between the **gateStartPoint** and **gateStopPoint**.

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function. The general meaning of the status code is as follows: **Value** / **Meaning** 0 / Success Positive Values / Warnings Negative Values / Errors

Trigger

Class/Panel Name	Function Name
Trigger	

Start Trigger	
Configure Digital Edge Start Trigger	niVNA_ConfigureDigitalEdgeStartTrigger
Configure Software Edge Start Trigger	niVNA_ConfigureSoftwareEdgeStartTrigger
Disable Start Trigger	niVNA_DisableStartTrigger
Step Trigger	
Configure Digital Edge Step Trigger	niVNA_ConfigureDigitalEdgeStepTrigger
Disable Step Trigger	niVNA_DisableStepTrigger
Send Software Edge Trigger	niVNA_SendSoftwareEdgeTrigger
Export Signal	niVNA_ExportSignal

Start Trigger

Class/Panel Name	Function Name
Start Trigger	
Configure Digital Edge Start Trigger	niVNA_ConfigureDigitalEdgeStartTrigger
Configure Software Edge Start Trigger	niVNA_ConfigureSoftwareEdgeStartTrigger
Disable Start Trigger	niVNA_DisableStartTrigger

niVNA_ConfigureDigitalEdgeStartTrigger

niVNA_ConfigureDigitalEdgeStartTrigger

niVNA_ConfigureDigitalEdgeStartTrigger(ViSession instrumentHandle, ViConstString triggerSource, ViInt32 triggerEdge);

Purpose

Configures the device to wait for a digital edge Start Trigger before starting a sweep.

Supported Devices: NI 5630/5632

Parameters

<table>
<tr><th colspan="3">Input</th></tr>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td>instrumentHandle</td><td>ViSession</td><td>Identifies your instrument handle.</td></tr>
<tr><td>triggerSource</td><td>ViConstString</td><td>Specifies the source of the digital edge for the trigger.

<table>
<tr><td>NIVNA_VAL_PFI0</td><td>The trigger is received on PFI 0.</td></tr>
<tr><td>NIVNA_VAL_PXI_TRIG0</td><td>The trigger is received on PXI trigger line 0.</td></tr>
<tr><td>NIVNA_VAL_PXI_TRIG1</td><td>The trigger is received on PXI trigger line 1.</td></tr>
<tr><td>NIVNA_VAL_PXI_TRIG2</td><td>The trigger is received on PXI trigger line 2.</td></tr>
<tr><td>NIVNA_VAL_PXI_TRIG3</td><td>The trigger is received on PXI trigger line 3.</td></tr>
<tr><td>NIVNA_VAL_PXI_TRIG4</td><td>The trigger is received on PXI trigger line 4.</td></tr>
<tr><td>NIVNA_VAL_PXI_TRIG5</td><td>The trigger is received on PXI trigger line 5.</td></tr>
<tr><td>NIVNA_VAL_PXI_TRIG6</td><td>The trigger is received on PXI trigger line 6.</td></tr>
<tr><td>NIVNA_VAL_PXI_TRIG7</td><td>The trigger is received on PXI trigger line 7.</td></tr>
</table></td></tr>
<tr><td>triggerEdge</td><td>ViInt32</td><td>Specifies the trigger edge on which the sweep starts.

<table>
<tr><td>NIVNA_VAL_TRIGGER_EDGE_RISING</td><td>The sweep starts on the rising edge of the trigger.</td></tr>
<tr><td>NIVNA_VAL_TRIGGER_EDGE_FALLING</td><td>The sweep starts on</td></tr>
</table></td></tr>
</table>

		the falling edge of the trigger.

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function. The general meaning of the status code is as follows: **Value** / **Meaning** 0 / Success Positive Values / Warnings Negative Values / Errors

niVNA_ConfigureSoftwareEdgeStartTrigger

niVNA_ConfigureSoftwareEdgeStartTrigger

niVNA_ConfigureSoftwareEdgeStartTrigger(ViSession instrumentHandle);

Purpose

Configures the device to wait for a software Start Trigger before starting measurement.

The device waits until you call the niVNA_SendSoftwareEdgeTrigger function to assert the trigger.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.

Return Value

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>returnValue</td><td>ViStatus</td><td>Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred.

To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function.

The general meaning of the status code is as follows:

<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>0</td><td>Success</td></tr>
<tr><td>Positive Values</td><td>Warnings</td></tr>
<tr><td>Negative Values</td><td>Errors</td></tr>
</table>
</td></tr>
</table>

niVNA_DisableStartTrigger

niVNA_DisableStartTrigger

niVNA_DisableStartTrigger(ViSession instrumentHandle);

Purpose

Configures the device to start measuring without waiting for a Start Trigger. This function is necessary only if you want to disable a configured Start Trigger.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function. The general meaning of the status code is as follows: **Value** | **Meaning**

		0 Success Positive Values Warnings Negative Values Errors

Step Trigger

Class/Panel Name	Function Name
Step Trigger	
Configure Digital Edge Step Trigger	niVNA_ConfigureDigitalEdgeStepTrigger
Disable Step Trigger	niVNA_DisableStepTrigger

niVNA_ConfigureDigitalEdgeStepTrigger

niVNA_ConfigureDigitalEdgeStepTrigger

niVNA_ConfigureDigitalEdgeStepTrigger(ViSession instrumentHandle, ViConstString triggerSource, ViInt32 triggerEdge);

Purpose

Configures the device to wait for a digital edge Step Trigger before every individual point measurement within a sweep. The device expects ***N*** Step Triggers per sweep, where ***N*** is the number of points configured for the sweep.

Supported Devices: NI 5632

Parameters

Input

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>instrumentHandle</td><td>ViSession</td><td>Identifies your instrument handle.</td></tr>
<tr><td>triggerSource</td><td>ViConstString</td><td>Specifies the source of the digital edge for the trigger.

<table><tr><td>NIVNA_VAL_PFI0</td><td>The trigger is received on PFI 0.</td></tr><tr><td>NIVNA_VAL_PXI_TRIG0</td><td>The trigger is received on PXI trigger line 0.</td></tr><tr><td>NIVNA_VAL_PXI_TRIG1</td><td>The trigger is received on PXI trigger line 1.</td></tr><tr><td>NIVNA_VAL_PXI_TRIG2</td><td>The trigger is received on PXI trigger line 2.</td></tr><tr><td>NIVNA_VAL_PXI_TRIG3</td><td>The trigger is received on PXI trigger line 3.</td></tr><tr><td>NIVNA_VAL_PXI_TRIG4</td><td>The trigger is received on PXI trigger line 4.</td></tr><tr><td>NIVNA_VAL_PXI_TRIG5</td><td>The trigger is received on PXI trigger line 5.</td></tr><tr><td>NIVNA_VAL_PXI_TRIG6</td><td>The trigger is received on PXI trigger line 6.</td></tr><tr><td>NIVNA_VAL_PXI_TRIG7</td><td>The trigger is received on PXI trigger line 7.</td></tr></table></td></tr>
<tr><td>triggerEdge</td><td>ViInt32</td><td>Specifies the trigger edge on which the sweep starts.

<table><tr><td>NIVNA_VAL_TRIGGER_EDGE_RISING</td><td>The sweep starts on the rising edge of the trigger.</td></tr><tr><td>NIVNA_VAL_TRIGGER_EDGE_FALLING</td><td>The sweep starts on the falling edge of the trigger.</td></tr></table></td></tr>
</table>

Return Value

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>returnValue</td><td>ViStatus</td><td>Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred.

To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function.

The general meaning of the status code is as follows:

<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>0</td><td>Success</td></tr>
<tr><td>Positive Values</td><td>Warnings</td></tr>
<tr><td>Negative Values</td><td>Errors</td></tr>
</table>
</td></tr>
</table>

niVNA_DisableStepTrigger

niVNA_DisableStepTrigger

niVNA_DisableStepTrigger(ViSession instrumentHandle);

Purpose

Configures the device to sweep without waiting for a Step Trigger to take an individual point measurement. This function is necessary only if you want to disable a configured Step Trigger.

Supported Devices: NI 5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function. The general meaning of the status code is as follows: **Value** — **Meaning** 0 — Success Positive Values — Warnings Negative Values — Errors

niVNA_SendSoftwareEdgeTrigger

niVNA_SendSoftwareEdgeTrigger(ViSession instrumentHandle);

Purpose

Asserts a software trigger. You must first use the `niVNA_ConfigureSoftwareEdgeStartTrigger` function to configure the device to use a software trigger.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function. The general meaning of the status code is as follows: **Value** — **Meaning** 0 — Success Positive Values — Warnings Negative Values — Errors

niVNA_ExportSignal

niVNA_ExportSignal(ViSession instrumentHandle, ViInt32 signal, ViConstString signalIdentifier, ViConstString outputTerminal);

Purpose

Routes signals to the specified output terminal.

Supported Devices: NI 5632

Parameters

<table>
<tr><th colspan="3">Input</th></tr>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td>instrumentHandle</td><td>ViSession</td><td>Identifies your instrument handle.</td></tr>
<tr><td>signal</td><td>ViInt32</td><td>Specifies the type of signal to route.

<table><tr><td>NIVNA_VAL_EVENT_READY_FOR_START (0)</td><td>Ready for Start Event</td></tr><tr><td>NIVNA_VAL_EVENT_READY_FOR_STEP (1)</td><td>Ready for Step Event</td></tr><tr><td>NIVNA_VAL_EVENT_END_OF_STEP (2)</td><td>End of Step Event</td></tr><tr><td>NIVNA_VAL_EVENT_END_OF_SWEEP (3)</td><td>End of Sweep Event</td></tr></table></td></tr>
<tr><td>signalIdentifier</td><td>ViConstString</td><td>Specifies a particular instance of a trigger. NI-VNA does not currently support this parameter. The value of this parameter must always be an empty string ("").</td></tr>
<tr><td>outputTerminal</td><td>ViConstString</td><td>Specifies the terminal where the signal will be exported. You can also choose not to export any signal.

<table><tr><td>NIVNA_VAL_DO_NOT_EXPORT_STR ("")</td><td>Do not export signal</td></tr><tr><td>NIVNA_VAL_PFI0 ("PFI0")</td><td>The signal is exported to PFI 0.</td></tr><tr><td>NIVNA_VAL_PXI_TRIG0 ("PXI_Trig0")</td><td>The signal is exported to PXI trigger line 0.</td></tr></table></td></tr>
</table>

		NIVNA_VAL_PXI_TRIG1 ("PXI_Trig1") — The signal is exported to PXI trigger line 1. NIVNA_VAL_PXI_TRIG2 ("PXI_Trig2") — The signal is exported to PXI trigger line 2. NIVNA_VAL_PXI_TRIG3 ("PXI_Trig3") — The signal is exported to PXI trigger line 3. NIVNA_VAL_PXI_TRIG4 ("PXI_Trig4") — The signal is exported to PXI trigger line 4. NIVNA_VAL_PXI_TRIG5 ("PXI_Trig5") — The signal is exported to PXI trigger line 5. NIVNA_VAL_PXI_TRIG6 ("PXI_Trig6") — The signal is exported to PXI trigger line 6. NIVNA_VAL_PXI_TRIG7 ("PXI_Trig7") — The signal is exported to PXI trigger line 7.

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function. The general meaning of the status code is as follows: **Value** — **Meaning** 0 — Success Positive Values — Warnings

		Negative Values | Errors

Set/Get Attribute

Class/Panel Name	Function Name
Set/Get Attribute	
Set Attribute	
Set Attribute ViInt32	niVNA_SetAttributeViInt32
Set Attribute ViInt64	niVNA_SetAttributeViInt64
Set Attribute ViReal64	niVNA_SetAttributeViReal64
Set Attribute ViString	niVNA_SetAttributeViString
Set Attribute ViBoolean	niVNA_SetAttributeViBoolean
Set Attribute ViSession	niVNA_SetAttributeViSession
Get Attribute	
Get Attribute ViInt32	niVNA_GetAttributeViInt32
Get Attribute ViInt64	niVNA_GetAttributeViInt64
Get Attribute ViReal64	niVNA_GetAttributeViReal64
Get Attribute ViString	niVNA_GetAttributeViString
Get Attribute ViBoolean	niVNA_GetAttributeViBoolean
Get Attribute ViSession	niVNA_GetAttributeViSession
Invalidate All Attributes	niVNA_InvalidateAllAttributes

Set Attribute

Class/Panel Name	Function Name
Set Attribute	

Set Attribute ViInt32	niVNA_SetAttributeViInt32
Set Attribute ViInt64	niVNA_SetAttributeViInt64
Set Attribute ViReal64	niVNA_SetAttributeViReal64
Set Attribute ViString	niVNA_SetAttributeViString
Set Attribute ViBoolean	niVNA_SetAttributeViBoolean
Set Attribute ViSession	niVNA_SetAttributeViSession

niVNA_SetAttributeViInt32

niVNA_SetAttributeViInt32

niVNA_SetAttributeViInt32(ViSession instrumentHandle, ViConstString channel, ViAttr attributeID, ViInt32 value);

Purpose

Sets the value of a ViInt32 attribute.

Use this low-level function to set the values of inherent IVI attributes and instrument-specific attributes.

NI-VNA contains high-level functions that set most of the instrument attributes. NI recommends you use the high-level functions as much as possible. High-level functions handle order dependencies and multithread-locking for you.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.
channel	ViConstString	Specifies the channel for which this operation is targeted.

attributeID	ViAttr	Pass the ID of an attribute.
value	ViInt32	Pass the value to which you want to set the attribute.

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function. The general meaning of the status code is as follows: **Value** / **Meaning** 0 / Success Positive Values / Warnings Negative Values / Errors

niVNA_SetAttributeViInt64

niVNA_SetAttributeViInt64

niVNA_SetAttributeViInt64(ViSession instrumentHandle, ViConstString channel, ViAttr attributeID, ViInt64 value);

Purpose

Sets the value of a ViInt64 attribute.

Use this low-level function to set the values of inherent IVI attributes and instrument-specific attributes.

NI-VNA contains high-level functions that set most of the instrument attributes. NI recommends you use the high-level functions as much as possible. High-level functions handle order dependencies and multithread-locking for you.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.
channel	ViConstString	Specifies the channel for which this operation is targeted.
attributeID	ViAttr	Pass the ID of an attribute.
value	ViInt64	Pass the value to which you want to set the attribute.

Return Value

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>returnValue</td><td>ViStatus</td><td>Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred.

To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function.

The general meaning of the status code is as follows:

<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>0</td><td>Success</td></tr>
<tr><td>Positive Values</td><td>Warnings</td></tr>
</table>
</td></tr>
</table>

		Negative Values: Errors

niVNA_SetAttributeViReal64

niVNA_SetAttributeViReal64

niVNA_SetAttributeViReal64(ViSession instrumentHandle, ViConstString channel, ViAttr attributeID, ViReal64 value);

Purpose

Sets the value of a ViReal64 attribute.

Use this low-level function to set the values of inherent IVI attributes and instrument-specific attributes.

NI-VNA contains high-level functions that set most of the instrument attributes. NI recommends you use the high-level functions as much as possible. High-level functions handle order dependencies and multithread-locking for you.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.
channel	ViConstString	Specifies the channel for which this operation is targeted.
attributeID	ViAttr	Pass the ID of an attribute.
value	ViReal64	Pass the value to which you want to set the attribute.

Return Value

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>returnValue</td><td>ViStatus</td><td>Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred.

To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function.

The general meaning of the status code is as follows:

<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>0</td><td>Success</td></tr>
<tr><td>Positive Values</td><td>Warnings</td></tr>
<tr><td>Negative Values</td><td>Errors</td></tr>
</table>
</td></tr>
</table>

niVNA_SetAttributeViString

niVNA_SetAttributeViString

niVNA_SetAttributeViString(ViSession instrumentHandle, ViConstString channel, ViAttr attributeID, ViConstString value);

Purpose

Sets the value of a ViString attribute.

Use this low-level function to set the values of inherent IVI attributes and instrument-specific attributes.

NI-VNA contains high-level functions that set most of the instrument attributes. NI

recommends you use the high-level functions as much as possible. High-level functions handle order dependencies and multithread-locking for you.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.
channel	ViConstString	Specifies the channel for which this operation is targeted.
attributeID	ViAttr	Pass the ID of an attribute.
value	ViConstString	Pass the value to which you want to set the attribute.

Return Value

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>returnValue</td><td>ViStatus</td><td>Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred.

To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function.

The general meaning of the status code is as follows:

<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>0</td><td>Success</td></tr>
<tr><td>Positive Values</td><td>Warnings</td></tr>
<tr><td>Negative Values</td><td>Errors</td></tr>
</table>
</td></tr>
</table>

niVNA_SetAttributeViBoolean

niVNA_SetAttributeViBoolean

niVNA_SetAttributeViBoolean(ViSession instrumentHandle, ViConstString channel, ViAttr attributeID, ViBoolean value);

Purpose

Sets the value of a ViBoolean attribute.

Use this low-level function to set the values of inherent IVI attributes and instrument-specific attributes.

NI-VNA contains high-level functions that set most of the instrument attributes. NI recommends you use the high-level functions as much as possible. High-level functions handle order dependencies and multithread-locking for you.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.
channel	ViConstString	Specifies the channel for which this operation is targeted.
attributeID	ViAttr	Pass the ID of an attribute.
value	ViBoolean	Pass the value to which you want to set the attribute.

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code

		from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function. The general meaning of the status code is as follows: **Value** / **Meaning** 0 / Success Positive Values / Warnings Negative Values / Errors

niVNA_SetAttributeViSession

niVNA_SetAttributeViSession

niVNA_SetAttributeViSession(ViSession instrumentHandle, ViConstString channel, ViAttr attributeID, ViSession value);

Purpose

Sets the value of a ViSession attribute.

Use this low-level function to set the values of inherent IVI attributes and instrument-specific attributes.

NI-VNA contains high-level functions that set most of the instrument attributes. NI recommends you use the high-level functions as much as possible. High-level functions handle order dependencies and multithread-locking for you.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.
channel	ViConstString	Specifies the channel for which this operation is targeted.
attributeID	ViAttr	Pass the ID of an attribute.
value	ViSession	Pass the value to which you want to set the attribute.

Return Value

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>returnValue</td><td>ViStatus</td><td>Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred.

To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function.

The general meaning of the status code is as follows:

<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>0</td><td>Success</td></tr>
<tr><td>Positive Values</td><td>Warnings</td></tr>
<tr><td>Negative Values</td><td>Errors</td></tr>
</table>
</td></tr>
</table>

Get Attribute

Class/Panel Name	Function Name
Get Attribute	

Get Attribute ViInt32	niVNA_GetAttributeViInt32
Get Attribute ViInt64	niVNA_GetAttributeViInt64
Get Attribute ViReal64	niVNA_GetAttributeViReal64
Get Attribute ViString	niVNA_GetAttributeViString
Get Attribute ViBoolean	niVNA_GetAttributeViBoolean
Get Attribute ViSession	niVNA_GetAttributeViSession

niVNA_GetAttributeViInt32

niVNA_GetAttributeViInt32

niVNA_GetAttributeViInt32(ViSession instrumentHandle, ViConstString channel, ViAttr attributeID, ViInt32 *value);

Purpose

Queries the value of a ViInt32 attribute.

You can use this low-level function to get the values of inherent IVI attributes, and instrument-specific attributes.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.
channel	ViConstString	Specifies the channel for which this operation is targeted.
attributeID	ViAttr	Pass the ID of an attribute.
Output		

Name	Type	Description
value	ViInt32*	Returns the current value of the attribute. Pass the address of a ViInt32 variable.

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function. The general meaning of the status code is as follows: **Value** / **Meaning** 0 / Success Positive Values / Warnings Negative Values / Errors

niVNA_GetAttributeViInt64

niVNA_GetAttributeViInt64

niVNA_GetAttributeViInt64(ViSession instrumentHandle, ViConstString channel, ViAttr attributeID, ViInt64 *value);

Purpose

Queries the value of a ViInt64 attribute.

You can use this low-level function to get the values of inherent IVI attributes, and instrument-specific attributes.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.
channel	ViConstString	Specifies the channel for which this operation is targeted.
attributeID	ViAttr	Pass the ID of an attribute.
Output		
Name	Type	Description
value	ViInt64*	Returns the current value of the attribute. Pass the address of a ViInt64 variable.

Return Value

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>returnValue</td><td>ViStatus</td><td>Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred.

To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function.

The general meaning of the status code is as follows:

<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>0</td><td>Success</td></tr>
<tr><td>Positive Values</td><td>Warnings</td></tr>
</table>
</td></tr>
</table>

		Negative Values | Errors

niVNA_GetAttributeViReal64

niVNA_GetAttributeViReal64

niVNA_GetAttributeViReal64(ViSession instrumentHandle, ViConstString channel, ViAttr attributeID, ViReal64 *value);

Purpose

Queries the value of a ViReal64 attribute.

You can use this low-level function to get the values of inherent IVI attributes, and instrument-specific attributes.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.
channel	ViConstString	Specifies the channel for which this operation is targeted.
attributeID	ViAttr	Pass the ID of an attribute.
Output		
Name	Type	Description
value	ViReal64*	Returns the current value of the attribute. Pass the address of a ViReal64 variable.

Return Value

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>returnValue</td><td>ViStatus</td><td>Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred.

To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function.

The general meaning of the status code is as follows:

<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>0</td><td>Success</td></tr>
<tr><td>Positive Values</td><td>Warnings</td></tr>
<tr><td>Negative Values</td><td>Errors</td></tr>
</table>
</td></tr>
</table>

niVNA_GetAttributeViString

niVNA_GetAttributeViString

niVNA_GetAttributeViString(ViSession instrumentHandle, ViConstString channel, ViAttr attributeID, ViInt32 bufferSize, ViChar value[]);

Purpose

Queries the value of a ViString attribute.

You can use this low-level function to get the values of inherent IVI attributes, and instrument-specific attributes.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.
channel	ViConstString	Specifies the channel for which this operation is targeted.
attributeID	ViAttr	Pass the ID of an attribute.
bufferSize	ViInt32	Pass the number of bytes in the ViChar buffer you specify for the attribute value parameter. If you pass a negative number, the function copies the value to the buffer regardless of the number of bytes in the value. If you pass 0, you can pass VI_NULL for the attribute value buffer parameter.
Output		
Name	Type	Description
value	ViChar[]	Returns the current value of the attribute. Pass the address of a ViString variable.

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function. The general meaning of the status code is as follows: **Value** | **Meaning**

		0 Success Positive Values Warnings Negative Values Errors

niVNA_GetAttributeViBoolean

niVNA_GetAttributeViBoolean

niVNA_GetAttributeViBoolean(ViSession instrumentHandle, ViConstString channel, ViAttr attributeID, ViBoolean *value);

Purpose

Queries the value of a ViBoolean attribute.

You can use this low-level function to get the values of inherent IVI attributes, and instrument-specific attributes.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.
channel	ViConstString	Specifies the channel for which this operation is targeted.
attributeID	ViAttr	Pass the ID of an attribute.
Output		
Name	Type	Description

value	ViBoolean*	Returns the current value of the attribute. Pass the address of a ViBoolean variable.

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function. The general meaning of the status code is as follows: **Value** / **Meaning** 0 / Success Positive Values / Warnings Negative Values / Errors

niVNA_GetAttributeViSession

niVNA_GetAttributeViSession

niVNA_GetAttributeViSession(ViSession instrumentHandle, ViConstString channel, ViAttr attributeID, ViSession *value);

Purpose

Queries the value of a ViSession attribute.

You can use this low-level function to get the values of inherent IVI attributes, and

instrument-specific attributes.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.
channel	ViConstString	Specifies the channel for which this operation is targeted.
attributeID	ViAttr	Pass the ID of an attribute.
Output		
Name	Type	Description
value	ViSession*	Returns the current value of the attribute. Pass the address of a ViSession variable.

Return Value

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>returnValue</td><td>ViStatus</td><td>Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred.

To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function.

The general meaning of the status code is as follows:

<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>0</td><td>Success</td></tr>
<tr><td>Positive Values</td><td>Warnings</td></tr>
</table>
</td></tr>
</table>

		Negative Values: Errors

niVNA_InvalidateAllAttributes

niVNA_InvalidateAllAttributes(ViSession instrumentHandle, ViConstString channel);

Purpose

Sets the cached attribute value to invalid. If you query the attribute value after you call this function, NI-VNA queries the value from the device.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.
channel	ViConstString	Specifies the channel for which this operation is targeted.

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function. The general meaning of the status code is as follows:

		Value / **Meaning** 0 / Success Positive Values / Warnings Negative Values / Errors

Connector Configuration

Class/Panel Name	**Function Name**
Connector Configuration	
Configure Connector	niVNA_ConfigureConnector

niVNA_ConfigureConnector

niVNA_ConfigureConnector(ViSession instrumentHandle, ViConstString port, ViInt32 connectorGender, ViInt32 connectorType);

Purpose

Configures the connector on the manual calibration standard. Selecting correct values for the parameters in this function ensures that calibration quality is optimized because the calibration characterization (resistance, capacitance, or inductance) is as accurate as possible.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.

<table>
<tr><td>port</td><td>ViConstString</td><td>Specifies the port to which to apply the settings. Valid port specifiers are "<code>port1</code>" or "<code>port2</code>".</td></tr>
<tr><td>connectorGender</td><td>ViInt32</td><td>Specifies the connector gender.

<table><tr><td>NIVNA_VAL_CONNECTOR_GENDER_MALE</td><td>The port uses a male connector.</td></tr><tr><td>NIVNA_VAL_CONNECTOR_GENDER_FEMALE</td><td>The port uses a female connector.</td></tr></table></td></tr>
<tr><td>connectorType</td><td>ViInt32</td><td>Specifies the connector type.

<table><tr><td>NIVNA_VAL_CONNECTOR_TYPE_N</td><td>The port uses an N-type connector.</td></tr><tr><td>NIVNA_VAL_CONNECTOR_TYPE_K</td><td>The port uses a K-type connector.</td></tr><tr><td>NIVNA_VAL_CONNECTOR_TYPE_GPC</td><td>The port uses a GPC 3.5 connector.</td></tr><tr><td>NIVNA_VAL_CONNECTOR_TYPE_SMA</td><td>The port uses an SMA connector.</td></tr></table></td></tr>
</table>

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function. The general meaning of the status code is as follows:

		Value / **Meaning** 0 / Success Positive Values / Warnings Negative Values / Errors

niVNA_PerformAutoReferencePlaneExtension

niVNA_PerformAutoReferencePlaneExtension(ViSession instrumentHandle, ViConstString port);

Purpose

Automatically determines the reference plane extension length and the reference plane extension time by measuring the connected manual short/open standards. Ensure that the manual short or open standard is connected to the RF cable connected to your device, and perform at least one frequency sweep operation before you call this function.

The corresponding attributes for these parameters are set to the value that is measured by this function.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.
port	ViConstString	Specifies the port to which to apply the settings. Valid port specifiers are "`port1`" or "`port2`".

Return Value

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>returnValue</td><td>ViStatus</td><td>Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred.

To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function.

The general meaning of the status code is as follows:

<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>0</td><td>Success</td></tr>
<tr><td>Positive Values</td><td>Warnings</td></tr>
<tr><td>Negative Values</td><td>Errors</td></tr>
</table>
</td></tr>
</table>

Measurement

Class/Panel Name	Function Name
Measurement	
Sweep	
Initiate	niVNA_Initiate
Commit	niVNA_Commit
Abort	niVNA_Abort
Is Sweep Done	niVNA_IsSweepDone
Wait For Sweep Done	niVNA_WaitForSweepDone
Fetch Sweep Measurement F64	niVNA_FetchSweepMeasurementF64
Fetch Sweep Measurement Complex	niVNA_FetchSweepMeasurementComplex

Fetch Sweep Measurement Into File	niVNA_FetchSweepMeasurementIntoFile
Fetch Time Domain Transform F64	niVNA_FetchTimeDomainTransformF64
Fetch Time Domain Gate Shape F64	niVNA_FetchTimeDomainGateShapeF64
Fetch Receiver Measurement F64	niVNA_FetchReceiverMeasurementF64
Fetch Receiver Measurement Complex	niVNA_FetchReceiverMeasurementComplex
Calibration	
Init User Calibration	niVNA_InitUserCal
Perform Calibration Action	niVNA_PerformCalibrationAction
Perform TRL Calibration Action	niVNA_PerformTRLCalibrationAction
Apply Adapter Removal Calibration	niVNA_ApplyAdapterRemovalCalibration
Get Calibration State	niVNA_GetCalibrationState
Is Calibration Action Done	niVNA_IsCalibrationActionDone
Wait For Calibration Action Done	niVNA_WaitForCalibrationActionDone
Abort Calibration Action	niVNA_AbortCalibrationAction
Close User Calibration	niVNA_CloseUserCal
Load Calibration And Configuration Data From File	niVNA_LoadCalibrationAndConfigurationDataFromFil
Save Calibration And Configuration Data To File	niVNA_SaveCalibrationAndConfigurationDataToFile
Load User Defined Manual Calibration Kit Data From File	niVNA_LoadUserDefinedManualCalibrationKitDataFro
AutoCal	
Query AutoCal Box Info	niVNA_QueryAutoCalBoxInfo
Load AutoCal Characterization Data From File	niVNA_LoadAutoCalCharacterizationDataFromFile

Sweep

Class/Panel Name	Function Name
Sweep	
Initiate	niVNA_Initiate
Commit	niVNA_Commit

Abort	niVNA_Abort
Is Sweep Done	niVNA_IsSweepDone
Wait For Sweep Done	niVNA_WaitForSweepDone

niVNA_Initiate

niVNA_Initiate(ViSession instrumentHandle);

Purpose

Starts the measurement. Use this function in conjunction with the `niVNA_WaitForSweepDone` function to wait for the sweep measurement to complete and the niVNA fetch functions to retrieve acquired data. If this function runs successfully, NI-VNA transitions to the Running state.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function. The general meaning of the status code is as follows:

		Value / **Meaning** 0 / Success Positive Values / Warnings Negative Values / Errors

niVNA_Commit

niVNA_Commit(ViSession instrumentHandle);

Purpose

Completes any remaining attribute validation and commits the settings to hardware. This function commits all routes, and this action may cause glitching PXI or PFI lines. This function also begins locking to a reference frequency if you have configured one using the NIVNA_ATTR_REFERENCE_FREQUENCY_SOURCE attribute. If the commit action is successful, NI-VNA transitions to the Committed state.

Calling this function before you call the niVNA_Initiate function is optional, but it has the following benefits:

- Routes are committed, so signals are imported.
- Any Reference Clock and external clock circuits are phase-locked.
- Any subsequent `niVNA_Initiate` function calls can run faster because the device is already configured.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description

instrumentHandle	ViSession	Identifies your instrument handle.

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function. The general meaning of the status code is as follows: **Value** / **Meaning** 0 / Success Positive Values / Warnings Negative Values / Errors

niVNA_Abort

niVNA_Abort(ViSession instrumentHandle);

Purpose

Stops a measurement previously started with the niVNA_Initiate function. Calling this function is optional, unless you want to stop a measurement before it is complete. If this function runs successfully, NI-VNA transitions to the Done state (if the sweep is complete) or the Committed state (if this call stops a sweep in progress).

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function. The general meaning of the status code is as follows: **Value** — **Meaning** 0 — Success Positive Values — Warnings Negative Values — Errors

niVNA_IsSweepDone

niVNA_IsSweepDone(ViSession instrumentHandle, ViBoolean *isDone);

Purpose

Checks the status of the measurement. Use this function to check for any errors that may occur during acquisition or to check whether the device is done measuring the

sweep. If the sweep is complete, NI-VNA transitions to the Done state.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.
Output		
Name	Type	Description
isDone	ViBoolean*	Returns whether the operation has completed.

Return Value

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>returnValue</td><td>ViStatus</td><td>Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred.

To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function.

The general meaning of the status code is as follows:

<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>0</td><td>Success</td></tr>
<tr><td>Positive Values</td><td>Warnings</td></tr>
<tr><td>Negative Values</td><td>Errors</td></tr>
</table>
</td></tr>
</table>

niVNA_WaitForSweepDone

niVNA_WaitForSweepDone(ViSession instrumentHandle, ViReal64 timeout);

Purpose

Waits for the sweep measurement to complete. If the measurement does not complete within the specified **timeout**, this function returns a timeout error. If this function runs successfully, NI-VNA transitions to the Done state.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.
timeout	ViReal64	Specifies the time to wait for the function to complete operation before returning an error. A negative value specifies that NI-VNA waits indefinitely. This value is expressed in seconds (s).

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function. The general meaning of the status code is as follows:

		Value / **Meaning** 0 / Success Positive Values / Warnings Negative Values / Errors

niVNA_FetchSweepMeasurementF64

niVNA_FetchSweepMeasurementF64(ViSession instrumentHandle, ViConstString channel, ViInt32 measurement, ViInt32 format, ViReal64 timeout, ViReal64 data[], ViReal64 scalePoints[], ViInt32 numberOfElements, ViInt32 *readCount);

Purpose

Fetches floating 64-bit data and scale points from the last measurement.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.
channel	ViConstString	Specifies the channel for which this operation is targeted.
measurement	ViInt32	Specifies the measurement for which to fetch data. The NI 5630 supports only the NIVNA_VAL_MEASUREMENT_S11 and NIVNA_VAL_MEASUREMENT_S21 values. For more information about S-parameters, refer to the Vector Network Analyzer Measurements topic.

		<table><tr><td>NIVNA_VAL_MEASUREMENT_S11</td><td>NI-VNA fetches S11 measurement data.</td></tr><tr><td>NIVNA_VAL_MEASUREMENT_S21</td><td>NI-VNA fetches S21 measurement data.</td></tr><tr><td>NIVNA_VAL_MEASUREMENT_S12</td><td>NI-VNA fetches S12 measurement data.</td></tr><tr><td>NIVNA_VAL_MEASUREMENT_S22</td><td>NI-VNA fetches S22 measurement data.</td></tr></table>
format	ViInt32	Specifies the format in which to fetch the data. <table><tr><td>NIVNA_VAL_FORMAT_LOGMAG_DB</td><td>NI-VNA fetches the data in log magnitude format. NI-VNA log magnitude data is presented in decibels (20 log (data)).</td></tr><tr><td>NIVNA_VAL_FORMAT_LINMAG</td><td>NI-VNA fetches the data in linear magnitude format. NI-VNA linear magnitude data is presented as a linear ratio.</td></tr><tr><td>NIVNA_VAL_FORMAT_PHASE_DEG</td><td>NI-VNA fetches the data in phase format. NI-VNA phase datais presented in degrees of phase shift of the response (port 1 or 2) with respect to the stimulus (port 1). Phase data is presented in a wrapped format (+/-180 degree segments).</td></tr><tr><td>NIVNA_VAL_FORMAT_VSWR</td><td>NI-VNA fetches the data</td></tr></table>

		in voltage standing wave ratio (VSWR) format. NI-VNA magnitude data is presented as a linear ratio of (maximum RF voltage/minimum RF voltage).
		NIVNA_VAL_FORMAT_GROUPDELAY: NI-VNA fetches the data in group delay format. Group delay expresses the rate of change of phase with respect to frequency (dp/df) over a specified range of frequency points, known as a window, expressed as time delay. The units are seconds.
timeout	ViReal64	Specifies the time to wait for the function to complete operation before returning an error. A negative value specifies that NI-VNA waits indefinitely. This value is expressed in seconds (s).
numberOfElements	ViInt32	Specifies the number of elements to fetch.
Output		
Name	Type	Description
data	ViReal64[]	Returns the gate shape data.
scalePoints	ViReal64[]	Returns the sweep points that were included in the measurement. For a frequency sweep operation, this parameter returns the frequencies at which **data** was taken. For a time-domain operation, this parameter returns the time or distance. For a power sweep operation, this parameter returns the power values at which the **data** was taken.
readCount	ViInt32*	Returns the number of data points.

Return Value

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>returnValue</td><td>ViStatus</td><td>Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred.

To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function.

The general meaning of the status code is as follows:

<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>0</td><td>Success</td></tr>
<tr><td>Positive Values</td><td>Warnings</td></tr>
<tr><td>Negative Values</td><td>Errors</td></tr>
</table>
</td></tr>
</table>

niVNA_FetchSweepMeasurementComplex

niVNA_FetchSweepMeasurementComplex(ViSession instrumentHandle, ViConstString channel, ViInt32 measurement, ViInt32 format, ViReal64 timeout, NIComplexNumber data[], ViReal64 scalePoints[], ViInt32 numberOfElements, ViInt32 *readCount);

Purpose

Fetches complex data and scale points from the last measurement.

Supported Devices: NI 5630/5632

Parameters

Input

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>instrumentHandle</td><td>ViSession</td><td>Identifies your instrument handle.</td></tr>
<tr><td>channel</td><td>ViConstString</td><td>Specifies the channel for which this operation is targeted.</td></tr>
<tr><td>measurement</td><td>ViInt32</td><td>Specifies the measurement for which to fetch data. The NI 5630 supports only the NIVNA_VAL_MEASUREMENT_S11 and NIVNA_VAL_MEASUREMENT_S21 values. For more information about S-parameters, refer to the Vector Network Analyzer Measurements topic.

<table><tr><td>NIVNA_VAL_MEASUREMENT_S11</td><td>NI-VNA fetches S11 measurement data.</td></tr><tr><td>NIVNA_VAL_MEASUREMENT_S21</td><td>NI-VNA fetches S21 measurement data.</td></tr><tr><td>NIVNA_VAL_MEASUREMENT_S12</td><td>NI-VNA fetches S12 measurement data.</td></tr><tr><td>NIVNA_VAL_MEASUREMENT_S22</td><td>NI-VNA fetches S22 measurement data.</td></tr></table></td></tr>
<tr><td>format</td><td>ViInt32</td><td>Specifies the format in which to fetch the data.

<table><tr><td>NIVNA_VAL_FORMAT_IMPEDANCE</td><td>NI-VNA fetches the data in impedance format. The impedance is expressed in ohms. NI-VNA magnitude and phase data are</td></tr></table></td></tr>
</table>

		presented as a complex real and imaginary Cartesian number pair in ohms.
		NIVNA_VAL_FORMAT_COEFFICIENT — NI-VNA fetches the data as the reflection (S11 and S22) or transmission (for S21 and S12) coefficient (gamma). The coefficient is related to impedance according to the formula ***Impedance = Zo * (1 + gamma)/(1 - gamma)***, where ***Zo*** is assumed to be 50 ohms.
timeout	ViReal64	Specifies the time to wait for the function to complete operation before returning an error. A negative value specifies that NI-VNA waits indefinitely. This value is expressed in seconds (s).
numberOfElements	ViInt32	Specifies the number of elements to fetch.
Output		
Name	Type	Description
data	NIComplexNumber[]	Returns the measurement data.
scalePoints	ViReal64[]	Returns the sweep points that were included in the measurement. For a frequency sweep operation, this parameter returns the frequencies at which **data** was

		taken. For a time-domain operation, this parameter returns the time or distance. For a power sweep operation, this parameter returns the power values at which the **data** was taken.
readCount	ViInt32*	Returns the number of data points.

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function. The general meaning of the status code is as follows: **Value** / **Meaning** 0 / Success Positive Values / Warnings Negative Values / Errors

niVNA_FetchSweepMeasurementIntoFile

niVNA_FetchSweepMeasurementIntoFile(ViSession instrumentHandle, ViConstString channel, ViConstString filePath, ViInt32 fileType, ViBoolean overwrite, ViReal64 timeout);

Purpose

Fetches the data from the device directly into a one- or two-port S-parameter file from

the latest measurement.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.
channel	ViConstString	Specifies the channel for which this operation is targeted.
filePath	ViConstString	Specifies the absolute file path into which to save the data.
fileType	ViInt32	specifies the type of file into which to save the data. NIVNA_VAL_DATA_FILE_TYPE_S1P: The data is saved in a one-port S-parameter file. NIVNA_VAL_DATA_FILE_TYPE_S2P: The data is saved in a two-port S-parameter file.
overwrite	ViBoolean	Specifies whether to overwrite the existing file, if one exists.
timeout	ViReal64	Specifies the time to wait for the function to complete operation before returning an error. A negative value specifies that NI-VNA waits indefinitely. This value is expressed in seconds (s).

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function. The general meaning of the status code is as follows:

		Value / **Meaning** 0 / Success Positive Values / Warnings Negative Values / Errors

niVNA_FetchTimeDomainTransformF64

niVNA_FetchTimeDomainTransformF64(ViSession instrumentHandle, ViConstString channel, ViInt32 measurement, ViInt32 format, ViReal64 timeout, ViReal64 data[], ViReal64 scalePoints[], ViInt32 numberOfElements, ViInt32 *readCount);

Purpose

Fetches floating 64-bit data and fetches scale points from the time-domain analysis transform.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.
channel	ViConstString	Specifies the channel for which this operation is targeted.
measurement	ViInt32	Specifies the measurement for which to fetch data. The NI 5630 supports only the NIVNA_VAL_MEASUREMENT_S11 and NIVNA_VAL_MEASUREMENT_S21 values. For more information about S-parameters, refer to the Vector Network Analyzer Measurements topic.

		<table><tr><td>NIVNA_VAL_MEASUREMENT_S11</td><td>NI-VNA fetches S11 measurement data.</td></tr><tr><td>NIVNA_VAL_MEASUREMENT_S21</td><td>NI-VNA fetches S21 measurement data.</td></tr><tr><td>NIVNA_VAL_MEASUREMENT_S12</td><td>NI-VNA fetches S12 measurement data.</td></tr><tr><td>NIVNA_VAL_MEASUREMENT_S22</td><td>NI-VNA fetches S22 measurement data.</td></tr></table>
format	ViInt32	Specifies the format in which to fetch the data. <table><tr><td>NIVNA_VAL_FORMAT_LOGMAG_DB</td><td>NI-VNA fetches the data in log magnitude format. NI-VNA log magnitude data is presented in decibels (20 log (data)).</td></tr><tr><td>NIVNA_VAL_FORMAT_LINMAG</td><td>NI-VNA fetches the data in linear magnitude format. NI-VNA linear magnitude data is presented as a linear ratio.</td></tr><tr><td>NIVNA_VAL_FORMAT_PHASE_DEG</td><td>NI-VNA fetches the data in phase format. NI-VNA phase data is presented in degrees of phase shift of the response (port 1 or 2) with respect to the stimulus (port 1). Phase data is presented in a wrapped format (+/-180 degree segments).</td></tr><tr><td>NIVNA_VAL_FORMAT_VSWR</td><td>NI-VNA fetches the data in voltage standing wave</td></tr></table>

		<table><tr><td></td><td>ratio (VSWR) format. NI-VNA magnitude data is presented as a linear ratio of (maximum RF voltage/minimum RF voltage).</td></tr><tr><td>NIVNA_VAL_FORMAT_REAL</td><td>NI-VNA fetches the data in real format. NI-VNA real data is presented as the real element of the complex data. The data is unitless.</td></tr></table>
timeout	ViReal64	Specifies the time to wait for the function to complete operation before returning an error. A negative value specifies that NI-VNA waits indefinitely. This value is expressed in seconds (s).
numberOfElements	ViInt32	Specifies the number of elements to fetch.
Output		
Name	Type	Description
data	ViReal64[]	Returns the gate shape data.
scalePoints	ViReal64[]	Returns the sweep points that were included in the measurement. For a frequency sweep operation, this parameter returns the frequencies at which **data** was taken. For a time-domain operation, this parameter returns the time or distance. For a power sweep operation, this parameter returns the power values at which the **data** was taken.
readCount	ViInt32*	Returns the number of data points.

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred.

		To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function. The general meaning of the status code is as follows: **Value** / **Meaning** 0 / Success Positive Values / Warnings Negative Values / Errors

niVNA_FetchTimeDomainGateShapeF64

niVNA_FetchTimeDomainGateShapeF64(ViSession instrumentHandle, ViConstString channel, ViInt32 measurement, ViInt32 format, ViReal64 timeout, ViReal64 data[], ViReal64 scalePoints[], ViInt32 numberOfElements, ViInt32 *readCount);

Purpose

Fetches floating 64-bit data and fetches scale points from the time-domain analysis gate shape.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.
channel	ViConstString	Specifies the channel for which this operation is targeted.
measurement	ViInt32	Specifies the measurement for which to fetch data. The NI

		5630 supports only the NIVNA_VAL_MEASUREMENT_S11 and NIVNA_VAL_MEASUREMENT_S21 values. For more information about S-parameters, refer to the Vector Network Analyzer Measurements topic. NIVNA_VAL_MEASUREMENT_S11 — NI-VNA fetches S11 measurement data. NIVNA_VAL_MEASUREMENT_S21 — NI-VNA fetches S21 measurement data. NIVNA_VAL_MEASUREMENT_S12 — NI-VNA fetches S12 measurement data. NIVNA_VAL_MEASUREMENT_S22 — NI-VNA fetches S22 measurement data.
format	ViInt32	Specifies the format in which to fetch the data. NIVNA_VAL_FORMAT_LOGMAG_DB — NI-VNA fetches the data in log magnitude format. NI-VNA log magnitude data is presented in decibels (20 log (data)). NIVNA_VAL_FORMAT_LINMAG — NI-VNA fetches the data in linear magnitude format. NI-VNA linear magnitude data is presented as a linear ratio. NIVNA_VAL_FORMAT_PHASE_DEG — NI-VNA fetches the data in phase format. NI-VNA phase data is presented in degrees of phase shift of the response (port 1 or 2) with respect to the stimulus (port 1). Phase data is presented in a

		<table><tr><td></td><td>wrapped format (+/-180 degree segments).</td></tr><tr><td>NIVNA_VAL_FORMAT_VSWR</td><td>NI-VNA fetches the data in voltage standing wave ratio (VSWR) format. NI-VNA magnitude data is presented as a linear ratio of (maximum RF voltage/minimum RF voltage).</td></tr><tr><td>NIVNA_VAL_FORMAT_REAL</td><td>NI-VNA fetches the data in real format. NI-VNA real data is presented as the real element of the complex data. The data is unitless.</td></tr></table>
timeout	ViReal64	Specifies the time to wait for the function to complete operation before returning an error. A negative value specifies that NI-VNA waits indefinitely. This value is expressed in seconds (s).
numberOfElements	ViInt32	Specifies the number of elements to fetch.
Output		
Name	Type	Description
data	ViReal64[]	Returns the gate shape data.
scalePoints	ViReal64[]	Returns the sweep points that were included in the measurement. For a frequency sweep operation, this parameter returns the frequencies at which **data** was taken. For a time-domain operation, this parameter returns the time or distance. For a power sweep operation, this parameter returns the power values at which the **data** was taken.
readCount	ViInt32*	Returns the number of data points.

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function. The general meaning of the status code is as follows: **Value** / **Meaning** 0 / Success Positive Values / Warnings Negative Values / Errors

niVNA_FetchReceiverMeasurementF64

niVNA_FetchReceiverMeasurementF64(ViSession instrumentHandle, ViConstString channel, ViConstString receiverPort, ViInt32 receiver, ViInt32 format, ViConstString drivingPort, ViReal64 timeout, ViReal64 data[], ViReal64 scalePoints[], ViInt32 numberOfElements, ViInt32 *readCount);

Purpose

Fetches the receiver measurement and scale points in floating 64-bit data from the last measurement.

Supported Devices: NI 5632

Parameters

<table>
<tr><th colspan="3">Input</th></tr>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td>instrumentHandle</td><td>ViSession</td><td>Identifies your instrument handle.</td></tr>
<tr><td>channel</td><td>ViConstString</td><td>Specifies the channel for which this operation is targeted.</td></tr>
<tr><td>receiverPort</td><td>ViConstString</td><td>Specifies the port from which the measurement is performed. Valid specifiers are "<code>port1</code>" or "<code>port2</code>".</td></tr>
<tr><td>receiver</td><td>ViInt32</td><td>Specifies the receiver on which the measurement is performed. For more information about test and reference receivers, refer to the Test and Reference Receiver Calibration topic.<table><tr><td>NIVNA_VAL_RECEIVER_TEST</td><td>NI-VNA fetches the raw receiver measurements from the test receiver.</td></tr><tr><td>NIVNA_VAL_RECEIVER_REFERENCE</td><td>NI-VNA fetches the raw receiver measurements from the reference receiver.</td></tr></table></td></tr>
<tr><td>format</td><td>ViInt32</td><td>Specifies the format in which to fetch the data.<table><tr><td>NIVNA_VAL_FORMAT_LOGMAG_DB</td><td>NI-VNA fetches the data in log magnitude format. NI-VNA log magnitude data is presented in decibels (20 log (data)).</td></tr><tr><td>NIVNA_VAL_FORMAT_LINMAG</td><td>NI-VNA fetches the data in linear magnitude format. NI-VNA linear magnitude data is presented as a linear</td></tr></table></td></tr>
</table>

		<table><tr><td></td><td>ratio.</td></tr><tr><td>NIVNA_VAL_FORMAT_PHASE_DEG</td><td>NI-VNA fetches the data in phase format. NI-VNA phase datais presented in degrees of phase shift of the response (port 1 or 2) with respect to the stimulus (port 1). Phase data is presented in a wrapped format (+/-180 degree segments).</td></tr><tr><td>NIVNA_VAL_FORMAT_VSWR</td><td>NI-VNA fetches the data in voltage standing wave ratio (VSWR) format. NI-VNA magnitude data is presented as a linear ratio of (maximum RF voltage/minimum RF voltage).</td></tr><tr><td>NIVNA_VAL_FORMAT_GROUPDELAY</td><td>NI-VNA fetches the data in group delay format. Group delay expresses the rate of change of phase with respect to frequency (dp/df) over a specified range of frequency points, known as a window, expressed as time delay. The units are seconds.</td></tr></table>
drivingPort	ViConstString	Specifies the port from which the power is sourced. Valid specifiers are "`port1`" or "`port2`".
timeout	ViReal64	Specifies the time to wait for the function to complete operation before returning an error. A negative value specifies that NI-VNA waits indefinitely. This value is expressed in

		seconds (s).
numberOfElements	ViInt32	Specifies the number of elements to fetch.
Output		
Name	Type	Description
data	ViReal64[]	Returns the gate shape data.
scalePoints	ViReal64[]	Returns the sweep points that were included in the measurement. For a frequency sweep operation, this parameter returns the frequencies at which **data** was taken. For a time-domain operation, this parameter returns the time or distance. For a power sweep operation, this parameter returns the power values at which the **data** was taken.
readCount	ViInt32*	Returns the number of data points.

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function. The general meaning of the status code is as follows: **Value** / **Meaning** 0 / Success Positive Values / Warnings Negative Values / Errors

niVNA_FetchReceiverMeasurementComplex

niVNA_FetchReceiverMeasurementComplex(ViSession instrumentHandle, ViConstString channel, ViConstString receiverPort, ViInt32 receiver, ViInt32 format, ViConstString drivingPort, ViReal64 timeout, NIComplexNumber data[], ViReal64 scalePoints[], ViInt32 numberOfElements, ViInt32 *readCount);

Purpose

Fetches the receiver measurement in complex data, and fetches scale points from the last sweep measurement.

Supported Devices: NI 5632

Parameters

<table>
<tr><td colspan="3">Input</td></tr>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td>instrumentHandle</td><td>ViSession</td><td>Identifies your instrument handle.</td></tr>
<tr><td>channel</td><td>ViConstString</td><td>Specifies the channel for which this operation is targeted.</td></tr>
<tr><td>receiverPort</td><td>ViConstString</td><td>Specifies the port from which the measurement is performed. Valid specifiers are "<code>port1</code>" or "<code>port2</code>".</td></tr>
<tr><td>receiver</td><td>ViInt32</td><td>Specifies the receiver on which the measurement is performed. For more information about test and reference receivers, refer to the Test and Reference Receiver Calibration topic.<table><tr><td>NIVNA_VAL_RECEIVER_TEST</td><td>NI-VNA fetches the raw receiver measurements from the test receiver.</td></tr><tr><td>NIVNA_VAL_RECEIVER_REFERENCE</td><td>NI-VNA fetches</td></tr></table></td></tr>
</table>

		the raw receiver measurements from the reference receiver.
format	ViInt32	Specifies the format in which to fetch the data.
		NIVNA_VAL_FORMAT_IMPEDANCE: NI-VNA fetches the data in impedance format. The impedance is expressed in ohms. NI-VNA magnitude and phase data are presented as a complex real and imaginary Cartesian number pair in ohms.
		NIVNA_VAL_FORMAT_COEFFICIENT: NI-VNA fetches the data as the reflection (S11 and S22) or transmission (for S21 and S12) coefficient (gamma). The coefficient is related to impedance according to the formula ***Impedance =***

		Zo * (1 + gamma)/(1 - gamma), where ***Zo*** is assumed to be 50 ohms.
drivingPort	ViConstString	Specifies the port from which the power is sourced. Valid specifiers are "`port1`" or "`port2`".
timeout	ViReal64	Specifies the time to wait for the function to complete operation before returning an error. A negative value specifies that NI-VNA waits indefinitely. This value is expressed in seconds (s).
numberOfElements	ViInt32	Specifies the number of elements to fetch.
Output		
Name	Type	Description
data	NIComplexNumber[]	Returns the measurement data.
scalePoints	ViReal64[]	Returns the sweep points that were included in the measurement. For a frequency sweep operation, this parameter returns the frequencies at which **data** was taken. For a time-domain operation, this parameter returns the time or distance. For a power sweep operation, this parameter returns the power values at which the **data** was taken.
readCount	ViInt32*	Returns the number of data points.

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function.

		The general meaning of the status code is as follows: **Value** / **Meaning** 0 — Success Positive Values — Warnings Negative Values — Errors

Calibration

Class/Panel Name	Function Name
Calibration	
Init User Calibration	niVNA_InitUserCal
Perform Calibration Action	niVNA_PerformCalibrationAction
Perform TRL Calibration Action	niVNA_PerformTRLCalibrationAction
Apply Adapter Removal Calibration	niVNA_ApplyAdapterRemovalCalibration
Get Calibration State	niVNA_GetCalibrationState
Is Calibration Action Done	niVNA_IsCalibrationActionDone
Wait For Calibration Action Done	niVNA_WaitForCalibrationActionDone
Abort Calibration Action	niVNA_AbortCalibrationAction
Close User Calibration	niVNA_CloseUserCal
Load Calibration And Configuration Data From File	niVNA_LoadCalibrationAndConfigurationDataFromFil
Save Calibration And Configuration Data To File	niVNA_SaveCalibrationAndConfigurationDataToFile
Load User Defined Manual Calibration Kit Data From File	niVNA_LoadUserDefinedManualCalibrationKitDataFro
AutoCal	
Query AutoCal Box Info	niVNA_QueryAutoCalBoxInfo
Load AutoCal Characterization Data From File	niVNA_LoadAutoCalCharacterizationDataFromFile

niVNA_InitUserCal

niVNA_InitUserCal(ViSession instrumentHandle, ViConstString channel, ViInt32 calibrationType);

Purpose

Initializes user calibration.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.
channel	ViConstString	Specifies the channel for which this operation is targeted.
calibrationType	ViInt32	Specifies the type of calibration that is performed. NIVNA_VAL_CALIBRATION_TYPE_2PORT_SOLT — NI-VNA is configured to perform a two-port, one-path short/open/load/through (SOLT) calibration. NIVNA_VAL_CALIBRATION_TYPE_1PORT_SOL — NI-VNA is configured to perform a one-port short/open/load (SOL) calibration.

		NIVNA_VAL_CALIBRATION_TYPE_REFLECTION_RESPONSE	NI-VNA is configured t perform a reflection-response calibration.
		NIVNA_VAL_CALIBRATION_TYPE_TRANSMISSION_RESPONSE	NI-VNA is configured t perform a transmissio response calibration.
		NIVNA_VAL_CALIBRATION_TYPE_TRL	NI-VNA is configured t perform a fu two-port through-reflect-line (TRL) calibration.
		NIVNA_VAL_CALIBRATION_TYPE_POWER	NI-VNA is configured t perform a power calibration, including source and receiver calibration.
		NIVNA_VAL_CALIBRATION_TYPE_ADAPTER_REMOVAL	NI-VNA is configured t apply a calibration that de-embeds an adapter.

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function. The general meaning of the status code is as follows: **Value** — **Meaning** 0 — Success Positive Values — Warnings Negative Values — Errors

niVNA_PerformCalibrationAction

niVNA_PerformCalibrationAction(ViSession instrumentHandle, ViConstString portList, ViInt32 actionType);

Purpose

Performs one calibration action on the specified **portList** or **actionType**.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description

instrumentHandle	ViSession	Identifies your instrument handle.
portList	ViConstString	Specifies the ports on which to perform the specified calibration action. If yo set **actionType** to `NIVNA_VAL_CALIBRATION_ACTION_SOURCE_POWE R_CAL` or NIVNA_VAL_CALIBRATION_ACTION_REFERENCE_RECEIVER_CAL, t valid port list specifier is "`port1`" or "`port2`". If you set **actionType** to `NIV A_VAL_CALIBRATION_ACTION_TEST_RECEIVER_CAL`, the valid port specifier is "`port1,port2`" to use the source on PORT 1 and calibrate the receiver on PORT 2 or "`port2,port1`" to use the source on PORT 2 and calibrate the receiver on PORT 1.
actionType	ViInt32	Specifies the type of calibration action to perform. NIVNA_VAL_CALIBRATION_ACTION_OPEN — NI-VNA performs calibratio using the manual open standard. NIVNA_VAL_CALIBRATION_ACTION_SHORT — NI-VNA performs calibratio using the manual short standard. NIVNA_VAL_CALIBRATION_ACTION_LOAD — NI-VNA performs calibratio using the manual load standard. NIVNA_VAL_CALIBRATION_ACTION_THRU — NI-VNA performs calibratio using the manual through

	standard.
NIVNA_VAL_CALIBRATION_ACTION_ISOLATION	NI-VNA performs calibratio using the manual load standard measure leakage from two terminate ports.
NIVNA_VAL_CALIBRATION_ACTION_AUTOCAL_STANDARD	NI-VNA performs standard automati calibratio using an AutoCal module.
NIVNA_VAL_CALIBRATION_ACTION_AUTOCAL_TRUE_THRU	NI-VNA performs standard automati calibratio using an AutoCal module and an external through connectio between the port cables.
NIVNA_VAL_CALIBRATION_ACTION_SOURCE_POWER_CAL	NI-VNA performs

			source power calibratio using a power meter.
		NIVNA_VAL_CALIBRATION_ACTION_TEST_RECEIVER_CAL	NI-VNA performs test receiver power calibratio using a through standard.
		NIVNA_VAL_CALIBRATION_ACTION_REFERENCE_RECEIVER_CAL	NI-VNA performs reference receiver power calibratio using a match loa standard match connecti

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function.

		The general meaning of the status code is as follows: **Value** / **Meaning** 0 — Success Positive Values — Warnings Negative Values — Errors

niVNA_PerformTRLCalibrationAction

niVNA_PerformTRLCalibrationAction(ViSession instrumentHandle, ViConstString portList, ViConstString standard);

Purpose

Performs one through-reflect-line (TRL) calibration action on the specified **portList** and **standard**.

Supported Devices: NI 5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.
portList	ViConstString	Specifies the ports on which to perform the specified calibration action.
standard	ViConstString	Specifies the standard on which to perform the calibration action. Valid standard specifiers include "`standard1`," "`standard2`," "`standard3`," "`standard4`," "`standard5`," and "`standard6`". For more information about valid specifiers, refer to the TRL

		Calibration (`TRLCal.c`) Example.

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function. The general meaning of the status code is as follows: **Value** / **Meaning** 0 / Success Positive Values / Warnings Negative Values / Errors

niVNA_ApplyAdapterRemovalCalibration

niVNA_ApplyAdapterRemovalCalibration(ViSession instrumentHandle, ViConstString portList, ViConstString firstPortCalFilePath, ViConstString secondPortCalFilePath, ViReal64 adapterLength);

Purpose

Loads the calibration files, specifies the adapter length, and prepares the device for performing adapter removal calibration.

Supported Devices: NI 5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.
portList	ViConstString	Specifies the ports on which to perform the specified calibration action.
firstPortCalFilePath	ViConstString	Specifies the saved calibration file path with the adapter connected to the first port in the **portList**.
secondPortCalFilePath	ViConstString	Specifies the saved calibration file path with the adapter connected to the second port in the **portList**.
adapterLength	ViReal64	Specifies the electrical length of the adapter, in seconds.

Return Value

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>returnValue</td><td>ViStatus</td><td>Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred.

To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function.

The general meaning of the status code is as follows:

<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>0</td><td>Success</td></tr>
<tr><td>Positive Values</td><td>Warnings</td></tr>
<tr><td>Negative Values</td><td>Errors</td></tr>
</table>
</td></tr>
</table>

niVNA_GetCalibrationState

niVNA_GetCalibrationState(ViSession instrumentHandle, ViConstString channel, ViInt32 *state);

Purpose

Returns the state of the user calibration.

Supported Devices: NI 5630/5632

Parameters

<table>
<tr><td colspan="3">Input</td></tr>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td>instrumentHandle</td><td>ViSession</td><td>Identifies your instrument handle.</td></tr>
<tr><td>channel</td><td>ViConstString</td><td>Specifies the channel for which this operation is targeted.</td></tr>
<tr><td colspan="3">Output</td></tr>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td>state</td><td>ViInt32*</td><td>Returns the state of the calibration.

The relationship between the NIVNA_ATTR_CORRECTION_ENABLED attribute and the state parameter is shown in the following table.
<table>
<tr><th rowspan="2">Correction Enabled</th><th colspan="3">State</th></tr>
<tr><th>Uncalibrated</th><th>Calibrated</th><th>Calibrating</th></tr>
<tr><td>FALSE</td><td>NI-VNA does not have a valid set of calibration data for the current device configuration.</td><td>NI-VNA does have a valid set of calibration data for the current device configuration;</td><td>N/A</td></tr>
</table></td></tr>
</table>

			Measurement data is ***not*** corrected using calibration data.	however, measurement data is ***not*** corrected using calibration data.	
		TRUE	NI-VNA does ***not*** have a valid set of calibration data for the current device configuration; however, measurement data ***is*** modified using (potentially invalid) calibration data.	NI-VNA ***does*** have a valid set of calibration data for the current device configuration. Measurement data ***is*** corrected using calibration data.	N/A

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function. The general meaning of the status code is as follows: **Value** — **Meaning** 0 — Success Positive Values — Warnings

		Negative Values: Errors

niVNA_IsCalibrationActionDone

niVNA_IsCalibrationActionDone(ViSession instrumentHandle, ViConstString channel, ViBoolean *isDone);

Purpose

Checks the status of the user calibration action. Use this function to check for any errors that may occur during the calibration action or to check whether the last calibration action is complete.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.
channel	ViConstString	Specifies the channel for which this operation is targeted.
Output		
Name	Type	Description
isDone	ViBoolean*	Returns whether the operation has completed.

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code

		from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function. The general meaning of the status code is as follows: **Value** / **Meaning** 0 / Success Positive Values / Warnings Negative Values / Errors

niVNA_WaitForCalibrationActionDone

niVNA_WaitForCalibrationActionDone(ViSession instrumentHandle, ViConstString channel, ViReal64 timeout);

Purpose

Waits for the last user calibration action to complete. If the calibration action does not complete before the specified timeout, this function returns a timeout error.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.
channel	ViConstString	Specifies the channel for which this operation is targeted.
timeout	ViReal64	Specifies the time to wait for the function to complete operation

		before returning an error. A negative value specifies that NI-VNA waits indefinitely. This value is expressed in seconds (s).

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function. The general meaning of the status code is as follows: **Value** / **Meaning** 0 / Success Positive Values / Warnings Negative Values / Errors

niVNA_AbortCalibrationAction

niVNA_AbortCalibrationAction(ViSession instrumentHandle, ViConstString channel);

Purpose

Aborts the user calibration action that is in progress.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.
channel	ViConstString	Specifies the channel for which this operation is targeted.

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function. The general meaning of the status code is as follows: **Value** / **Meaning** 0 / Success Positive Values / Warnings Negative Values / Errors

niVNA_CloseUserCal

niVNA_CloseUserCal(ViSession instrumentHandle, ViConstString channel, ViInt32 action);

Purpose

Closes the user calibration session.

Supported Devices: NI 5630/5632

Parameters

<table>
<tr><th colspan="3">Input</th></tr>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td>instrumentHandle</td><td>ViSession</td><td>Identifies your instrument handle.</td></tr>
<tr><td>channel</td><td>ViConstString</td><td>Specifies the channel for which this operation is targeted.</td></tr>
<tr><td>action</td><td>ViInt32</td><td>Specifies the action to perform when closing the user calibration.

<table><tr><td>NIVNA_VAL_USER_CALIBRATION_ABORT</td><td>NI-VNA discards the calibration constants calculated during the user calibration.</td></tr><tr><td>NIVNA_VAL_USER_CALIBRATION_COMMIT</td><td>NI-VNA commits the calibration constants to the device and transitions the device to the Calibrated calibration state. After you commit the values, you should not change some device attributes for this session.</td></tr></table></td></tr>
</table>

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function. The general meaning of the status code is as follows: **Value** / **Meaning** 0 / Success Positive Values / Warnings Negative Values / Errors

niVNA_LoadCalibrationAndConfigurationDataFror

niVNA_LoadCalibrationAndConfigurationDataFromFile(ViSession instrumentHandle, ViConstString channel, ViConstString filePath);

Purpose

Loads user calibration and configuration data from a file.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description

instrumentHandle	ViSession	Identifies your instrument handle.
channel	ViConstString	Specifies the channel for which this operation is targeted.
filePath	ViConstString	Specifies the absolute file path from which to load the calibration and configuration data.

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function. The general meaning of the status code is as follows: **Value** / **Meaning** 0 / Success Positive Values / Warnings Negative Values / Errors

niVNA_SaveCalibrationAndConfigurationDataToFi

niVNA_SaveCalibrationAndConfigurationDataToFile(ViSession instrumentHandle, ViConstString channel, ViConstString filePath, ViBoolean overwrite);

Purpose

Saves the current user calibration and configuration data into a file.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.
channel	ViConstString	Specifies the channel for which this operation is targeted.
filePath	ViConstString	Specifies the absolute file path into which to save the user calibration data.
overwrite	ViBoolean	Specifies whether to overwrite the existing file, if one exists.

Return Value

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>returnValue</td><td>ViStatus</td><td>Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred.

To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function.

The general meaning of the status code is as follows:

<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>0</td><td>Success</td></tr>
<tr><td>Positive Values</td><td>Warnings</td></tr>
<tr><td>Negative Values</td><td>Errors</td></tr>
</table>
</td></tr>
</table>

niVNA_LoadUserDefinedManualCalibrationKitData

niVNA_LoadUserDefinedManualCalibrationKitDataFromFile(ViSession instrumentHandle, ViConstString channel, ViConstString filePath);

Purpose

Loads the user-defined manual calibration characterization data.

Specify the user-defined manual calibration kit data using the NI-VNA SFP. Refer to the User Calibration topic for more information about specifying manual calibration kit data.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.
channel	ViConstString	Specifies the channel for which this operation is targeted.
filePath	ViConstString	Specifies the absolute file path from which to load the calibration and configuration data.

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function.

		The general meaning of the status code is as follows: **Value** / **Meaning** 0 — Success Positive Values — Warnings Negative Values — Errors

AutoCal

Class/Panel Name	Function Name
AutoCal	
Query AutoCal Box Info	niVNA_QueryAutoCalBoxInfo
Load AutoCal Characterization Data From File	niVNA_LoadAutoCalCharacterizationDataFromFile

niVNA_QueryAutoCalBoxInfo

niVNA_QueryAutoCalBoxInfo

niVNA_QueryAutoCalBoxInfo(ViSession instrumentHandle, ViConstString channel, ViChar serialNumber[], ViInt32 serialNumberBufferSize, ViChar type[], ViInt32 typeBufferSize);

Purpose

Queries the AutoCal module information, including the **serialNumber** and **type**.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.
channel	ViConstString	Specifies the channel for which this operation is targeted.
serialNumberBufferSize	ViInt32	Specifies the size of the buffer for the **serialNumber** parameter.
typeBufferSize	ViInt32	Specifies the size of the **type** array.
Output		
Name	Type	Description
serialNumber	ViChar[]	Returns the serial number of the AutoCal module.
type	ViChar[]	Returns the type of the AutoCal module.

Return Value

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>returnValue</td><td>ViStatus</td><td>Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred.

To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function.

The general meaning of the status code is as follows:

<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>0</td><td>Success</td></tr>
<tr><td>Positive Values</td><td>Warnings</td></tr>
<tr><td>Negative Values</td><td>Errors</td></tr>
</table>
</td></tr>
</table>

niVNA_LoadAutoCalCharacterizationDataFro

niVNA_LoadAutoCalCharacterizationDataFromFile

niVNA_LoadAutoCalCharacterizationDataFromFile(ViSession instrumentHandle, ViConstString channel, ViConstString filePath);

Purpose

Loads the AutoCal calibration characterization data.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.
channel	ViConstString	Specifies the channel for which this operation is targeted.
filePath	ViConstString	Specifies the absolute file path from which to load the AutoCal characterization data.

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function. The general meaning of the status code is as follows:

		Value / **Meaning** 0 / Success Positive Values / Warnings Negative Values / Errors

Utility

Class/Panel Name	Function Name
Utility	
Self test	niVNA_self_test
Reset	niVNA_reset
Reset With Defaults	niVNA_ResetWithDefaults
Disable	niVNA_Disable
Revision query	niVNA_revision_query
Feature Activation	
Activate Feature	niVNA_ActivateFeature
Query Feature Activation Status	niVNA_QueryFeatureActivationStatus
Error Info	
Error query	niVNA_error_query
Error message	niVNA_error_message
Get Error	niVNA_GetError
Clear Error	niVNA_ClearError
Get Next Coercion Record	niVNA_GetNextCoercionRecord
Interchangeability Info	
Get Next Interchange Warning	niVNA_GetNextInterchangeWarning
Reset Interchange Check	niVNA_ResetInterchangeCheck

Clear Interchange Warnings	niVNA_ClearInterchangeWarnings
Locking	
Lock Session	niVNA_LockSession
Unlock Session	niVNA_UnlockSession

niVNA_self_test

niVNA_self_test(ViSession instrumentHandle, ViInt16 *testResult, ViChar testMessage[]);

Purpose

Performs a self-test on the NI-VNA device and returns the test result. This function performs a simple series of tests verifying that the NI-VNA device is powered on and responding.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.
Output		
Name	Type	Description
testResult	ViInt16*	Returns the value returned from the device self-test. A value of 0 indicates that the device is powered on and responding. A value of 1 indicates that the device failed the self test.
testMessage	ViChar[]	Returns the self-test response string from the NI-VNA device. You must pass a ViChar array with at least 256 bytes to this parameter.

Return Value

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>returnValue</td><td>ViStatus</td><td>Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred.

To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function.

The general meaning of the status code is as follows:

<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>0</td><td>Success</td></tr>
<tr><td>Positive Values</td><td>Warnings</td></tr>
<tr><td>Negative Values</td><td>Errors</td></tr>
</table>
</td></tr>
</table>

niVNA_reset

niVNA_reset(ViSession instrumentHandle);

Purpose

Resets the device to a known initialization state.

Supported Devices: NI 5630/5632

Parameters

<table>
<tr><th colspan="3">Input</th></tr>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td>instrumentHandle</td><td>ViSession</td><td>Identifies your instrument handle.</td></tr>
</table>

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function. The general meaning of the status code is as follows: **Value** — **Meaning** 0 — Success Positive Values — Warnings Negative Values — Errors

niVNA_ResetWithDefaults

niVNA_ResetWithDefaults(ViSession instrumentHandle);

Purpose

Resets the device to a known initialization state with defaults as specified for the device in the **optionString** parameter in the niVNA_InitWithOptions function.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description

instrumentHandle	ViSession	Identifies your instrument handle.

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function. The general meaning of the status code is as follows: **Value** / **Meaning** 0 / Success Positive Values / Warnings Negative Values / Errors

niVNA_Disable

niVNA_Disable(ViSession instrumentHandle);

Purpose

Places the instrument in a quiescent state where it has minimal or no impact on the system to which it is connected.

	**Note** This function is currently not supported.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function. The general meaning of the status code is as follows: **Value** / **Meaning** 0 / Success Positive Values / Warnings Negative Values / Errors

niVNA_revision_query

niVNA_revision_query(ViSession instrumentHandle, ViChar instrumentDriverRevision[], ViChar firmwareRevision[]);

Purpose

Queries the revision of the device.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.
Output		
Name	Type	Description
instrumentDriverRevision	ViChar[]	Returns the instrument driver revision information. You must pass a ViChar array with at least 256 bytes to this parameter.
firmwareRevision	ViChar[]	Returns firmware revision information. You must pass a ViChar array with at least 256 bytes to this parameter.

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function. The general meaning of the status code is as follows:

Value	**Meaning**
0	Success
Positive Values	Warnings

		Negative Values: Errors

Feature Activation

Class/Panel Name	Function Name
Feature Activation	
Activate Feature	niVNA_ActivateFeature
Query Feature Activation Status	niVNA_QueryFeatureActivationStatus

niVNA_ActivateFeature

niVNA_ActivateFeature(ViSession instrumentHandle, ViInt32 feature, ViConstString activationKey);

Purpose

Activates a feature for the specified device.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.
feature	ViInt32	Specifies the feature for which NI-VNA queries the activation status.

		NIVNA_VAL_FEATURE_TIME_DOMAIN: Activates time-domain analysis on the device.
activationKey	ViConstString	Specifies the key you use to activate certain features. NI provides this key after you purchase the option.

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function. The general meaning of the status code is as follows: **Value** / **Meaning** 0 / Success Positive Values / Warnings Negative Values / Errors

niVNA_QueryFeatureActivationStatus

niVNA_QueryFeatureActivationStatus(ViSession instrumentHandle, ViInt32 feature, ViBoolean *isActivated);

Purpose

Queries the activation status of a certain feature.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.
feature	ViInt32	Specifies the feature for which NI-VNA queries the activation status. NIVNA_VAL_FEATURE_TIME_DOMAIN — Activates time-domain analysis on the device.
Output		
Name	Type	Description
isActivated	ViBoolean*	Returns the activation status of the feature.

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function. The general meaning of the status code is as follows:

		Value / **Meaning** 0 / Success Positive Values / Warnings Negative Values / Errors

Error Info

Class/Panel Name	Function Name
Error Info	
Error query	niVNA_error_query
Error message	niVNA_error_message
Get Error	niVNA_GetError
Clear Error	niVNA_ClearError

niVNA_error_query

niVNA_error_query(ViSession instrumentHandle, ViStatus *errorCode, ViChar errorMessage[]);

Purpose

Returns the last error code and message.

Note This function is currently not supported.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.
Output		
Name	Type	Description
errorCode	ViStatus*	Passes the **returnValue** parameter that is returned from any NI-VNA function. The default value is 0 (VI_SUCCESS).
errorMessage	ViChar[]	Returns the user-readable message string that corresponds to the status code you specify. You must pass a ViChar array with at least 256 bytes to this parameter.

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function. The general meaning of the status code is as follows: **Value** — **Meaning** 0 — Success Positive Values — Warnings Negative Values — Errors

niVNA_error_message

niVNA_error_message(ViSession instrumentHandle, ViStatus errorCode, ViChar errorMessage[]);

Purpose

Converts a status code returned by an NI-VNA function into a user-readable string.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.
errorCode	ViStatus*	Passes the **returnValue** parameter that is returned from any NI-VNA function. The default value is 0 (VI_SUCCESS).
Output		
Name	Type	Description
errorMessage	ViChar[]	Returns the user-readable message string that corresponds to the status code you specify. You must pass a ViChar array with at least 256 bytes to this parameter.

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the

		error information from NI-VNA, call the niVNA_ClearError function. The general meaning of the status code is as follows: **Value** / **Meaning** 0 / Success Positive Values / Warnings Negative Values / Errors

niVNA_GetError

niVNA_GetError(ViSession instrumentHandle, ViStatus *errorCode, ViInt32 bufferSize, ViChar description[]);

Purpose

Purpose

Retrieves and then clears the IVI error information for the session or the current execution thread.

Note If the **bufferSize** parameter is 0, this function does not clear the error information. By passing 0 to **bufferSize**, you can determine the buffer size required to read the entire error description string. You can then call this function again with a sufficiently large buffer.

If you specify a valid IVI session for the **vi** parameter, this function retrieves and then clears the error information for the session. If you pass VI_NULL for **vi**, this function retrieves and then clears the error information for the current execution thread. If **vi** is an invalid session, this function does nothing and returns an error. Normally, the error information describes the first error that occurred since you last called this function or the niVNA_ClearError function.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.
bufferSize	ViInt32	Specifies the size of the buffer into which to acquire the error message.
Output		
Name	Type	Description
errorCode	ViStatus*	Passes the **returnValue** parameter that is returned from any NI-VNA function. The default value is 0 (VI_SUCCESS).
description	ViChar[]	Returns the error description for the IVI session or execution thread. If there is no description, the `niVNA_GetError` function returns an empty string. The buffer must contain at least as many elements as the value you specify with **bufferSize**. If the error description, including the terminating NULL byte, contains more bytes than you indicate with bufferSize, the function copies **bufferSize** - 1 bytes into the buffer, places an ASCII NULL byte at the end of the buffer, and returns the size of the buffer that you must pass to get the entire value. For example, if the value is "123456" and the **bufferSize** is 4, the function places "123" into the buffer and returns 7. If you pass 0 to **bufferSize**, you can pass VI_NULL for this parameter.

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function.

		The general meaning of the status code is as follows: **Value** / **Meaning** 0 — Success Positive Values — Warnings Negative Values — Errors

niVNA_ClearError

niVNA_ClearError(ViSession instrumentHandle);

Purpose

Clears the last error code.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the

		error information from NI-VNA, call the niVNA_ClearError function. The general meaning of the status code is as follows: **Value** / **Meaning** 0 / Success Positive Values / Warnings Negative Values / Errors

niVNA_GetNextCoercionRecord

niVNA_GetNextCoercionRecord(ViSession instrumentHandle, ViInt32 bufferSize, ViChar record[]);

Purpose

Obtains the coercion information associated with the IVI session.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.
bufferSize	ViInt32	Specifies the size of the buffer into which to acquire the interchangeability record.
Output		
Name	Type	Description
record	ViChar[]	record ViChar[] Returns the next coercion record for the IVI session. If there are no coercion records, the function returns an empty

		string. The buffer must contain at least as many elements as the value you specify with **bufferSize**. If the next coercion record string, including the terminating NULL byte, contains more bytes than you indicate with **bufferSize**, the function copies **bufferSize** - 1 bytes into the buffer, places an ASCII NULL byte at the end of the buffer, and returns the size of the buffer you must pass to get the entire value. For example, if the value is "123456" and the **bufferSize** is 4, the function places "123" into the buffer and returns 7. This parameter returns an empty string if no coercion records remain for the session.

Return Value

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>returnValue</td><td>ViStatus</td><td>Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred.

To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function.

The general meaning of the status code is as follows:

<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>0</td><td>Success</td></tr>
<tr><td>Positive Values</td><td>Warnings</td></tr>
<tr><td>Negative Values</td><td>Errors</td></tr>
</table>
</td></tr>
</table>

Interchangeability Info

Class/Panel Name	Function Name

Interchangeability Info	
Get Next Interchange Warning	niVNA_GetNextInterchangeWarning
Reset Interchange Check	niVNA_ResetInterchangeCheck
Clear Interchange Warnings	niVNA_ClearInterchangeWarnings

niVNA_GetNextInterchangeWarning

niVNA_GetNextInterchangeWarning(ViSession instrumentHandle, ViInt32 bufferSize, ViChar warnString[]);

Purpose

This function returns the interchangeability warnings associated with the IVI session.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.
bufferSize	ViInt32	Indicates the number of bytes in the ViChar array you specify for **warnString**. If the current value of the attribute, including the terminating NULL byte, is larger than the size you indicate in **bufferSize**, the function copies **bufferSize** - 1 bytes into the buffer, places an ASCII NULL byte at the end of the buffer, and returns the size of the buffer that you must pass to get the entire value. For example, if the value is "123456" and the **bufferSize** is 4, the function places "123" into the buffer and returns 7. If you pass a negative number to this parameter, the function copies the value to the buffer regardless of the number of bytes in the value.

Output		
Name	Type	Description
warnString	ViChar[]	Returns the oldest interchange warning for the IVI session. This parameter returns an empty string if there are no interchange warnings remaining for the session. The buffer must contain at least as many elements as the value you specify with **bufferSize**. If the next interchangeability warning string, including the terminating NULL byte, contains more bytes than you indicate with **bufferSize**, the function copies **bufferSize** - 1 bytes into the buffer, places an ASCII NULL byte at the end of the buffer, and returns the size of the buffer that you must pass to get the entire value. For example, if the value is "123456" and the **bufferSize** is 4, the function places "123" into the buffer and returns 7.

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function. The general meaning of the status code is as follows: **Value** — **Meaning** 0 — Success Positive Values — Warnings Negative Values — Errors

niVNA_ResetInterchangeCheck

niVNA_ResetInterchangeCheck(ViSession instrumentHandle);

Purpose

Resets the interchangeability checking algorithms of the IVI specific driver.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.

Return Value

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>returnValue</td><td>ViStatus</td><td>Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred.

To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function.

The general meaning of the status code is as follows:

<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>0</td><td>Success</td></tr>
<tr><td>Positive Values</td><td>Warnings</td></tr>
<tr><td>Negative Values</td><td>Errors</td></tr>
</table>
</td></tr>
</table>

niVNA_ClearInterchangeWarnings

niVNA_ClearInterchangeWarnings(ViSession instrumentHandle);

Purpose

Clears the list of interchangeability warnings that the IVI specific driver maintains.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.

Return Value

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>returnValue</td><td>ViStatus</td><td>Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred.

To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function.

The general meaning of the status code is as follows:

<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>0</td><td>Success</td></tr>
<tr><td>Positive Values</td><td>Warnings</td></tr>
<tr><td>Negative Values</td><td>Errors</td></tr>
</table>
</td></tr>
</table>

Locking

Class/Panel Name	Function Name
Locking	
Lock Session	niVNA_LockSession
Unlock Session	niVNA_UnlockSession

niVNA_LockSession

niVNA_LockSession(ViSession instrumentHandle, ViBoolean *callerHasLock);

Purpose

Obtains a multithread lock on the instrument session. Before doing so, this function waits until all other execution threads have released their locks on the instrument session.

Other threads might have obtained a lock on this session in the following ways:

- Your application already called this function.
- A call to NI-VNA locked the session.

After the call to this function returns successfully, no other threads can access the instrument session until you call the niVNA_UnlockSession function. Use the niVNA_LockSession function and the niVNA_UnlockSession function around a sequence of calls to NI-VNA functions if you require that the NI-VNA device retain its settings through the end of the sequence.

You can safely make nested calls to the niVNA_LockSession function within the same thread. To completely unlock the session, balance each call to the niVNA_LockSession function with a call to the niVNA_UnlockSession function. If, however, you use **callerHasLock** in all calls to the niVNA_LockSession function and the niVNA_UnlockSession function within a function, the IVI Library locks the session only once within the function regardless of the number of calls you make to the niVNA_LockSession function. Locking the session only once allows you to call the

niVNA_UnlockSession function just once at the end of the function.

Supported Devices: NI 5630/5632

Parameters

<table>
<tr><td colspan="3">Input</td></tr>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td>instrumentHandle</td><td>ViSession</td><td>Identifies your instrument handle.</td></tr>
<tr><td colspan="3">Output</td></tr>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td>callerHasLock</td><td>ViBoolean*</td><td>Keeps track of whether you obtain a lock and therefore need to unlock the session in complex functions. Pass the address of a local ViBoolean variable. In the declaration of the local variable, initialize it to VI_FALSE. Pass the address of the same local variable to any other calls you make to this function or the niVNA_UnlockSession function in the same function.

This parameter serves as a convenience. If you do not want to use this parameter, pass VI_NULL.

The niVNA_LockSession function and the niVNA_UnlockSession function each inspect the current value and take the following actions:
<table>
<tr><td rowspan="2">niVNA_LockSession</td><td>VI_TRUE</td><td>The niVNA_LockSession function does not lock the session again.</td></tr>
<tr><td>VI_FALSE</td><td>The niVNA_LockSession function obtains the lock and sets the value of the parameter to VI_TRUE.</td></tr>
<tr><td>niVNA_UnlockSession</td><td>VI_FALSE</td><td>The niVNA_UnlockSession function does not attempt to unlock the session.</td></tr>
</table>
</td></tr>
</table>

		<table><tr><td></td><td>VI_TRUE</td><td>The niVNA_UnlockSession function releases the lock and sets the value of the parameter to VI_FALSE.</td></tr></table> Thus, you can call the niVNA_UnlockSession function at the end of your function regardless of whether you actually have the lock.

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function. The general meaning of the status code is as follows: <table><tr><th>Value</th><th>Meaning</th></tr><tr><td>0</td><td>Success</td></tr><tr><td>Positive Values</td><td>Warnings</td></tr><tr><td>Negative Values</td><td>Errors</td></tr></table>

niVNA_UnlockSession

niVNA_UnlockSession(ViSession instrumentHandle, ViBoolean *callerHasLock);

Purpose

Releases a lock obtained on an NI-VNA device session by calling the niVNA_LockSession function. Refer to the niVNA_LockSession function for additional information on session locks.

Supported Devices: NI 5630/5632

Parameters

<table>
<tr><td colspan="3">Input</td></tr>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td>instrumentHandle</td><td>ViSession</td><td>Identifies your instrument handle.</td></tr>
<tr><td colspan="3">Output</td></tr>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td>callerHasLock</td><td>ViBoolean*</td><td>Keeps track of whether you obtain a lock and therefore need to unlock the session in complex functions. Pass the address of a local ViBoolean variable. In the declaration of the local variable, initialize it to VI_FALSE. Pass the address of the same local variable to any other calls you make to this function or the niVNA_UnlockSession function in the same function.

This parameter serves as a convenience. If you do not want to use this parameter, pass VI_NULL.

The niVNA_LockSession function and the niVNA_UnlockSession function each inspect the current value and take the following actions:

<table>
<tr><td rowspan="2">niVNA_LockSession</td><td>VI_TRUE</td><td>The niVNA_LockSession function does not lock the session again.</td></tr>
<tr><td>VI_FALSE</td><td>The niVNA_LockSession function obtains the lock and sets the value of the</td></tr>
</table>
</td></tr>
</table>

		parameter to VI_TRUE.
niVNA_UnlockSession	VI_FALSE	The niVNA_UnlockSession function does not attempt to unlock the session.
	VI_TRUE	The niVNA_UnlockSession function releases the lock and sets the value of the parameter to VI_FALSE.

Thus, you can call the niVNA_UnlockSession function at the end of your function regardless of whether you actually have the lock.

Return Value

Name	Type	Description
returnValue	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred. To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function. The general meaning of the status code is as follows: **Value** — **Meaning** 0 — Success Positive Values — Warnings Negative Values — Errors

niVNA_close

niVNA_close(ViSession instrumentHandle);

Purpose

Closes the session to the device.

Supported Devices: NI 5630/5632

Parameters

Input		
Name	Type	Description
instrumentHandle	ViSession	Identifies your instrument handle.

Return Value

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>returnValue</td><td>ViStatus</td><td>Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to an NI-VNA function to determine if an error has occurred.

To obtain a text description of the status code and additional information about the error condition, call the niVNA_GetError function. To clear the error information from NI-VNA, call the niVNA_ClearError function.

The general meaning of the status code is as follows:

<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>0</td><td>Success</td></tr>
<tr><td>Positive Values</td><td>Warnings</td></tr>
<tr><td>Negative Values</td><td>Errors</td></tr>
</table>
</td></tr>
</table>

NIVNA_ATTR_EXPORTED_READY_FOR_START_EVE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	Device	None	niVNA_ExportSignal

Description

Specifies the terminal where the signal is exported. You can also choose to not export a signal.

Defined Values:

NIVNA_VAL_DO_NOT_EXPORT_STR	The signal is not exported.
NIVNA_VAL_PFI0	The signal is exported to PFI 0, which is the external signal input terminal on the module front panel.
NIVNA_VAL_PXI_TRIG0	The signal is exported to PXI trigger line 0.
NIVNA_VAL_PXI_TRIG1	The signal is exported to PXI trigger line 1.
NIVNA_VAL_PXI_TRIG2	The signal is exported to PXI trigger line 2.
NIVNA_VAL_PXI_TRIG3	The signal is exported to PXI trigger line 3.
NIVNA_VAL_PXI_TRIG4	The signal is exported to PXI trigger line 4.
NIVNA_VAL_PXI_TRIG5	The signal is exported to PXI trigger line 5.
NIVNA_VAL_PXI_TRIG6	The signal is exported to PXI trigger line 6.
NIVNA_VAL_PXI_TRIG	The signal is exported to PXI trigger line 7.

7	

Default Value: NIVNA_VAL_DO_NOT_EXPORT_STR

Supported Devices: NI 5632

NIVNA_ATTR_READY_FOR_START_EVENT_LEVEL

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	Device	None	N/A

Description

Specifies the active level for the Ready For Start Event.

Defined Values:

NIVNA_VAL_TRIGGER_LEVEL_ACTIVE_LOW (0)	Specifies a low digital level.

Default Value: NIVNA_VAL_TRIGGER_LEVEL_ACTIVE_LOW

Supported Devices: NI 5632

NIVNA_ATTR_EXPORTED_READY_FOR_STEP_EVEI

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	Device	None	niVNA_ExportSignal

Description

Specifies the terminal where the signal is exported. You can also choose to not export a signal.

Defined Values:

NIVNA_VAL_DO_NOT_EXPORT_STR	The signal is not exported.
NIVNA_VAL_PFI0	The signal is exported to PFI 0, which is the external signal input terminal on the module front panel.
NIVNA_VAL_PXI_TRIG0	The signal is exported to PXI trigger line 0.
NIVNA_VAL_PXI_TRIG1	The signal is exported to PXI trigger line 1.
NIVNA_VAL_PXI_TRIG2	The signal is exported to PXI trigger line 2.
NIVNA_VAL_PXI_TRIG3	The signal is exported to PXI trigger line 3.
NIVNA_VAL_PXI_TRIG4	The signal is exported to PXI trigger line 4.
NIVNA_VAL_PXI_TRIG5	The signal is exported to PXI trigger line 5.
NIVNA_VAL_PXI_TRIG6	The signal is exported to PXI trigger line 6.
NIVNA_VAL_PXI_TRIG7	The signal is exported to PXI trigger line 7.

Default Value: NIVNA_VAL_DO_NOT_EXPORT_STR

Supported Devices: NI 5632

NIVNA_ATTR_READY_FOR_STEP_EVENT_LEVEL

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	Device	None	N/A

Description

Specifies the active level for the Ready For Step Event.

Defined Values:

NIVNA_VAL_TRIGGER_LEVEL_ACTIVE_LOW (0)	Specifies a low digital level.

Default Value: NIVNA_VAL_TRIGGER_LEVEL_ACTIVE_LOW

Supported Devices: NI 5632

NIVNA_ATTR_EXPORTED_END_OF_STEP_EVENT_

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	Device	None	niVNA_ExportSignal

Description

Specifies the terminal where the signal is exported. You can also choose to not export a signal.

Defined Values:

NIVNA_VAL_DO_NOT_EXPORT_STR	The signal is not exported.
NIVNA_VAL_PFI0	The signal is exported to PFI 0, which is the external signal input terminal on the module front panel.
NIVNA_VAL_PXI_TRIG0	The signal is exported to PXI trigger line 0.
NIVNA_VAL_PXI_TRIG1	The signal is exported to PXI trigger line 1.
NIVNA_VAL_PXI_TRIG2	The signal is exported to PXI trigger line 2.
NIVNA_VAL_PXI_TRIG3	The signal is exported to PXI trigger line 3.
NIVNA_VAL_PXI_TRIG4	The signal is exported to PXI trigger line 4.
NIVNA_VAL_PXI_TRIG5	The signal is exported to PXI trigger line 5.
NIVNA_VAL_PXI_TRIG6	The signal is exported to PXI trigger line 6.
NIVNA_VAL_PXI_TRIG7	The signal is exported to PXI trigger line 7.

Default Value: NIVNA_VAL_DO_NOT_EXPORT_STR

Supported Devices: NI 5632

NIVNA_ATTR_END_OF_STEP_EVENT_EDGE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	Device	None	N/A

Description

Specifies the active edge for the End Of Step Event.

Defined Values:

NIVNA_VAL_TRIGGER_EDGE_FALLING (0)	The end of the step is signaled by a falling edge.

Default Value: NIVNA_VAL_TRIGGER_EDGE_FALLING

Supported Devices: NI 5632

NIVNA_ATTR_EXPORTED_END_OF_SWEEP_EVENT

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	Device	None	niVNA_ExportSignal

Description

Specifies the terminal where the signal is exported. You can also choose to not export a signal.

Defined Values:

NIVNA_VAL_DO_NOT_EXPORT_STR	The signal is not exported.
NIVNA_VAL_PFI0	The signal is exported to PFI 0, which is the external signal input terminal on the module front panel.
NIVNA_VAL_PXI_TRIG0	The signal is exported to PXI trigger line 0.

NIVNA_VAL_PXI_TRIG1	The signal is exported to PXI trigger line 1.
NIVNA_VAL_PXI_TRIG2	The signal is exported to PXI trigger line 2.
NIVNA_VAL_PXI_TRIG3	The signal is exported to PXI trigger line 3.
NIVNA_VAL_PXI_TRIG4	The signal is exported to PXI trigger line 4.
NIVNA_VAL_PXI_TRIG5	The signal is exported to PXI trigger line 5.
NIVNA_VAL_PXI_TRIG6	The signal is exported to PXI trigger line 6.
NIVNA_VAL_PXI_TRIG7	The signal is exported to PXI trigger line 7.

Default Value: NIVNA_VAL_DO_NOT_EXPORT_STR

Supported Devices: NI 5632

NIVNA_ATTR_END_OF_SWEEP_EVENT_EDGE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	Device	None	N/A

Description

Specifies the active edge for the End Of Sweep Event.

Defined Values:

NIVNA_VAL_TRIGGER_EDGE_FALLING (0)	The end of the sweep is signaled by a falling edge.

Default Value: NIVNA_VAL_TRIGGER_EDGE_FALLING

Supported Devices: NI 5632

NIVNA_ATTR_START_TRIGGER_TYPE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	Device	None	N/A

Description

Specifies the Start Trigger type.

Defined Values:

NIVNA_VAL_TRIGGER_NONE (0)	A Start Trigger is not configured. The sweep measurement starts immediately after you call the niVNA_Initiate function.
NIVNA_VAL_TRIGGER_DIGITAL_EDGE (1)	The sweep measurement starts after the device receives the configured digital edge from the selected source. The source of the digital edge is specified with the NIVNA_ATTR_START_TRIGGER_SOURCE attribute.
NIVNA_VAL_TRIGGER_SOFTWARE_EDGE (2)	The sweep measurement starts after you call the niVNA_SendSoftwareEdgeTrigger function. You can assert the software trigger by calling the niVNA_SendSoftwareEdgeTrigger.

Supported Devices: NI 5630/5632

NIVNA_ATTR_START_TRIGGER_SOURCE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	Device	None	niVNA_ConfigureDigitalEdgeStartTrigger

Description

Specifies the Start Trigger source.

Defined Values:

NIVNA_VAL_PFI0	The trigger is received from PFI 0, which is the external trigger input terminal on the module front panel.
NIVNA_VAL_PXI_TRIG0	The trigger is received on PXI trigger line 0.
NIVNA_VAL_PXI_TRIG1	The trigger is received on PXI trigger line 1.
NIVNA_VAL_PXI_TRIG2	The trigger is received on PXI trigger line 2.
NIVNA_VAL_PXI_TRIG3	The trigger is received on PXI trigger line 3.
NIVNA_VAL_PXI_TRIG4	The trigger is received on PXI trigger line 4.
NIVNA_VAL_PXI_TRIG5	The trigger is received on PXI trigger line 5.
NIVNA_VAL_PXI_TRIG6	The trigger is received on PXI trigger line 6.
NIVNA_VAL_PXI_TRIG7	The trigger is received on PXI trigger line 7.

Supported Devices: NI 5630/5632

NIVNA_ATTR_START_TRIGGER_EDGE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	Device	None	niVNA_ConfigureDigitalEdgeStartTrigger

Description

Specifies the active edge for the Start Trigger. This attribute is used only when the NIVNA_ATTR_START_TRIGGER_TYPE attribute is set to NIVNA_VAL_TRIGGER_DIGITAL_EDGE.

Defined Values:

NIVNA_VAL_TRIGGER_EDGE_RISING (1)	The sweep measurement starts on the rising edge of the signal.
NIVNA_VAL_TRIGGER_EDGE_FALLING (0)	The sweep measurement starts on the falling edge of the signal.

Supported Devices: NI 5630/5632

NIVNA_ATTR_STEP_TRIGGER_TYPE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	Device	None	N/A

Description

Specifies the Step Trigger type.

Defined Values:

NIVNA_VAL_TRIGGER_NONE (0)	A Step Trigger is not configured. The sweep measurement starts immediately after you call the niVNA_Initiate function.
NIVNA_VAL_TRIGGER_DIGITAL_EDGE (1)	The sweep measurement starts after the device receives the configured digital edge from the selected source. The source of the digital edge is specified with the NIVNA_ATTR_STEP_TRIGGER_SOURCE attribute.

Default Value: NIVNA_VAL_TRIGGER_NONE

Supported Devices: NI 5632

NIVNA_ATTR_STEP_TRIGGER_SOURCE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	Device	None	niVNA_ConfigureDigitalEdgeStepTrigger

Description

Specifies the Step Trigger source.

Defined Values:

NIVNA_VAL_PFI0	The trigger is received from PFI 0, which is the external trigger input terminal on the module front panel.

NIVNA_VAL_PXI_TRIG0	The trigger is received on PXI trigger line 0.
NIVNA_VAL_PXI_TRIG1	The trigger is received on PXI trigger line 1.
NIVNA_VAL_PXI_TRIG2	The trigger is received on PXI trigger line 2.
NIVNA_VAL_PXI_TRIG3	The trigger is received on PXI trigger line 3.
NIVNA_VAL_PXI_TRIG4	The trigger is received on PXI trigger line 4.
NIVNA_VAL_PXI_TRIG5	The trigger is received on PXI trigger line 5.
NIVNA_VAL_PXI_TRIG6	The trigger is received on PXI trigger line 6.
NIVNA_VAL_PXI_TRIG7	The trigger is received on PXI trigger line 7.

Supported Devices: NI 5632

NIVNA_ATTR_STEP_TRIGGER_EDGE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	Device	None	niVNA_ConfigureDigitalEdgeStepTrigger

Description

Specifies the active edge for the Step Trigger. This attribute is used only when the NIVNA_ATTR_STEP_TRIGGER_TYPE attribute is set to NIVNA_VAL_TRIGGER_DIGITAL_EDGE.

Defined Values:

NIVNA_VAL_TRIGGER_EDGE_RISING (1)	The sweep measurement starts on the rising edge of the signal.
NIVNA_VAL_TRIGGER_EDGE_FALLING (0)	The sweep measurement starts on the falling edge of the signal.

Supported Devices: NI 5632

NIVNA_ATTR_AVERAGING_COUNT

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	Channel	None	None

Description

Specifies the averaging count for sweep averaging. Sweep averaging can be enabled by setting this attribute to a value greater than 1. Specify the type of sweep averaging using the NIVNA_ATTR_AVERAGING_TYPE attribute.

Supported Devices: NI 5630/5632

NIVNA_ATTR_AVERAGING_TYPE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	None

Description

Specifies the type of sweep averaging.

Defined Values:

`NIVNA_VAL_AVERAGING_POINTBYPOINT` (0)	The NI vector network analyzer uses point-by-point averaging. Point-by-point averaging takes the specified number of measurements at each point of the sweep.
`NIVNA_VAL_AVERAGING_SWEEPBYSWEEP` (1)	The NI vector network analyzer uses sweep-by-sweep averaging. Sweep-by-sweep averaging performs the given number of sweeps and then averages each point.

Supported Devices: NI 5630/5632

NIVNA_ATTR_SMOOTHING_APERTURE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Channel	None	None

Description

Specifies the smoothing aperture for the formatted data. Smoothing is applied to the formatted data after sweep averaging and error correction (calibration term application).

Units: percentage

Valid Range: 0 to 20

Supported Devices: NI 5630/5632

NIVNA_ATTR_IF_BANDWIDTH

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Channel	None	None

Description

Specifies the intermediate frequency (IF) bandwidth for the NI-VNA device.

Valid Values

NI 5630—10 Hz, 30 Hz, 100 Hz, 300 Hz, 1 kHz, 3 kHz, 10 kHz, or 30 kHz

NI 5632—10 Hz, 30 Hz, 50 Hz, 70 Hz, 100 Hz, 300 Hz, 500 Hz, 700 Hz, 1 kHz, 3 kHz, 5 kHz, 7 kHz, 10 kHz, or 30 kHz, 50 kHz, 70 kHz, 100 kHz, 300 kHz, or 500 kHz

Default Value: 1 kHz

Supported Devices: NI 5630/5632

NIVNA_ATTR_OUTPUT_POWER

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Port	N/A	None

Description

Specifies the output power for the active port.

Units: dBm

Valid Ranges:

NI 5630—(-25) to 5, in 0.5 dBm increments

NI 5632—(-30) to 10, in 0.01 dBm increments

Default Value: 5

Supported Devices: NI 5630/5632

NIVNA_ATTR_REFERENCE_FREQUENCY_SOURCE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	N/A	None	None

Description

Specifies the source for the reference frequency.



Note The NI 5630 does not support the `NIVNA_VAL_FREQUENCY_REFERENCE_PXI_CLK100` value as a frequency source.

Defined Values:

`NIVNA_VAL_REF_IN`	The reference signal received from the REF IN connector on the module front panel is used as the source for the frequency reference signal.
`NIVNA_VAL_FREQUENC`	The reference signal received from the PXI_CLK100 terminal on the NI

Y_REFERENCE_PXI_CLK100	5632 PXI Express backplane is used as the source for the frequency reference signal.
NIVNA_VAL_ONBOARD_REFERENCE	The reference signal received from the onboard Reference Clock is used as the source for the frequency reference signal.

Supported Devices: NI 5630/5632

NIVNA_ATTR_REFERENCE_FREQUENCY

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	RO	N/A	None	None

Description

Returns the reference frequency.

Valid Values:

NI 5630—10 MHz

NI 5632—10 MHz when you set the NIVNA_ATTR_REFERENCE_FREQUENCY_SOURCE attribute to NIVNA_VAL_REF_IN, 100 MHz when you set the NIVNA_ATTR_REFERENCE_FREQUENCY_SOURCE attribute to NIVNA_VAL_FREQUENCY_REFERENCE_PXI_CLK100.

Units: Hz

Default Value: 10 MHz

Supported Devices: NI 5630/5632

NIVNA_ATTR_REFPLANE_EXTENSION_DISTANCE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Port	None	None

Description

Specifies the length of the reference plane extension. Port extension parameters can aid with adapter removal, thus providing more accurate measurement.

Units: meters

Supported Devices: NI 5630/5632

NIVNA_ATTR_REFPLANE_EXTENSION_TIME

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Port	None	None

Description

Specifies the time it takes for the signal to travel across the reference plane extension. Port extension parameters can aid with adapter removal, thus providing more accurate measurement.

Units: seconds (s)

Supported Devices: NI 5630/5632

NIVNA_ATTR_REFPLANE_EXTENSION_DIELECTRIC

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	Port	None	None

Description

Specifies the reference plane extension dielectric type. Port extension parameters can aid with adapter removal, thus providing more accurate measurement.

Defined Values:

NIVNA_VAL_REFPLANE_EXTENSION_DIELECTRIC_AIR (0)	The reference plane extension uses air as the dielectric, which is associated with a dielectric constant of 1.000649.
NIVNA_VAL_REFPLANE_EXTENSION_DIELECTRIC_POLYTHYLENE (1)	The reference plane extension uses a polyethylene dielectric, which is associated with a dielectric constant of 2.26.
NIVNA_VAL_REFPLANE_EXTENSION_DIELECTRIC_TEFLON (2)	The reference plane extension uses a teflon dielectric, which is associated with a dielectric constant of 2.1.
NIVNA_VAL_REFPLANE_EXTENSION_DIELECTRIC_MICRO_TEFLON (3)	The reference plane extension uses a microteflon dielectric, which is associated with a dielectric constant of 1.69.
NIVNA_VAL_REFPLANE_EXTENSION_DIELECTRIC_USER_VALUE (4)	The reference plane extension uses the value you specify for the NIVNA_ATTR_REFPLANE_EXTENSION_DIELECTRIC_USER_VALUE attribute as the dielectric value.

Supported Devices: NI 5630/5632

NIVNA_ATTR_REFPLANE_EXTENSION_DIELECTRIC

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Port	None	None

Description

Specifies the reference plane extension dielectric user value.

Supported Devices: NI 5630/5632

NIVNA_ATTR_REFPLANE_EXTENSION_INSERTION

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Port	None	None

Description

Specifies the reference plane extension dielectric insertion loss. Port extension parameters can aid with adapter removal and fixture de-embedding, thus providing more accurate measurement.

Units: dB

Supported Devices: NI 5630/5632

NIVNA_ATTR_SWEEP_MODE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	Channel	None	niVNA_ConfigureSweep

Description

Specifies the sweep mode for the NI vector network analyzers.

Note The NI 5630 supports only the NIVNA_VAL_SWEEP_FREQUENCY mode. For more information about sweep modes, refer to the NI 5632 Segmented Sweep topic.

Note You can also configure the NIVNA_VAL_SWEEP_FREQUENCY_BASED_SEGMENTED and NIVNA_VAL_SWEEP_INDEX_BASED_SEGMENTED modes in the NI-VNA Soft Front Panel.

Defined Values:

NIVNA_VAL_SWEEP_FREQUENCY (0)	The device sweeps across a range of frequencies with constant power.
NIVNA_VAL_SWEEP_POWER (1)	The device sweeps across a range of powers with constant frequency.
NIVNA_VAL_SWEEP_FREQUENCY_BASED_SEGMENTED (2)	The device sweeps across a list of monotonically increasing frequencies.
NIVNA_VAL_SWEEP_INDEX_BASED_SEGMENTED (3)	The device sweeps across a list of frequencies.

Default Value: NIVNA_VAL_SWEEP_FREQUENCY

Supported Devices: NI 5630/5632

NIVNA_ATTR_SWEEP_NUMBER_OF_POINTS

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	Channel	None	niVNA_ConfigureSweep

Description

Specifies the number of points for the sweep measurement.

Valid Ranges:

NI 5630—1 to 3,201

NI 5632—1 to 20,001

Default Value: 201

Supported Devices: NI 5630/5632

NIVNA_ATTR_SWEEP_FREQUENCY_START

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Channel	None	niVNA_ConfigureFrequencyStartStop

Description

Specifies the start frequency for the frequency sweep.

Valid Values:

NI 5630—≥10 MHz

NI 5632—≥300 kHz

Units: Hz

Default Value: 10 MHz

Supported Devices: NI 5630/5632

NIVNA_ATTR_SWEEP_FREQUENCY_STOP

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Channel	None	niVNA_ConfigureFrequencyStartStop

Description

Specifies the stop frequency for the frequency sweep.

Valid Values:

NI 5630—≤6 GHz

NI 5632—≤8.5 GHz

Units: Hz

Default Value: 6 GHz

Supported Devices: NI 5630/5632

NIVNA_ATTR_POWER_SWEEP_FREQUENCY

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Channel	None	None

Description

Specifies the output frequency for power sweep.

Units: Hz

Valid Range: 300 kHz to 8.5 GHz, in 0.1 Hz increments

Default Value: 300 kHz

Supported Devices: NI 5632

NIVNA_ATTR_SWEEP_POWER_START

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Port	None	`niVNA_ConfigurePowerStartStop`

Description

Specifies the start power for the power sweep.

Units: dBm

Default Value: -30.0

Valid Values: ≥-30

Supported Devices: NI 5632

NIVNA_ATTR_SWEEP_POWER_STOP

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Port	None	niVNA_ConfigurePowerStartStop

Description

Specifies the stop power for the power sweep.

Units: dBm

Default Value: 10

Valid Values: ≤10

Supported Devices: NI 5632

NIVNA_ATTR_SOURCE_CALIBRATION_POWER_OF

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Port	None	None

Description

Specifies the power offset to be applied during source power calibration. The offset applies to all sweep types.

Power offset occurs when you place a device with gain or loss between the source port and the device under test (DUT). You can then place the power meter at the reference plane, and the gain or loss is removed out of the system during calibration so that the actual power applied to the DUT is the port power plus the offset. This attribute affects only the output of the VNA after a power calibration.

Units: dB

Default Value: 0

Valid Range: -100 to 100

Supported Devices: NI 5632

NIVNA_ATTR_SWEEP_STEP_DELAY

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Channel	None	None

Description

Specifies the delay for each sweep point. The sweep step delay allows, for example, synchronization with external devices, such as antenna rotors.

Units: seconds (s)

Supported Devices: NI 5630/5632

NIVNA_ATTR_FORMAT_PHASE_OFFSET

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Channel	None	None

Description

Specifies the phase offset that is applied to the formatted data. This attribute merely shifts the phase of the returned data by the specified amount. The reference plane configuration is not affected.

Units: degrees

Supported Devices: NI 5630/5632

NIVNA_ATTR_S11_ENABLED

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	Channel	None	None

Description

Specifies whether the S11 measurement is enabled. If you disable the S11 measurement, you cannot read the S11 parameter.



Note If you enable the appropriate S-parameters for your measurement, the VNA is capable of optimizing measurement time.

VI_TRUE	S11 measurement is enabled.
VI_FALSE	S11 measurement is disabled.

Supported Devices: NI 5630/5632

NIVNA_ATTR_S21_ENABLED

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	N/A	None	None

Description

Specifies whether the S21 measurement is enabled. If you disable the S21 measurement, you cannot read the S21 parameter.



Note If you enable the appropriate S-parameters for your measurement, the VNA is capable of optimizing measurement time.

VI_TRUE	S21 measurement is enabled.
VI_FALSE	S21 measurement is disabled.

Supported Devices: NI 5630/5632

NIVNA_ATTR_S12_ENABLED

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	Channel	None	None

Description

Specifies whether the S12 measurement is enabled. If you disable the S12 measurement, you cannot read the S12 parameter.



Note If you enable the appropriate S-parameters for your measurement, the VNA is capable of optimizing measurement time.

VI_TRUE	S12 measurement is enabled.
VI_FALSE	S12 measurement is disabled.

Supported Devices: NI 5632

NIVNA_ATTR_S22_ENABLED

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	Channel	None	None

Description

Specifies whether the S22 measurement is enabled. If you disable the S22 measurement, you cannot read the S22 parameter.



Note If you enable the appropriate S-parameters for your measurement, the VNA is capable of optimizing measurement time.

VI_TRUE	S22 measurement is enabled.
VI_FALSE	S22 measurement is disabled.

Supported Devices: NI 5632

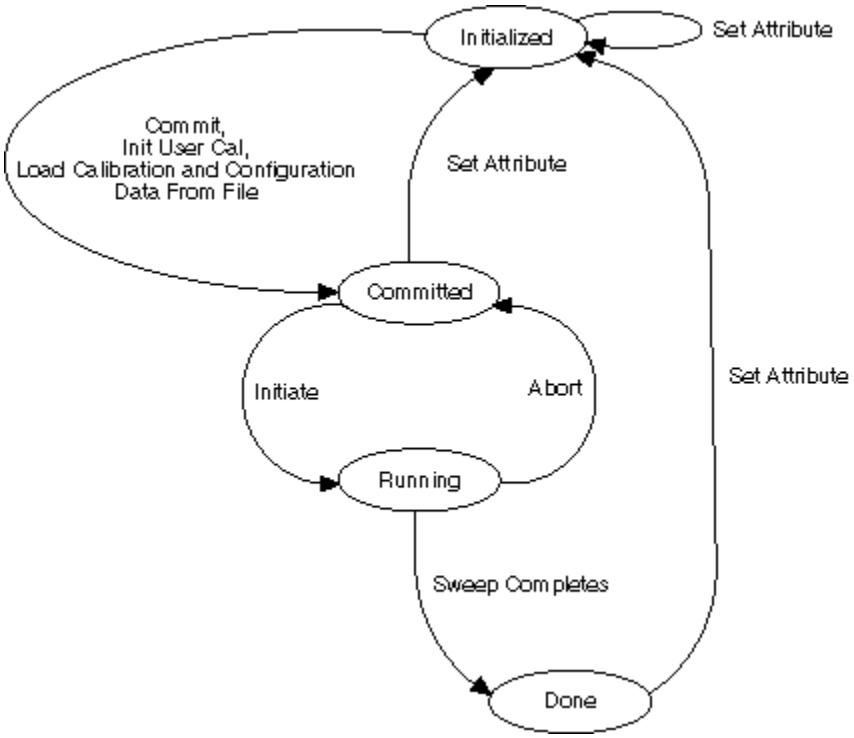
NIVNA_ATTR_DRIVER_STATE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	RO	N/A	None	None

Description

Returns the state of NI-VNA. The possible states are shown in the following diagram.



Defined Values:

NIVNA_VAL_DRIVER_STATE_INITIALIZED (1)	The driver has been initialized.
NIVNA_VAL_DRIVER_STATE_COMMITTED (4)	The driver settings have been committed.

NIVNA_VAL_DRIVER_STATE_RUNNING (5)	The driver is running.
NIVNA_VAL_DRIVER_STATE_DONE (6)	The driver has finished acquiring data.

Supported Devices: NI 5630/5632

NIVNA_ATTR_CORRECTION_ENABLED

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	Channel	None	None

Description

Specifies whether NI-VNA applies calibration data to the raw measurements from the hardware. Changing attribute values after calibration can result in inappropriate correction and inaccurate measurements. For best results, calibrate after the device is fully configured and take measurements without changing any configuration values.

Note This attribute is valid only for RF calibrations and does not apply to power calibrations including, source power calibration, test receiver calibration, and reference receiver calibration.

The relationship between the NIVNA_ATTR_CORRECTION_ENABLED and the NIVNA_ATTR_CALIBRATION_CURRENT_STATE attribute is shown in the following table.

Correction Enabled	Calibration State		
	Uncalibrated	Calibrated	Calibrating
FALSE	NI-VNA does ***not*** have a valid set of	NI-VNA ***does*** have a valid set of	N/A

	calibration data for the current device configuration. Measurement data is ***not*** corrected using calibration data.	calibration data for the current device configuration; however, measurement data is ***not*** corrected using calibration data.	
TRUE	NI-VNA does ***not*** have a valid set of calibration data for the current device configuration; however, measurement data ***is*** modified using (potentially invalid) calibration data.	NI-VNA ***does*** have a valid set of calibration data for the current device configuration. Measurement data ***is*** corrected using calibration data.	N/A

Note This attribute is automatically set to `VI_TRUE` when calibration is successfully loaded from a file, performed manually, or performed using the AutoCal module.

Supported Devices: NI 5630/5632

NIVNA_ATTR_CALIBRATION_CURRENT_STATE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	RO	N/A	None	None

Description

Returns the calibration state. The calibration state indicates whether NI-VNA has calibration data for the current configuration.

The relationship between the `NIVNA_ATTR_CORRECTION_ENABLED` and the `NIVNA_ATTR_CALIBRATION_CURRENT_STATE` attribute is shown in the following table.

Correction Enabled	Calibration State		
	Uncalibrated	**Calibrated**	**Calibrating**
FALSE	NI-VNA does ***not*** have a valid set of	NI-VNA ***does*** have a valid set of	N/A

	calibration data for the current device configuration. Measurement data is ***not*** corrected using calibration data.	calibration data for the current device configuration; however, measurement data is ***not*** corrected using calibration data.	
TRUE	NI-VNA does ***not*** have a valid set of calibration data for the current device configuration; however, measurement data ***is*** modified using (potentially invalid) calibration data.	NI-VNA ***does*** have a valid set of calibration data for the current device configuration. Measurement data ***is*** corrected using calibration data.	N/A

Defined Values:

`NIVNA_VAL_CALIBRATION_STATE_UNCALIBRATED` (0)	Indicates that the NI vector network analyzer is not calibrated.
`NIVNA_VAL_CALIBRATION_STATE_CALIBRATING` (1)	Indicates that the NI vector network analyzer is performing calibration.
`NIVNA_VAL_CALIBRATION_STATE_CALIBRATED` (2)	Indicates that the NI vector network analyzer is calibrated.

Supported Devices: NI 5630/5632

NIVNA_ATTR_RF_CALIBRATION_TYPE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	RO	Channel	None	None

Description

Returns the type of RF calibration performed on the device.

You must set the following S-parameter attributes so that the NIVNA_ATTR_RF_CALIBRATION_TYPE attribute can determine the appropriate calibration port.

- NIVNA_ATTR_S11_ENABLED
- NIVNA_ATTR_S12_ENABLED
- NIVNA_ATTR_S21_ENABLED
- NIVNA_ATTR_S22_ENABLED

The following table lists the RF calibration types and corresponding S-parameter attribute settings.

RF Calibration Type	Port Source	S11 Enabled	S21 Enabled	S12 Enabled	S22 Enabled
Two-Port	PORT 1/PORT 2	TRUE	TRUE	TRUE	TRUE
Two-Port, One-Path	PORT 1	TRUE	TRUE	FALSE	FALSE
Two-Port, One-Path	PORT 2	FALSE	FALSE	TRUE	TRUE
One-Port	PORT 1	TRUE	FALSE	FALSE	FALSE
One-Port	PORT 2	FALSE	FALSE	FALSE	TRUE
Reflection	PORT 1	TRUE	FALSE	FALSE	FALSE
Reflection	PORT 2	FALSE	FALSE	FALSE	TRUE
Reflection	PORT 1/PORT 2	TRUE	FALSE	FALSE	TRUE
Transmission	PORT 1	FALSE	TRUE	FALSE	FALSE
Transmission	PORT 2	FALSE	FALSE	TRUE	FALSE
Transmission	PORT 1/PORT 2	FALSE	TRUE	TRUE	FALSE

Defined Values:

NIVNA_VAL_CALIBRATION_TYPE_2PORT_SOLT (0)	The NI vector network analyzer performs two-port, one-path short/open/load/thru (SOLT) calibration.

NIVNA_VAL_CALIBRATION_TYPE_1PORT_SOL (1)	The NI vector network analyzer performs one-port short/open/load (SOL) calibration.
NIVNA_VAL_CALIBRATION_TYPE_REFLECTION_RESPONSE (2)	The NI vector network analyzer performs one-port reflection response calibration.
NIVNA_VAL_CALIBRATION_TYPE_TRANSMISSION_RESPONSE (3)	The NI vector network analyzer performs transmission response calibration.
NIVNA_VAL_CALIBRATION_TYPE_TRL (4)	The NI vector network analyzer performs a full two-port through-reflect-line (TRL) calibration.



Note NIVNA_VAL_CALIBRATION_TYPE_TRL is a valid value for only the NI 5632.

Default Value: NIVNA_VAL_CALIBRATION_TYPE_2PORT_SOLT

Supported Devices: NI 5630/5632

NIVNA_ATTR_ALLOW_CHANGES_WHILE_CALIBRA

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	N/A	None	None

Description

Specifies whether changes are allowed after the device is calibrated.

Supported Devices: NI 5630/5632

NIVNA_ATTR_CALIBRATION_HIGH_SENSITIVITY_E

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	N/A	None	None

Description

Specifies whether high-sensitivity mode is enabled while calibrating.

Supported Devices: NI 5630

NIVNA_ATTR_POWER_METER_NAME

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	Device	None	None

Description

Specifies the resource name of the external power meter used for power calibration.

To determine the power meter name, use the NI-VNA Soft Front Panel (SFP). Within the NI-VNA SFP, select **Power**»**Power Cal**»**Perform Cal** from the Softkey Menu. The power meter name, or COM port name, is listed in the drop-down menu under the **Model_SerialNumber_ComPort** control.

Supported Devices: NI 5632

NIVNA_ATTR_POWER_COMPENSATION_SLOPE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Port	None	None

Description

Specifies the slope compensation for frequency sweep-based power calibration.

Units: dB/GHz

Default Value: 0

Supported Devices: NI 5632

NIVNA_ATTR_DE_EMBEDDING_ENABLED

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	RO	Channel	None	None

Description

Returns whether de-embedding is enabled.

For more information about de-embedding a test fixture, refer to the NI 5632 De-Embedding topic.

Defined Values:

NIVNA_VAL_DE_EMBEDDING_DISABLED (0)	De-embedding is disabled.
NIVNA_VAL_DE_EMBEDDING_ENABLED (1)	De-embedding is enabled.

Supported Devices: NI 5632

NIVNA_ATTR_RF_CALIBRATION_STATUS

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViBoolean	RO	Channel	None	None

Description

Indicates the RF calibration status of the channel. If RF calibration is available, use the NIVNA_ATTR_CORRECTION_ENABLED attribute to apply an RF calibration.

Defined Values:

VI_TRUE (0)	RF calibration is available.
VI_FALSE (1)	RF calibration is not available.

Default Value: VI_FALSE

Supported Devices: NI 5632

NIVNA_ATTR_SOURCE_POWER_CALIBRATION_ST

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViBoolean	RO	Port	None	None

Description

Indicates the source power calibration status of the specified port.

Defined Values:

`VI_TRUE` (0)	Calibration is applied.
`VI_FALSE` (1)	Calibration is not applied.

Default Value: `VI_FALSE`

Supported Devices: NI 5632

NIVNA_ATTR_TEST_RECEIVER_CALIBRATION_STA

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViBoolean	RO	Port	None	None

Description

Indicates the test receiver power calibration status of the specified receiver.

Defined Values:

`VI_TRUE` (0)	Calibration is applied.
`VI_FALSE` (1)	Calibration is not applied.

Supported Devices: NI 5632

NIVNA_ATTR_REFERENCE_RECEIVER_CALIBRATIC

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViBoolean	RO	Port	None	None

Description

Indicates the reference receiver power calibration status of the specified receiver.

Defined Values:

`VI_TRUE` (0)	Calibration is applied.
`VI_FALSE` (1)	Calibration is not applied.

Supported Devices: NI 5632

NIVNA_ATTR_CALIBRATION_THRU_LINE_TYPE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	Channel	None	None

Description

Specifies the through line type for manual calibration. A ***through line*** is a direct cable connection between the test ports, with no intervening connectors.

Note The NI 5630 supports only the `NIVNA_VAL_CALIBRATION_THRU_LINE_TYPE_THRU` mode.

Defined Values:

`NIVNA_VAL_CALIBRATION_THRU_LINE_TYPE_THRU` (0)	The VNA uses a through line for manual calibration.
`NIVNA_VAL_CALIBRATION_THRU_LINE_TYPE_RECIPROCAL` (1)	The VNA uses a reciprocal line for manual calibration.

Default Value: `NIVNA_VAL_CALIBRATION_THRU_LINE_TYPE_THRU`

Supported Devices: NI 5630/5632

NIVNA_ATTR_CALIBRATION_THRU_LINE_LENGTH

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Channel	None	None

Description

Specifies the through line length for manual calibration. A ***through line*** is a direct cable connection between the test ports, with no intervening connectors.

Units: mm

Default Value: 0

Supported Devices: NI 5632

NIVNA_ATTR_CALIBRATION_THRU_LINE_IMPEDA

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Channel	None	None

Description

Specifies the through line impedance for manual calibration. A ***through line*** is a direct cable connection between the test ports, with no intervening connectors.

Units: ohms

Default Value: 50

Supported Devices: NI 5632

NIVNA_ATTR_CALIBRATION_THRU_LINE_LOSS

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions

ViReal64	R/W	Channel	None	None

Description

Specifies the line loss of the through line for manual calibration. A ***through line*** is a direct cable connection between the test ports, with no intervening connectors.

Use the NIVNA_ATTR_CALIBRATION_THRU_REF_FREQUENCY attribute to specify the frequency at which line loss occurs.

Units: dB/mm

Default Value: 0

Supported Devices: NI 5632

NIVNA_ATTR_CALIBRATION_THRU_REF_FREQUEN

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Channel	None	None

Description

Specifies the through line reference frequency for manual calibration. A ***through line*** is a direct cable connection between the test ports, with no intervening connectors.

Use the NIVNA_ATTR_CALIBRATION_THRU_LINE_LOSS attribute to specify line loss at the frequency specified for this attribute.

Units: Hz

Default Value: 0

Supported Devices: NI 5632

NIVNA_ATTR_CONNECTOR_GENDER

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	Port	None	niVNA_ConfigureConnector

Description

Specifies the port connector gender.

Defined Values:

NIVNA_VAL_CONN_GENDER_MALE (0)	The port has a male connector.
NIVNA_VAL_CONN_GENDER_FEMALE (1)	The port has a female connector.

Supported Devices: NI 5630/5632

NIVNA_ATTR_CONNECTOR_TYPE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	Port	None	niVNA_ConfigureConnector

Description

Specifies the port connector type.

Defined Values:

NIVNA_VAL_CONN_TYPE_N (0)	The port uses an N-type connector.
NIVNA_VAL_CONN_TYPE_K (1)	The port uses a K-type connector.
NIVNA_VAL_CONN_TYPE_GPC (2)	The port uses a GPC 3.5 connector.
NIVNA_VAL_CONN_TYPE_SMA (3)	The port uses an SMA connector.

Supported Devices: NI 5630/5632

NIVNA_ATTR_AUTOCAL_TRUE_THRU_LENGTH

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Channel	None	None

Description

Specifies the true through length for the AutoCal. A ***true through*** is a direct cable connection between the test ports, with no intervening connectors. If desired, you can remove the AutoCal module for the through calibration step, and you can supply the parameters of the true through connection using this attribute.

Units: mm

Supported Devices: NI 5630/5632

NIVNA_ATTR_AUTOCAL_TRUE_THRU_IMPEDANCE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Channel	None	None

Description

Specifies a true through impedance for the AutoCal. A ***true through*** is a direct cable connection between the test ports, with no intervening connectors. If desired, you can remove the AutoCal module for the through calibration step, and you can specify the parameters of the true through connection using this attribute.

Units: ohms

Supported Devices: NI 5630/5632

NIVNA_ATTR_AUTOCAL_TRUE_THRU_LINE_LOSS

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Channel	None	None

Description

Specifies a line loss for the true through for AutoCal. A ***true through*** is a direct cable connection between the test ports, with no intervening connectors. If desired, you can

remove the AutoCal module for the through calibration step, and you can specify the parameters of the true through connection using this attribute.

Units: dB/mm

Supported Devices: NI 5630/5632

NIVNA_ATTR_AUTOCAL_TRUE_THRU_REF_FREQU

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Channel	None	None

Description

Specifies the true through reference frequency for the AutoCal. A ***true through*** is a direct cable connection between the test ports, with no intervening connectors. If desired, you can remove the AutoCal module for the through calibration step, and you can specify the parameters of the true through connection using this attribute.

Units: Hz

Supported Devices: NI 5630/5632

NIVNA_ATTR_AUTOCAL_BOX_ORIENTATION

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	Channel	None	None

Description

Specifies the orientation of the AutoCal module. If you set this attribute incorrectly, the AutoCal assurance fails. Enable AutoCal assurance using the NIVNA_ATTR_AUTOCAL_ASSURANCE_ENABLED attribute.

Defined Values:

<table>
<tr>
<td>NIVNA_VAL_AUTOCAL_BOX_ORIENTATION_A (1)</td>
<td>Specifies that either the VNA PORT 1 connects to the left port on the AutoCal module or PORT 2 connects to the right port on the AutoCal module, as indicated in the following table. An X indicates that the specified port is not applicable.
<table>
<tr><th>Calibration</th><th>Port 1</th><th>Port 2</th></tr>
<tr><td>1 Port (S11)</td><td>Left</td><td>X</td></tr>
<tr><td>1 Port (S22)</td><td>X</td><td>Right</td></tr>
<tr><td>2 Port, 2 Port 1 Path</td><td>Left</td><td>Right</td></tr>
</table>
</td>
</tr>
<tr>
<td>NIVNA_VAL_AUTOCAL_BOX_ORIENTATION_B (0)</td>
<td>Specifies that either the VNA PORT 1 connects to the right port on the AutoCal module or PORT 2 connects to the left port on the AutoCal module, as indicated in the following table. An X indicates that the specified port is not applicable.
<table>
<tr><th>Calibration</th><th>Port 1</th><th>Port 2</th></tr>
<tr><td>1 Port (S11)</td><td>Right</td><td>X</td></tr>
<tr><td>1 Port (S22)</td><td>X</td><td>Left</td></tr>
<tr><td>2 Port, 2 Port 1 Path</td><td>Right</td><td>Left</td></tr>
</table>
</td>
</tr>
<tr>
<td>NIVNA_VAL_AUTOCAL_BOX_ORIENTATION_AUTO_SENSE (2)</td>
<td>Specifies that the VNA determines the orientation of the AutoCal module. The supported calibration is two-port SOLT.</td>
</tr>
</table>

Supported Devices: NI 5630/5632

NIVNA_ATTR_AUTOCAL_BOX_COMMUNICATION_I

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	Channel	None	None

Description

Specifies the communication port to which the AutoCal module is connected.

Supported Devices: NI 5630/5632

NIVNA_ATTR_AUTOCAL_ASSURANCE_ENABLED

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	Channel	None	None

Description

Specifies whether an assurance step is automatically executed for the AutoCal. The assurance step verifies whether AutoCal ran successfully. If enabled, the assurance step is performed at the time that the `niVNA_CloseUserCal` function is called.

Supported Devices: NI 5630/5632

NIVNA_ATTR_TRL_CALIBRATION_REFERENCE_PL

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	Channel	None	None

Description

Specifies the location of the reference plane for TRL calibration.

Defined Values:

`NIVNA_VAL_TRL_CALIBRATION_REF_PLANE_MIDDLE_OF_STANDARD1` (0)	The reference plane is located at the middle of the "standard1" line standard.
`NIVNA_VAL_TRL_CALIBRATION_REF_PLANE_ENDS_OF_STANDARD1` (1)	The reference plane is located at the ends of the "standard1" line standard.

Default Value: `NIVNA_VAL_TRL_CALIBRATION_REF_PLANE_MIDDLE_OF_STANDARD1`

Supported Devices: NI 5632

NIVNA_ATTR_TRL_CALIBRATION_NUMBER_OF_B

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	Channel	None	None

Description

Specifies the number of frequency bands to use for TRL calibration.

Default Value: 1

Valid Values: 1 or 2

Supported Devices: NI 5632

NIVNA_ATTR_TRL_CALIBRATION_STANDARD_TYP

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	Standard	None	None

Description

Specifies the type of standard used for TRL calibration.

Defined Values:

`NIVNA_VAL_TRL_CALIBRATION_STANDARD_TYPE_SHORT` (0)	The standard is a short-like device.
`NIVNA_VAL_TRL_CALIBRATION_STANDARD_TYPE_OPEN` (1)	The standard is an open-like device.
`NIVNA_VAL_TRL_CALIBRATION_STANDARD_TYPE_LINE` (2)	The standard is a line device.

NIVNA_VAL_TRL_CALIBRATION_STANDARD_TYPE_MATCH (3)	The standard is a matched load.

Default Value: NIVNA_VAL_TRL_CALIBRATION_STANDARD_TYPE_LINE

Supported Devices: NI 5632

NIVNA_ATTR_TRL_CALIBRATION_STANDARD_LEN

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Standard	None	None

Description

Specifies the length of the TRL calibration standard.

If you set the NIVNA_ATTR_TRL_CALIBRATION_STANDARD_TYPE attribute to NIVNA_VAL_TRL_CALIBRATION_STANDARD_TYPE_LINE, the NIVNA_ATTR_TRL_CALIBRATION_STANDARD_LENGTH attribute corresponds to line length in meters. If you set the NIVNA_ATTR_TRL_CALIBRATION_STANDARD_TYPE attribute to NIVNA_VAL_TRL_CALIBRATION_STANDARD_TYPE_OPEN or NIVNA_VAL_TRL_CALIBRATION_STANDARD_TYPE_SHORT, the NIVNA_ATTR_TRL_CALIBRATION_STANDARD_LENGTH attribute corresponds to the offset.

Note NI-VNA ignores this attribute if you set the NIVNA_ATTR_TRL_CALIBRATION_STANDARD_TYPE attribute to NIVNA_VAL_TRL_CALIBRATION_STANDARD_TYPE_MATCH.

Units: meters

Default Value: 0

Supported Devices: NI 5632

NIVNA_ATTR_TRL_CALIBRATION_STANDARD_LINE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Standard	None	None

Description

Specifies the line loss of the active standard.

The line loss is proportional to the square root of (***f***/***f0***), where ***f0*** is the `NIVNA_ATTR_TRL_CALIBRATION_STANDARD_LINE_LOSS_FREQUENCY` attribute value and ***f*** is the frequency list defined by the **startFrequency** and **stopFrequency** parameters of the `niVNA_ConfigureFrequencyStartStop` function. If you set the line loss frequency to 0, NI-VNA assumes that the line loss is constant across frequencies. This attribute is valid only if you set the `NIVNA_ATTR_TRL_CALIBRATION_STANDARD_TYPE` attribute to `NIVNA_VAL_TRL_CALIBRATION_STANDARD_TYPE_LINE`.

Units: dB/mm

Default Value: 0

Supported Devices: NI 5632

NIVNA_ATTR_TRL_CALIBRATION_STANDARD_LINE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Standard	None	None

Description

Specifies the frequency at which the NIVNA_ATTR_TRL_CALIBRATION_STANDARD_LINE_LOSS attribute value is valid.

The line loss is proportional to the square root of (***f***/***f0***), where ***f0*** is the NIVNA_ATTR_TRL_CALIBRATION_STANDARD_LINE_LOSS_FREQUENCY attribute value and ***f*** is the frequency list defined by the **startFrequency** and **stopFrequency** parameters of the niVNA_ConfigureFrequencyStartStop function. If you set the line loss frequency to 0, NI-VNA assumes that the line loss is constant across frequencies. This attribute is valid only if you set the NIVNA_ATTR_TRL_CALIBRATION_STANDARD_TYPE attribute to NIVNA_VAL_TRL_CALIBRATION_STANDARD_TYPE_LINE.

Units: Hz

Default Value: 0

Supported Devices: NI 5632

NIVNA_ATTR_TRL_CALIBRATION_BAND_STANDAR

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	Band	None	None

Description

Specifies the list of TRL calibration standards to include in a TRL calibration band.

Default Value: " "

Valid Values: "standard1", "standard2", "standard3", "standard4", "standard5", or "standard6"

Supported Devices: NI 5632

NIVNA_ATTR_TRL_CALIBRATION_BAND_BREAK_P

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	Band	None	None

Description

Specifies whether the band breakpoint for TRL calibration is calculated automatically or specified manually using the `NIVNA_ATTR_TRL_CALIBRATION_BAND_BREAK_POINT_FREQUENCY` attribute.

The band breakpoint is the frequency at which the TRL calibration stops using the standards associated with one frequency band and begins using the standards associated with the next band.

Defined Values:

`NIVNA_VAL_TRL_CALIBRATION_MANUAL_BAND_BREAK_POINT` (0)	The band breakpoint is manually specified.
`NIVNA_VAL_TRL_CALIBRATION_AUTO_BAND_BREAK_POINT` (1)	The band breakpoint is calculated automatically.

Default Value: `NIVNA_VAL_TRL_CALIBRATION_AUTO_BAND_BREAK_POINT`

Supported Devices: NI 5632

NIVNA_ATTR_TRL_CALIBRATION_BAND_BREAK_P

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Band	None	None

Description

Specifies the manual frequency band breakpoint at which the TRL calibration stops using the standards associated with this frequency band and uses the subsequent band standards instead.

Writing to this value specifies the manual band breakpoint frequency. Reading from this value returns the automatically calculated value when you set the NIVNA_ATTR_TRL_CALIBRATION_BAND_BREAK_POINT_TYPE attribute to NIVNA_VAL_TRL_CALIBRATION_AUTO_BAND_BREAK_POINT and returns the manually specified value when you set the NIVNA_ATTR_TRL_CALIBRATION_BAND_BREAK_POINT_TYPE attribute to NIVNA_VAL_TRL_CALIBRATION_MANUAL_BAND_BREAK_POINT.

Units: Hz

Default Value: 2 GHz

Supported Devices: NI 5632

NIVNA_ATTR_CALIBRATION_FILE_INTENDED_FOR

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	N/A	None	None

Description

Specifies whether the created calibration file is optimized for use with the NI-VNA soft front panel (SFP).

Default Value: `VI_FALSE`

Supported Devices: NI 5630/5632

NIVNA_ATTR_EXT_CAL_YEAR

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	RO	N/A	None	None

Description

Returns the year of the last external calibration.

Supported Devices: NI 5630/5632

NIVNA_ATTR_EXT_CAL_MONTH

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	RO	N/A	None	None

Description

Returns the month of the last external calibration. The returned value is expressed as an integer. For example, December is represented as 12.

Supported Devices: NI 5630/5632

NIVNA_ATTR_EXT_CAL_DAY

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	RO	N/A	None	None

Description

Returns the day of the last external calibration.

Supported Devices: NI 5630/5632

NIVNA_ATTR_EXT_CAL_HOUR

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	RO	N/A	None	None

Description

Returns the hour of the last external calibration. The time returned is 24-hour (military) local time; for example, if the device was calibrated at 2:00 PM, this function returns 14 as the value of this attribute.

Supported Devices: NI 5630/5632

NIVNA_ATTR_EXT_CAL_MINUTE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	RO	N/A	None	None

Description

Returns the minute of the last external calibration.

Supported Devices: NI 5630/5632

NIVNA_ATTR_EXT_CAL_TEMPERATURE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	RO	N/A	None	None

Description

Returns the temperature of the last external calibration.

Units: degrees Celsius

Supported Devices: NI 5630/5632

NIVNA_ATTR_EXT_CAL_INTERVAL

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	RO	N/A	None	None

Description

Returns the interval since the last external calibration. Complete the steps in the external calibration procedure for the device if the specified interval has elapsed since the last external calibration date.

Units: months

Supported Devices: NI 5630/5632

NIVNA_ATTR_USER_CAL_YEAR

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	RO	N/A	None	None

Description

Returns the year of the last user calibration.

Supported Devices: NI 5630/5632

NIVNA_ATTR_USER_CAL_MONTH

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	RO	N/A	None	None

Description

Returns the month of the last user calibration. It is expressed as an integer. For example, December is represented as 12.

Supported Devices: NI 5630/5632

NIVNA_ATTR_USER_CAL_DAY

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	RO	N/A	None	None

Description

Returns the day of the last user calibration.

Supported Devices: NI 5630/5632

NIVNA_ATTR_USER_CAL_HOUR

Specific Attribute

Data	Access	Applies to	Coercion	High Level Functions

type				
ViInt32	RO	N/A	None	None

Description

Returns the hour of the last user calibration.

Supported Devices: NI 5630/5632

NIVNA_ATTR_USER_CAL_MINUTE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	RO	N/A	None	None

Description

Returns the minute of the last user calibration.

Supported Devices: NI 5630/5632

NIVNA_ATTR_USER_CAL_TEMPERATURE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	RO	N/A	None	None

Description

Returns the temperature at which the last user calibration was performed.

Supported Devices: NI 5630/5632

NIVNA_ATTR_TIME_DOMAIN_TYPE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	Channel	None	None

Description

Specifies the type of time-domain analysis.

Both lowpass types require the frequencies to be linearly spaced (harmonic frequencies). Lowpass types provide higher time or distance resolution and also extra information as to the type of impedance mismatch (inductive, resistive, or capacitive) encountered in the DUT. When either of the lowpass modes is selected, NI-VNA does not automatically recalculate and change the frequency to be harmonic.

`NIVNA_VAL_TIME_DOMAIN_TYPE_BANDPASS_IMPULSE` (0)	The device performs a time-domain bandpass analysis with simulated stimulus-impulse response. Select this option for general-purpose operation when making time-domain measurements. This option allows you to use any frequency range and is a good fit for measuring band-limited devices, such as bandpass filters.
`NIVNA_VAL_TIME_DOMAIN_TYPE_LOWPASS_IMPULSE` (1)	The device performs a time-domain lowpass analysis with simulated stimulus-impulse response.
`NIVNA_VAL_TIME_DOMAIN_TYPE_LOWPASS_STEP` (2)	The device performs a time-domain lowpass analysis with simulated stimulus step response.
`NIVNA_VAL_TIME_DOMAIN_TYPE_BANDPASS_PHASOR_IMP`	The device performs a time-domain bandpass analysis with simulated phasor stimulus-impulse response.

ULSE (3)	

Supported Devices: NI 5630/5632

NIVNA_ATTR_TIME_DOMAIN_TRIP_TYPE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	Channel	None	None

Description

Specifies the trip type for time-domain analysis. Trip type refers to the path the signal takes through the DUT. Setting this attribute determines how the frequency domain data is interpreted as time-domain data.

This attribute is valid only if you have activated the Time Domain option on your NI RF vector network analyzer.

NIVNA_VAL_TIME_DOMAIN_TRIP_AUTO (0)	The device determines the trip type based on the S-parameter being measured, and it generates one-way times for both reflection and transmission.
NIVNA_VAL_TIME_DOMAIN_TRIP_ROUND_TRIP (1)	The device interprets the frequency domain data as traveling on a round-trip path, which means the signal is transmitted to and received from the DUT. Select NIVNA_VAL_TIME_DOMAIN_TRIP_ROUND_TRIP for reflective measurements.
NIVNA_VAL_TIME_DOMAIN_TRIP_ONE_WAY (2)	The device interprets the frequency domain data as traveling on a one-way path, which means the signal as it is transmitted through the DUT. Select NIVNA_VAL_TIME_DOMAIN_TRIP_ONE_WAY for transmission measurements.

Default Value: NIVNA_VAL_TIME_DOMAIN_TRIP_AUTO

Supported Devices: NI 5630/5632

NIVNA_ATTR_TIME_DOMAIN_IMPULSE_WIDTH_3D

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	RO	Channel	None	None

Description

Returns the calculated impulse width from time-domain analysis. This width implies the resolution of the time-domain analysis calculations.

This attribute is valid only if you have activated the Time Domain option on your NI RF vector network analyzer and set the `NIVNA_ATTR_TIME_DOMAIN_TYPE` attribute to either `NIVNA_VAL_TIME_DOMAIN_TYPE_LOWPASS_IMPULSE` or `NIVNA_VAL_TIME_DOMAIN_TYPE_BANDPASS_IMPULSE`.

Supported Devices: NI 5630/5632

NIVNA_ATTR_TIME_DOMAIN_ALIAS_FREE_RANGE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	RO	Channel	None	None

Description

Returns the calculated alias-free range from the time-domain analysis. This value is calculated after each change to the frequency points.

This attribute is valid only if you have activated the Time Domain option on your NI RF vector network analyzer.

Supported Devices: NI 5630/5632

NIVNA_ATTR_TIME_DOMAIN_UNITS

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	Channel	None	None

Description

Specifies the units for time-domain analysis.

This attribute is valid only if you have activated the Time Domain option on your NI RF vector network analyzer.

`NIVNA_VAL_TIME_DOMAIN_UNITS_TIME` (0)	The time-domain analysis is interpreted in terms of time.
`NIVNA_VAL_TIME_DOMAIN_UNITS_DISTANCE` (1)	The time-domain analysis is interpreted in terms of distance.

Supported Devices: NI 5630/5632

NIVNA_ATTR_TIME_DOMAIN_START

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Channel	None	`niVNA_ConfigureTimeDomainStartStopDistance` `niVNA_ConfigureTimeDomainStartStopTime`

Description

Specifies the start point for the time-domain analysis.

This attribute is valid only if you have activated the Time Domain option on your NI RF vector network analyzer.

Units: Seconds (s) or meters (m) depending on the NIVNA_ATTR_TIME_DOMAIN_UNITS attribute value.

Supported Devices: NI 5630/5632

NIVNA_ATTR_TIME_DOMAIN_STOP

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Channel	None	niVNA_ConfigureTimeDomainStartStopDistance niVNA_ConfigureTimeDomainStartStopTime

Description

Specifies the stop point for the time-domain analysis.

This attribute is valid only if you have activated the Time Domain option on your NI RF vector network analyzer.

Units: Seconds (s) or meters (m) depending on the NIVNA_ATTR_TIME_DOMAIN_UNITS attribute value.

Supported Devices: NI 5630/5632

NIVNA_ATTR_TIME_DOMAIN_WINDOW_SHAPE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	Channel	None	None

Description

Specifies the window shape for time-domain analysis.

This attribute is valid only if you have activated the Time Domain option on your NI RF vector network analyzer and set the NIVNA_ATTR_TIME_DOMAIN_TYPE attribute to either of the lowpass types.

NIVNA_VAL_TIME_DOMAIN_WINDOW_RECTANGULAR (0)	NI-VNA uses a rectangular window for time-domain analysis. This window provides the maximum resolution but the highest side lobes.
NIVNA_VAL_TIME_DOMAIN_WINDOW_HAMMING (1)	NI-VNA uses a Hamming window for time-domain analysis. This window provides moderate side lobes and resolution. This value is the default.
NIVNA_VAL_TIME_DOMAIN_WINDOW_BLACKMAN (2)	NI-VNA uses a Blackman window for time-domain analysis. This window provides low side lobes and low resolution.
NIVNA_VAL_TIME_DOMAIN_WINDOW_BLACKMAN_HARRIS (3)	NI-VNA uses a Blackman-Harris window for time-domain analysis. This window provides the minimum side lobes, but low resolution.
NIVNA_VAL_TIME_DOMAIN_WINDOW_KAISERBESSEL (4)	NI-VNA uses a Kaiser-Bessel window for time-domain analysis. When you select this value, you can adjust the window using the NIVNA_ATTR_TIME_DOMAIN_WINDOW_KAISER_BESSEL_BETA attribute.
NIVNA_VAL_TIME_DOMAIN_WINDOW_DOLPHCHEBYSHE	NI-VNA uses a Dolph-Chebyshev window for time-domain analysis. When you select this value, you can adjust the window using the NIVNA_ATTR_TIME_DOMAIN_WINDOW_DOLPH_CHEBYSHEV_GAMMA attribute.

V (5)	

Supported Devices: NI 5630/5632

NIVNA_ATTR_TIME_DOMAIN_WINDOW_KAISER_B

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Channel	None	None

Description

Specifies the Kaiser-Bessel window beta value for time-domain analysis.

Note If you want a window with a lower side lobe, use a wider gate, and vice versa.

This attribute is valid only if you have activated the Time Domain option on your NI RF vector network analyzer, set the `NIVNA_ATTR_TIME_DOMAIN_TYPE` attribute to either of the lowpass types, and set the `NIVNA_ATTR_TIME_DOMAIN_WINDOW_SHAPE` attribute value to `NIVNA_VAL_TIME_DOMAIN_WINDOW_KAISERBESSEL`.

Supported Devices: NI 5630/5632

NIVNA_ATTR_TIME_DOMAIN_WINDOW_DOLPH_C

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Channel	None	None

Description

Specifies the Dolph-Chebyshev window gamma value for time-domain analysis.

Note If you want a window with a lower side lobe, use a wider gate, and vice versa.

This attribute is valid only if you have activated the Time Domain option on your NI RF vector network analyzer, set the NIVNA_ATTR_TIME_DOMAIN_TYPE attribute to either of the lowpass types, and set the NIVNA_ATTR_TIME_DOMAIN_WINDOW_SHAPE attribute value to NIVNA_VAL_TIME_DOMAIN_WINDOW_DOLPHCHEBYSHEV.

Supported Devices: NI 5630/5632

NIVNA_ATTR_TIME_DOMAIN_GATE_SHAPE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	Channel	None	None

Description

Specifies the gate shape for the time-domain analysis.

This attribute is valid only if you have activated the Time Domain option on your NI RF vector network analyzer.

NIVNA_VAL_TIME_DOMAIN_GATE_SHAPE_RECTANGULAR (0)	NI-VNA uses a rectangular shape for the time-domain analysis gate. This gate shape provides the maximum resolution but the highest side lobes.
NIVNA_VAL_TIME_DOMAIN_GATE_SHAPE_HAMMING (1)	NI-VNA uses a Hamming shape for the time-domain analysis gate. This gate shape provides moderate side lobes and resolution. This value is the

	default.
NIVNA_VAL_TIME_DOMAIN_GATE_SHAPE_BLACKMAN (2)	NI-VNA uses a Blackman shape for the time-domain analysis gate. This gate shape provides low side lobes and low resolution.
NIVNA_VAL_TIME_DOMAIN_GATE_SHAPE_BLACKMAN_HARRIS (3)	NI-VNA uses a Blackman-Harris shape for the time-domain analysis gate. This gate shape provides the minimum side lobes, but low resolution.
NIVNA_VAL_TIME_DOMAIN_GATE_SHAPE_KAISERBESSEL (4)	NI-VNA uses a Kaiser-Bessel shape for the time-domain analysis gate. When you select this value, you can adjust the window using the NIVNA_ATTR_TIME_DOMAIN_GATE_KAISER_BESSEL_BETA attribute.
NIVNA_VAL_TIME_DOMAIN_GATE_SHAPE_DOLPHCHEBYSHEV (5)	NI-VNA uses a Dolph-Chebyshev shape for the time-domain analysis gate. When you select this value, you can adjust the window using the NIVNA_ATTR_TIME_DOMAIN_GATE_DOLPH_CHEBYSHEV_GAMMA attribute.

Supported Devices: NI 5630/5632

NIVNA_ATTR_TIME_DOMAIN_GATE_ENABLE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	Channel	None	None

Description

Specifies whether the gate is enabled for the time-domain analysis.

This attribute is valid only if you have activated the Time Domain option on your NI RF vector network analyzer.

Supported Devices: NI 5630/5632

NIVNA_ATTR_TIME_DOMAIN_GATE_INCLUSIVE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	Channel	None	None

Description

Specifies the scope of the gated time/distance span. If you set this attribute to `VI_TRUE`, the data outside the gated span is suppressed. If you set this attribute to `VI_FALSE`, the data inside the gated span is suppressed.

This attribute is valid only if you have activated the Time Domain option on your NI RF vector network analyzer.

Default Value: `VI_TRUE`

Supported Devices: NI 5630/5632

NIVNA_ATTR_TIME_DOMAIN_GATE_START

NIVNA_ATTR_TIME_DOMAIN_GATE_START

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Channel	None	`niVNA_ConfigureTimeDomainGateStartStopDis`

				tance niVNA_ConfigureTimeDomainGateStartStopTime

Description

Specifies the starting point for the time domain gate.

This attribute is valid only if you have activated the Time Domain option on your NI RF vector network analyzer.

Units: Seconds (s) or meters (m)

Supported Devices: NI 5630/5632

NIVNA_ATTR_TIME_DOMAIN_GATE_STOP

NIVNA_ATTR_TIME_DOMAIN_GATE_STOP

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Channel	None	niVNA_ConfigureTimeDomainGateStartStopDistance niVNA_ConfigureTimeDomainGateStartStopTime

Description

Specifies the stopping point for the time domain gate.

This attribute is valid only if you have activated the Time Domain option on your NI RF vector network analyzer.

Units: Seconds (s) or meters (m)

Supported Devices: NI 5630/5632

NIVNA_ATTR_TIME_DOMAIN_GATE_KAISER_BESS

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Channel	None	None

Description

Specifies the Kaiser-Bessel beta value for the time-domain analysis gate.



Note If you use a window with a lower side lobe, use a wider gate, and vice versa.

This attribute is valid only if you have activated the Time Domain option on your NI RF vector network analyzer, set the `NIVNA_ATTR_TIME_DOMAIN_TYPE` attribute to either of the lowpass types, and set the `NIVNA_ATTR_TIME_DOMAIN_GATE_SHAPE` attribute value to `NIVNA_VAL_TIME_DOMAIN_GATE_SHAPE_KAISERBESSEL`.

Supported Devices: NI 5630/5632

NIVNA_ATTR_TIME_DOMAIN_GATE_DOLPH_CHEE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Channel	None	None

Description

Specifies the Dolph-Chebyshev gamma value for the time-domain analysis gate.

	Note If you use a window with a lower side lobe, use a wider gate, and vice versa.

This attribute is valid only if you have activated the Time Domain option on your NI RF vector network analyzer, set the NIVNA_ATTR_TIME_DOMAIN_TYPE attribute to either of the lowpass types, and set the NIVNA_ATTR_TIME_DOMAIN_GATE_SHAPE attribute value to NIVNA_VAL_TIME_DOMAIN_GATE_SHAPE_DOLPHCHEBYSHEV.

Supported Devices: NI 5630/5632

NIVNA_ATTR_TIME_DOMAIN_DC_TERM_EXTRAPC

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	Channel	None	None

Description

Specifies the DC term extrapolation type for lowpass time-domain calculations.

This attribute is valid only if you have activated the Time Domain option on your NI RF vector network analyzer and set the NIVNA_ATTR_TIME_DOMAIN_TYPE attribute to either of the lowpass types.

NIVNA_VAL_TIME_DOMAIN_DC_TERM_EXTRAPOLATION_AUTO (0)	NI-VNA automatically extrapolates the DC term from existing frequency data used in the time-domain analysis.
NIVNA_VAL_TIME_DOMAIN_DC_TERM_EXTRAPOLATION_USER (1)	If you select this value, you must supply the DC term used in the time-domain analysis with the NIVNA_ATTR_TIME_DOMAIN_DC_TERM_EXTRAPOLATION_USER_VALUE attribute.

Supported Devices: NI 5630/5632

NIVNA_ATTR_TIME_DOMAIN_DC_TERM_EXTRAPC

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	Channel	None	None

Description

Specifies the DC term extrapolation method for lowpass time-domain calculations.

This attribute is valid only if you have activated the Time Domain option on your NI RF vector network analyzer and set the `NIVNA_ATTR_TIME_DOMAIN_TYPE` attribute to either of the lowpass types.

`NIVNA_VAL_TIME_DOMAIN_DC_TERM_EXTRAPOLATION_METHOD_PHASE_ONLY` (0)	Assumes the magnitude remains constant, so only phase information is extrapolated. This option is commonly used when using long cables to connect the VNA to your DUT.
`NIVNA_VAL_TIME_DOMAIN_DC_TERM_EXTRAPOLATION_METHOD_LOG_MAG_AND_PHASE` (1)	Extrapolates both magnitude and phase information.

Supported Devices: NI 5630/5632

NIVNA_ATTR_TIME_DOMAIN_DC_TERM_EXTRAPC

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	RO	Channel	None	None

Description

Specifies the coefficient that is used for the DC term extrapolation calculations. This value corresponds with the DC impedance of the DUT.

This attribute is valid only if you have activated the Time Domain option on your NI RF vector network analyzer and set the `NIVNA_ATTR_TIME_DOMAIN_TYPE` attribute to either of the lowpass types.

Supported Devices: NI 5630/5632

NIVNA_ATTR_TIME_DOMAIN_DC_TERM_EXTRAPC

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Channel	None	None

Description

Specifies the DC term value for lowpass time-domain calculations.

This attribute is valid only if you have activated the Time Domain option on your NI RF vector network analyzer and set the `NIVNA_ATTR_TIME_DOMAIN_TYPE` attribute to either of the lowpass types.

Supported Devices: NI 5630/5632

NIVNA_ATTR_DEVICE_PXI_CHASSIS

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions

ViInt32	RO	N/A	None	None

Description

Returns the identifier for the PXI chassis in which the NI vector network analyzer is installed.

Supported Devices: NI 5630/5632

NIVNA_ATTR_DEVICE_PXI_SLOT

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	RO	N/A	None	None

Description

Returns the number for the PXI slot in which the NI vector network analyzer is installed.

Supported Devices: NI 5630/5632

NIVNA_ATTR_DEVICE_TEMPERATURE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	RO	N/A	None	None

Description

Returns the current temperature of the NI-VNA device.

Units: degrees Celsius

Supported Devices: NI 5630/5632

NIVNA_ATTR_DEVICE_SERIAL_NUMBER

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	RO	N/A	None	None

Description

Returns the serial number of the NI-VNA device.

Supported Devices: NI 5630/5632

NIVNA_ATTR_NUMBER_OF_PORTS

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	RO	N/A	None	None

Description

Returns the number of ports for the NI vector network analyzer device.

Supported Devices: NI 5630/5632

NIVNA_ATTR_CACHE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	N/A	None	None

Description

Specifies whether to cache the value of attributes. When caching is enabled, NI-VNA tracks the current NI-VNA device settings and avoids sending redundant commands to the device.

NI-VNA can always cache or never cache particular attributes, regardless of the setting of this attribute.

Use `niVNA_InitWithOptions` to override the default value.

Supported Devices: NI 5630/5632

NIVNA_ATTR_INTERCHANGE_CHECK

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	N/A	None	None

Description

Specifies whether to perform interchangeability checking and retrieve interchangeability warnings.

	Note Interchangeability check is unsupported.

Supported Devices: NI 5630/5632

NIVNA_ATTR_DRIVER_SETUP

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	RO	N/A	None	None

Description

The DriverSetup string is used to set the initial values for attributes that are specific to NI-VNA. The format of the DriverSetup string is: ***Tag: Value***. ***Tag*** is the name of the ***DriverSetup*** string attribute. ***Value*** is the value set to the attribute. To set multiple attributes, separate their assignments with a semicolon.

Supported Devices: NI 5630/5632

NIVNA_ATTR_QUERY_INSTRUMENT_STATUS

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	N/A	None	None

Description

Specifies whether NI-VNA queries the NI-VNA device status after each operation. Querying the device status is useful for debugging. After you validate your program, you can set this attribute to `VI_FALSE` to disable status checking and maximize

performance.

NI-VNA can choose to ignore status checking for particular attributes, regardless of the setting of this attribute.



Note Use `niVNA_InitWithOptions` function to override this value.

Supported Devices: NI 5630/5632

NIVNA_ATTR_RANGE_CHECK

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	N/A	None	None

Description

Specifies whether to validate attribute values and function parameters. If enabled, NI-VNA validates the parameter values that you pass to NI-VNA functions. Range checking parameters is very useful for debugging. After you validate your program, you can set this attribute to VI_FALSE to disable range checking and maximize performance.



Note Use `niVNA_InitWithOptions` function to override this value.

Supported Devices: NI 5630/5632

NIVNA_ATTR_RECORD_COERCIONS

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions

ViBoolean	R/W	N/A	None	None

Description

Specifies whether the IVI engine keeps a list of the value coercions it makes for integer and real type attributes.

	**Note** This attribute is currently not supported.

Supported Devices: NI 5630/5632

NIVNA_ATTR_SIMULATE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViBoolean	RO	N/A	None	None

Description

Specifies whether or not to simulate NI-VNA I/O operations. This attribute is useful for debugging applications without using hardware. After a session is opened, you cannot change the simulation state. Use the `niVNA_InitWithOptions` function to enable simulation.

	**Note** NI-VNA does not support simulation.

Supported Devices: NI 5630/5632

NIVNA_ATTR_SPECIFIC_DRIVER_DESCRIPTION

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	RO	N/A	None	None

Description

Returns a string that contains a brief description of NI-VNA.

Supported Devices: NI 5630/5632

NIVNA_ATTR_SPECIFIC_DRIVER_PREFIX

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	RO	N/A	None	None

Description

Returns a string that contains the prefix for NI-VNA. The name of each user-callable function in NI-VNA starts with this prefix.

Supported Devices: NI 5630/5632

NIVNA_ATTR_SPECIFIC_DRIVER_VENDOR

Specific Attribute

Data	Access	Applies to	Coercion	High Level Functions

type				
ViString	RO	N/A	None	None

Description

Returns a string that contains the name of the vendor that supplies NI-VNA.

Supported Devices: NI 5630/5632

NIVNA_ATTR_SPECIFIC_DRIVER_REVISION

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	RO	N/A	None	None

Description

Returns a string that contains additional version information about NI-VNA.

Supported Devices: NI 5630/5632

NIVNA_ATTR_SPECIFIC_DRIVER_CLASS_SPEC_MA

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	RO	N/A	None	None

Description

Returns the major version number of the class specification with which NI-VNA is

compliant.

Supported Devices: NI 5630/5632

NIVNA_ATTR_SPECIFIC_DRIVER_CLASS_SPEC_MI

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	RO	N/A	None	None

Description

Returns the minor version number of the class specification with which NI-VNA is compliant.

Supported Devices: NI 5630/5632

NIVNA_ATTR_GROUP_CAPABILITIES

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	RO	N/A	None	None

Description

Returns a string that contains a comma-separated list of class-extension groups that NI-VNA implements.

Supported Devices: NI 5630/5632

NIVNA_ATTR_SUPPORTED_INSTRUMENT_MODEL

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	RO	N/A	None	None

Description

Returns a comma-separated list of supported devices.

Supported Devices: NI 5630/5632

NIVNA_ATTR_INSTRUMENT_MANUFACTURER

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	RO	N/A	None	None

Description

Returns a string that contains the name of the manufacturer for the NI-VNA device you are currently using.

Supported Devices: NI 5630/5632

NIVNA_ATTR_INSTRUMENT_MODEL

Specific Attribute

Data	Access	Applies to	Coercion	High Level Functions

type				
ViString	RO	N/A	None	None

Description

Returns a string that contains the model number or name of the NI-VNA device that you are currently using.

Supported Devices: NI 5630/5632

NIVNA_ATTR_INSTRUMENT_FIRMWARE_REVISION

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	RO	N/A	None	None

Description

Returns a string that contains the firmware revision information for the NI-VNA device you are currently using.

Supported Devices: NI 5630/5632

NIVNA_ATTR_LOGICAL_NAME

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	RO	N/A	None	None

Description

Returns the logical name you specified when opening the current IVI session. You may pass a logical name to `niVNA_init` or the `niVNA_InitWithOptions` function. The IVI Configuration Utility must contain an entry for the logical name. The logical name entry refers to a driver session section in the IVI Configuration file. The driver session section specifies a physical device and initial user options.

Supported Devices: NI 5630/5632

NIVNA_ATTR_IO_RESOURCE_DESCRIPTOR

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	RO	N/A	None	None

Description

Indicates the resource name NI-VNA uses to identify the physical device. If you initialize NI-VNA with a logical name, this attribute contains the resource name that corresponds to the entry in the IVI Configuration Utility.

If you initialize NI-VNA with the resource name, this attribute contains that value.

Supported Devices: NI 5630/5632

NIVNA_ATTR_SOURCE_DRIVING_STATE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	Port	None	None

Description

Specifies whether the port source is active or inactive.

Defined Values:

NIVNA_VAL_SOURCE_DRIVING_AUTOMATIC (0)	The port source is automatically activated or deactivated during a sweep, depending on the S-parameters measured during the sweep.
NIVNA_VAL_SOURCE_DRIVING_ACTIVE (1)	The port source is active during a sweep.
NIVNA_VAL_SOURCE_DRIVING_INACTIVE (2)	The port source is inactive during a sweep.

Default Value: 0

Supported Devices: NI 5632

NIVNA_ATTR_SOURCE_FREQUENCY_MULTIPLIER

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Port	None	niVNA_ConfigureFrequencyOffsetEquation

Description

Specifies the multiplier value in the source frequency offset equation.

The frequency offset equation is (***M*** * ***f***) + ***b***, where ***M*** is the multiplier, ***f*** is the frequency list defined by the start and stop frequency properties, and ***b*** is the offset defined by the NIVNA_ATTR_SOURCE_FREQUENCY_OFFSET attribute.

Note If you specify values for ***M*** and ***b*** that, after calculation, result in a value that is out of the frequency range, NI-VNA returns an error.

Default Value: 1

Supported Devices: NI 5632

NIVNA_ATTR_SOURCE_FREQUENCY_OFFSET

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Port	None	`niVNA_ConfigureFrequencyOffsetEquation`

Description

Specifies the offset value in the source frequency offset equation.

The frequency offset equation is (***M*** * ***f***) + ***b***, where ***M*** is the multiplier, ***f*** is the frequency list defined by the start and stop frequency properties, and ***b*** is the offset.

Note If you specify values for ***M*** and ***b*** that, after calculation, result in a value that is out of the frequency range, NI-VNA returns an error.

Default Value: 0

Supported Devices: NI 5632

NIVNA_ATTR_RECEIVER_FREQUENCY_MULTIPLIER

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions

ViReal64	R/W	Channel	None	niVNA_ConfigureFrequencyOffsetEquation

Description

Specifies the multiplier value in the receiver frequency offset equation.

The frequency offset equation is (***M*** * ***f***) + ***b***, where ***M*** is the multiplier, ***f*** is the frequency list defined by the start and stop frequency properties, and ***b*** is the offset defined by the NIVNA_ATTR_RECEIVER_FREQUENCY_OFFSET attribute.

Note If you specify values for ***M*** and ***b*** that, after calculation, result in a value that is out of the frequency range, NI-VNA returns an error.

Default Value: 1

Supported Devices: NI 5632

NIVNA_ATTR_RECEIVER_FREQUENCY_OFFSET

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Channel	None	niVNA_ConfigureFrequencyOffsetEquation

Description

Specifies the offset value in the receiver frequency offset equation.

The frequency offset equation is (***M*** * ***f***) + ***b***, where ***M*** is the multiplier, ***f*** is the frequency list defined by the start and stop frequency properties, and ***b*** is the offset.

Note If you specify values for ***M*** and ***b*** that, after calculation, result in a value that is out of the frequency range, NI-VNA returns an error.

Default Value: 0

Supported Devices: NI 5632

© 2025 National Instruments Corporation.