



# S5105 Series Microwave Multifunctional Analyzer Programming Manual



Saluki Technology Inc.

## The manual applies to the microwave analyzer of the following models:

- S5105D Microwave Multifunctional Analyzer (30kHz - 18GHz)
- S5105E Microwave Multifunctional Analyzer (30kHz - 26.5GHz)
- S5105F Microwave Multifunctional Analyzer (30kHz - 40GHz)

## Standard pack and accessories:

Part No.	Name
1	Main machine
2	Standard three-core power cord
3	Power adapter
4	Rechargeable lithium-ion battery
5	CD (user manual, programming manual, USB driver, program-controlled function library, program-controlled example, and installation file required for program-controlled function library)
6	Certificate of Conformity

## Options of the S5105 series microwave analyzer in addition to standard accessories:

Part No.	Name	Description
S5105-S02	Antenna Test (Software)	For testing RL, VSWR, Break point of Cable and Antenna.
S5105-S03	Vector Voltmeter (Software)	For testing cable phase shift and electrical length.
S5105-S04	USB Power Measurements (Software)	External USB Power probe can conduct precise measurement of continuous wave signal. (Need to reprovision USB power sensors)
S5105-S05	Power Detection (Software)	Receiving external signal at spectrum input port in order to measure signal power.
S5105-S06	Field Strength Measurements (Software)	For field strength measurements. (Need to reprovision antenna)
S5105-S07	GPS Positioning (Software)	Provide geographical information such as longitude, latitude and altitude. (Including GPS antenna)
S5105-S08	Electronic Calibration	Software, should buy e-cal module separately (For VNA, C&A, Vector Voltmeter function)
S5105-H01	Rechargeable Lithium-ion Battery	Backup battery
S5105-H02	AC-DC Adapter	Backup
S5105-H04	S31101A N-type male Calibration Kit, DC - 18GHz	Calibrate for Vector Network Analysis, Antenna Test and Vector
S5105-H05	S31101B N-type Female Calibration Kit, DC - 18GHz	
S5105-H06	S31121 3.5mm Calibration Kit, DC - 26.5GHz	
S5105-H07	S31123 2.4mm Calibration Kit, DC - 40GHz	
S5105-H08	N (M-M) Calibration Cable	Calibration or Cable Test
S5105-H09	N (F-M) Calibration Cable	

Part No.	Name	Description
S5105-H10	3.5mm (F-F) Calibration Cable	Calibration or Cable Test
S5105-H11	3.5mm (F-M) Calibration Cable	
S5105-H12	2.4mm (F-F) Calibration Cable	
S5105-H13	2.4mm (F-M) Calibration Cable	
S5105-H14	S87230 USB Power Sensor, 9kHz-6GHz	For High-precision Power Measurement
S5105-H15	S87231 USB Power Sensor, 10MHz-18GHz	
S5105-H16	S87232 USB Power Sensor, 50MHz-26.5GHz	
S5105-H17	S87233 USB Power Sensor, 50MHz-40GHz	
S5105-H18	S89101A Antenna, 10kHz-20MHz	For Field Strength Measurement
S5105-H19	S89101B Antenna, 20MHz-200MHz	
S5105-H20	S89101C Antenna, 200MHz-500MHz	
S5105-H21	S89101D Antenna, 500MHz-4000MHz	
S5105-H22	S89901 Antenna, 1GHz-18GHz	
S5105-H23	S89401 Antenna Amplifier, 10kHz-4GHz	
S5105-H24	S71522D Attenuator (40dB, 25W)	For High Power Measurement
S5105-H25	S71523C Attenuator (40dB, 50W)	
S5105-H26	S71524C Attenuator (40dB, 100W)	
S5105-H27	S71101 Adapter, N(F)-N(F)	For Switching Between Connectors
S5105-H28	S71115 Adapter, 3.5mm(M)-N(F)	
S5105-H29	S71116 Adapter, 3.5mm(M)-N(M)	
S5105-H30	S71117 Adapter, 3.5mm(F)-N(M)	
S5105-H31	S81101 Adapter, N(M)-N(F)	
S5105-H32	Soft Backpack	For Carrying
S5105-H33	Aluminum Carrying Case	For Transportation
S5105-H34	Waterproof Safety Box	For Transportation
S5105-H35	S89901 Antenna Handle	Used with option H22 antenna
S5105-H37	S20403 Electrical Calibration Kit	Frequency: 10MHz - 26.5GHz
S5105-H38	S20404 Electrical Calibration Kit	Frequency: 10MHz - 50GHz
S5105-H39	S87302FZ Test Cable	3.5/3.5-female to female, 60cm
S5105-H40	S87302FE Test Cable	3.5/3.5-female to male, 60cm
S5105-H41	S87302AZ Test Cable	N/N-male to male, 60cm

Part No.	Name	Description
S5105-H42	S87302BA Test Cable	N/N-female to male, 60cm
S5105-H43	S87234D USB Peak/Average Power Sensor	50MHz-18GHz, for peak power measurement
S5105-H44	S87234E USB Peak/Average Power Sensor	50MHz-26.5GHz, for peak power measurement
S5105-H45	S87234F USB Peak/Average Power Sensor	50MHz-40GHz, for peak power measurement

## Preface

Thanks for choosing S5105 Series Microwave Multifunctional Analyzer produced by Saluki Technology Inc. Please read this programming manual carefully for your convenience.

We devote ourselves to meeting your demands, providing you high-quality measuring instrument and the best after-sales service. We persist with “superior quality and considerate service”, and are committed to offering satisfactory products and service for our clients.

## Manual No.

S5105-03-02

## Version

Rev01 2019.12

Saluki Technology

## Manual Authorization

The information contained in this Programming Manual is subject to change without notice. The power to interpret the contents of and terms used in this Manual rests with Saluki.

Saluki Tech owns the copyright of this Programming Manual which should not be modified or tampered by any organization or individual, or reproduced or transmitted for the purpose of making profit without its prior permission, otherwise Saluki will reserve the right to investigate and affix legal liability of infringement.

## Product Quality Certificate

The product meets the indicator requirements of the manual at the time of delivery. Calibration and measurement are completed by the measuring organization with qualifications specified by the state, and relevant data are provided for reference.

## Quality/Environment Management

Research, development, manufacturing and testing of the product comply with the requirements of the quality and environmental management system.

## Contacts

Service Tel:	886.909 602 109
Website:	<a href="http://www.salukitec.com">www.salukitec.com</a>
Email:	<a href="mailto:sales@salukitec.com">sales@salukitec.com</a>
Address:	No. 367 Fuxing N Road, Taipei 105, Taiwan (R.O.C.)

## Contents

Chapter I Description of SCPI Commands.....	1
Instrument Connection.....	1
IEEE 488.2 Common Commands.....	1
:CALCulate:PARAmeter:DEFine <string>.....	2
:CALCulate:RELative[:MAGNitude]?.....	2
:CALCulate:RELative[:MAGNitude]:AUTO <string>.....	2
:CALCulate[:SElected]:DATA:FDATA?.....	3
:CALCulate[:SElected]:DATA:FMEM?.....	3
:CALCulate[:SElected]:DATA:SDATA?.....	3
:CALCulate[:SElected]:DATA:SMEM?.....	3
:CALCulate[:SElected]:FORMat <string>.....	4
:CALCulate[:SElected]:LIMit:BEEP <string>.....	4
:CALCulate[:SElected]:LIMit:EDIT:ADD.....	4
:CALCulate[:SElected]:LIMit:EDIT:SElect <int>.....	5
:CALCulate[:SElected]:LIMit:EDIT:DElete.....	5
:CALCulate[:SElected]:LIMit:EDIT:CLEar.....	5
:CALCulate[:SElected]:LIMit:POINts?.....	5
:CALCulate[:SElected]:LIMit:EDIT:X <double>.....	5
:CALCulate[:SElected]:LIMit:EDIT:Y <double>.....	6
:CALCulate[:SElected]:LIMit:TEST <string>.....	6
:CALCulate[:SElected]:LIMit:PASS?.....	6
:CALCulate[:SElected]:MARKer{n}:[STATe] <string>.....	6
:CALCulate[:SElected]:MARKer:AOFF.....	7
:CALCulate[:SElected]:MARKer{n}:FCOunt[:STATe] <string>.....	7
:CALCulate[:SElected]:MARKer:FCOunt:X?.....	7
:CALCulate[:SElected]:MARKer{n}:FUNctIon:MAXimum.....	7
:CALCulate[:SElected]:MARKer{n}:FUNctIon:MINimum.....	7
:CALCulate[:SElected]:MARKer{n}:FUNctIon:PEAK.....	8
:CALCulate[:SElected]:MARKer{n}:FUNctIon:PLEft.....	8
:CALCulate[:SElected]:MARKer{n}:FUNctIon:PNEXt.....	8
:CALCulate[:SElected]:MARKer{n}:FUNctIon:PRIGHt.....	8
:CALCulate[:SElected]:MARKer{n}:NOISe <string>.....	9
:CALCulate[:SElected]:MARKer{n}:SET <string>.....	9
:CALCulate[:SElected]:MARKer{n}:X <double>.....	9
:CALCulate[:SElected]:MARKer{n}:Y?.....	9
:CALCulate[:SElected]:MATH:FUNctIon <string>.....	10
:CALCulate[:SElected]:MATH:MEMorize.....	10
:CALCulate[:SElected]:SMOothing:APERture <double>.....	10

:CALCulate[:SElected]:SMOothing[:STATe] <string>.....	11
:CALCulate[:SElected]:TRANSform:CLOSs <double>.....	11
:CALCulate[:SElected]:TRANSform:DISTance:STARt <double>.....	11
:CALCulate[:SElected]:TRANSform:DISTance:STOP <double>.....	11
:CALCulate[:SElected]:TRANSform:DISTance:UNIT <string>.....	11
:CALCulate[:SElected]:TRANSform:TIME:STARt <double>.....	12
:CALCulate[:SElected]:TRANSform:TIME:STATe <string>.....	12
:CALCulate[:SElected]:TRANSform:TIME:STOP <double>.....	12
:CALCulate[:SElected]:TRANSform:VFACtor <double>.....	12
:CALCulate[:SElected]:TRANSform:WINDow <string>.....	12
:CALibration:ZERO.....	13
:CALibration:ZERO:STATe? .....	13
:DISPlay:WINDow:ANALog:LOWer <double>.....	14
:DISPlay:WINDow:ANALog:UPPer <double>.....	14
:DISPlay:WINDow:TRACe:Y[:SCALE]:AUTO.....	14
:DISPlay:WINDow:TRACe:Y[:SCALE]:TOP <double>.....	14
:DISPlay:WINDow:TRACe:Y[:SCALE]:BOTTom <double>.....	14
:DISPlay:WINDow:TRACe:Y[:SCALE]:PDIVision <double>.....	15
:DISPlay:WINDow:TRACe:Y[:SCALE]:RLEVel <double>.....	15
:FORMat[:DATA] <string>.....	15
:INITiate:CONTinuous <string>.....	15
:INITiate[:IMMEDIATE].....	16
:INITiate:HOLD.....	16
:INSTrument:CATalog?.....	16
:INSTrument[:SElect] <string>.....	16
:MMEMory:DELeTe:STATe <string>.....	17
:MMEMory:DELeTe:STATe:ALL.....	17
:MMEMory:DELeTe:TRACe <string>.....	17
:MMEMory:DELeTe:TRACe:ALL.....	18
:MMEMory:LOAD:STATe <string>.....	18
:MMEMory:LOAD:TRACe <string>.....	18
:MMEMory:LOCation <string>.....	18
:MMEMory:STORe:SCReen <string>.....	19
:MMEMory:STORe:STATe <string>.....	19
:MMEMory:STORe:TRACe <string>.....	19
[:SENSe]:ACPR:BW:ADJChbw <double>.....	19
[:SENSe]:ACPR:BW:MAINchbw <double>.....	19
[:SENSe]:ACPR:CHPower:MAINChpower?.....	20
[:SENSe]:ACPR:CHPower:UPCHpower?.....	20
[:SENSe]:ACPR:CHPower: DOWNChpower?.....	20
[:SENSe]:ACPR:ACPR:UPCHacpr?.....	20

[[:SENSe]:ACPR:ACPR: DOWNchacpr?.....	20
[[:SENSe]:ACPR:BW:SPACE <double>.....	21
[[:SENSe]:ACPR[:STATe] <string>.....	21
[[:SENSe]:AMPLitude:ALIGNment:NOW.....	21
[[:SENSe]:FST:ANTenna:OFF.....	21
[[:SENSe]:FST [:STATe] <string>.....	21
[[:SENSe]:FST:ANTenna <string>.....	22
[[:SENSe]:AMPLitude:SCALE <string>.....	22
[[:SENSe]:AMPLitude:UNIT <string>.....	22
[[:SENSe]:AVERage:COUNT <int>.....	22
[[:SENSe]:AVERage:CLEar.....	23
[[:SENSe]:AVERage:STATe <string>.....	23
[[:SENSe]:BANDwidth:RESolution <double>.....	23
[[:SENSe]:BANDwidth:RESolution:AUTO <string>.....	23
[[:SENSe]:BANDwidth:RESolution:RATio <int>.....	23
[[:SENSe]:BANDwidth:VIDeo <double>.....	24
[[:SENSe]:BANDwidth:VIDeo:AUTO <string>.....	24
[[:SENSe]:BANDwidth:VIDeo:RATio <int>.....	24
[[:SENSe]:BANDwidth <double>.....	24
[[:SENSe]:CORRection:COLLect[:ACQuire]:ISO <string>.....	24
[[:SENSe]:CORRection:COLLect[:ACQuire]:LOAD{n}.....	25
[[:SENSe]:CORRection:COLLect[:ACQuire]:OPEN{n}.....	25
[[:SENSe]:CORRection:COLLect[:ACQuire]:SHORt{n}.....	25
[[:SENSe]:CORRection:COLLect[:ACQuire]:THUR <string>.....	26
[[:SENSe]:CORRection:COLLect[:ACQuire]:CKIT:LABel <string>.....	26
[[:SENSe]:CORRection:COLLect[:ACQuire]:CKIT:MATCh <string>.....	26
[[:SENSe]:CORRection:COLLect:METHod <string>.....	26
[[:SENSe]:CORRection:COLLect:DONE.....	27
[[:SENSe]:CORRection <string>.....	27
[[:SENSe]:CORRection:VALid.....	27
[[:SENSe]:CORRection:GAIN <double>.....	28
[[:SENSe]:CORRection:GAIN:STATe <string>.....	28
[[:SENSe]:CHPW:STATe <string>.....	28
[[:SENSe]:CHPW:CHBW <double>.....	28
[[:SENSe]:CHPW:TPWR?.....	28
[[:SENSe]:CHPW:PSDR?.....	29
[[:SENSe]:DATA?.....	29
[[:SENSe]:DETector:FUNCTion <string>.....	29
[[:SENSe]:DETector:FUNCTion:AUTO <string>.....	29
[[:SENSe]:FREQUency <double>.....	30
[[:SENSe]:FREQUency:CENTer <double>.....	30



[:SENSe]:FREQUency:SPAN <double>.....	30
[:SENSe]:FREQUency:SPAN:FULL.....	30
[:SENSe]:FREQUency:SPAN:PREVious.....	30
[:SENSe]:FREQUency:SPAN:ZERO.....	31
[:SENSe]:FREQUency:STARt <double>.....	31
[:SENSe]:FREQUency:STOP <double>.....	31
[:SENSe]:FREQUency:CW <double>.....	31
[:SENSe]:OBW:METHod <string>.....	31
[:SENSe]:OBW:OBW?.....	32
[:SENSe]:OBW:PPOW <double>.....	32
[:SENSe]:OBW[:STATe] <string>.....	32
[:SENSe]:OBW:XDB <double>.....	32
[:SENSe]:POWER[:RF]:ATTenuation <double>.....	32
[:SENSe]:POWER[:RF]:ATTenuation:AUTO <string>.....	33
[:SENSe]:POWER[:RF]:GAIN[:STATe] <string>.....	33
[:SENSe]:SWEep:POINts <int>.....	33
[:SENSe]:SWEep:TIME <double>.....	33
[:SENSe]:SWEep:TIME:AUTO <string>.....	33
[:SENSe]:IA[:STATe] <string>.....	34
[:SENSe]:IA:REDLimit < double>.....	34
[:SENSe]:IA:BLUelimit <double>.....	34
[:SENSe]:IA:MODE <string>.....	34
[:SENSe]:IA:CLEar.....	34
:SOURce:POWER:ALC:MAN <double>.....	35
:SOURce:POWER:ALC:MODE <string>.....	35
:SOURce:TYPE <string>.....	35
[:SYSTem]:GPS <string>.....	35
[:SYSTem]:GPS:DATA?.....	36
[:SYSTem]:GPS:RECeive[:STATe]?.....	36
[:SYSTem]:GPS:RST.....	36
[:SYSTem]:GPS:STATe?.....	36
:TRACe{n}:DATA?.....	37
:TRACe{n}:TYPE <string>.....	37
Chapter II Description of Secondary Development Library Functions.....	38
Section I Construction of the Development Environment.....	38
I Instrument Connection Commands.....	39
Starting the Instrument.....	39
Turning off the Device.....	39
II IEEE488.2 General Commands.....	39
QueryIDN.....	39
Reset.....	40

III Commands of the Math Sub-system.....	40
SetMeasTarget.....	40
QueryMeasTarget.....	41
QueryRefVal.....	41
SetRefOn.....	42
QueryRefOn.....	42
ReadCurTrace (Processed).....	42
ReadMemTrace (Processed).....	43
ReadCurTrace.....	43
ReadMemTrace.....	44
SetMeasFormat.....	44
QueryMeasFormat.....	45
SetAlarmOn.....	46
QueryAlarmOn.....	46
SeLmtTestOn.....	46
QueryLmtTestOn.....	47
QueryLmtPass.....	47
QueryLmtPtNum.....	47
LmtAddPt.....	48
LmtClear.....	48
LmtDelPt.....	48
LmtSelectPt.....	49
QueryLmtSelectPt.....	49
SetLmtPtX.....	49
QueryLmtPtX.....	50
SetLmtPtY.....	50
QueryLmtPtY.....	51
SetMkrState.....	51
QueryMkrState.....	51
SetMkrAOff.....	52
SetMkrCounterSwitch.....	52
QueryMkrCounterSwitch.....	53
QueryMkrCounterXValue.....	53
SearchMkrToMax.....	54
SearchMkrToMin.....	54
SearchMkrToPeak.....	54
SearchMkrToPeakLeft.....	55
SearchMkrToSubPeak.....	55
SearchMkrToPeakRight.....	55
SetMkrNoiseSwitch.....	56
QueryMkrNoiseSwitch.....	56

SetMkrTo (Marker ->)	57
SetMkrXVal	57
QueryMkrXVal	58
QueryMkrYVal	58
SetTraceMathFunc	59
QueryTraceMathFunc	59
TraceToMemory	60
SetSmoothAper	60
QuerySmoothAper	61
SetSmoothOn	61
QuerySmoothOn	61
SetCableLoss	62
QueryCableLoss	62
SetStartDist	62
QueryStartDist	63
SetStopDist	63
QueryStopDist	63
SetDTFUnit	64
QuerySetDTFUnit	64
SetTimeTransStartTime	65
QueryTimeTrandStartTime	65
SetTimeTransStopTime	65
QueryTimeTransStopTime	66
SetTimeTransSwitch	66
QueryTimeTransSwitch	66
SetVFactor	67
QueryVFactor	67
SetWinFunc	68
QueryWinFunc	68
IV Commands of the Calibration Sub-system	69
SetCalibZero	69
QueryCalibZero	69
V Commands of the Display Sub-system	70
SetScaleTop	70
QueryScaleTop	70
SetScaleBottom	71
QueryScaleBottom	71
SetScaleMax	71
QueryScaleMax	72
SetScaleMin	72
QueryScaleMin	73

---

AutoScale.....	73
SetScalePDiv.....	73
QueryScalePDiv.....	74
SetAmplitudeRef (Ref level ).....	74
QueryAmplitudeRef (Ref level ).....	74
VI Commands of the Trigger Sub-system.....	75
SetSwpType.....	75
QuerySwpType.....	75
TrigerSwp.....	76
HoldSwp.....	76
VII Commands of the Device Sub-system.....	76
QueryInstCatalog.....	76
SetInstSel.....	77
QueryInstSel.....	77
VIII Commands of the Memory Sub-system.....	78
DelStateFile.....	78
DelAllStateFile.....	78
DelTraceFile.....	79
DelAllTraceFile.....	79
DelPictureFile.....	79
DelAllPictureFile.....	80
LoadStateFile.....	80
LoadTraceFile.....	80
SetFileLocation.....	81
QueryFileLocation.....	81
StoreScreen.....	82
StoreStateFile.....	82
StoreTraceFile.....	82
IX Commands of the Sensor Subsystem.....	83
ACPRSetSwitch.....	83
ACPRQuerySwitch.....	83
ACPRSetMainCHBW.....	84
ACPRQueryMainCHBW.....	84
ACPRSetAdjCHBW.....	84
ACPRQueryAdjCHBW.....	85
ACPRSetCHSpace.....	85
ACPRQueryCHSpace.....	86
ACPRQueryMainCHPower.....	86
ACPRQueryUpCHPower.....	86
ACPRQueryDownCHPower.....	87
ACPRQueryUpACPR.....	87

---

ACPRQueryDownACPR.....	87
SASStartZeroCal.....	88
FSTSetAntenaOff (Field Strength).....	88
STSetSwitch.....	88
FSTQuerySwitch.....	89
FSTSetAntena.....	89
FSTQueryAntena.....	89
SetAmplitudeScaleType.....	90
QueryAmplitudeScaleType.....	90
SetAmplitudeUnit.....	91
QueryAmplitudeUnit.....	91
SetAvgOn.....	92
QueryAvgOn.....	92
SetAvgFactor.....	92
QueryAvgFactor.....	93
ClearAvgCount.....	93
SetRBW.....	93
QueryRBW.....	94
SetVBW.....	94
QueryVBW.....	95
SetRBWAuto.....	95
QueryRBWAuto.....	95
SetVBWAuto.....	96
QueryVBWAuto.....	96
SetSPANRBW_Ratio.....	96
QuerySPANRBW_Ratio.....	97
SetRBWVBW_Ratio.....	97
QueryRBWVBW_Ratio.....	97
SetIFBW.....	98
QueryIFBW.....	98
SetCalMethod.....	99
QueryCalMethod.....	99
CalCollIso.....	100
QueryCalCollIso.....	100
CalCollLoad.....	101
QueryCalCollLoad.....	101
CalCollOpen.....	102
QueryCalCollOpen.....	102
CalCollShort.....	102
QueryCalCollShort.....	103
CalCollThru.....	103

---

QueryCalCollThru.....	104
SetCalKit.....	104
QueryCalKit.....	105
CalCollFinish.....	105
SetCalOn.....	105
QueryCalOn.....	106
QueryCalValid.....	106
SetMCalKitMatch.....	106
SetOffset.....	107
QueryOffset.....	107
SetOffsetOn.....	107
QueryOffsetOn.....	108
CHPWSetSwitch.....	108
CHPWQuerySwitch.....	109
CHPWSetChBw.....	109
CHPWQueryChBw.....	109
CHPWQueryChPower.....	110
CHPWQueryPowerDensity.....	110
SetDetectorType.....	110
QueryDetectorType.....	111
SetDetectorAuto.....	111
QueryDetectorAuto.....	112
SetFreq.....	112
QueryFreq.....	113
SetCenterFreq.....	113
QueryCenterFreq.....	113
SetSpan.....	114
QuerySpan.....	114
SetFullSpan.....	114
SetZeroSpan.....	115
SetLastSpan.....	115
SetStartFreq.....	115
QueryStartFreq.....	116
SetStopFreq.....	116
QueryStopFreq.....	116
SetCWFreq.....	117
QueryCWFreq.....	117
OBWSetSwitch.....	118
OBWQuerySwitch.....	118
OBWSetMethod.....	118
OBWQueryMethod.....	119

---

OBWSetPercent.....	119
OBWQueryPercent.....	120
OBWSetXdBValue.....	120
OBWQueryXdBValue.....	120
OBWQueryOBWValue.....	121
SetAttenuator.....	121
QueryAttenuator.....	121
SetAttenuatorAuto.....	122
QueryAttenuatorAuto.....	122
SetPreAmpSwitch.....	122
QueryPreAmpSwitch.....	123
SetSwpPoints.....	123
QuerySwpPoints.....	124
SetSwpTime.....	124
QuerySwpTime.....	124
SetAutoSwpTimeOn.....	125
QueryAutoSwpTimeOn.....	125
IASetSwitch.....	126
IAQuerySwitch.....	126
IASetRedLimit.....	126
IAQueryRedLimit.....	127
IASetBlueLimit.....	127
IAQueryBlueLimit.....	127
IASetMode.....	128
IAQueryMode.....	128
IAClear.....	129
QueryMeasData.....	129
X Commands of the Source Sub-system.....	129
SetPortOutputSweepType.....	129
QueryPortOutputSweepType.....	130
SetPortOutputMode.....	130
QueryPortOutputMode.....	131
SetPortOutputManualPwr.....	131
QueryPortOutputManualPwr.....	131
XI Commands of the Trace Sub-system.....	132
ReadSATrace.....	132
SetSATraceStatus.....	132
QuerySATraceStatus.....	133
XII Sub-system Commands of the System.....	133
SetGPSOn.....	133
QueryGPSOn.....	134

---

QueryGPSState.....	134
QueryGPSReceiveState.....	135
GPS Cold Start.....	135
QueryGPSData.....	135



## Chapter I Description of SCPI Commands

### Instrument Connection

Before using SCPI commands or secondary development libraries for remote control of the instrument, it is necessary to connect the instrument with a computer via a network cable, with the IPs of the computer and the instrument configured to ensure normal communication between them. Remote control with USB cable is not supported at present.

### IEEE 488.2 Common Commands

#### \*CLS - Clear Status

Clear the instrument status, namely: Empty error queue and all event register. At the same time cancel all pending \*OPC commands and query commands.

#### \*IDN? - Identify

Return the unique instrument identification string, which varies with the different models.

#### \*OPC - Operation Complete Command

Set OPC bit of the standard event status register, after completing all pending overlapping commands (e.g.: Primary sweep or Default command, etc.).

#### \*OPC? - Operation Complete Query

Return letter "1" after completing all pending overlapping commands.

#### \*RST - Reset

Perform reset operation, cancel all pending \*OPC commands or query commands, the function of which is same as SYSTem:PRESet command. The nonvolatile memory contents of the instrument are not lost.

#### \*WAI - Wait

The instrument processes the new commands after completion of processing all the pending overlapping commands.

**:CALCulate:PARAmeter:DEFine <string>**

**(Read-Write)** Set or query current measurement parameter type.

**Applicable Mode:** Network Analyzer, CAT, VVM

**Parameters:** string, value description:

Instrument mode	Set command parameter (string)	Query command returned value (int)	Description
Network analyze	S11	0	Positive reflection measurement
	S21	1	Positive transmission measurement
	S12	2	Reverse transmission measurement
	S22	3	Reverse reflection measurement
	A1, B1, R1, A2, B2, R2	4 (A1), 5 (B1), 6 (R1), 7 (A2), 8 (B2), 9 (R2)	Advanced measurement parameters
Antenna test	S11	0	Reflection measurement
	A1, B1, R1	4 (A1), 5 (B1), 6 (R1)	Advanced measurement parameters
Vector voltmeter (VVM)	S11 (1)	0	Port 1 reflection measurement
	S21 (2)	1	Port 1 transmission measurement
	S12 (3)	2	Port 2 transmission measurement
	S22 (4)	3	Port 2 reflection measurement

**Example:** :CALC:PAR:DEF S11

**Query Syntax:** :CALC:PAR:DEF?

**Default:** 0 (S11)

**Returned value:** Value (int)

**:CALCulate:RELative[:MAGNitude]?**

**(Read only)** query the saved related measured value (valid when Relative is turned on).

**Applicable Mode:** USB Power Meter

**Parameter:** None

**Example:** :CALC:REL?

**Query Syntax:** :CALC:REL?

**Default:** None

**Returned value:** Value (double) (dBm)

**:CALCulate:RELative[:MAGNitude]:AUTO <string>**

**(Read-Write)** Set or query current relative measurement to be on or off.

**Applicable Mode:** Power Monitor, USB Power Meter, VVM

**Parameters:** string, value: OFF, ON

**Example:** :CALC:REL:AUTO ON

**Query Syntax:** :CALC:REL:AUTO?

**Default:** OFF

**Returned value:** Value (int): 0 (Off), 1 (On)

**:CALCulate[:SElected]:DATA:FDATA?**

**(Read only)** Read the current trace data (the data after format conversion and such processing as averaging, transformation, format conversion and smoothing, etc.).

**Applicable Mode:** Network Analyzer, CAT

**Parameter:** None

**Example:** :CALC:DATA:FDATA?

**Query Syntax:** :CALC:DATA:FDATA?

**Default:** None

**Returned value:** Numerical array (float), in non-Smith and polar coordinate measurement formats, it is an array (index, measurement data) arranged in sequence; In Smith or polar coordinate measurement format, it is an array (real part of measurement data, imaginary part of measurement data) arranged in sequence.

**:CALCulate[:SElected]:DATA:FMEM?**

**(Read only)** Read the saved trace data (the data after format conversion and such processing as averaging, transformation, format conversion and smoothing, etc.).

**Applicable Mode:** Network Analyzer, CAT

**Parameter:** None

**Example:** :CALC:DATA:FMEM?

**Query Syntax:** :CALC:DATA:FMEM?

**Default:** None

**Returned value:** Numerical array (float), in non-Smith and polar coordinate measurement formats, it is an array (index, measurement data) arranged in sequence; In Smith or polar coordinate measurement format, it is an array (real part of measurement data, imaginary part of measurement data) arranged in sequence.

**:CALCulate[:SElected]:DATA:SDATA?**

**(Read only)** Read the current trace data (the original data after calibration and averaging but not receiving such processing as transformation, format conversion and smoothing, etc.).

**Applicable Mode:** Network Analyzer, CAT

**Parameter:** None

**Example:** :CALC:DATA:SDATA?

**Query Syntax:** :CALC:DATA:SDATA?

**Default:** None

**Returned value:** Numerical array (float) in the format of an array arranged in sequence (real part of original data, imaginary part of original data).

**:CALCulate[:SElected]:DATA:SMEM?**

**(Read only)** Read the saved trace data (the original data after calibration and averaging but not receiving such processing as transformation, format conversion and smoothing, etc.).

**Applicable Mode:** Network Analyzer, CAT

**Parameter:** None

**Example:** :CALC:DATA:SMEM?

**Query Syntax:** :CALC:DATA:SMEM?

**Default:** None

**Returned value:** Numerical array (float) in the format of an array arranged in sequence (real part of original data, imaginary part of original data).

**:CALCulate[:SElected]:FORMat <string>**

**(Read-Write)** Set or query the current measurement format (**vector voltage measurement format is only valid for reflection measurement**).

**Applicable Mode:** Network Analyzer, CAT, VVM

**Parameters:** string, value description:

Measurement mode	Set command parameter (string)	Query command returned value (int)	Description
Vector network analyzer	MLOG	0	Logarithm magnitude
	MLIN	1	Linear magnitude
	VSWR	2	Voltage standing wave ratio
	PHASe	3	Phase
	DELay	4	Delay
	SMITH	5	Smith
	POLar	6	Polar
Antenna test	VSWR	2	Voltage standing wave ratio
	PHASe	3	phase
	DELay	4	Delay
	SMITH	5	Smith
	POLar	6	Polar
	CLOSs	7	Cable loss
	RLOSs	8	Return loss
	DTFSwr	9	SWR standing wave ratio
	DTFRI	10	DTF return loss
Vector voltmeter	DB	11	DB
	VSWR	2	Voltage standing wave ratio
	REIM	12	Impedance

**Example:** :CALC:FORM MLOG

**Query Syntax:** :CALC:FORM?

**Default:** Network analyzer MLOG

**Returned value:** Value (int)

**:CALCulate[:SElected]:LIMit:BEEP <string>**

**(Read-Write)** Query or set audio alarm on / off upon limits.

**Applicable Mode:** Network Analyzer, CAT, Spectrum Analyzer

**Parameters:** string, value: OFF, ON

**Example:** :CALC:LIM:BEEP ON

**Query Syntax:** :CALC:LIM:BEEP?

**Default:** OFF

**Returned value:** Value (int): 0 (Off), 1 (On)

**:CALCulate[:SElected]:LIMit:EDIT:ADD**

**(Write only)** Add limit points, each limit line supporting up to 50 limit points.

**Applicable Mode:** Network Analyzer, CAT, Spectrum Analyzer

**Parameter:** None

**Example:** :CALC:SEL:LIM:EDIT:ADD

**Query Syntax:** None

**Default:** None

**Returned value:** None

**:CALCulate[:SElected]:LIMit:EDIT:SElect <int>**

**(Read-Write)** Designate current limit point through indexing, with the index starting from 0.

**Applicable Mode:** Network Analyzer, CAT, Spectrum Analyzer

**Parameters:** Value (int), range: 0 ~ (number of limit points -1)

**Example:** :CALC:SEL:LIM:EDIT:SEL 1

**Query Syntax:** :CALC:SEL:LIM:EDIT:SEL?

**Default:** None

**Returned value:** Value (int), index value of current limit point

**:CALCulate[:SElected]:LIMit:EDIT:DElete**

**(write only)** Delete current limit point.

**Applicable Mode:** Network Analyzer, CAT, Spectrum Analyzer

**Parameter:** None

**Example:** :CALC:SEL:LIM:EDIT:DEL

**Query Syntax:** None

**Default:** None

**Returned value:** None

**:CALCulate[:SElected]:LIMit:EDIT:CLEar**

**(Write only)** Delete all limit points

**Applicable Mode:** Network Analyzer, CAT, Spectrum Analyzer

**Parameter:** None

**Example:** :CALC:LIM:EDIT:CLE

**Query Syntax:** None

**Returned value:** None

**:CALCulate[:SElected]:LIMit:POINts?**

**(Read only)** Get current number of limit points

**Applicable Mode:** Network Analyzer, CAT, Spectrum Analyzer

**Parameter:** None

**Example:** :CALC:LIM:POIN?

**Default:** 0

**Returned value:** Value (int)

**:CALCulate[:SElected]:LIMit:EDIT:X <double>**

**(Read-Write)** Query or set X value of current limit point. In frequency domain measurement, the unit is hertz (Hz), in time domain measurement, the unit is second (s), and in DTF measurement, the unit is meter (M).

**Applicable Mode:** Network Analyzer, CAT, Spectrum Analyzer

**Parameter:** Value (double), the range is the test range in the domain measured currently.

**Example:** :CALC:LIM:EDIT:X 100000

**Query Syntax:** :CALC:LIM:EDIT:X?

**Default:** None

**Returned value:** Value (double)

**:CALCulate[:SElected]:LIMit:EDIT:Y <double>**

**(Read-Write)** Query or set amplitude of the current limit point.

**Applicable Mode:** Network Analyzer, CAT, Spectrum Analyzer

**Parameter:** Value (double), range: -500 ~ 500 (This range is irrelevant to the measurement format in Network Analyzer and CAT modes. But in the Spectrum Analyzer mode, the unit is dBm)

**Example:** :CALC:SEL:LIM:EDIT:Y 10

**Query Syntax:** :CALC:SEL:LIM:EDIT:Y?

**Default:** None

**Returned Value:** Value (double)

**:CALCulate[:SElected]:LIMit:TEST <string>**

**(Read-Write)** Limit line test switch.

**Applicable Mode:** Network Analyzer, CAT, Spectrum Analyzer

**Parameters:** string, value: OFF, ON

**Example:** :CALC:SEL:LIM:TEST ON

**Query Syntax:** :CALC:SEL:LIM:TEST?

**Default:** OFF

**Returned Value:** Value (int): 0 (OFF), 1 (ON)

**:CALCulate[:SElected]:LIMit:PASS?**

**(Read only)** Whether the limit test is passed

**Applicable Mode:** Network Analyzer, CAT, Spectrum Analyzer

**Parameter:** None

**Example:** :CALC:SEL:LIMit:PASS?

**Query Syntax:** :CALC:SEL:LIMit:PASS?

**Default:** None

**Returned Value:** Value (int), 0 (Fail), 1 (Pass)

**:CALCulate[:SElected]:MARKer{n}:[STATe] <string>**

**(Read-Write)** Query or set state of Marker n. N may be set to 1, 2, 3, 4, 5, 6, 7 or 8, respectively indicating marker 1, 2, 3, 4, 5, 6, 7 or 8. n is 1 if not specified.

**Applicable Mode:** Spectrum Analyzer, Network Analyzer, CAT

**Parameter:** string, value:

Set command parameter (string)	Query command returned value (int)	Description
OFF	0	Marker off
NORMAL	1	Normal Mkr mode
DELTA	2	Delta Mkr mode

**Example:** :CALC:MARK1:STAT OFF

**Query Syntax:** :CALC:MARK1:STAT?

**Default:** OFF

**Returned value:** Value (int)

**:CALCulate[:SElected]:MARKer:AOff**

**(Write only)** All markers off.

**Applicable Mode:** Spectrum Analyzer, Network Analyzer, CAT

**Parameter:** None

**Example:** :CALC:MARK:AOff

**Query Syntax:** None

**Default:** None

**Returned Value:** None

**:CALCulate[:SElected]:MARKer{n}:FCOunt[:STATe] <string>**

**(Read-Write)** Set or query counter on / off (set to Normal Mkr when the counter function of turned on, with only one marker counter turned on). N may be set to 1, 2, 3, 4, 5, 6, 7 or 8, respectively indicating marker 1, 2, 3, 4, 5, 6, 7 or 8. n is 1 if not specified.

**Applicable Mode:** Spectrum Analyzer

**Parameters:** string, value: OFF, ON

**Example:** :CALC:MARK1:FCO ON

**Query Syntax:** :CALC:MARK1:FCO?

**Default:** OFF

**Returned Value:** Value (int): 0 (OFF), 1 (ON)

**:CALCulate[:SElected]:MARKer:FCOunt:X?**

**(Read only)** Query frequency value on the counter (return to 0 if counter is not started).

**Applicable Mode:** Spectrum Analyzer

**Parameter:** None.

**Example:** :CALC:MARK:FCO:X?

**Query Syntax:** :CALC:MARK:FCO:X?

**Default:** None

**Returned Value:** Value (double)

**:CALCulate[:SElected]:MARKer{n}:FUNcTion:MAXimum**

**(Write only)** Search for max. value of Marker n (turn on marker if it's not turned on). N may be set to 1, 2, 3, 4, 5, 6, 7 or 8, respectively indicating marker 1, 2, 3, 4, 5, 6, 7 or 8. n is 1 if not specified.

**Applicable Mode:** Spectrum Analyzer, Network Analyzer, CAT

**Parameter:** None

**Example:** :CALC:MARK1:FUNC:MAX

**Query Syntax:** None

**Default:** None

**Returned value:** None

**:CALCulate[:SElected]:MARKer{n}:FUNcTion:MINimum**

**(Write only)** Search for min. value of Marker n (turn on marker if it's not turned on). N may be set to 1, 2,

3, 4, 5, 6, 7 or 8, respectively indicating marker 1, 2, 3, 4, 5, 6, 7 or 8. n is 1 if not specified.

**Applicable Mode:** Spectrum Analyzer, Network Analyzer, CAT

**Parameter:** None

**Example:** :CALC:MARK1:FUNC:MIN

**Query Syntax:** None

**Default:** None

**Returned value:** None

**:CALCulate[:SElected]:MARKer{n}:FUNCtion:PEAK**

**(Write only)** Search for peak value of marker (**turn on marker if it's not turned on**). N may be set to 1, 2, 3, 4, 5, 6, 7 or 8, respectively indicating marker 1, 2, 3, 4, 5, 6, 7 or 8. n is 1 if not specified.

**Applicable Mode:** Spectrum Analyzer

**Parameter:** None

**Example:** :CALC:MARK1:FUNC:PEAK

**Query Syntax:** None

**Default:** None

**Returned value:** None

**:CALCulate[:SElected]:MARKer{n}:FUNCtion:PLEft**

**(Write only)** search for left peak of marker (**turn on marker if it's not turned on**). N may be set to 1, 2, 3, 4, 5, 6, 7 or 8, respectively indicating marker 1, 2, 3, 4, 5, 6, 7 or 8. n is 1 if not specified.

**Applicable Mode:** Spectrum Analyzer

**Parameter:** None

**Example:** :CALC:MARK1:FUNC:PLEF

**Query Syntax:** None

**Default:** None

**Returned value:** None

**:CALCulate[:SElected]:MARKer{n}:FUNCtion:PNEXt**

**(Write only)** Search for Sub Peak of marker (**turn on marker if it's not turned on**). N may be set to 1, 2, 3, 4, 5, 6, 7 or 8, respectively indicating marker 1, 2, 3, 4, 5, 6, 7 or 8. n is 1 if not specified.

**Applicable Mode:** Spectrum Analyzer

**Parameter:** None

**Example:** :CALC:MARK1:FUNC:PNEX

**Query Syntax:** None

**Default:** None

**Returned value:** None

**:CALCulate[:SElected]:MARKer{n}:FUNCtion:PRIGHt**

**(Write only)** Search for right peak of marker (**turn on marker if it's not turned on**). N may be set to 1, 2, 3, 4, 5, 6, 7 or 8, respectively indicating marker 1, 2, 3, 4, 5, 6, 7 or 8. n is 1 if not specified.

**Applicable Mode:** Spectrum Analyzer

**Parameter:** None

**Example:** :CALC:MARK1:FUNC:PRIG

**Query Syntax:** None



**Default:** None

**Returned value:** None

**:CALCulate[:SElected]:MARKer{n}:NOISe <string>**

**(Read-Write)** Set or query noise marker (**turn on marker if it's not turned on**). N may be set to 1, 2, 3, 4, 5, 6, 7 or 8, respectively indicating marker 1, 2, 3, 4, 5, 6, 7 or 8. n is 1 if not specified.

**Applicable Mode:** Spectrum Analyzer

**Parameter:** string, value: OFF (Off), ON (On)

**Example:** :CALC:MARK1:NOIS ON

**Query Syntax:** :CALC:MARK1:NOIS?

**Default:** OFF

**Returned value:** Value (int): 0 (Off), 1 (On)

**:CALCulate[:SElected]:MARKer{n}:SET <string>**

**(Write only)** Set marker function (Mkr→) set the current marker position as frequency parameter (**turn on marker if it's not turned on**). N may be set to 1, 2, 3, 4, 5, 6, 7 or 8, respectively indicating marker 1, 2, 3, 4, 5, 6, 7 or 8. n is 1 if not specified.

**Applicable Mode:** Spectrum Analyzer

**Parameter:** string, value:

Parameter Value	Description
START	Start frequency
STOP	Stop frequency
CENTER	Center frequency
STEP	Step frequency

**Example:** :CALC:MARK1:SET START

**Query Syntax:** None

**Default:** None

**Returned Value:** None

**:CALCulate[:SElected]:MARKer{n}:X <double>**

**(Read-Write)** Set or query marker X value (**invalid if marker is not turned on**). N may be set to 1, 2, 3, 4, 5, 6, 7 or 8, respectively indicating marker 1, 2, 3, 4, 5, 6, 7 or 8. n is 1 if not specified. In time domain measurement, the unit is second (s), in frequency domain measurement, the unit is Hertz (Hz), and in DTF measurement, the unit is the current distance unit.

**Applicable Mode:** Spectrum Analyzer, Network Analyzer, CAT

**Parameter:** Value (double), the range is the current test range.

**Example:** :CALC:MARK1:X 10000

**Query Syntax:** :CALC:MARK1:X?

**Default:** When normal marker is turned on, the marker is set as the start index point.

When delta marker is turned on, the delta marker position is the same to that of the normal marker.

**Returned Value:** Value (double)

**:CALCulate[:SElected]:MARKer{n}:Y?**

**(Read only)** Query marker amplitude (**return to 0 if marker is not turned on**). N may be set to 1, 2, 3, 4, 5, 6, 7 or 8, respectively indicating marker 1, 2, 3, 4, 5, 6, 7 or 8. n is 1 if not specified.

**Applicable Mode:** Spectrum Analyzer, Network Analyzer, CAT

**Parameter:** None

**Example:** :CALC:MARK1:Y?

**Query Syntax:** :CALC:MARK1:Y?

**Default:** None

**Returned Value:** Two values (double, double) in Network Analyzer and CAT modes and the real part and the imaginary part of the complex number measurement results in Smith or Polar coordinates. The value is the sweep index value and measured value in other formats.

In the Spectrum Analyzer mode, there are two values (double, double), with the former being the current measured value (adopting the current amplitude unit), and the latter fixed to 0.0.

**:CALCulate[:SElected]:MATH:FUNCtion <string>**

**(Read-Write)** Set or query trace display and calculation modes.

**Applicable Mode:** Network Analyzer, CAT

**Parameters:** string, value description:

Set command Parameter (string)	Query command returned value (int)	Description
DATA	0	No trace calculation, current trace displayed
MEM	1	No trace calculation, saved trace displayed (Save Trace must be valid)
AND	2	No trace calculation, current and saved traces displayed (Save Trace must be valid)
SUB	3	Current trace-saved trace, current trace displayed (Save Trace must be valid)
ADD	4	Current trace+saved trace, current trace displayed (Save Trace must be valid)
DIV	5	Current trace / saved trace, current trace displayed (Save Trace must be valid)
SRCMATCH	6	Display source matching results (Save Trace must be valid)

**Example:** :CALC:MATH:FUNC DATA

**Query Syntax:** :CALC:MATH:FUNC?

**Default:** DATA

**Returned Value:** Value (int), see the table above for detailed definition

**:CALCulate[:SElected]:MATH:MEMorize**

**(Write only)** Current trace saved to Save Trace.

**Applicable Mode:** Network Analyzer, CAT

**Parameter:** None

**Example:** :CALC:MATH:MEM

**Query Syntax:** None

**Default:** None

**Returned value:** None

**:CALCulate[:SElected]:SMOothing:APERture <double>**

**(Read-Write)** Set or query smoothing aperture.

**Applicable Mode:** Network Analyzer, CAT

**Parameter:** Value (double)(%), range: 0.01 ~ 20.00

**Example:** :CALC:SMO:APER 10.00

**Query Syntax:** :CALC:SMO:APER?

**Default:** 10.00

**Returned Value:** Value (double)

**:CALCulate[:SElected]:SMOothing[:STATe] <string>**

**(Read-Write)** Set or query Smoothing On / off.

**Applicable Mode:** Network Analyzer, CAT

**Parameters:** string, value: OFF, ON

**Example:** :CALC:SMO ON

**Query Syntax:** :CALC:SMO?

**Default:** OFF

**Returned Value:** Value (int): 0 (OFF), 1 (ON)

**:CALCulate[:SElected]:TRANSform:CLOSs <double>**

**(Read-Write)** Set or query cable loss of DTF (the unit is relevant to the selected DTF unit, dB/m or dB/ft).

**Applicable Mode:** CAT

**Parameter:** Value (double), range: 0.000 ~ 30.000 (dB/m)

**Example:** :CALC:TRAN:CLOS 0.1

**Query Syntax:** :CALC:TRAN:CLOS?

**Default:** 0

**Returned Value:** Value (double)

**:CALCulate[:SElected]:TRANSform:DISTance:STARt <double>**

**(Read-Write)** Set or query start distance of DTF.

**Applicable Mode:** CAT

**Parameter:** Value (double), the same unit to the distance unit of the current DTF, range: 0 ~ max distance. The max. distance is determined by Span, Speed Factor and Number of Points.

**Example:** :CALC:TRAN:DIST:STAR 1

**Query Syntax:** :CALC:TRAN:DIST:STAR?

**Default:** 0 m

**Returned Value:** Value (double)

**:CALCulate[:SElected]:TRANSform:DISTance:STOP <double>**

**(Read-Write)** Set or query stop distance of DTF.

**Applicable Mode:** CAT

**Parameter:** Value (double), the same unit to the distance unit of the current DTF, range: 0 ~ max distance. The max. distance is determined by Span, Speed Factor and Number of Points.

**Example:** :CALC:TRAN:DIST:STOP 0.1

**Query Syntax:** :CALC:TRAN:DIST:STOP?

**Default:** The max distance at the parameter currently swept

**Returned Value:** Value (double)

**:CALCulate[:SElected]:TRANSform:DISTance:UNIT <string>**

**(Read-Write)** Set or query unit type of DTF.

**Applicable Mode:** CAT

**Parameter:** string, value: METER (Metric), FEET (English)

**Example:** :CALC:TRAN:DIST:UNIT METER

**Query Syntax:** :CALC:TRAN:DIST:UNIT?

**Default:** METERS

**Returned Value:** Value (int): 0 (METER), 1 (FEET).

**:CALCulate[:SElected]:TRANSform:TIME:STARt <double>**

**(Read-Write)** Set or query start time of time domain.

**Applicable Mode:** Network Analyzer

**Parameter:** Value (double)(ns), range: 0 ~ Max. test time. Max. test time is determined by current span and number of points.

**Example:** :CALC:TRAN:TIME:STAR 100

**Query Syntax:** :CALC:TRAN:TIME:STAR?

**Default:** 0 ns

**Returned Value:** Value (double) (ns)

**:CALCulate[:SElected]:TRANSform:TIME:STATe <string>**

**(Read-Write)** Set or query time domain on / off.

**Applicable Mode:** Network Analyzer

**Parameters:** string, value: OFF, ON

**Example:** :CALC:TRAN:TIME:STAT ON

**Query Syntax:** :CALC:TRAN:TIME:STAT?

**Default:** 0 (OFF)

**Returned value:** Value (int): 0 (Off), 1 (On)

**:CALCulate[:SElected]:TRANSform:TIME:STOP <double>**

**(Read-Write)** Set or query stop time of time domain, in the unit of nanoseconds (ns).

**Applicable Mode:** Network Analyzer

**Parameter:** Value (double)(ns), range: 0 ~ Max. test time. Max. test time is determined by current span and number of points.

**Example:** :CALC:TRAN:TIME:STOP 0.1

**Query Syntax:** :CALC:TRAN:TIME:STOP?

**Default:** Max. stop time at the current parameter.

**Returned Value:** Value (double) (ns)

**:CALCulate[:SElected]:TRANSform:VFACtor <double>**

**(Read-Write)** Set or query the velocity of time domain (Network Analyzer) or DTF (CAT).

**Applicable Mode:** Network Analyzer, CAT

**Parameter:** Value (double), range: 0.01 ~ 1.00

**Example:** :CALC:TRAN:VFAC 1

**Query Syntax:** :CALC:TRAN:VFAC?

**Default:** 1.0

**Returned Value:** Value (double)

**:CALCulate[:SElected]:TRANSform:WINDow <string>**

**(Read-Write)** Set or query window function type of time domain (Network Analyzer) or DTF (CAT).

**Applicable Mode:** Network Analyzer, CAT

**Parameter:** string, parameter value description:

Instrument mode	Set command Parameter (string)	Query command Returned value (int)	Description
Network analyze	MAXIMUM	0	Max
	NORMAL	1	Normal
	MINIMUM	2	Min
	MANUAL	3	Man
Antenna test	RECT	4	Rectangular Window
	HANNING	5	Hanning Window
	HAMMING	6	Hamming Window
	BLACKMAN	7	Blackman Window
	KAISER	8	Kaiser Window

**Example:** :CALC:TRAN:WIND RECT

**Query Syntax:** :CALC:TRAN:WIND?

**Default:** Network Analyzer mode, 1 (Normal);  
CAT mode, 4 (Rectangular Window)

**Returned Value:** Value (int), see the table above for detailed definition

**:CALibration:ZERO**

**(Write only)** Start USB Power Meter zeroing.

**Applicable Mode:** USB Power Meter

**Parameter:** None

**Example:** :CAL:ZERO

**Query Syntax:** None

**Default:** None

**Returned value:** None

**:CALibration:ZERO:STATe?**

**(Read only)** USB Power Meter zeroing state query.

**Applicable Mode:** USB Power Meter

**Parameter:** None

**Example:** :CAL:ZERO:STAT?

**Default:** 0 (normal)

**Returned Value:** Value (int), see the table above for detailed definition:

Query command Returned value (int)	Description
0	Normal
1	Zeroing
2	Zeroing completed
3	Zeroing failed

**:DISPlay:WINDow:ANALog:LOWer <double>**

**(Read-Write)** Set or query mix scale.

**Applicable Mode:** Power Monitor, USB Power Meter

**Parameter:** Value (double), dBm as the unit, range: -70 ~ 25.

**Example:** :DISP:WIND:ANAL:LOW -60

**Query Syntax:** :DISP:WIND:ANAL:LOW?

**Default:** -70 dBm

**Returned Value:** Value (double)

**:DISPlay:WINDow:ANALog:UPPer <double>**

**(Read-Write)** Set or query max scale.

**Applicable Mode:** Power Monitor, USB Power Meter

**Parameter:** Value (double), dBm as the unit, range: -70 ~ 25

**Example:** :DISP:WIND:ANAL:UPP 20

**Query Syntax:** :DISP:WIND:ANAL:UPP?

**Default:** 30 dBm

**Returned Value:** Value (double)

**:DISPlay:WINDow:TRACe:Y[:SCALe]:AUTO**

**(Write only)** Set to Auto Scale.

**Applicable Mode:** Network Analyzer, CAT, VVM, USB Power Meter

**Parameter:** None

**Example:** :DISP:WIND:TRAC:Y:AUTO

**Query Syntax:** None

**Default:** None

**Returned value:** None

**:DISPlay:WINDow:TRACe:Y[:SCALe]:TOP <double>**

**(Read-Write)** Query or set top scale.

**Applicable Mode:** Network Analyzer, CAT

**Parameter:** Value (double) (dB), scale range:

-500 dB ~ 500 dB (Log)

-10U ~ 100 U (Linear)

-1,000 ns ~ 1,000 ns (Group Delay)

-450 ~ 450° (Phase)

-100 ~ 100 (VSWR).

0.1 will be automatically added when set to the min value.

**Example:** :DISP:WIND:TRAC:Y:TOP 0.1

**Query Syntax:** :DISP:WIND:TRAC:Y:TOP?

**Default:** None

**Returned Value:** Value (double)

**:DISPlay:WINDow:TRACe:Y[:SCALe]:BOTTom <double>**

**(Read-Write)** Query or set bottom scale.

**Applicable Mode:** Network Analyzer, CAT

**Parameter:** Value (double) (dB), scale range:

- 500 dB ~ 500 dB (Log)
- 10 U ~ 100 U (Linear)
- 1,000 ns ~ 1,000 ns (Group Delay)
- 450 ~ 450° (Phase)
- 100 ~ 100 (VSWR).

0.1 will be automatically subtracted when set to Max min value.

**Example:** :DISP:WIND:TRAC:Y:BOTT 0.1

**Query Syntax:** :DISP:WIND:TRAC:Y:BOTT?

**Default:** None

**Returned Value:** Value (double)

**:DISPlay:WINDow:TRACe:Y[:SCALe]:PDIVision <double>**

**(Read-Write)** Query or set Scale / Div.

**Applicable Mode:** Spectrum Analyzer (only valid when amplitude unit is set to dBm).

**Parameter:** Value (double) (dB), range: 0.1 ~ 20

**Example:** :DISP:WIND:TRAC:Y:PDIV 0.1

**Query Syntax:** :DISP:WIND:TRAC:Y:PDIV?

**Default:** None

**Returned Value:** Value (double)

**:DISPlay:WINDow:TRACe:Y[:SCALe]:RLEVel <double>**

**(Read-Write)** Query or set reference level (reference value).

**Applicable Mode:** Spectrum Analyzer

**Parameter:** Value (double) (dBm), value range: -150 ~ 30

**Example:** :DISP:WIND:TRAC:Y:RLEV -10

**Query Syntax:** :DISP:WIND:TRAC:Y:RLEV?

**Default:** 0dBm

**Returned Value:** Value (double), dBm as the unit

**:FORMat[:DATA] <string>**

**(Read-Write)** Query or set data format.

**Applicable Mode:** All modes

**Parameter:** string, value: ASC (character), HEX (numeric value)

**Example:** :FORM ASC

**Query Syntax:** :FORM?

**Default:** ASC

**Returned value:** Value (int): 0 (ASC), 1 (HEX)

**:INITiate:CONTInuous <string>**

**(Read-Write)** Query or set sweep type.

**Applicable Mode:** Spectrum Analyzer, Network Analyzer, CAT

**Parameter:** string, value: OFF (Swp Once), ON (Swp Cont)

**Example:** :INIT:CONT OFF

**Query Syntax:** :INIT:CONT?

**Default:** ON

**Returned Value:** Value (int): 0 (Swp Once), 1 (Swp Cons)

**:INITiate[:IMMediate]**

**(Write only)** Trigger sweep once (valid only in Swp Once.).

**Applicable Mode:** Spectrum Analyzer, Network Analyzer, CAT

**Parameter:** None.

**Example:** :INIT;

**Query Syntax:** None

**Default:** None

**Returned value:** None

**:INITiate:HOLD**

**(Write only)** Hold the current sweep state.

**Applicable Mode:** Spectrum Analyzer, Network Analyzer, CAT

**Parameter:** None.

**Example:** :INIT:HOLD

**Query Syntax:** None

**Default:** None

**Returned value:** None

**:INSTrument:CATalog?**

**(Read only)** Query or set available instrument working mode. With **:INST:CAT**, you can query available working modes of the instrument.

**Applicable Mode:** All modes

**Parameter:** None

**Example:** :INST:CAT?

**Query Syntax:** :INST:CAT?

**Default:** 0 x 03 (Valid in Spectrum Analyzer and Network Analyzer modes)

**Returned Value:** Value(int). This value is converted from a 32-bit binary number. If a bit value is 1, the mode corresponding to that bit is valid. Relationship between measurements mode and bits:

Measurement mode	Bit
Antenna test	Bit 1
Spectrum Analyzer	Bit 2
Network analyze	Bit 3
Power Meter	Bit 4
USB Power Meter	Bit 5
Vector voltmeter (VVM)	Bit 6
Signal source	Bit 7

**:INSTrument[:SElect] <string>**

**(Read-Write)** Query or set current working mode of the instrument, with **:INST:CAT**, you can query



available working modes of the instrument.

**Applicable Mode:** All modes

**Parameter:** string, value description:

Set command Parameter (string)	Query command Returned value (int)	Description
SA	0	Spectrum Analyzer mode
VNA	1	Network Analyzer mode
CAT	2	Antenna test
PM	3	Power meter
USBPM	4	USB Power Meter
VVM	5	Vector voltmeter (VVM)
SG	6	Signal Source Measurement mode

**Example:** :INST NA;

**Query Syntax:** :INST?

**Default:** The mode in which the instrument was upon last shutdown

**Returned Value:** Value (int), see the table above for detailed definition

**:MMEMory:DELeTe:STATe <string>**

**(Write only)** Delete state file in the current mode (the command is invalid if the file does not exist and is valid only for the current storage location).

**Applicable Mode:** All modes

**Parameter:** State file name

**Example:** :MMEM:DEL:STAT set1

**Query Syntax:** None

**Default:** None

**Returned Value:** None

**:MMEMory:DELeTe:STATe:ALL**

**(Write only)** Delete all state files in the current mode.

**Applicable Mode:** All modes

**Parameter:** None

**Example:** :MMEM:DEL:STAT:ALL

**Query Syntax:** None

**Default:** None

**Returned Value:** None

**:MMEMory:DELeTe:TRACe <string>**

**(Write only)** Delete trace file in the current mode (the command is invalid if the file does not exist and is valid only for the current storage location).

**Applicable Mode:** Network Analyzer, CAT

**Parameter:** Trace file name

**Example:** :MMEM:DEL:TRAC rc1

**Query Syntax:** None

**Default:** None

**Returned Value:** None

**:MMEMory:DELeTe:TRACe:ALL**

**(Write only)** Delete all trace files in the current mode.

**Applicable Mode** Network Analyzer, CAT **Parameter:**

None

**Example:** :MMEM:DEL:TRAC:ALL

**Query Syntax:** None

**Default:** None

**Returned Value:** None

**:MMEMory:LOAD:STATe <string>**

**(Write only)** Load state file in the current mode (the command is invalid if the file does not exist and is valid only for the current storage location).

**Applicable Mode:** All modes

**Parameter:** State file name

**Example:** :MMEM:LOAD:STAT set1

**Query Syntax:** None

**Default:** None

**Returned Value:** None

**:MMEMory:LOAD:TRACe <string>**

**(Write only)** Load trace files in the current mode (the command is invalid if the file does not exist, and is valid only for the current storage location, loading trace state).

**Applicable Mode:** Network Analyzer, CAT

**Parameter:** Trace file name

**Example:** :MMEM:LOAD:TRAC trc1

**Query Syntax:** None

**Default:** None

**Returned Value:** None

**:MMEMory:LOCation <string>**

**(Read-Write)** Query or set storage location.

**Applicable Mode:** All modes

**Parameters:** string, value description:

Set command Parameter (string)	Query command Returned value (int)	Description
INT	0	internal storage
USB	1	USB storage
SDCARD	2	SD card

**Example:** :MMEM:LOC USB

**Query Syntax:** :MMEM:LOC?

**Default:** INT

**Returned Value:** Value (int), see the table above for detailed definition

**:MMEMory:STORe:SCReen <string>**

**(Write only)** Copy the screen and save the snapshot of the current screen into a file (the file will overwrite an existing file and is valid only for the current storage location).

**Applicable Mode:** All modes

**Parameter:** Screen shot file name.

**Example:** :MMEM:STOR:SCR pic1

**Query Syntax:** None

**Default:** None

**Returned Value:** None

**:MMEMory:STORe:STATe <string>**

**(Write only)** Save the state in the current mode as a file (the file will overwrite an existing file and is valid only for the current storage location).

**Applicable Mode:** All modes

**Parameter:** State file name.

**Example:** :MMEM:STOR:STAT set1

**Query Syntax:** None

**Default:** None

**Returned Value:** None

**:MMEMory:STORe:TRACe <string>**

**(Write only)** Save the trace in the current mode as a file (the file will overwrite an existing file and is valid only for the current storage location), and the state will be saved at the same time.

**Applicable Mode:** Network Analyzer, CAT

**Parameter:** Trace file name.

**Example:** :MMEM:STOR:TRAC trc1

**Query Syntax:** None

**Default:** None

**Returned Value:** None

**[[:SENSe]:ACPR:BW:ADJChbw <double>**

**(Read-Write)** Set or query ACPR and Adj Ch BW.

**Applicable Mode:** Spectrum Analyzer

**Parameter:** Value (double), Hertz (Hz) as the unit, value range: 0 ~ (current span - 2\*channel spacing)

**Example:** :ACPR:BW:ADJC 3000000

**Query Syntax:** :ACPR:BW:ADJC?

**Default:** 10000000 (10 MHz)

**Returned Value:** Value (double)

**[[:SENSe]:ACPR:BW:MAINchbw <double>**

**(Read-Write)** Set or query bandwidth of the main channel of ACPR.

**Applicable Mode:** Spectrum Analyzer

**Parameter:** Value (double), Hertz (Hz) as the unit, value range: Min span ~ max span (related to specific models).

**Example:** :ACPR:MAIN 3000000

**Query Syntax:** :ACPR:MAIN?

**Default:** 10 MHz

**Returned Value:** Value (double)

**[[:SENSe]:ACPR:CHPower:MAINchpower?**

**(Read only)** Query main channel power of ACPR.

**Applicable Mode:** Spectrum Analyzer **Parameter:**

None

**Example:** :ACPR:CHP:MAIN?

**Query Syntax:** :ACPR:CHP:MAIN?

**Default:** None

**Returned Value:** Value (double)

**[[:SENSe]:ACPR:CHPower:UPCHpower?**

**(Read only)** Query upper channel power of ACPR.

**Applicable Mode:** Spectrum Analyzer

**Parameter:** None

**Example:** :ACPR:CHP:UPCH?

**Query Syntax:** :ACPR:CHP:UPCH?

**Default:** None

**Returned Value:** Value (double)

**[[:SENSe]:ACPR:CHPower: DOWNchpower?**

**(Read only)** Query lower channel power of ACPR.

**Applicable Mode:** Spectrum Analyzer

**Parameter:** None

**Example:** :ACPR:CHP: DOWN?

**Query Syntax:** :ACPR:CHP: DOWN?

**Default:** None

**Returned Value:** Value (double)

**[[:SENSe]:ACPR:ACPR:UPCHacpr?**

**(Read only)** Query upper channel ACPR.

**Applicable Mode:** Spectrum Analyzer

**Parameter:** None

**Example:** :ACPR:ACPR:UPCH?

**Query Syntax:** :ACPR:ACPR:UPCH?

**Default:** None

**Returned Value:** Value (double)

**[[:SENSe]:ACPR:ACPR: DOWNchacpr?**

**(Read only)** Query lower channel ACPR.

**Applicable Mode:** Spectrum Analyzer

**Parameter:** None

**Example:** :ACPR: ACPR: DOWN?

**Query Syntax:** :ACPR: ACPR: DOWN?

**Default:** None

**Returned Value:** Value (double)

**[[:SENSe]:ACPR:BW:SPACe <double>**

**(Read-Write)** Set or query channel spacing of ACPR.

**Applicable Mode:** Spectrum Analyzer

**Parameter:** Value (double), Hertz (Hz) as the unit, value range: 0 ~ (current span-main channel bandwidth) / 2

**Example:** :ACP:SPAC 3000000

**Query Syntax:** :ACP:SPAC?

**Default:** 3 MHz

**Returned Value:** Value (double) (Hz)

**[[:SENSe]:ACPR[:STATe] <string>**

**(Read-Write)** Set or query ACPR on / off (**other measurement functions will be disabled if this function is enabled**).

**Applicable Mode:** Spectrum Analyzer

**Parameters:** string, value: OFF, ON

**Example:** :ACPR ON

**Query Syntax:** :ACPR ?

**Default:** OFF

**Returned Value:** Value (int): 0 (OFF), 1 (ON)

**[[:SENSe]:AMPLitude:ALIGnment:NOW**

**(Write only)** Zero calibration (**Do not repeat the zero calibration during the process of zero calibration**).

**Applicable Mode:** Spectrum Analyzer

**Parameter:** None

**Example:** :AMPL:ALIG:NOW

**Query Syntax:** None

**Default:** None

**Returned Value:** None

**[[:SENSe]:FST:ANTenna:OFF**

**(Write only)** Turn off antenna loading and set to No-antenna state.

**Applicable Mode:** Spectrum Analyzer

**Parameter:** None

**Example:** :SENS:FST:ANT:OFF

**Query Syntax:** None

**Default:** None

**Returned value:** None

**[[:SENSe]:FST [:STATe] <string>**

**(Read-Write)** Set or query Field Strength Meter on / off

**Applicable Mode:** Spectrum Analyzer

**Parameters:** string, value: OFF, ON

**Example:** :FST ON

**Query Syntax:** :FST?

**Default:** OFF

**Returned Value:** Value (int): 0 (OFF), 1 (ON)

**[[:SENSe]:FST:ANTenna <string>**

**(Read-Write)** Set or query antenna

**Applicable Mode:** Spectrum Analyzer

**Parameter:** string

**Example:** :FST:ANT?

**Query Syntax:** :FST:ANT?

**Default:** None

**Returned Value:** string

**[[:SENSe]:AMPLitude:SCALe <string>**

**(Read-Write)** Query or set scale type.

**Applicable Mode:** Spectrum Analyzer

**Parameter:** string, value: LOG(logarithm ), LIN(linear)

**Example:** :AMPL:SCAL LOG

**Query Syntax:** :AMPL:SCAL?

**Default:** LOG

**Returned Value:** Value(int): 0 (LOG), 1 (LIN).

**[[:SENSe]:AMPLitude:UNIT <string>**

**(Read-Write)** Query or set amplitude unit.

**Applicable Mode:** Spectrum Analyzer

**Parameters:** string, value description:

Set command Parameter (string)	Query command Returned value (int)	Description
dBm	0	dBm as the unit
dBmV	1	dBmV as the unit
dBuV	2	dBuV as the unit
Volts	3	Volt as the unit
Watts	4	Watt as the unit

**Example:** :AMPL:UNIT dBm

**Query Syntax:** :AMPL:UNIT?

**Default:** dBm

**Returned Value:** Value (int), see the table above for detailed definition

**[[:SENSe]:AVERage:COUNT <int>**

**(Read-Write)** Query or set average count.

**Applicable Mode:** Spectrum Analyzer, Network Analyzer, CAT, Power Meter

**Parameter:** Value (int), the range is 2 ~ 999

**Example:** :AVER:COUN 16

**Query Syntax:** :AVER:COUN?

**Default:** 16

**Returned Value:** Value (int)

**[[:SENSe]:AVERage:CLEar**

**(Write only)** Count current averaging from 1.

**Applicable Mode:** Spectrum Analyzer, Network Analyzer, CAT, Power Meter

**Parameter:** None

**Example:** :AVER:CLE

**Query Syntax:** None

**Returned Value:** None

**[[:SENSe]:AVERage:STATe <string>**

**(Read-Write)** Query or set averaging on / off.

**Applicable Mode:** Spectrum Analyzer, Network Analyzer, CAT, Power Meter

**Parameters:** string, value: OFF, ON

**Example:** :AVER:STAT OFF

**Query Syntax:** :AVER:STAT?

**Default:** OFF

**Returned Value:** Value (int): 0 (OFF), 1 (ON)

**[[:SENSe]:BANDwidth:RESolution <double>**

**(Read-Write)** Query or set resolution bandwidth (RBW).

**Applicable Mode:** Spectrum Analyzer, Power Meter

**Parameter:** Value (double), Hz as the unit, value range: 1 ~ 5000000 (set valid value to 1 Hz ~ 5 MHz, with the discrete value at the step of 1, 3, 10)

**Example:** :BAND:RES 300000

**Query Syntax:** :BAND:RES?

**Default:** 5 MHz

**Returned Value:** Value (double) (Hz)

**[[:SENSe]:BANDwidth:RESolution:AUTO <string>**

**(Read-Write)** Query or set RBW Auto On / Off. When set to Auto, RBW will adjust RBW by bandwidth according to the ratio of SPAN / RBW

**Applicable Mode:** Spectrum Analyzer, Power Meter

**Parameter:** string, value: OFF (Man), ON (Auto)

**Example:** :BAND:AUTO 300000

**Query Syntax:** :BAND:AUTO?

**Default:** 1 (Auto)

**Returned Value:** Value (int): 0 (OFF), 1 (ON)

**[[:SENSe]:BANDwidth:RESolution:RATio <int>**

**(Read-Write)** Query or set Span / RBW value.

**Applicable Mode:** Spectrum Analyzer, Power Meter

**Parameter:** Value (int), value range: 1 ~ 500

**Example:** :BAND:RES:RAT 100

**Query Syntax:** :BAND:RES:RAT?

**Default:** 100

**Returned Value:** Value (int)

**[[:SENSe]:BANDwidth:VIDeo <double>**

**(Read-Write)** Query or set video bandwidth (VBW).

**Applicable Mode:** Spectrum Analyzer, Power Meter

**Parameter:** Value (double) (Hz), value range: 1 ~ 5000000 (set valid value to 1 Hz ~ 5 MHz, with the discrete value at the step of 1, 3, 10)

**Example:** :BAND:VID 300000

**Query Syntax:** :BAND:VID?

**Default:** 5MHz

**Returned Value:** Value (double) (Hz)

**[[:SENSe]:BANDwidth:VIDeo:AUTO <string>**

**(Read-Write)** Query or set query or set VBW Auto On / Off. When set to Auto, the video bandwidth will be adjusted according to the ratio of RBW / VBW.

**Applicable Mode:** Spectrum Analyzer, Power Meter

**Parameter:** string, value: OFF (Man), ON (Auto)

**Example:** :BAND:VID:AUTO 300000

**Query Syntax:** :BAND:VID:AUTO?

**Default:** Auto

**Returned Value:** Value (int): 0 (OFF), 1 (ON)

**[[:SENSe]:BANDwidth:VIDeo:RATio <int>**

**(Read-Write)** Query or set RBW / VBW value.

**Applicable Mode:** Spectrum Analyzer, Power Meter

**Parameter:** Value (int), value range: 1 ~ 100

**Example:** :BAND:VID:RAT 1

**Query Syntax:** :BAND:VID:RAT?

**Default:** 1

**Returned Value:** Value (int)

**[[:SENSe]:BANDwidth <double>**

**(Read-Write)** Set or query IF BW.

**Applicable Mode:** Network Analyzer, CAT

**Parameter:** Value (double) (Hz), value range: 1 Hz ~ 100 kHz, with the discrete value at the step of 1, 3, 10.

**Example:** :BAND 1000

**Query Syntax:** :BAND?

**Default:** 1 kHz

**Returned Value:** Value (double) (Hz)

**[[:SENSe]:CORRection:COLLect[:ACQuire]:ISO <string>**



**(Read-Write)** Collect isolation calibration data, query whether isolation calibration data collection is completed.

**Applicable Mode:** Network Analyzer, CAT, VVM

**Parameters:** string, value description:

Set command Parameter (string)	Description
AUTO	Automatically collect positive isolation and reverse isolation calibration coefficient during full 2 ports calibration, and automatically collect isolation data in the measurement type during response and isolation calibration
IGNore	Ignore isolation
POS	Positive isolation
RES	Reserve isolation

**Example:** :CORR:COLL:ISO AUTO

**Query Syntax:** :CORR:COLL:ISO?

**Default:** None

**Returned Value:** Value (int): 0 (Undone), 1 (Done)

**[:SENSe]:CORRection:COLLect[:ACQuire]:LOAD{n}**

**(Read-Write)** Collect load calibration data, query whether load calibration data collection is completed.. n is the port number, which can be set to 1 and 2, respectively representing port 1 load and port 2 load. n is 1 if not specified.

**Applicable Mode:** Network Analyzer, CAT, VVM

**Parameter:** None

**Example:** :CORR:COLL:LOAD

**Query Syntax:** :CORR:COLL:LOAD1?

**Default:** None

**Returned Value:** Value (int): 0 (Undone), 1 (Done)

**[:SENSe]:CORRection:COLLect[:ACQuire]:OPEN{n}**

**(Read-Write)** Collect open circuit device calibration data. n is the port number, which can be set to 1 and 2, respectively representing port 1 load and port 2 load. n is 1 if not specified.

**Applicable Mode:** Network Analyzer, CAT, VVM

**Parameter:** None

**Example:** :CORR:COLL:OPEN2

**Query Syntax:** :CORR:COLL:OPEN2?

**Default:** None

**Returned Value:** Value (int): 0 (Undone), 1 (Done)

**[:SENSe]:CORRection:COLLect[:ACQuire]:SHORt{n}**

**(Read-Write)** Collect short circuit device calibration data, query whether short circuit calibration data collection is completed. n is the port number, which can be set to 1 and 2, respectively representing port 1 load and port 2 load. n is 1 if not specified.

**Applicable Mode:** Network Analyzer, CAT, VVM

**Parameter:** None

**Example:** :CORR:COLL:SHOR

**Query Syntax:** :CORR:COLL:SHOR?

**Default:** None

**Returned Value:** Value (int) 0 (Undone), 1 (Done)

**[[:SENSe]:CORRection:COLLect[:ACQuire]:THUR <string>**

**(Read-Write)** Collect THRU calibration data, query whether THRU calibration data collection is completed.

**Applicable Mode:** Network Analyzer, VVM

**Parameter:** string, parameter value description:

Set command Parameter (string)	Description
AUTO	Automatically collect positive matching, positive transmission, reverse transmission, and reserve matching calibration coefficients during full 2 ports calibration, automatically collect relevant data during frequency response, response and isolation calibration THRU data in the measurement type
S11	collect positive matching calibration data(valid only in Full 2 Ports mode)
S21	collect positive transmission calibration data(valid only in Full 2 Ports mode)
S12	collect negative transmission calibration data(valid only in Full 2 Ports mode)
S22	collect negative matching calibration data(valid only in Full 2 Ports mode)

**Example:** :CORR:COLL:THUR AUTO

**Query Syntax:** :CORR:COLL:THUR?

**Default:** None

**Returned Value:** Value (int): 0 (Undone), 1 (Done)

**[[:SENSe]:CORRection:COLLect[:ACQuire]:CKIT:LABel <string>**

**(Read-Write)** Set or query the currently used calibration kit.

**Applicable Mode:** Network Analyzer, CAT, VVM

**Parameter:** string, calibration kit name: AV31101A, AV20201A, AV20201

**Example:** :CORR:COLL:CKIT:LAB AV31101A

**Query Syntax:** :CORR:COLL:CKIT:LAB?

**Default:** AV31101A

**Returned Value:** string

**[[:SENSe]:CORRection:COLLect[:ACQuire]:CKIT:MATCh <string>**

**(Write only)** Set current matching type.

**Applicable Mode:** Network Analyzer, CAT, VVM

**Parameter:** string, parameter value description: MALE (male matching), FEMale (female matching)

**Example:** :CORR:COLL:CKIT:MATC FEM

**Query Syntax:** None

**Default:** None

**Returned Value:** None

**[[:SENSe]:CORRection:COLLect:METHod <string>**

**(Read-Write)** Set or query calibration type

**Applicable Mode:** Network Analyzer, CAT, VVM

**Parameter:** string, value description:

Instrument mode	Set Parameter Value (string)	Query command Returned value (int)	Description
Network analyze	NONE	0	None
	ERES1	1	Resp.&Iso@S12
	ERES2	2	Resp.&Iso@S21
	OPEN1	3	Freq Resp. S11 Open
	OPEN2	4	Freq Resp. S22 Open
	SHOR1	5	Freq Resp. S11 Short
	SHOR2	6	Freq Resp. S22 Short
	THRU1	7	Freq Resp. S12 THRU
	THRU2	8	Freq Resp. S21 THRU
	SOLR	9	Full 2 ports calibration
	SOLT1	10	S11 single port calibration
	SOLT2	11	S22 single port calibration
Antenna test	SOLT1	10	S11 single port
Vector voltmeter (VVM)	SOLR	9	Full 2 ports calibration
	SOLT1	11	S11 single port calibration
	SOLT2	12	S22 single port calibration

**Example:** :CORR:COLL:METH SOLR

**Query Syntax:** :CORR:COLL:METH?

**Default:** None

**Returned Value:** Value (int)

**[[:SENSE]:CORRection:COLLect:DONE**

**(Write only)** Set calibration to complete.

**Applicable Mode:** Network Analyzer

**Parameter:** None

**Example:** :CORR:COLL:DONE

**Query Syntax:** None

**Default:** None

**Returned Value:** None

**[[:SENSE]:CORRection <string>**

**(Read-Write)** Set or query calibration On / Off.

**Applicable Mode:** Network Analyzer, CAT, VVM

**Parameter:** string, value: OFF (Off), ON (On)

**Example:** :CORR ON

**Query Syntax:** :CORR ON

**Default:** OFF

**Returned Value:** Value (int): 0 (OFF), 1 (ON)

**[[:SENSE]:CORRection:VALid**

**(Read only)** Query calibration coefficient validity.

**Applicable Mode:** Network Analyzer, CAT, VVM

**Parameter:** None

**Example:** :CORR:VALid?

**Default:** OFF

**Returned Value:** Value (int): 0 (Invalid), 1 (Valid)

**[[:SENSe]:CORRection:GAIN <double>**

**(Read-Write)** Set or query measurement offset.

**Applicable Mode:** Power Monitor, USB Power Meter

**Parameter:** Value (double) (dB), range: -50 ~ 50

**Example:** :CORR:GAIN -5

**Query Syntax:** :CORR:GAIN?

**Default:** 0

**Returned Value:** Value (double) (dB)

**[[:SENSe]:CORRection:GAIN:STATe <string>**

**(Read-Write)** Set or query offset On / Off.

**Applicable Mode:** Power Monitor, USB Power Meter

**Parameter:** Offset On / Off: OFF (Off), ON (On)

**Example:** :CORR:GAIN2:STAT ON

**Query Syntax:** :CORR:GAIN2:STAT?

**Default:** OFF

**Returned value:** Value (int): 0 (Off), 1 (On)

**[[:SENSe]:CHPW:STATe <string>**

**(Read-Write)** Set or query channel power On / Off.

**Applicable Mode:** Spectrum Measure

**Parameter:** string, value: OFF (Man), ON (Auto)

**Example:** :CHPW:CHBW ON

**Query Syntax:** :CHPW:CHBW?

**Default:** OFF

**Returned value:** Value (int): 0 (Off), 1 (On)

**[[:SENSe]:CHPW:CHBW <double>**

**(Read-Write)** Set or query channel power BW.

**Applicable Mode:** Spectrum Measure

**Parameter:** Value (double) (Hz), the range is the Span value range of the current model.

**Example:** :CHPW:CHBW 10000

**Query Syntax:** :CHPW:CHBW?

**Default:** None

**Returned Value:** Value (double) (Hz)

**[[:SENSe]:CHPW:TPWR?**

**(Read only)** Query channel power

**Applicable Mode:** Spectrum Measure

**Default:** None

**Returned value:** Value (double) (dBm)

**[[:SENSe]:CHPW:PSDR?**

**(Read only)** Query channel power density.

**Applicable Mode:** Spectrum Measure

**Parameter:** None.

**Default:** None

**Returned Value:** Value (double)

**[[:SENSe]:DATA?**

**(Read only)** Query measured value.

**Applicable Mode:** Power Monitor, USB Power Meter

**Parameter:** None

**Example:** :DATA?

**Query Syntax:** :DATA?

**Default:** None

**Returned Value:** Value (double) (dBm)

**[[:SENSe]:DETECTOR:FUNCTION <string>**

**(Read-Write)** Query or set detection type.

**Applicable Mode:** Spectrum Analyzer

**Parameter:** string, parameter value description:

Set command Parameter (string)	Query command Returned value (int)	Description
POS	0	Positive peak value
NEG	1	Negative peak value
SAM	2	Sample
NOR	3	Normal
AVG	4	Average
RMS	5	Root-mean-square

**Example:** :DET:FUNC NOR

**Query Syntax:** :DET:FUNC?

**Default:** NORM

**Returned Value:** Value (int)

**[[:SENSe]:DETECTOR:FUNCTION:AUTO <string>**

**(Read-Write)** Query or set auto detection On / Off. In auto detection mode, the instrument will automatically select detection type according to different measurements.

**Applicable Mode:** Spectrum Analyzer

**Parameter:** string, value: OFF (Man), ON (Auto)

**Example:** :DET:FUNC:AUTO OFF

**Query Syntax:** :DET:FUNC:AUTO?

**Default:** Auto

**Returned Value:** Value (int): 0 (OFF), 1 (ON)

**[[:SENSe]:FREQuency <double>**

**(Read-Write)** Query or set frequency. In the signal source mode, it is the CW frequency value.

**Applicable Mode:** USB Power Meter, VVM, Signal Source mode

**Parameter:** Value (double) (Hz), it is the detection frequency range of the power probe in the USB Power Meter mode. In the VVM mode and the Signal Source mode, it is the frequency value of the current model.

**Example:** :FREQ 10000

**Query Syntax:** :FREQ?

**Default:** None

**Returned Value:** Value (double) (Hz)

**[[:SENSe]:FREQuency:CENTer <double>**

**(Read-Write)** Query or set center frequency.

**Applicable Mode:** Spectrum Analyzer, Network Analyzer, CAT, Power Meter

**Parameter:** Value (double) (Hz), value range: 0 ~ max sweep frequency (different for different models)

**Example:** :FREQ:CENT 10000

**Query Syntax:** :FREQ:CENT?

**Default:** None

**Returned Value:** Value (double) (Hz)

**[[:SENSe]:FREQuency:SPAN <double>**

**(Read-Write)** Query or set Span.

**Applicable Mode:** Spectrum Analyzer, Network Analyzer, CAT, Power Meter

**Parameter** Value (double) (Hz), value range: 0 ~ Max Span(different for different models).

**Example:** :FREQ:SPAN 10000

**Query Syntax:** :FREQ:SPAN?

**Default:** Max Span of the current model

**Returned Value:** Value (double) (Hz)

**[[:SENSe]:FREQuency:SPAN:FULL**

**(Write only)** Set to Full Span.

**Applicable Mode:** Spectrum Analyzer, Power Meter

**Parameter:** None

**Example:** :FREQ:SPAN:FULL

**Query Syntax:** None

**Returned Value:** None

**[[:SENSe]:FREQuency:SPAN:PREVIOUS**

**(Write only)** Set to the previous Span.

**Applicable Mode:** Spectrum Analyzer, Power Meter

**Parameter:** None

**Example:** :FREQ:SPAN:PREV

**Query Syntax:** None

**Returned Value:** None

**[[:SENSe]:FREQuency:SPAN:ZERO**

**(Write only)** Set to zero Span.

**Applicable Mode:** Spectrum Analyzer

**Parameter:** None

**Example:** :FREQ:SPAN:ZERO

**Query Syntax:** None

**Returned Value:** None

**[[:SENSe]:FREQuency:STARt <double>**

**(Read-Write)** Query or set start frequency.

**Applicable Mode:** Spectrum Analyzer, Network Analyzer, CAT, Power Meter, Signal Source

**Parameter:** Value (double) (Hz), value range: Min sweep frequency ~ max sweep frequency (different sweep ranges for different models)

**Example:** :FREQ:STAR 10000

**Query Syntax:** :FREQ:STAR?

**Default:** Min sweep frequency

**Returned Value:** Value (double) (Hz)

**[[:SENSe]:FREQuency:STOP <double>**

**(Read-Write)** Query or set stop frequency.

**Applicable Mode:** Spectrum Analyzer, Network Analyzer, CAT, Power Meter, Signal Source

**Parameter:** Value (double) (Hz), value range: Min sweep frequency ~ max sweep frequency (different sweep ranges for different models)

**Example:** :FREQ:STOP 10000

**Query Syntax:** :FREQ:STOP?

**Default:** Max sweep frequency

**Returned Value:** Value (double) (Hz)

**[[:SENSe]:FREQuency:CW <double>**

**(Read-Write)** Query or set Signal Source frequency.

**Applicable Mode:** Signal Source

**Parameter:** Value (double) (Hz), it is the frequency value range of the current model

**Example:** :FREQ:CW 10000

**Query Syntax:** :FREQ:CW?

**Default:** Center frequency in max sweep range

**Returned Value:** Value (double) (Hz)

**[[:SENSe]:OBW:METHod <string>**

**(Read-Write)** Set or query measurement method of OBW function.

**Applicable Mode:** Spectrum Analyzer

**Parameter:** string, value: PPOW (percentage), XDB (xdB)

**Example:** :OBW:METH XDB

**Query Syntax:** :OBW:METH?

**Default:** PPOW

**Returned Value:** Value (int): 0 (PPOW), 1 (XDB)

**[[:SENSe]:OBW:OBW?**

**(Read only)** Query OBW value (valid after OBW is on and a sweep is completed).

**Applicable Mode:** Spectrum Analyzer

**Parameter:** None

**Example:** :OBW:OBW?

**Default:** None

**Returned Value:** Value (double) (Hz)

**[[:SENSe]:OBW:PPOW <double>**

**(Read-Write)** Set or query the value corresponding to the OBW percentage.

**Applicable Mode:** Spectrum Analyzer

**Parameter:** Value (double), range: 0.1 ~ 0.9999

**Example:** :OBW:PPOW 0.90

**Query Syntax:** :OBW:PPOW?

**Default:** 0.90

**Returned Value:** Value (double)

**[[:SENSe]:OBW[:STATe] <string>**

**(Read-Write)** Set or query OBW function measurement On / Off.

**Applicable Mode:** Spectrum Analyzer

**Parameters:** string, value: OFF, ON

**Example:** :OBW ON

**Query Syntax:** :OBW?

**Default:** 0 (Off)

**Returned value:** Value (int): 0 (Off), 1 (On)

**[[:SENSe]:OBW:XDB <double>**

**(Read-Write)** Set or query OBW XdB value.

**Applicable Mode:** Spectrum Analyzer

**Parameter:** Value (double) (dB), range: -100 ~ -0.1

**Example:** :OBW:XDB -3

**Query Syntax:** :OBW:XDB?

**Default:** -3

**Returned Value:** Value (double) (dB)

**[[:SENSe]:POWer[:RF]:ATTenuation <double>**

**(Read-Write)** Query or set attenuation value.

**Applicable Mode:** Spectrum Analyzer

**Parameter:** Value (double) (dB), range: 0 ~ 31

**Example:** :POW:ATT 10

**Query Syntax:** :POW:ATT?

**Default:** 10

**Returned Value:** Value (double)



**[[:SENSe]:POWer[:RF]:ATTenuation:AUTO <string>**

**(Read-Write)** Query or set attenuation auto On / Off. When attenuation Auto is turned on, the instrument will set corresponding attenuation according to the reference value automatically.

**Applicable Mode:** Spectrum Analyzer

**Parameter:** string, value: OFF (Off / Auto), ON (On / Auto)

**Example:** :POW:ATT:AUTO ON

**Query Syntax:** :POW:ATT:AUTO?

**Default:** 1 (Auto)

**Returned Value:** Value (int): 0 (Man), 1 (Auto)

**[[:SENSe]:POWer[:RF]:GAIN[:STATe] <string>**

**(Read-Write)** Query or set pre-amplifier to On / Off.

**Applicable Mode:** Spectrum Analyzer, Power Meter

**Parameters:** string, value: OFF, ON

**Example:** :POW:GAIN OFF

**Query Syntax:** :POW:GAIN?

**Default:** 0 (Off)

**Returned value:** Value (int): 0 (Off), 1 (On)

**[[:SENSe]:SWEep:POINts <int>**

**(Read-Write)** Set or query sweep points.

**Applicable Mode:** Network Analyzer, CAT, Signal Source mode

**Parameter:** Value (int), value range: 0 ~ 10001

**Example:** :SWE:POIN 201

**Query Syntax:** :SWE:POIN?

**Default:** 201

**Returned Value:** Value (int)

**[[:SENSe]:SWEep:TIME <double>**

**(Read-Write)** Set or query sweep time of (linear) sweep.

**Applicable Mode:** Spectrum Analyzer, Network Analyzer, CAT

**Parameter:** Value (double) (ms), value range:

In the Spectrum Analyzer mode, at zero Span, the value can be set to: 10 us ~ 600 s, and set as follows when at a non-zero Span: 1 ms ~ 200 s.

In the Network Analyzer and CAT modes, the Min sweep is determined by the settings of sweep points and IF BW, and the max sweep is: Points\*100(s).

**Example:** :SWE:TIME 100

**Query Syntax:** :SWE:TIME?

**Default:** Min sweep time

**Returned Value:** Value (double) (ms)

**[[:SENSe]:SWEep:TIME:AUTO <string>**

**(Read-Write)** Set or query auto On / Off of sweep time for linear sweep.

**Applicable Mode:** Spectrum Analyzer, Network Analyzer, CAT

**Parameters:** string, value: OFF, ON

**Example:** :SWE:TIME:AUTO ON

**Query Syntax:** :SWE:TIME:AUTO?

**Default:** 1 (On)

**Returned value:** Value (int): 0 (Off), 1 (On)

**[[:SENSe]:IA:[:STATe] <string>**

**(Read-Write)** Set or query IA On / Off

**Applicable Mode:** Spectrum mode

**Parameters:** string, value: OFF, ON

**Example:** :IA ON

**Query Syntax:** :IA?

**Default:** 0 (Off)

**Returned value:** Value (int): 0 (Off), 1 (On)

**[[:SENSe]:IA:REDLimit < double>**

**(Read-Write)** Set or query IA red limit value

**Applicable Mode:** Spectrum mode

**Parameter:** Value (double) (dBm), value range: -500 ~ 500

**Example:** :IA:REDLimit 10

**Query Syntax:** :IA:REDLimit?

**Default:** None

**Returned value:** Value (double) (dBm)

**[[:SENSe]:IA:BLUelimit <double>**

**(Read-Write)** Set or query IA blue limit value.

**Applicable Mode:** Spectrum mode

**Parameter:** Value (double) (dBm), value range: -500 ~ 500

**Example:** :IA:BLUelimit 10

**Query Syntax:** :IA:BLUelimit?

**Default:** None

**Returned value:** Value (double) (dBm)

**[[:SENSe]:IA:MODE <string>**

**(Read-Write)** Set or query IA mode.

**Applicable Mode:** Spectrum mode

**Parameter:** string, value: SPECTROGRAM (Spec), WATERFALL (Wtf)

**Example:** :IA:MODE SPECTROGRAM

**Query Syntax:** :IA:MODE?

**Default:** 0 (Spec)

**Returned Value:** Value (int): 0 (Spec), 1 (Wtf)

**[[:SENSe]:IA:CLEar**

**(Write only)** Set to Clear in the IA mode.

**Applicable Mode:** Spectrum mode

**Parameter:** None.

**Example:** :IA:CLear

**Query Syntax:** None

**Default:** None

**Returned Value:** None

**:SOURce:POWer:ALC:MAN <double>**

**(Read-Write)** Set or query manual power value in the Power mode

**Applicable Mode:** Network Analyzer, CAT, Signal Source

**Parameter:** Value (double) (dBm), value range: -50 ~ 10

**Example:** :SOUR:POW:ALC:MAN -5

**Query Syntax:** :SOUR:POW:ALC:MAN?

**Default:**

**Returned value:** Value (double) (dBm)

**:SOURce:POWer:ALC:MODE <string>**

**(Read-Write)** Set or query the Power mode

**Applicable Mode:** Network Analyzer, CAT, Signal Source

**Parameter:** string, parameter value description:

Set command Parameter (string)	Query command Returned value (int)	Description
HIGH	0	High power
LOW	1	Low power
MAN	2	Manual power

**Example:** :SOUR:POW:ALC HIGH

**Query Syntax:** :SOUR:POW:ALC?

**Default:** HIGH

**Returned Value:** Value (int)

**:SOURce:TYPE <string>**

**(Read-Write)** Set or query output mode.

**Applicable Mode:** Signal Source

**Parameter:** string, value: CW (CW Freq), SWEEP (sweep freq)

**Example:** :SOUR:TYPE CW

**Query Syntax:** :SOUR:STAT?

**Default:** 0 (CW)

**Returned Value:** Value (int): 0 (CW), 1 (SWEEP)

**[:SYSTem]:GPS <string>**

**(Read-Write)** Query or set GPS On / Off.

**Applicable Mode:** All modes

**Parameters:** string, value: OFF, ON

**Example:** :GPS ON;

**Query Syntax:** :GPS?

**Default:** 0 (OFF)

**Returned Value:** Value (int): 0 (OFF), 1 (ON)

**[:SYSTem]:GPS:DATA?**

**(Read only)** Return the current GPS data in the following format: <Longitude>, <Latitude>, <Altitude>, <TimeUTC>, e.g.: 38 28'11.22" N,122 42'13.23" W,152,06/28/2010 23:35:38. Return the following if there's no data: --,--,--,--.

**Applicable Mode:** All modes

**Parameter:** None

**Example:** :GPS:DATA?

**Query Syntax:** :GPS:DATA?

**Default:** None

**Returned Value:** string

**[:SYSTem]:GPS:RECEive[:STATe]?**

**(Read only)** Query GPS receiver state and whether there's data.

**Applicable Mode:** All modes

**Parameter:** None

**Example:** :GPS:REC?

**Query Syntax:** :GPS:REC?

**Default:** None

**Returned Value:** Value (int): 0 (with data), 1 (with data)

**[:SYSTem]:GPS:RST**

**(Write only)** GPS cold start, sometimes, for example, when in places where the signal is extremely poor, there is no way to receive the GPS signal for a long time, and sometimes there is no way to change the location. At this time, choose cold start to run the new star search positioning will make the starting speed faster. In this case, you can choose cold start to let the module search for galaxy positioning again..

**Applicable Mode:** All modes

**Parameter:** None

**Example:** :GPS:RST

**Query Syntax:** None

**Default:** None

**Returned Value:** None

**[:SYSTem]:GPS:STATe?**

**(Read only)** Query GPS state.

**Applicable Mode:** All modes

**Parameter:** None

**Example:** :GPS:STAT?

**Query Syntax:** :GPS:STAT?

**Default:** None

**Returned Value:** Value (int), value description:

Query command Returned value (int)	Description
0	Positioning not performed

1	Differential positioning not performed
2	Differential positioning
3	Invalid PPS
4	Estimating

### **:TRACe{n}:DATA?**

**(Read only)** Query trace data. n is the trace number, which can be set to 1, 2 and 3, respectively representing Trace 1, Trace 2 and Trace 3. n is 1 if not specified.

**Applicable Mode:** Spectrum Analyzer

**Parameter:** None

**Example:** :TRAC1:DATA?

**Query Syntax:** :TRAC1:DATA?

**Default:** None

**Returned Value:** Value (float) array, measurement points arranged in order, dBm as the unit

### **:TRACe{n}:TYPE <string>**

**(Read-Write)** Query or set trace state. n is the trace number, which can be set to 1, 2 and 3, respectively representing Trace 1, Trace 2 and Trace 3. n is 1 if not specified.

**Applicable Mode:** Spectrum Analyzer

**Parameter:** string, parameter value description:

Set command Parameter (string)	Query command Returned value (int)	Description
REFRESH	0	Refresh trace
MAXKEEP	1	Max Hold
MINKEEP	2	Min Hold
CURKEEP	3	Hold Trace
HIDE	4	Hide Trace

**Example:** :TRAC2:TYPE REFRESH

**Query Syntax:** :TRAC2:TYPE?

**Default:** REFRESH

**Returned Value:** Value (int), see the table above for detailed definition.

## Chapter II Description of Secondary Development Library Functions

For the convenience of users, we have packaged the SCPI commands and made them into a dynamic link library. Users can easily set or query the S5105D/E/F microwave analyzer by calling the dynamic link library, which is suitable for users to build an automatic testing system. (Note: This dynamic library is generated under LabWindows / CVI 2010 programming environment, and NI's VISA library is used as the communication interface)

### Section I Construction of the Development Environment

Before using the secondary development library, a development environment should be set up, including three steps of installing the required NI-Visa library files, installing the USB driver of the instrument, and adding the secondary development library files. The required files can be found under the "secondary development" folder on the attached CD. This folder includes:

- 1) Standard NI Visa library installation package: NIVISAruntime.msi;
- 2) NI Visa library files: visa32.lib, visa32.dll, visa.h, visatype.h;
- 3) S5105DEF secondary development library files: S5105X.h, AV4957X.lib and AV4957X.dll;
- 4) Related variable definition header files for secondary development; ScpiCommonDef.h;
- 5) S5105DEF Usb drivers: USBDriver.inf, USBDriver\_vista.inf.

#### 1. NI Visa library installation

You can directly install NIVISAruntime.msi, and after the installation, the directory is included in the project. You can also use the NI Visa library files in the CD: visa32.lib, visa32.dll, visa.h, visatype.h. Place the four files in the project directory. Note that visa32.dll and the executable program should be in the same directory.

#### 2 Usb driver installation

If the network cable is used for remote control, the driver does not need to be installed.

If you want to use a USB cable for remote control or secondary development, you need to install the USB driver. The installation method is to double-click USBDriver.inf to install automatically, and select the correct instrument driver through the device manager when connecting the instrument.

#### 3 Adding secondary development library files

Add the four files, av4957xh, av4957x.dll, av4957x.lib, and scpicommondef.h to the project, and then you can control the instrument with the functions in AV5105.h.

## Section II Function Description

### I Instrument Connection Commands

#### Starting the Instrument

**ViStatus \_VI\_FUNC av4957x\_Open (ViRsrc resourceName, ViSession\* handle)**

**Applicable Mode:** All modes

**Function Usage:**

Starting the Instrument

This function is the first function to be called when accessing the instrument driver. It completes the following initialization operations:

According to the interface and logical address information specified by the parameter resourceName, open the handle of the module and connect with the wave comprehensive tester to establish a data channel.

Return handle to identify the module in a subsequent call to the instrument driver function.

**Note that when the instrument is turned on, the Return Parameter format of the instrument will be set to the numerical format. please do not manually set it to the character format during the process, otherwise an error will occur when using the query function.**

**Parameter list:**

handle

Variable type: ViSession\*

Instrument handle returned by the function, communicating with the instrument.

resourceName

Variable type: ViRsrc

Instrument resource character string, AV5105DEF

TCP resource character string is "TCPIP::10.42.130.253::5000::SOCKET", the underlined part is the default IP address of the instrument. If the IP of the instrument is changed, the actual IP shall prevail.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

#### Turning off the Device

**ViStatus \_VI\_FUNC av4957x\_Close (ViSession handle)**

**Function Usage:**

Turn off the instrument; after controlling the instrument, it is required to recall this function to turn off the instrument.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

### II IEEE488.2 General Commands

#### QueryIDN

**ViStatus \_VI\_FUNC av4957x\_QueryIDN (ViSession handle, ViChar IDN[])**

**Function Usage:**

Query instrument ID string.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

**IDN**

Variable type: **ViChar**[][]

The instrument ID string sent from the instrument, in the format of "manufacturer, model, serial number, application version number".

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**Reset**

**ViStatus \_VI\_FUNC av4957x\_Reset (ViSession handle)**

**Function Usage:**

Restore the current working mode of the instrument to the existing default state.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**III Commands of the Math Sub-system**

**SetMeasTarget**

**ViStatus \_VI\_FUNC av4957x\_SetMeasTarget (ViSession handle, ViUInt32 nVal)**

**Applicable Mode:** Network Analyzer, CAT, or VVM

**Function Usage:**

Set measurement type.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nVal

Variable type: int

Measurement type, 0 (S11), 1 (S21), 2 (S12), 3 (S22), 4 (A1), 5 (B1), 6 (R1), 7 (A2), 8 (B2), 9 (R2).

Instrument mode	Set Parameter	Measurement type
Network analyze	S11	Positive reflection measurement
	S21	Positive transmission measurement



	S12	Reverse transmission measurement
	S22	Reverse reflection measurement
	A1, B1, R1, A2, B2, R2	Advanced measurement parameters
Antenna test	S11	Reflection measurement
	A1, B1, R1	Advanced measurement parameters
Vector voltmeter (VVM)	S11	Port 1 reflection measurement
	S21	Port 1 transmission measurement
	S12	Port 2 transmission measurement
	S22	Port 2 reflection measurement

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryMeasTarget**

**ViStatus \_VI\_FUNC av4957x\_QueryMeasTarget (ViSession handle, ViUInt32 nVal[])**

**Applicable Mode:** Network Analyzer, CAT, or VVM

**Function Usage:**

Query measurement type.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nVal

Variable type: ViUInt32 []

Measurement type, consistent with the variable description in "SetMeasTarget".

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryRefVal**

**ViStatus \_VI\_FUNC av4957x\_QueryRefVal (ViSession handle, ViReal64 fVal[])**

**Applicable Mode:** USB Power Meter

**Function Usage:**

Query relative measurement value.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64 []

Relative measured value (dBm).

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetRefOn**

**ViStatus \_VI\_FUNC av4957x\_SetRefOn (ViSession handle, ViBoolean bVal)**

**Applicable Mode:** Power Meter, USB Power Meter, or VVM

**Function Usage:**

Set relative measurement to On / Off.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

bVal

Variable type: ViBoolean

0 means on, and 1 means off.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryRefOn**

**ViStatus \_VI\_FUNC av4957x\_QueryRefOn (ViSession handle, ViBoolean bVal[])**

**Applicable Mode:** Power Meter, USB Power Meter, or VVM

**Function Usage:**

Query relative measurement On / Off status.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

bVal

Variable type: ViBoolean []

0 means on, 1 means off.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**ReadCurTrace (Processed)**

**ViStatus \_VI\_FUNC av4957x\_ReadCurTrace\_Processed (ViSession handle, float pData[], ViInt32 nSize[])**

**Applicable Mode:** Network Analyzer or CAT

**Function Usage:**

Read current trace data (real data after averaging and smoothing, still a complex number in Smith and polar coordinates formats), and in non-Smith and non-polar coordinates formats, set every two numbers as one group, wherein the first number represents the sweep point index and the second represents the trace data; In Smith or polar coordinates, each two numbers are set as a group, representing the real part and imaginary part of a data.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

pData

Variable type: float[]

Trace data stores an array pointer that meets the received trace data size.

nSize

Variable type: ViInt32 []

The actual number of trace data read.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**ReadMemTrace (Processed)**

**ViStatus \_VI\_FUNC av4957x\_ReadMemTrace\_Processed (ViSession handle, float pData[], ViInt32 nSize[])**

**Applicable Mode:** Network Analyzer or CAT

**Function Usage:**

Read memory trace data (real data after averaging and smoothing, still a complex number in Smith and polar coordinates formats), and in non-Smith and non-polar coordinates formats, set every two numbers as one group, wherein the first number represents the sweep point index and the second represents the trace data; In Smith or polar coordinates, each two numbers are set as a group, representing the real part and imaginary part of a data.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

pData

Variable type: float[]

Trace data stores an array pointer that meets the received trace data size.

nSize

Variable type: ViInt32 []

The actual number of trace data read.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**ReadCurTrace**

**iStatus \_VI\_FUNC av4957x\_ReadCurTrace (ViSession handle, float pData[], ViInt32 nSize[]);**

**Applicable Mode:** Network Analyzer or CAT

**Function Usage:**

Read the current trace data (which is the original data after calibration and averaging but not receiving such processing as transformation, format conversion and smoothing, etc.). Every two as a group, representing the real and imaginary parts of a data.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

pData

Variable type: float[]

Trace data stores an array pointer that meets the received trace data size.

nSize

Variable type: ViInt32 []

The number of trace data read.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**ReadMemTrace**

**ViStatus \_VI\_FUNC av4957x\_ReadMemTrace (ViSession handle, float pData[], ViInt32 nSize[])**

**Applicable Mode:** Network Analyzer or CAT

**Function Usage:**

Read the saved trace data (which the data after format conversion and such processing as averaging, transformation, format conversion and smoothing, etc.). Every two as a group, representing the real and imaginary parts of a data.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

pData

Variable type: float[]

Trace data stores an array pointer that meets the received trace data size.

nSize

Variable type: ViInt32 []

The number of trace data read.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetMeasFormat**

**ViStatus \_VI\_FUNC av4957x\_SetMeasFormat (ViSession handle, ViUInt32 nVal)**

**Applicable Mode:** Network Analyzer, CAT, or VVM

**Function Usage:**

Set measurement format, in CAT mode; if the current measurement type is not Re / Im, it will not be switched to the Re / Im mode directly; but it will be switched directly to Re / Im in the Network Analyzer mode.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nVal

Measurement types, 0 (MLOG), 1 (MLIN), 2 (VSWR), 3 (PHASe), 4 (DElay), 5 (SMITh), 6 (POLAr), 7 (CLOSs), 8 (RLOSs), 9 (DTFSwr), 10 (DTFRI), 11 (DB), 12 (REIM).

Measurement mode	Parameter	Measurement type
Vector network analyzer	MLOG	Logarithm magnitude
	MLIN	Linear magnitude
	VSWR	Voltage Standing Wave Ratio
	PHASe	phase
	DElay	Delay
	SMITh	Smith
	POLAr	Polar
Antenna test	VSWR	Voltage Standing Wave Ratio
	PHASe	phase
	DElay	Delay
	SMITh	Smith
	POLAr	Polar
	CLOSs	Cable loss
	RLOSs	Return loss
	DTFSwr	SWR Standing wave ratio
	DTFRI	DTF return loss
Vector voltmeter	DB	DB
	VSWR	Voltage Standing Wave Ratio
	REIM	Impedance

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryMeasFormat**

**ViStatus \_VI\_FUNC av4957x\_QueryMeasFormat (ViSession handle, ViUInt32 nVal[])**

**Applicable Mode:** Network Analyzer, CAT, or VVM

**Function Usage:**

Query measurement format.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nVal

Variable type: ViUInt32[]

Measurement format, consistent with the variable description in "SetMeasTarget".

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

### SetAlarmOn

**ViStatus \_VI\_FUNC av4957x\_SetAlarmOn (ViSession handle, ViBoolean bVal)**

**Applicable Mode:** Network Analyzer, CAT, or Spectrum Analyzer

#### Function Usage:

Set limit alarm to on / off. If the audio alarm is turned on, when the limit test switch is turned on and the test fails, the buzzer of the instrument will give a short audio alarm "Beep" after every sweep.

#### Parameter list:

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

bOn

Variable type: ViBoolean

0 means on, 1 means off.

#### Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

### QueryAlarmOn

**ViStatus \_VI\_FUNC av4957x\_QueryAlarmOn (ViSession handle, ViBoolean bVal[])**

**Applicable Mode:** Network Analyzer, CAT, or Spectrum Analyzer

#### Function Usage:

Query limit alarm on / off status. If the audio alarm is turned on, when the limit test switch is turned on and the test fails, the buzzer of the instrument will give a short audio alarm "Beep" after every sweep.

#### Parameter list:

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

bOn

Variable type: ViBoolean[]

0 means on, 1 means off.

#### Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

### SeLmtTestOn

**ViStatus \_VI\_FUNC av4957x\_SeLmtTestOn (ViSession handle, ViBoolean bVal)**

**Applicable Mode:** Network Analyzer, CAT, or Spectrum Analyzer

#### Function Usage:

Set lower limit test to On / Off.

#### Parameter list:

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

bVal

Variable type: ViBoolean

0 means on, 1 means off.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryLmtTestOn**

**ViStatus \_VI\_FUNC av4957x\_QueryLmtTestOn (ViSession handle, ViBoolean bVal[])**

**Applicable Mode:** Network Analyzer, CAT, or Spectrum Analyzer

**Function Usage:**

Query limit test On / Off status.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

bVal

Variable type: ViBoolean[]

0 means on, 1 means off.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryLmtPass**

**ViStatus \_VI\_FUNC av4957x\_QueryLmtPass (ViSession handle, ViBoolean bVal[])**

**Applicable Mode:** Network Analyzer, CAT, or Spectrum Analyzer

**Function Usage:**

Query whether the limit test is passed or not.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

bVal

Variable type: ViBoolean[]

0=fail, 1=pass.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryLmtPtNum**

**ViStatus \_VI\_FUNC av4957x\_QueryLmtPtNum (ViSession handle, ViUInt32 nVal[])**

**Applicable Mode:** Network Analyzer, CAT, or Spectrum Analyzer

**Function Usage:**

Query number of limit points.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nVal

Variable type: ViUInt32[]

Number of limit points.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**LmtAddPt**

**ViStatus \_VI\_FUNC av4957x\_LmtAddPt (ViSession handle)**

**Applicable Mode:** Network Analyzer, CAT, or Spectrum Analyzer

**Function Usage:**

Add limit points. Add up to 50 limit points

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**LmtClear**

**ViStatus \_VI\_FUNC av4957x\_LmtClear (ViSession handle)**

**Applicable Mode:** Network Analyzer, CAT, or Spectrum Analyzer

**Function Usage:**

Delete all limit points.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**LmtDelPt**

**ViStatus \_VI\_FUNC av4957x\_LmtDelPt (ViSession handle)**

**Applicable Mode:** Network Analyzer, CAT, or Spectrum Analyzer

**Function Usage:**

Delete the currently-selected limit point.

**Parameter list:**



handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**LmtSelectPt**

**ViStatus \_VI\_FUNC av4957x\_LmtSelectPt (ViSession handle, ViUInt32 nVal)**

**Applicable Mode:** Network Analyzer, CAT, or Spectrum Analyzer

**Function Usage:**

Set the limit point to be selected.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nVal

Variable type: ViUInt32

The index number (from 0) to be set as the current limit point.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryLmtSelectPt**

**ViStatus \_VI\_FUNC av4957x\_QueryLmtSelectPt (ViSession handle, ViUInt32 nVal[])**

**Applicable Mode:** Network Analyzer, CAT, or Spectrum Analyzer

**Function Usage:**

Query the selected limit point.

**Parameter list:**

instrumentHandle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nVal

Variable type: ViUInt32[]

Current limit point index (from 0).

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetLmtPtX**

**ViStatus \_VI\_FUNC av4957x\_SetLmtPtX (ViSession handle, ViReal64 fVal)**

**Applicable Mode:** Network Analyzer, CAT, or Spectrum Analyzer

**Function Usage:**

Set the X value of the current limit point.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64

X value of limit point (Hz as the unit in frequency domain measurement, s in time domain measurement and current distance in DTF measurement). The range is the test range of the current measurement domain.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryLmtPtX**

**ViStatus \_VI\_FUNC av4957x\_QueryLmtPtX (ViSession handle, ViReal64 fVal[])**

**Applicable Mode:** Network Analyzer or CAT

**Function Usage:**

Query X value of the current limit point.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64[]

X value of limit point (Hz as the unit in frequency domain measurement, s in time domain measurement and current distance in DTF measurement).

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetLmtPtY**

**ViStatus \_VI\_FUNC av4957x\_SetLmtPtY (ViSession handle, ViReal64 fVal)**

**Applicable Mode:** Network Analyzer or CAT

**Function Usage:**

Set Y value of the current limit point.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64

Y value of limit point. Value range: -500 ~ 500 (the range is irrelevant to the measurement format in Network Analyzer and CAT modes. But in the Spectrum Analyzer mode, the unit is dBm).

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

### QueryLmtPtY

**ViStatus \_VI\_FUNC av4957x\_QueryLmtPtY (ViSession handle, ViReal64 fVal[])**

**Applicable Mode:** Network Analyzer or CAT

**Function Usage:**

Query Y value of the current limit point.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64[]

Y value of limit point .

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

### SetMkrState

**ViStatus \_VI\_FUNC av4957x\_SetMkrState (ViSession handle, ViUInt32 nMkrId, ViUInt32 nMode)**

**Applicable Mode:** Spectrum Analyzer, Network Analyzer, or CAT

**Function Usage:**

Set the state of the specified marker, including Off, Normal, Delta.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nMkrId

Variable type: ViUInt32

Specify the index value of the marker, ranging from 1 to 8, indicating markers 1 to 8 respectively.

nMode

Variable type: ViUInt32

Marker state.

nMode value	Marker state
0	Mkr Off
1	Normal Mkr
2	Delta Mkr

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

### QueryMkrState

**ViStatus \_VI\_FUNC av4957x\_QueryMkrState (ViSession handle, ViUInt32 nMkrId, ViUInt32**

**nMode[])**

**Applicable Mode:** Spectrum Analyzer, Network Analyzer, or CAT

**Function Usage:**

Query the state of the specified marker.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nMkrId

Variable type: ViUInt32

Specify the index value of the marker, ranging from 1 to 8, indicating markers 1 to 8 respectively.

nMode

Variable type: ViUInt32[]

Marker state.

nMode value	Marker state
0	Mkr Off
1	Normal Mkr
2	Delta Mkr

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetMkrAOff**

**ViStatus \_VI\_FUNC av4957x\_SetMkrAOff (ViSession handle)**

**Applicable Mode:** Spectrum Analyzer, Network Analyzer, or CAT

**Function Usage:**

Turn off all markers in the current mode.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetMkrCounterSwitch**

**ViStatus \_VI\_FUNC av4957x\_SetMkrCounterSwitch (ViSession handle, ViUInt32 nMkrId, ViBoolean bVal)**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Set marker counter to On / Off in the current mode, and the set marker will be switched to the Normal state first.

**Note:** Only one marker counter can be turned on currently.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

bVal

Variable type: ViBoolean

On / Off, 0=Off, 1=On.

nMkrId

Variable type: ViUInt32

Specify the index value of the marker, ranging from 1 to 8, indicating markers 1 to 8 respectively.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryMkrCounterSwitch**

**ViStatus \_VI\_FUNC av4957x\_QueryMkrCounterSwitch (ViSession handle, ViUInt32 nMkrId, [])**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Query the marker On / Off state in the current mode.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

bVal

Variable type: ViBoolean[]

On / Off, 0=Off, 1=On.

nMkrId

Variable type: ViUInt32

Specify the index value of the marker, ranging from 1 to 8, indicating markers 1 to 8 respectively.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryMkrCounterXValue**

**ViStatus \_VI\_FUNC av4957x\_QueryMkrCounterXValue (ViSession handle, ViReal64 fVal[])**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Query counter frequency counts (**Invalid when counter is not on or not counting yet**).

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64[]

Return to the counter frequency obtained (Hz).

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SearchMkrToMax**

**ViStatus \_VI\_FUNC av4957x\_SearchMkrToMax (ViSession handle, ViUInt32 nMkrId)**

**Applicable Mode:** Spectrum Analyzer, Network Analyzer, or CAT

**Function Usage:**

Move the marker in the current mode to the position of Max value index.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nMkrId

Variable type: ViUInt32

Specify the index value of the marker, ranging from 1 to 8, indicating markers 1 to 8 respectively.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SearchMkrToMin**

**ViStatus \_VI\_FUNC av4957x\_SearchMkrToMin (ViSession handle, ViUInt32 nMkrId)**

**Applicable Mode:** Spectrum Analyzer, Network Analyzer, or CAT

**Function Usage:**

Move the marker in the current mode to the position of Min value index.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

Variable type: ViUInt32

Specify the index value of the marker, ranging from 1 to 8, indicating markers 1 to 8 respectively.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SearchMkrToPeak**

**ViStatus \_VI\_FUNC av4957x\_SearchMkrToPeak (ViSession handle, ViUInt32 nMkrId)**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Move the marker in the current mode to the position of Max. Peak value index.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nMkrId

Variable type: ViUInt32

Specify the index value of the marker, ranging from 1 to 8, indicating markers 1 to 8 respectively.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SearchMkrToPeakLeft**

**ViStatus \_VI\_FUNC av4957x\_SearchMkrToPeakLeft (ViSession handle, ViUInt32 nMkrId)**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Move the marker in the current mode to the left peak value index of the current position.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nMkrId

Variable type: ViUInt32

Specify the index value of the marker, ranging from 1 to 8, indicating markers 1 to 8 respectively.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SearchMkrToSubPeak**

**ViStatus \_VI\_FUNC av4957x\_SearchMkrToSubPeak(ViSession handle, ViUInt32 nMkrId)**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Move the marker in the current mode to the Sub-Peak value index of the current position.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nMkrId

Variable type: ViUInt32

Specify the index value of the marker, ranging from 1 to 8, indicating markers 1 to 8 respectively.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SearchMkrToPeakRight**

**ViStatus \_VI\_FUNC av4957x\_SearchMkrToPeakRight (ViSession handle, ViUInt32 nMkrId)**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Move the marker in the current mode to the right peak value index of the current position.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nMkrId

Variable type: ViUInt32

Specify the index value of the marker, ranging from 1 to 8, indicating markers 1 to 8 respectively.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetMkrNoiseSwitch**

**ViStatus \_VI\_FUNC av4957x\_SetMkrNoiseSwitch (ViSession handle, ViUInt32 nMkrId, ViBoolean bVal)**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Set the noise marker to On / Off in the current mode.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

bVal

Variable type: ViBoolean

On / Off, 0=Off, 1=On.

nMkrId

Variable type: ViUInt32

Specify the index value of the marker, ranging from 1 to 8, indicating markers 1 to 8 respectively.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryMkrNoiseSwitch**

**ViStatus \_VI\_FUNC QueryMkrNoiseSwitch (ViSession handle, ViUInt32 nMkrId, ViBoolean bVal[])**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Query the noise marker On / Off status in the current mode.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

bVal



Variable type: ViBoolean[]

On / Off, 0=Off, 1=On.

nMkrId

Variable type: ViUInt32

Specify the index value of the marker, ranging from 1 to 8, indicating markers 1 to 8 respectively.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetMkrTo (Marker ->)**

**ViStatus \_VI\_FUNC av4957x\_SetMkrTo (ViSession handle, ViUInt32 nMkrId, ViUInt32 nVal)**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Set marker function in the current mode (which is Mkr -> in the Spectrum Analyzer mode).

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nVal

Variable type: ViUInt32

Marker function type.

nVal	Function
0	Set marker frequency to start frequency
1	Set marker frequency to stop frequency
2	Set marker frequency to center frequency
3	Set marker frequency to step frequency

nMkrId

Variable type: ViUInt32

Specify the index value of the marker, ranging from 1 to 8, indicating markers 1 to 8 respectively.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetMkrXVal**

**ViStatus \_VI\_FUNC av4957x\_SetMkrXVal (ViSession handle, ViUInt32 nMkrId, ViReal64 fVal)**

**Applicable Mode:** Spectrum Analyzer, Network Analyzer, or CAT

**Function Usage:**

Set marker X value in the current mode. X may be negative when the marker is a delta marker.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64

Marker X value, Hz as the unit in the frequency domain mode, s in the time domain mode and the current distance in DTF measurement. The range is current test range.

nMkrId

Variable type: ViUInt32

Specify the index value of the marker, ranging from 1 to 8, indicating markers 1 to 8 respectively.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryMkrXVal**

**ViStatus \_VI\_FUNC av4957x\_QueryMkrXVal (ViSession handle, ViUInt32 nMkrId, ViReal64 fVal[])**

**Applicable Mode:** Spectrum Analyzer, Network Analyzer, or CAT

**Function Usage:**

Query marker X value in the current mode.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64[]

Marker X value, Hz as the unit in the frequency domain mode, s in the time domain mode and the current distance in DTF measurement.

nMkrId

Variable type: ViUInt32

Specify the index value of the marker, ranging from 1 to 8, indicating markers 1 to 8 respectively.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryMkrYVal**

**ViStatus \_VI\_FUNC av4957x\_QueryMkrYVal (ViSession handle, ViUInt32 nMkrId, ViReal64 fVal[])**

**Applicable Mode:** Spectrum Analyzer, Network Analyzer, or CAT

**Function Usage:**

Query marker Y value in the current mode.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64[]

marker Y value.

Two values (double, double) in Network Analyzer and CAT modes and the real part and the imaginary part of the complex number measurement results in Smith or Polar coordinates. The value is the sweep

index value and measured value in other formats.

In the Spectrum Analyzer mode, there are two values (double, double), with the former being the current measured value (adopting the current amplitude unit), and the latter fixed to 0.0

nMkrId

Variable type: ViUInt32

Specify the index value of the marker, ranging from 1 to 8, indicating markers 1 to 8 respectively.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetTraceMathFunc**

**ViStatus \_VI\_FUNC av4957x\_SetTraceMathFunc (ViSession handle, ViUInt32 nVal)**

**Applicable Mode:** Network Analyzer or CAT

**Function Usage:**

Set the trace calculation mode in the current mode.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nVal

Variable type: ViUInt32

Trace calculation and display mode.

nVal	Trace calculation and display mode.
0	No trace calculation, current trace displayed
1	No trace calculation, saved trace displayed (Save Trace must be valid)
2	No trace calculation, current and saved traces displayed (Save Trace must be valid)
3	Current trace-saved trace, displaying the final result (Save Trace must be valid)
4	Current trace+saved trace, displaying the final result (Save Trace must be valid)
5	Current trace / saved trace, displaying the final result (Save Trace must be valid)
6	Source matching calculation of current trace and saved trace, displaying the final result

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryTraceMathFunc**

**ViStatus \_VI\_FUNC av4957x\_QueryTraceMathFunc (ViSession handle, ViUInt32 nVal[])**

**Applicable Mode:** Network Analyzer or CAT

**Function Usage:**

Query the trace calculation mode in the current mode.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nVal

Variable type: ViUInt32[]

Trace calculation and display mode.

nVal	Trace calculation and display mode.
0	No trace calculation, current trace displayed
1	No trace calculation, saved trace displayed (Save Trace must be valid)
2	No trace calculation, current and saved traces displayed (Save Trace must be valid)
3	Current trace-saved trace, displaying the final result (Save Trace must be valid)
4	Current trace+saved trace, displaying the final result (Save Trace must be valid)
5	Current trace / saved trace, displaying the final result (Save Trace must be valid)
6	Source matching calculation of current trace and saved trace, displaying the final result

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**TraceToMemory**

**ViStatus \_VI\_FUNC av4957x\_TraceToMemory (ViSession handle)**

**Applicable Mode:** Network Analyzer or CAT

**Function Usage:**

Save current trace data in the current mode to memory.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetSmoothAper**

**ViStatus \_VI\_FUNC av4957x\_SetSmoothAper (ViSession handle, ViReal64 fVal)**

**Applicable Mode:** Network Analyzer or CAT

**Function Usage:**

Set smoothing aperture.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64

Smoothing aperture (%), value range: 0.01 ~ 20.00.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

### QuerySmoothAper

**ViStatus \_VI\_FUNC av4957x\_QuerySmoothAper (ViSession handle, ViReal64 fVal[])**

**Applicable Mode:** Network Analyzer or CAT

**Function Usage:**

Query smoothing aperture.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64[]

Smoothing aperture (%).

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

### SetSmoothOn

**ViStatus \_VI\_FUNC av4957x\_SetSmoothOn (ViSession handle, ViBoolean bVal)**

**Applicable Mode:** Network Analyzer or CAT

**Function Usage:**

Set smoothing to On / Off.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

bVal

Variable type: ViBoolean

0 means on, 1 means off.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

### QuerySmoothOn

**ViStatus \_VI\_FUNC av4957x\_QuerySmoothOn (ViSession handle, ViBoolean bVal[])**

**Applicable Mode:** Network Analyzer or CAT

**Function Usage:**

Query smoothing On / Off status.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

bVal

Variable type: ViBoolean[]

0 means on, 1 means off.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetCableLoss**

**ViStatus \_VI\_FUNC av4957x\_SetCableLoss (ViSession handle, ViReal64 fVal)**

**Applicable Mode:** CAT

**Function Usage:**

Set cable loss in DTF.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64

Cable loss (the unit, related to the current distance unit, is dB/m or dB/inch), value range: 0.000 ~ 30.000 (dB/m).

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryCableLoss**

**ViStatus \_VI\_FUNC av4957x\_QueryCableLoss (ViSession handle, ViReal64 fVal[])**

**Applicable Mode:** CAT

**Function Usage:**

Query the cable loss in DTF.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64[]

Cable loss value (the unit, related with the current DTF distance unit, is dB/m or dB/inch).

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetStartDist**

**ViStatus \_VI\_FUNC av4957x\_SetStartDist (ViSession handle, ViReal64 fVal)**

**Applicable Mode:** CAT

**Function Usage:**

Set start distance in DTF.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64

Start distance (the unit is the current DTF distance unit).

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryStartDist**

**av4957x\_QueryStartDist (ViSession handle, ViReal64 fVal[])**

**Applicable Mode:** CAT

**Function Usage:**

Query the start distance in DTF. Value range: 0 ~ Max distance. The max. distance is determined by Span, Speed Factor and Number of Points.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64[]

Start distance (the unit is the current DTF distance unit).

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetStopDist**

**ViStatus \_VI\_FUNC av4957x\_SetStopDist (ViSession handle, ViReal64 fVal)**

**Applicable Mode:** CAT

**Function Usage:**

Set the stop distance in DTF.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64

Stop distance (the unit is the current DTF distance unit), value range: 0 ~ Max distance. The max. distance is determined by Span, Speed Factor and Number of Points.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryStopDist**

---

**ViStatus \_VI\_FUNC av4957x\_QueryStopDist (ViSession handle, ViReal64 fVal[])****Applicable Mode:** CAT**Function Usage:**

Query the stop distance in DTF.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64[]

Stop distance (the unit is the current DTF distance unit).

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetDTFUnit****ViStatus \_VI\_FUNC av4957x\_SetDTFUnit (ViSession handle, ViUInt32 nVal)****Applicable Mode:** CAT**Function Usage:**

Set the DTF distance unit.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nVal

Variable type: ViUInt32

DTF unit: 0 for metric, 1 for English.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QuerySetDTFUnit****ViStatus \_VI\_FUNC av4957x\_QuerySetDTFUnit (ViSession handle, ViUInt32 nVal[])****Applicable Mode:** CAT**Function Usage:**

Query the DTF distance unit.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nVal

Variable type: ViUInt32[]

DTF unit, 0 for metric, 1 for English.



**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetTimeTransStartTime**

**ViStatus \_VI\_FUNC av4957x\_SetTimeTransStartTime (ViSession handle, ViReal64 fVal)**

**Applicable Mode:** Network Analyzer

**Function Usage:**

Set time domain start time.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64

Start time (ns), value range: 0 ~ Max test time. Max. test time is determined by current span and number of points.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryTimeTrandStartTime**

**ViStatus \_VI\_FUNC av4957x\_QueryTimeTransStartTime (ViSession handle, ViReal64 fVal[])**

**Applicable Mode:** Network Analyzer

**Function Usage:**

Query time domain start time.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64[]

Start time (ns).

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetTimeTransStopTime**

**ViStatus \_VI\_FUNC av4957x\_SetTimeTransStopTime (ViSession handle, ViReal64 fVal)**

**Applicable Mode:** Network Analyzer

**Function Usage:**

Set time domain stop time.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64

Stop time (ns), value range: 0 ~ Max test time. Max. test time is determined by current span and number of points.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryTimeTransStopTime**

**av4957x\_QueryTimeTransStopTime (ViSession handle, [])**

**Applicable Mode:** Network Analyzer

**Function Usage:**

Query time domain stop time.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64[]

Stop time (ns).

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetTimeTransSwitch**

**ViStatus \_VI\_FUNC av4957x\_SetTimeTransSwitch (ViSession handle, ViBoolean bVal)**

**Applicable Mode:** Network Analyzer

**Function Usage:**

Set time domain to On / Off (**which cannot be on in List Sweep**).

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

bVal

Variable type: ViBoolean

0 means on, 1 means off.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryTimeTransSwitch**

**ViStatus \_VI\_FUNC av4957x\_QueryTimeTransSwitch (ViSession handle, ViBoolean bVal[])**

**Applicable Mode:** Network Analyzer

**Function Usage:**

Query time domain On / Off status.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

bVal

Variable type: ViBoolean[]

0 means on, 1 means off.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetVFactor**

**ViStatus \_VI\_FUNC av4957x\_SetVFactor (ViSession handle, ViReal64 fVal)**

**Applicable Mode:** Network Analyzer or CAT

**Function Usage:**

Set velocity factor in DTF or time domain.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64

Velocity factor, range 0.001 ~ 1.0.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryVFactor**

**ViStatus \_VI\_FUNC av4957x\_QueryVFactor (ViSession handle, ViReal64 fVal[])**

**Applicable Mode:** Network Analyzer or CAT

**Function Usage:**

Query velocity factor in DTF or time domain.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64[]

Velocity factor.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

### SetWinFunc

**ViStatus \_VI\_FUNC av4957x\_SetWinFunc (ViSession handle, ViUInt32 nVal)**

**Applicable Mode:** Network Analyzer or CAT

#### Function Usage:

Set window function in DTF or time domain.

#### Parameter list:

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nVal

Variable type: ViUInt32

Window function type.

Instrument mode	nVal	Window function
Network analyze	0	Max
	1	Normal
	2	Min
	3	Man
Antenna test	4	Rectangular Window
	5	Hanning Window
	6	Hamming Window
	7	Blackman Window
	8	Kaiser Window

#### Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

### QueryWinFunc

**ViStatus \_VI\_FUNC av4957x\_QueryWinFunc (ViSession handle, ViUInt32 nVal[])**

**Applicable Mode:** Network Analyzer or CAT

#### Function Usage:

Query window function in DTF or time domain.

#### Parameter list:

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nVal

Variable type: ViUInt32[]

Window function type.

Instrument mode	nVal	Window function
Network analyze	0	Max
	1	Normal
	2	Min
	3	Man
Antenna test	4	Rectangular Window
	5	Hanning Window
	6	Hamming Window
	7	Blackman Window
	8	Kaiser Window

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**IV Commands of the Calibration Sub-system**

**SetCalibZero**

**ViStatus \_VI\_FUNC av4957x\_SetCalibZero (ViSession handle)**

**Applicable Mode:** USB Power Meter

**Function Usage:**

Start USB Power Meter zeroing (**Do not repeat zeroing during zeroing**).

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryCalibZero**

**ViStatus \_VI\_FUNC av4957x\_QueryCalibZero (ViSession handle, ViUInt32 nVal[])**

**Applicable Mode:** USB Power Meter

**Function Usage:**

Query whether USB Power Meter zeroing is successful (query is not possible during zeroing).

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nVal

Variable type: ViUInt32

Zeroing state.

nVal	Zeroing state
0	Normal
1	Zeroing
2	Zeroing completed
3	Zeroing failed

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**V Commands of the Display Sub-system**

**SetScaleTop**

**ViStatus\_VI\_FUNC av4957x\_SetScaleTop (ViSession handle, ViReal64 fVal)**

**Applicable Mode:** CAT or Network Analyzer

**Function Usage:**

Set top value of scale.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64

Set top value of scale. Scale range:

-500 dB ~ 500 dB (Log)

-10 U ~ 100 U (Linear)

-1000 ns ~ 1000 ns (Group Delay)

-450 ~ 450° (Phase)

-100 ~ 100 (VSWR).

0.1 will be automatically added when set to the min value.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryScaleTop**

**ViStatus\_VI\_FUNC av4957x\_QueryScaleTop (ViSession handle, ViReal64 fVal[])**

**Applicable Mode:** Network Analyzer or CAT

**Function Usage:**

Query top value of scale.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64[]

Top scale value.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetScaleBottom****ViStatus \_VI\_FUNC av4957x\_SetScaleBottom(ViSession handle, ViReal64 fVal)****Applicable Mode:** Network Analyzer or CAT**Function Usage:**

Set bottom value of scale.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64

Bottom value of scale. Scale range:

-500 dB ~ 500 dB (Log)

-10 U ~ 100 U (Linear)

-1000 ns ~ 1000 ns (Group Delay)

-450 ~ 450° (Phase)

-100 ~ 100 (VSWR).

0.1 will be automatically subtracted when set to Max min value.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryScaleBottom****ViStatus \_VI\_FUNC av4957x\_QueryScaleBottom (ViSession handle, ViReal64 fVal[])****Applicable Mode:** Network Analyzer or CAT**Function Usage:**

Query bottom value of scale.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64[]

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetScaleMax****ViStatus \_VI\_FUNC av4957x\_SetScaleMax (ViSession handle, ViReal64 fVal)**

**Applicable Mode:** USB Power Meter, Power Meter

**Function Usage:**

Set Max value of scale.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64

Max value of scale , dBm as the unit, value range: -70 ~ 25.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryScaleMax**

**ViStatus \_VI\_FUNC av4957x\_QueryScaleMax (ViSession handle, ViReal64 fVal[])**

**Applicable Mode:** USB Power Meter, Power Meter

**Function Usage:**

Query max value of scale.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64[]

Max value of scale.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetScaleMin**

**ViStatus \_VI\_FUNC av4957x\_SetScaleMin(ViSession handle, ViReal64 fVal)**

**Applicable Mode:** USB Power Meter, Power Meter

**Function Usage:**

Set min value of scale.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64

Min value of scale, dBm as the unit, value range: -70 ~ 25.

**Returned Value:**



Returned value indicates the execution result of the function: 0 means success, and minus means failure.

### QueryScaleMin

**ViStatus \_VI\_FUNC av4957x\_QueryScaleMin (ViSession handle, ViReal64 fVal[])**

**Applicable Mode:** USB Power Meter, Power Meter

**Function Usage:**

Query min value of scale.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64[]

Min value of scale.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

### AutoScale

**ViStatus \_VI\_FUNC av4957x\_AutoScale (ViSession handle)**

**Applicable Mode:** Network Analyzer, CAT, USB Power Meter, or Power Meter

**Function Usage:**

The instrument automatically adjusts the display scale to suit the observation according to the measured value range.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

### SetScalePDiv

**ViStatus \_VI\_FUNC av4957x\_SetScalePDiv (ViSession handle, ViReal64 fVal)**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Set scale.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64

Scale / Div, dB as the unit, value range: 0.1 ~ 20.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryScalePDiv**

**ViStatus \_VI\_FUNC av4957x\_QueryScalePDiv (ViSession handle, ViReal64 fVal[])**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Query scale.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Scale / Div.

Variable type: ViReal64[]

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetAmplitudeRef (Ref level )**

**ViStatus \_VI\_FUNC av4957x\_SetAmplitudeRef (ViSession handle, ViReal64 fVal)**

**Applicable Mode:** Spectrum Analyzer or Power Meter

**Function Usage:**

Set reference level value (dBm).

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64

Reference level (dBm), value range: -150 ~ 30.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryAmplitudeRef (Ref level )**

**ViStatus \_VI\_FUNC av4957x\_QueryAmplitudeRef (ViSession handle, ViReal64 fVal[])**

**Applicable Mode:** Spectrum Analyzer, Power Meter

**Function Usage:**

Query reference level value (dBm).

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64[]

Reference level (dBm).

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**VI Commands of the Trigger Sub-system**

**SetSwpType**

**ViStatus \_VI\_FUNC av4957x\_SetSwpType (ViSession handle, ViUInt32 nVal)**

**Applicable Mode:** Spectrum Analyzer, Network Analyzer or CAT

**Function Usage:**

Set sweep type.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nVal

Variable type: ViUInt32

Sweep type.

nVal	Sweep type
0	Swp Once
1	Swp Cont

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QuerySwpType**

**ViStatus \_VI\_FUNC av4957x\_QuerySwpType (ViSession handle, ViUInt32 nVal[])**

**Applicable Mode:** Spectrum Analyzer, Network Analyzer or CAT

**Function Usage:**

Query sweep type.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nVal

Variable type: ViUInt32[]

Sweep type.

nVal	Sweep type
------	------------

0	Swp Once
1	Swp Cont

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**TrigerSwp**

**ViStatus \_VI\_FUNC av4957x\_TrigerSwp (ViSession handle)**

**Applicable Mode:** Spectrum Analyzer, Network Analyzer, or CAT

**Function Usage:**

Switch to Swp Once, and trigger a single sweep.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**HoldSwp**

**ViStatus \_VI\_FUNC av4957x\_HoldSwp (ViSession handle)**

**Applicable Mode:** Spectrum Analyzer, Network Analyzer, or CAT

**Function Usage:**

Hold the current sweep.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**VII Commands of the Device Sub-system**

**QueryInstCatalog**

**ViStatus \_VI\_FUNC av4957x\_QueryInstCatalog (ViSession handle, ViUInt32 nVal[])**

**Applicable Mode:** All modes

**Function Usage:**

Query available instrument mode.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nVal

Variable type: ViUInt32[]

Available mode identification. This value is converted from a 32-bit binary number. If a bit value is 1, the mode corresponding to that bit is valid. Where:

Bit	Measurement mode
1	Antenna test
2	Spectrum Analyzer
3	Network analyze
4	Power Meter
5	USB Power Meter
6	Vector voltmeter (VVM)
7	Signal source

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetInstSel**

**ViStatus \_VI\_FUNC av4957x\_SetInstSel (ViSession handle, ViUInt32 nVal)**

**Applicable Mode:** All modes

**Function Usage:**

Set instrument mode.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nVal

Variable type: ViUInt32

Instrument mode identification.

nVal	Measurement mode
1	Antenna test
2	Spectrum Analyzer
3	Network analyze
4	Power Meter
5	USB Power Meter
6	Vector voltmeter (VVM)
7	Signal source

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryInstSel**

**ViStatus \_VI\_FUNC av4957x\_QueryInstSel (ViSession handle, ViUInt32 nVal[])**

**Applicable Mode:** All modes

**Function Usage:**

Query instrument mode.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nVal

Variable type: ViUInt32[]

Instrument mode identification.

nVal	Measurement mode
1	Antenna test
2	Spectrum Analyzer
3	Network analyze
4	Power Meter
5	USB Power Meter
6	Vector voltmeter (VVM)
7	Signal source

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**VIII Commands of the Memory Sub-system**

**DelStateFile**

**ViStatus \_VI\_FUNC av4957x\_DelStateFile (ViSession handle, ViChar\* pVal)**

**Applicable Mode:** All modes

**Function Usage:**

Delete the state files in the current mode (the command is invalid if the file does not exist and is valid only for the current storage location).

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

pVal

Variable type: ViChar\*

File name.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**DelAllStateFile**

**ViStatus \_VI\_FUNC av4957x\_DelAllStateFile (ViSession handle)**

**Applicable Mode:** All modes

**Function Usage:**

Delete all state files in the current mode.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**DelTraceFile**

**ViStatus \_VI\_FUNC av4957x\_DelTraceFile (ViSession handle, ViChar\* pVal)**

**Applicable Mode:** Network Analyzer or CAT

**Function Usage:**

Delete trace files in the current mode (**the command is invalid if the file does not exist** and is valid only for the current storage location).

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

pVal

Variable type: ViChar\*

File name.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**DelAllTraceFile**

**ViStatus \_VI\_FUNC av4957x\_DelAllTraceFile (ViSession handle)**

**Applicable Mode:** Network Analyzer or CAT

**Function Usage:**

Delete all trace files in the current mode.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**DelPictureFile**

**ViStatus \_VI\_FUNC av4957x\_DelPictureFile (ViSession handle, ViChar\* pVal)**

**Applicable Mode:** All modes

**Function Usage:**

Delete specified screenshots.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

pVal

Variable type: ViChar\*

File name.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**DelAllPictureFile**

**ViStatus \_VI\_FUNC av4957x\_DelAllPictureFile (ViSession handle)**

**Applicable Mode:** All modes

**Function Usage:**

Delete all screenshot files.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**LoadStateFile**

**ViStatus \_VI\_FUNC av4957x\_LoadStateFile (ViSession handle, ViChar\* pVal)**

**Applicable Mode:** All modes

**Function Usage:**

Recall state files in the current mode (**the command is invalid if the file does not exist** and is valid only for the current storage location).

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

pVal

Variable type: ViChar\*

File name.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**LoadTraceFile**

**ViStatus \_VI\_FUNC av4957x\_LoadTraceFile (ViSession handle, ViChar\* pVal)**

**Applicable Mode:** Network Analyzer or CAT

**Function Usage:**

Recall trace files in the current mode (**the command is invalid if the file does not exist and is valid only for the current storage location; in case of inconsistency between the current state and the loaded state, the**



recalled state will be loaded).

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

pVal

Variable type: ViChar\*

File name.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetFileLocation**

**ViStatus \_VI\_FUNC av4957x\_SetFileLocation (ViSession handle, ViUInt32 nVal)**

**Applicable Mode:** All modes

**Function Usage:**

Set storage location (which cannot be set to USB or SD card in the absence of USB storages or SD cards).

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nVal

Variable type: ViUInt32

Storage location.

nVal	Storage location
0	USB drive
1	SD card
2	Internal

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryFileLocation**

**ViStatus \_VI\_FUNC av4957x\_QueryFileLocation (ViSession handle, ViUInt32 nVal[])**

**Applicable Mode:** All modes

**Function Usage:**

Query the current storage location.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nVal

Variable type: ViUInt32[]

Storage location.

nVal	Storage location
0	USB drive
1	SD card
2	Internal

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**StoreScreen**

**ViStatus\_VI\_FUNC av4957x\_StoreScreen (ViSession handle, ViChar\* pVal)**

**Applicable Mode:** All modes

**Function Usage:**

Copy the screen, that is, to store the current screenshot as a file (the file will overwrite an existing file and is valid only for the current storage location).

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

pVal

Variable type: ViChar\*

File name.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**StoreStateFile**

**ViStatus\_VI\_FUNC av4957x\_StoreStateFile (ViSession handle, ViChar\* pVal)**

**Applicable Mode:** All modes

**Function Usage:**

Save the state in the current mode as a file (the file will overwrite an existing file and is valid only for the current storage location).

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

pVal

Variable type: ViChar\*

File name.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**StoreTraceFile**

---

**ViStatus \_VI\_FUNC av4957x\_StoreTraceFile (ViSession handle, ViChar\* pVal)**

**Applicable Mode:** Network Analyzer or CAT

**Function Usage:**

Save the trace in the current mode as a file (the file will overwrite an existing file and is valid only for the current storage location), and the state will be saved at the same time, and trace storage in the Spectrum Analyzer mode is also affected by settings of trace source.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

pVal

Variable type: ViChar\*

File name.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

### IX Commands of the Sensor Subsystem

#### ACPRSetSwitch

**ViStatus \_VI\_FUNC av4957x\_ACPRSetSwitch (ViSession handle, ViBoolean bVal)**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Set ACPR to On / Off.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

bVal

Variable type: ViBoolean

0 means on, 1 means off.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

#### ACPRQuerySwitch

**ViStatus \_VI\_FUNC av4957x\_ACPRQuerySwitch (ViSession handle, ViBoolean bVal[])**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Query ACPR On / Off status.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

bVal

Variable type: ViBoolean[]

0 means on, 1 means off.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**ACPRSetMainCHBW**

**ViStatus \_VI\_FUNC av4957x\_ACPRSetMainCHBW (ViSession handle, ViReal64 fVal)**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Set the main channel bandwidth of the ACPR function measurement in the Spectrum Analyzer mode.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64

Frequency value (Hz as the unit), value range: Min Span ~ Max Span (depending on specific model).

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**ACPRQueryMainCHBW**

**ViStatus \_VI\_FUNC av4957x\_ACPRQueryMainCHBW (ViSession handle, ViReal64 fVal[])**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Query the main channel bandwidth of the ACPR function measurement in the Spectrum Analyzer mode.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64[]

Frequency value (Hz).

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**ACPRSetAdjCHBW**

**ViStatus \_VI\_FUNC av4957x\_ACPRSetAdjCHBW (ViSession handle, ViReal64 fVal)**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Set the adjacent channel bandwidth of the ACPR function measurement in the Spectrum Analyzer mode.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64

Frequency value (Hz), value range: 0 ~ (current span - 2\* Channel spacing).

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**ACPRQueryAdjCHBW**

**ViStatus \_VI\_FUNC av4957x\_ACPRQueryAdjCHBW (ViSession handle, ViReal64 fVal[])**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Query the adjacent channel bandwidth of the ACPR function measurement in the Spectrum Analyzer mode.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64[]

Frequency value (Hz).

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**ACPRSetCHSpace**

**ViStatus \_VI\_FUNC av4957x\_ACPRSetCHSpace (ViSession handle, ViReal64 fVal)**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Set the channel spacing width of the ACPR function measurement in the Spectrum Analyzer mode.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64

Frequency value (Hz), value range: 0 ~ (current span-main channel bandwidth) / 2.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

### ACPRQueryCHSpace

**ViStatus\_VI\_FUNC av4957x\_ACPRQueryCHSpace (ViSession handle, ViReal64 fVal[])**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Query the channel spacing width of the ACPR function measurement in the Spectrum Analyzer mode.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64[]

Frequency value (Hz).

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

### ACPRQueryMainCHPower

**ViStatus\_VI\_FUNC av4957x\_ACPRQueryMainCHPower (ViSession handle, ViReal64 fVal[])**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Query the main channel power.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64[]

Main channel power (dBm).

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

### ACPRQueryUpCHPower

**ViStatus\_VI\_FUNC av4957x\_ACPRQueryUpCHPower (ViSession handle, ViReal64 fVal[])**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Query the upper channel power.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64[]

Upper channel power (dBm).

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**ACPRQueryDownCHPower**

**ViStatus \_VI\_FUNC av4957x\_ACPRQueryDownCHPower (ViSession handle, ViReal64 fVal[])**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Query the lower channel power.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64

Lower channel power (dBm).

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**ACPRQueryUpACPR**

**ViStatus \_VI\_FUNC av4957x\_ACPRQueryUpACPR (ViSession handle, ViReal64 fVal[]);**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Query the upper channel power ratio.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64[]

Upper channel power ratio.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**ACPRQueryDownACPR**

**ViStatus \_VI\_FUNC av4957x\_ACPRQueryDownACPR (ViSession handle, ViReal64 fVal[])**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Query the lower channel power ratio.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64

Lower channel power ratio.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SASStartZeroCal**

**ViStatus \_VI\_FUNC av4957x\_SASStartZeroCal (ViSession handle)**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Perform zero calibration.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**FSTSetAntenaOff (Field Strength)**

**ViStatus \_VI\_FUNC av4957x\_FSTSetAntenaOff (ViSession handle)**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Turn off antenna factor loading and set it to no-antenna factor state.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**STSetSwitch**

**ViStatus \_VI\_FUNC av4957x\_STSetSwitch (ViSession handle, ViBoolean bVal)**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Set field strength function measurement to On / Off.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.



bVal

Variable type: ViBoolean

0 means on, 1 means off.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**FSTQuerySwitch**

**ViStatus \_VI\_FUNC av4957x\_FSTQuerySwitch (ViSession handle, ViBoolean bVal[])**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Query field strength function measurement On / Off status.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

bVal

Variable type: ViBoolean[]

0 means on, 1 means off.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**FSTSetAntena**

**ViStatus \_VI\_FUNC av4957x\_FSTSetAntena (ViSession handle, ViChar\* pVal)**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Set field strength antenna factor.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

pVal

Variable type: ViChar\*

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**FSTQueryAntena**

**ViStatus \_VI\_FUNC av4957x\_FSTQueryAntena (ViSession handle, ViChar\* pVal)**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Query field strength antenna factor.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

pVal

Variable type: ViChar\*

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetAmplitudeScaleType**

**ViStatus \_VI\_FUNC av4957x\_SetAmplitudeScaleType (ViSession handle, ViUInt32 nVal)**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Set scale type.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nVal

Variable type: ViUInt32

Scale type.

nVal	Scale type
0	Logarithm scale
1	Linear scale

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryAmplitudeScaleType**

**ViStatus \_VI\_FUNC ViStatus \_VI\_FUNC av4957x\_QueryAmplitudeScaleType (ViSession handle, ViUInt32 nVal[])**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Query scale type.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nVal

Variable type: ViUInt32[]

Scale type.

nVal	Scale type
0	Logarithm scale

1	Linear scale
---	--------------

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetAmplitudeUnit**

**ViStatus \_VI\_FUNC av4957x\_SetAmplitudeUnit (ViSession handle, ViUInt32 nVal)**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Set amplitude unit.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nVal

Variable type: ViUInt32

Amplitude unit.

nVal	Amp Unit
0	dBm
1	dBmV
2	dBuV
3	Volts
4	Watts

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryAmplitudeUnit**

**ViStatus \_VI\_FUNC av4957x\_QueryAmplitudeUnit (ViSession handle, ViUInt32 nVal[])**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Query amplitude unit.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nVal

Variable type: ViUInt32[]

nVal	Amp Unit
0	dBm
1	dBmV
2	dBuV

3	Volts
4	Watts

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetAvgOn**

**ViStatus \_VI\_FUNC av4957x\_SetAvgOn (ViSession handle, ViBoolean bVal)**

**Applicable Mode:** Spectrum Analyzer, Network Analyzer, CAT, or Power Meter

**Function Usage:**

Set averaging to On / Off.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

bVal

Variable type: ViBoolean

0 means on, 1 means off.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryAvgOn**

**ViStatus \_VI\_FUNC av4957x\_QueryAvgOn (ViSession handle, ViBoolean nVal[])**

**Applicable Mode:** Spectrum Analyzer, Network Analyzer, CAT, or Power Meter

**Function Usage:**

Query average On / Off status.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nVal

Variable type: ViBoolean[]

0 means on, 1 means off.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetAvgFactor**

**ViStatus \_VI\_FUNC av4957x\_SetAvgFactor (ViSession handle, ViUInt32 nVal)**

**Applicable Mode:** Spectrum Analyzer, Network Analyzer, CAT, or Power Meter

**Function Usage:**

Set averaging factor.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nVal

Variable type: ViUInt32

Averaging factor, range 2 ~ 999.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryAvgFactor**

**ViStatus \_VI\_FUNC av4957x\_QueryAvgFactor (ViSession handle, ViUInt32 nVal[])**

**Applicable Mode:** Spectrum Analyzer, Network Analyzer, CAT, or Power Meter

**Function Usage:**

Query averaging count.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nVal

Variable type: ViUInt32[]

Averaging factor.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**ClearAvgCount**

**ViStatus \_VI\_FUNC av4957x\_ClearAvgCount (ViSession handle)**

**Applicable Mode:** Spectrum Analyzer, Network Analyzer, CAT, or Power Meter

**Function Usage:**

Restart averaging, with current averaging counting from 1.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetRBW**

**ViStatus \_VI\_FUNC av4957x\_SetRBW (ViSession handle, ViReal64 fVal)**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Set RBW in the Spectrum Analyzer mode.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64

Frequency value (Hz), value range: 1 ~ 5000000 (set valid value to 1 Hz ~ 5 MHz, with the discrete value at the step of 1, 3, 10).

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryRBW**

**ViStatus \_VI\_FUNC av4957x\_QueryRBW (ViSession handle, ViReal64 fVal[])**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Query RBW in the Spectrum Analyzer mode.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64[]

Frequency value (Hz).

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetVBW**

**ViStatus \_VI\_FUNC av4957x\_SetVBW (ViSession handle, ViReal64 fVal)**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Set VBW in the Spectrum Analyzer mode.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64

Frequency value (Hz), value range: 1 ~ 5000000 (set valid value to 1 Hz ~ 5 MHz, with the discrete value at the step of 1, 3, 10).

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

### QueryVBW

**ViStatus\_VI\_FUNC av4957x\_QueryVBW (ViSession handle, ViReal64 fVal[])**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Query VBW in the Spectrum Analyzer mode.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64[]

Frequency value (Hz).

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

### SetRBWAuto

**ViStatus\_VI\_FUNC av4957x\_SetRBWAuto (ViSession handle, ViBoolean bVal)**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Set RBW to be auto On / Off. When set to Auto, RBW will adjust RBW by bandwidth according to the ratio of SPAN / RBW.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

bVal

Variable type: ViBoolean

0 means on, 1 means off.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

### QueryRBWAuto

**ViStatus\_VI\_FUNC av4957x\_QueryRBWAuto (ViSession handle, ViBoolean bVal[])**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Query auto On / Off status of RBW.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

bVal

Variable type: ViBoolean[]

0 means on, 1 means off.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetVBWAuto**

**ViStatus \_VI\_FUNC av4957x\_SetVBWAuto (ViSession handle, ViBoolean bVal)**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Set auto on / off of VBW. When set to Auto, VBW will adjust RBW by resolution bandwidth according to the ratio of RBW / VBW.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

bVal

Variable type: ViBoolean

0 means on, 1 means off.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryVBWAuto**

**ViStatus \_VI\_FUNC av4957x\_QueryVBWAuto (ViSession handle, ViBoolean bVal[])**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Query auto On / Off status of VBW.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

bVal

Variable type: ViBoolean[]

0 means on, 1 means off.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetSPANRBW\_Ratio**

**ViStatus \_VI\_FUNC av4957x\_SetSPANRBW\_Ratio (ViSession handle, ViUInt32 nVal)**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Set the value of SPAN / RBW in the Spectrum Analyzer mode.



**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nVal

Variable type: ViUInt32

SPAN / RBW value, range 1 ~ 500.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QuerySPANRBW\_Ratio**

**ViStatus \_VI\_FUNC av4957x\_QuerySPANRBW\_Ratio (ViSession handle, ViUInt32 nVal[])**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Query the SPAN / RBW value in the Spectrum Analyzer mode.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nVal

Variable type: ViUInt32[]

SPAN / RBW value.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetRBWVBW\_Ratio**

**ViStatus \_VI\_FUNC av4957x\_SetRBWVBW\_Ratio (ViSession handle, ViUInt32 nVal)**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Set RBW / VBW value in the Spectrum Analyzer mode.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nVal

Variable type: ViUInt32

RBW / VBW value, range 1 ~ 100.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryRBWVBW\_Ratio**

**ViStatus\_VI\_FUNC av4957x\_QueryRBWVBW\_Ratio (ViSession handle, ViUInt32 nVal[])****Applicable Mode:** Spectrum Analyzer**Function Usage:**

Query RBW / VBW value in the Spectrum Analyzer mode.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nVal

Variable type: ViUInt32[]

RBW / VBW value.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetIFBW****ViStatus\_VI\_FUNC av4957x\_SetIFBW (ViSession handle, ViReal64 fVal)****Applicable Mode:** Network Analyzer or CAT**Function Usage:**

Set IF BW in the current mode.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64

Frequency value (Hz), value range: 1 Hz ~ 100 kHz, with the discrete value at the step of 1, 3, 10.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryIFBW****ViStatus\_VI\_FUNC av4957x\_QueryIFBW (ViSession handle, ViReal64 fVal[])****Applicable Mode:** Network Analyzer or CAT**Function Usage:**

Query IF BW in the current mode.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64[]

Frequency value (Hz).

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetCalMethod**

**ViStatus \_VI\_FUNC av4957x\_SetCalMethod (ViSession handle, ViUInt32 nVal)**

**Applicable Mode:** Network Analyzer, VVM, or CAT

**Function Usage:**

Set current calibration type.

**Parameter list:**

Handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nVal

Variable type: ViUInt32

Calibration type, which determines the calibration standard for the subsequent measurement. It is required to complete the measurement by recalling av4957x\_CalCollFinish (ViSession handle) according to the corresponding measurement standards in the table.

Instrument mode	Parameter	Calibration Type	Standard Steps
All modes	0	NONE	
Network analyze	1	Resp.&Iso@S12	THRU, ISO
	2	Resp.&Iso@S21	THRU, ISO
	3	Freq Resp. S11 Open	Open
	4	Freq Resp. S22 Open	Open
	5	Freq Resp. S11 Short	SHORT
	6	Freq Resp. S22 Short	SHORT
	7	Freq Resp. S12 THRU	THRU
	8	Freq Resp. S21 THRU	THRU
	9	Full 2 ports calibration	Reflection (P1 Open, Short, Load, P2 Open, Short, Load), Transmission (THRU), Iso
	11	S11 single port calibration	P1 Open, Short, Load
12	S22 single port calibration	P2 Open, Short, Load	
Antenna test	11	S11 single port	P1 Open, Short, Load
Vector voltmeter	9	Full 2 ports calibration	Reflection (P1 Open, Short, Load, P2 Open, Short, Load), Transmission (THRU), Iso
	11	S11 single port calibration	P1 Open, Short, Load
	12	S22 single port calibration	P2 Open, Short, Load

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryCalMethod**

**ViStatus \_VI\_FUNC av4957x\_QueryCalMethod (ViSession handle, ViUInt32 nVal[])**

**Applicable Mode:** Applicable Mode: Network Analyzer, CAT, or VVM

**Function Usage:**

Query calibration type.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nVal

Variable type: ViUInt32[]

Calibration type, see function description in “SetCalMethord” for parameter description.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**CalCollIso**

**ViStatus \_VI\_FUNC av4957x\_CalCollIso (ViSession handle, ViUInt32 iStep)**

**Applicable Mode:** Network Analyzer, VVM

**Function Usage:**

Collect isolation calibration data (which needs to be corresponding to calibration type; calibration type is only valid during response, isolation, or in the Full 2 Ports mode).

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

iStep

Variable type: ViUInt32

nVal	Collect isolation data type
0	Automatically collect positive isolation and reverse isolation calibration coefficient during full 2 ports calibration; automatically collect isolation data in the measurement type during response and isolation calibration
1	Ignore isolation (valid only in Full 2 Ports mode)
2	Positive isolation (valid only in Full 2 Ports mode)
3	Reverse isolation (valid only in Full 2 Ports mode)

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryCalCollIso**

**ViStatus \_VI\_FUNC av4957x\_QueryCalCollIso (ViSession handle, ViBoolean bVal[])**

**Applicable Mode:** Network Analyzer, VVM

**Function Usage:**

Query whether isolation calibration is done.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

bVal

Variable type: ViBoolean

0=Undone, 1=done.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**CalCollLoad**

**ViStatus \_VI\_FUNC av4957x\_CalCollLoad (ViSession handle, ViUInt32 idPort)**

**Applicable Mode:** Network Analyzer, CAT, or VVM

**Function Usage:**

Collect load calibration data (which needs to be corresponding to calibration type; calibration type is valid only in calibration modes of S11 single port, S22 single port and full 2 ports calibration).

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

idPort

Variable type: ViUInt32

Port number, value 1 or 2.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryCalCollLoad**

**ViStatus \_VI\_FUNC av4957x\_QueryCalCollLoad (ViSession handle, ViUInt32 idPort, ViBoolean bVal[])**

**Applicable Mode:** Network Analyzer, CAT, or VVM

**Function Usage:**

Query whether the load calibration is completed.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

idPort

Variable type: ViUInt32

Port number, value 1 or 2.

bVal

Variable type: ViBoolean

0=Undone, 1=done.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

### CalCollOpen

**ViStatus \_VI\_FUNC av4957x\_CalCollOpen (ViSession handle, ViUInt32 idPort)**

**Applicable Mode:** Network Analyzer, CAT, or VVM

#### Function Usage:

Collect open-circuit device calibration data (which needs to be corresponding to calibration type; calibration type is valid only in calibration modes of S11 single port, S22 single port, full 2 ports calibration and frequency response).

#### Parameter list:

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

idPort

Variable type: ViUInt32

Port number, value 1 or 2.

#### Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

### QueryCalCollOpen

**ViStatus \_VI\_FUNC av4957x\_QueryCalCollOpen (ViSession handle, ViUInt32 idPort, ViBoolean bVal[])**

**Applicable Mode:** Network Analyzer, CAT, or VVM

#### Function Usage:

Query whether the load calibration is completed.

#### Parameter list:

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

idPort

Variable type: ViUInt32

Port number, value 1 or 2.

bVal

Variable type: ViBoolean

0=Undone, 1=done.

#### Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

### CalCollShort

**ViStatus \_VI\_FUNC av4957x\_CalCollShort (ViSession handle, ViUInt32 idPort)**

**Applicable Mode:** Network Analyzer, CAT, or VVM

#### Function Usage:

Collect short circuit device calibration data (which needs to be corresponding to calibration type; calibration type is valid only in calibration modes of S11 single port, S22 single port, full 2 ports calibration and frequency response).

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

idPort

Variable type: ViUInt32

Port number, value 1 or 2.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryCalCollShort**

**ViStatus \_VI\_FUNC av4957x\_QueryCalCollShort (ViSession handle, ViUInt32 idPort, ViBoolean bVal[])**

**Applicable Mode:** Network Analyzer, CAT, or VVM

**Function Usage:**

Query whether the load calibration is completed.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

idPort

Variable type: ViUInt32

Port number, value 1 or 2.

bVal

Variable type: ViBoolean

0=Undone, 1=done.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**CalCollThru**

**ViStatus \_VI\_FUNC av4957x\_CalCollThru (ViSession handle, ViUInt32 iStep)**

**Applicable Mode:** Network Analyzer, VVM

**Function Usage:**

Collect THRU calibration data (which needs to be corresponding to calibration type; calibration type is valid only in calibration modes of frequency response, full 2 ports, response and isolation).

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

iStep

Variable type: ViUInt32

nVal	Collect THRU data type
0	Automatically collect positive matching, positive transmission, reverse transmission, and reserve matching calibration coefficients during full 2 ports calibration, automatically collect THRU data in the measurement type during frequency response, response and isolation calibration
1	Collect positive matching calibration data ( <b>valid only in Full 2 Ports mode</b> )
2	Collect positive transmission calibration data ( <b>valid only in Full 2 Ports mode</b> )
3	Collect negative transmission calibration data ( <b>valid only in Full 2 Ports mode</b> )
4	Collect negative matching calibration data ( <b>valid only in Full 2 Ports mode</b> )

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryCalCollThru**

**ViStatus \_VI\_FUNC av4957x\_QueryCalCollThru (ViSession handle, ViBoolean bVal[])**

**Applicable Mode:** Network Analyzer, VVM

**Function Usage:**

Query whether isolation calibration is done.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

bVal

Variable type: ViBoolean

0=Undone, 1=done.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetCalKit**

**ViStatus \_VI\_FUNC av4957x\_SetCalKit (ViSession handle, ViChar\* pVal)**

**Applicable Mode:** Network Analyzer, CAT, or VVM

**Function Usage:**

Set calibration kit type.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

pVal

Variable type: ViChar\*

Calibration kit, including: AV31101A, AV31101B, AV31121, AV31123, AV31101A/B.

**Returned Value:**



Returned value indicates the execution result of the function: 0 means success, and minus means failure.

### QueryCalKit

**ViStatus \_VI\_FUNC av4957x\_QueryCalKit (ViSession handle, ViChar \*pVal)**

**Applicable Mode:** Network Analyzer, CAT, or VVM

**Function Usage:**

Query calibration kit type.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

pVal

Variable type: ViChar\*

Calibration kit name.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

### CalCollFinish

**ViStatus \_VI\_FUNC av4957x\_CalCollFinish (ViSession handle)**

**Applicable Mode:** Network Analyzer, CAT, or VVM

**Function Usage:**

Complete calibration, calculate calibration coefficient, and enable the calibration on / off status (**which is valid after current calibration data collection**).

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

### SetCalOn

**ViStatus \_VI\_FUNC av4957x\_SetCalOn (ViSession handle, ViBoolean bVal)**

**Applicable Mode:** Network Analyzer, CAT, or VVM

**Function Usage:**

Set calibration to On / Off (**which can only be turned on after calibration data collection and calculation are completed**).

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

bVal

Variable type: ViBoolean

0 means on, 1 means off.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryCalOn**

**ViStatus \_VI\_FUNC av4957x\_QueryCalOn (ViSession handle, ViBoolean bVal[])**

**Applicable Mode:** Network Analyzer, CAT, or VVM

**Function Usage:**

Query calibration On / Off status.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

bVal

Variable type: ViBoolean[]

0 means on, 1 means off.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryCalValid**

**ViStatus \_VI\_FUNC av4957x\_QueryCalValid (ViSession handle, ViBoolean bVal[])**

**Applicable Mode:** Network Analyzer, CAT, or VVM

**Function Usage:**

Query validity of calibration coefficient.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

bVal

Variable type: ViBoolean[]

0 means invalid, 1 means valid.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetMCalKitMatch**

**ViStatus \_VI\_FUNC av4957x\_SetMCalKitMatch (ViSession handle, ViUInt32 nVal)**

**Applicable Mode:** Network Analyzer, CAT, or VVM

**Function Usage:**

Set matching mode of calibration kit. When the calibration kit is a combined calibration kit, it is required to recall this function to specify the currently used interface type.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nVal

Variable type: ViUInt32

0 means male, 1 means female.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetOffset**

**ViStatus \_VI\_FUNC av4957x\_SetOffset (ViSession handle, ViReal64 fVal)**

**Applicable Mode:** Power Meter, USB Power Meter

**Function Usage:**

Set offset value.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64

Offset (dB), value range: -50 ~ 50.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryOffset**

**ViStatus \_VI\_FUNC av4957x\_QueryOffset (ViSession handle, ViReal64 fVal[])**

**Applicable Mode:** Power Meter, USB Power Meter

**Function Usage:**

Query offset value.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64[]

Offset (dB).

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetOffsetOn**

**ViStatus \_VI\_FUNC av4957x\_SetOffsetOn (ViSession handle, ViBoolean bVal)**

**Applicable Mode:** Power Meter, USB Power Meter

**Function Usage:**

Set offset to On / Off.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

bVal

Variable type: ViBoolean

0 means on, 1 means off.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryOffsetOn**

**ViStatus \_VI\_FUNC av4957x\_QueryOffsetOn (ViSession handle, ViBoolean bVal[])**

**Applicable Mode:** Power Meter, USB Power Meter

**Function Usage:**

Query offset On / Off status.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

bVal

Variable type: ViBoolean[]

0 means on, 1 means off.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**CHPWSetSwitch**

**ViStatus \_VI\_FUNC av4957x\_CHPWSetSwitch (ViSession handle, ViBoolean bVal)**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Set channel power to On / Off.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

bVal

Variable type: ViBoolean

0 means on, 1 means off.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

### CHPWQuerySwitch

**ViStatus \_VI\_FUNC av4957x\_CHPWQuerySwitch (ViSession handle, ViBoolean bVal[])**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Query channel power On / Off status.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

bVal

Variable type: ViBoolean[]

0 means on, 1 means off.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

### CHPWSetChBw

**ViStatus \_VI\_FUNC av4957x\_CHPWSetChBw (ViSession handle, ViReal64 fVal)**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Set channel power BW value of the channel power function measurement in the Spectrum Analyzer mode.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64

Frequency value (Hz), which is the range for the span of the current model

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

### CHPWQueryChBw

**ViStatus \_VI\_FUNC av4957x\_CHPWQueryChBw (ViSession handle, ViReal64 fVal[])**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Query channel power BW value of the channel power function measurement in the Spectrum Analyzer mode.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64[]

Frequency value (Hz).

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**CHPWQueryChPower**

**ViStatus \_VI\_FUNC av4957x\_CHPWQueryChPower (ViSession handle, ViReal64 fVal[])**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Query the channel power value of the channel power function measurement in the Spectrum Analyzer mode (**which is valid when channel power is on and after a valid sweep**).

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64[]

Power value (dBm).

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**CHPWQueryPowerDensity**

**ViStatus \_VI\_FUNC av4957x\_CHPWQueryPowerDensity (ViSession handle, ViReal64 fVal[])**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Query the channel power density value of the channel power function measurement in the Spectrum Analyzer mode (**which is valid when channel power is on and after a valid sweep**).

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64[]

Power value (dBm/Hz).

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetDetectorType**

**ViStatus \_VI\_FUNC av4957x\_SetDetectorType (ViSession handle, ViUInt32 nVal)**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Set detection type.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nVal

Variable type: ViUInt32

nVal	Detection type
0	Positive peak detection
1	Negative peak detection
2	Sample detection
3	Normal (Rosenfeld) detection
4	Average detection
5	RMS detection

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryDetectorType**

**ViStatus \_VI\_FUNC av4957x\_QueryDetectorType (ViSession handle, ViUInt32 nVal[])**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Query detection type.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nVal

Variable type: ViUInt32[]

nVal	Detection type
0	Positive peak detection
1	Negative peak detection
2	Sample detection
3	Normal (Rosenfeld) detection
4	Average detection
5	RMS detection

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetDetectorAuto**

**ViStatus \_VI\_FUNC av4957x\_SetDetectorAuto (ViSession handle, ViBoolean bVal)**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Set detection to auto On / Off. In auto detection mode, the instrument will automatically select detection type according to different measurements.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

bVal

Variable type: ViBoolean

0 means on, 1 means off.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryDetectorAuto**

**ViStatus \_VI\_FUNC av4957x\_QueryDetectorAuto (ViSession handle, ViBoolean bVal[])**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Query detection auto On / Off status.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

bVal

Variable type: ViBoolean[]

0 means on, 1 means off.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetFreq**

**ViStatus \_VI\_FUNC av4957x\_SetFreq (ViSession handle, ViReal64 fVal)**

**Applicable Mode:** USB Power Meter, VVM, or Signal Source

**Function Usage:**

Set frequency value in the current mode.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64

Frequency value.



**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryFreq****ViStatus \_VI\_FUNC av4957x\_QueryFreq (ViSession handle, ViReal64 fVal[])****Applicable Mode:** USB Power Meter, VVM, or Signal Source**Function Usage:**

Query frequency value in the current mode.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64[]

Frequency value (double) (Hz), which is the frequency range of the power probe in the USB Power Meter mode. In the VVM mode and the Signal Source mode, it is the frequency value of the current model.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetCenterFreq****ViStatus \_VI\_FUNC av4957x\_SetCenterFreq (ViSession handle, ViReal64 fVal)****Applicable Mode:** Spectrum Analyzer, Network Analyzer, CAT, Power Meter, or Signal Source**Function Usage:**

Set center frequency value in the current mode.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64

Frequency value (Hz), value range: 0 ~ max sweep frequency (different for different models).

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryCenterFreq****ViStatus \_VI\_FUNC av4957x\_QueryCenterFreq (ViSession handle, ViReal64 fVal[])****Applicable Mode:** Spectrum Analyzer, Network Analyzer, CAT, Power Meter, or Signal Source**Function Usage:**

Query center frequency value in the current mode.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64[]

Frequency value (Hz).

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetSpan**

**ViStatus \_VI\_FUNC av4957x\_SetSpan (ViSession handle, ViReal64 fVal)**

**Applicable Mode:** Spectrum Analyzer, Network Analyzer, CAT, or Power Meter

**Function Usage:**

Set span in the current mode.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64

Frequency value (Hz), value range: 0 ~ Max Span (different Span Max values for different models).

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QuerySpan**

**ViStatus \_VI\_FUNC av4957x\_QuerySpan (ViSession handle, ViReal64 fVal[])**

**Applicable Mode:** Spectrum Analyzer, Network Analyzer, CAT, or Power Meter

**Function Usage:**

Query span in the current mode.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64[]

Frequency value (Hz).

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetFullSpan**

**ViStatus \_VI\_FUNC av4957x\_SetFullSpan (ViSession handle)**

**Applicable Mode:** Spectrum Analyzer or Power Meter

**Function Usage:**

Set to full span.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetZeroSpan****ViStatus \_VI\_FUNC av4957x\_SetZeroSpan (ViSession handle)**

**Applicable Mode:** Spectrum Analyzer or Power Meter

**Function Usage:**

Set to zero Span.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetLastSpan****ViStatus \_VI\_FUNC av4957x\_SetLastSpan (ViSession handle)**

**Applicable Mode:** Spectrum Analyzer or Power Meter

**Function Usage:**

Set to previous span.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetStartFreq****ViStatus \_VI\_FUNC av4957x\_SetStartFreq (ViSession handle, ViReal64 fVal)**

**Applicable Mode:** Spectrum Analyzer, Network Analyzer, CAT, Power Meter, or Signal Source

**Function Usage:**

Set start frequency value in the current mode.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64

Frequency value (Hz), value range: Min sweep frequency ~ max sweep frequency (different sweep ranges for different models).

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryStartFreq**

**ViStatus \_VI\_FUNC av4957x\_QueryStartFreq (ViSession handle, ViReal64 fVal[])**

**Applicable Mode:** Spectrum Analyzer, Network Analyzer, CAT, Power Meter, or Signal Source

**Function Usage:**

Query start frequency value in the current mode.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64[]

Frequency value (Hz).

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetStopFreq**

**ViStatus \_VI\_FUNC av4957x\_SetStopFreq (ViSession handle, ViReal64 fVal)**

**Applicable Mode:** Spectrum Analyzer, Network Analyzer, CAT, Power Meter, or Signal Source

**Function Usage:**

Set stop frequency value in the current mode.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64

Frequency value (Hz), value range: Min sweep frequency ~ max sweep frequency (different sweep ranges for different models).

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryStopFreq**

**ViStatus \_VI\_FUNC av4957x\_QueryStopFreq (ViSession handle, ViReal64 fVal[])**

**Applicable Mode:** Spectrum Analyzer, Network Analyzer, CAT, Power Meter, or Signal Source

**Function Usage:**

Query stop frequency value in the current mode.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64[]

Frequency value (Hz).

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetCWFreq**

**ViStatus \_VI\_FUNC av4957x\_SetCWFreq (ViSession handle, ViReal64 fVal)**

**Applicable Mode:** Signal source

**Function Usage:**

Set CW frequency in the Signal Source mode.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64

Frequency value (Hz), value range: Min sweep frequency ~ max sweep frequency (different sweep ranges for different models).

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryCWFreq**

**ViStatus \_VI\_FUNC av4957x\_QueryCWFreq (ViSession handle, ViReal64 fVal[])**

**Applicable Mode:** Signal Source

**Function Usage:**

Query CW frequency in the Signal Source mode.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64[]

Frequency value (Hz).

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

### OBWSetSwitch

**ViStatus \_VI\_FUNC av4957x\_OBWSetSwitch (ViSession handle, ViBoolean bVal)**

**Applicable Mode:** Applicable Mode: Spectrum Analyzer

#### Function Usage:

Set OBW function measurement to On / Off.

#### Parameter list:

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

bVal

Variable type: ViBoolean

0 means on, 1 means off.

#### Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

### OBWQuerySwitch

**ViStatus \_VI\_FUNC av4957x\_OBWQuerySwitch (ViSession handle, ViBoolean bVal[])**

**Applicable Mode:** Spectrum Analyzer

#### Function Usage:

Query OBW On / Off status.

#### Parameter list:

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

bVal

Variable type: ViBoolean[]

0 means on, 1 means off.

#### Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

### OBWSetMethod

**ViStatus \_VI\_FUNC av4957x\_OBWSetMethod (ViSession handle, ViUInt32 nVal)**

**Applicable Mode:** Spectrum Analyzer

#### Function Usage:

Set OBW measurement method. The percentage measurement method is to obtain the bandwidth of x% of the total power of the whole span, and the XdB measurement method is to obtain the bandwidths smaller than xdB on both sides of the max power value.

#### Parameter list:

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nVal

Variable type: ViUInt32

nVal	Measurement method
0	Percentage measurement
1	XdB measurement

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**OBWQueryMethod**

**ViStatus \_VI\_FUNC av4957x\_OBWQueryMethod (ViSession handle, ViUInt32 nVal[])**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Query OBW measurement method.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nVal

Variable type: ViUInt32[]

nVal	Measurement method
0	Percentage measurement
1	XdB measurement

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**OBWSetPercent**

**ViStatus \_VI\_FUNC av4957x\_OBWSetPercent (ViSession handle, ViReal64 fVal)**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Set the OBW percentage, which is valid in the percentage measurement method

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64

Percentage, range: 0.1 ~ 0.9999.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

### OBWQueryPercent

**ViStatus \_VI\_FUNC av4957x\_OBWQueryPercent (ViSession handle, ViReal64 fVal[])**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Query OBW percentage value.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64[]

Percentage value.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

### OBWSetXdBValue

**ViStatus \_VI\_FUNC av4957x\_OBWSetXdBValue (ViSession handle, ViReal64 fVal)**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Set the XdB value of OBW, which is valid in the XdB measurement method.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64

XdB value (dB), range: -100 ~ -0.1.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

### OBWQueryXdBValue

**ViStatus \_VI\_FUNC av4957x\_OBWQueryXdBValue (ViSession handle, ViReal64 fVal[])**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Query XdB value of OBW.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64[]



XdB value (dB).

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**OBWQueryOBWValue**

**ViStatus \_VI\_FUNC av4957x\_OBWQueryOBWValue (ViSession handle, ViReal64 fVal[])**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Query OBW value.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64[]

OBW value (Hz).

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetAttenuator**

**ViStatus \_VI\_FUNC av4957x\_SetAttenuator (ViSession handle, ViReal64 fVal)**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Set attenuation value.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64

Attenuation value (double) (dB), value range: 0 ~ 31.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryAttenuator**

**ViStatus \_VI\_FUNC av4957x\_QueryAttenuator (ViSession handle, ViReal64 fVal[])**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Query attenuation value.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64[]

Attenuation value.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetAttenuatorAuto**

**ViStatus \_VI\_FUNC av4957x\_SetAttenuatorAuto (ViSession handle, ViBoolean bVal)**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Set attenuation to auto On / Off. When attenuation Auto is turned on, the instrument will set corresponding attenuation according to the reference value automatically.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

bVal

Variable type: ViBoolean

0 means on, 1 means off.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryAttenuatorAuto**

**ViStatus \_VI\_FUNC av4957x\_QueryAttenuatorAuto (ViSession handle, ViBoolean bVal[])**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Query attenuation auto On / Off status.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

bVal

Variable type: ViBoolean[]

0 means on, 1 means off.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetPreAmpSwitch**

**ViStatus \_VI\_FUNC av4957x\_SetPreAmpSwitch (ViSession handle, ViBoolean bVal)**

**Applicable Mode:** Spectrum Analyzer or Power Meter

**Function Usage:**

Set pre-amplifier to On / Off. When set to On, it can improve the measurement accuracy of low-power signals, but it is better to turn it off when measuring high-power signals, otherwise it may lead to AD overloading of measurement.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

bVal

Variable type: ViBoolean

0 means on, 1 means off.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryPreAmpSwitch**

**ViStatus \_VI\_FUNC av4957x\_QueryPreAmpSwitch (ViSession handle, ViBoolean bVal[])**

**Applicable Mode:** Spectrum Analyzer or Power Meter

**Function Usage:**

Query pre-amplifier On / Off status.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

bVal

Variable type: ViBoolean[]

0 means on, 1 means off.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetSwpPoints**

**ViStatus \_VI\_FUNC av4957x\_SetSwpPoints (ViSession handle, ViUInt32 nVal)**

**Applicable Mode:** Network Analyzer, CAT, or Signal Source

**Function Usage:**

Set sweep points in the Linear Sweep mode.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nVal

Variable type: ViUInt32

Sweep points, range 11 ~ 10001.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QuerySwpPoints**

**ViStatus \_VI\_FUNC av4957x\_QuerySwpPoints (ViSession handle, ViUInt32 nVal[])**

**Applicable Mode:** Network Analyzer, CAT, or Signal Source

**Function Usage:**

Query sweep points in the Linear Sweep mode.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nVal

Variable type: ViUInt32[]

Sweep points.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetSwpTime**

**ViStatus \_VI\_FUNC av4957x\_SetSwpTime (ViSession handle, ViReal64 fVal)**

**Applicable Mode:** Spectrum Analyzer, Network Analyzer, or CAT

**Function Usage:**

Set sweep time in the current mode. Sweep time is the time required for the local oscillator tuning through the selected frequency interval. Sweep time directly affects the time required to complete a test, which does not include the dead time between the completion of a sweep and the start of the next sweep. Sweep time usually varies with Span, RBW and VBW. Sweep time cannot be set when RBW<=1kHz in the Spectrum Analyzer mode.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64

Time (ms).

In the Spectrum Analyzer mode, at zero Span, the value can be set to: 10 us ~ 600 s, and set as follows when at a non-zero Span: 1 ms ~ 200 s.

In the Network Analyzer and CAT modes, the Min sweep is determined by the settings of sweep points and IF BW, and the max sweep is: Points\* 100 (s).

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QuerySwpTime**

**ViStatus \_VI\_FUNC av4957x\_QuerySwpTime (ViSession handle, ViReal64 fVal[])****Applicable Mode:** Spectrum Analyzer, Network Analyzer, or CAT**Function Usage:**

Query the sweep time in the current mode.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64[]

Time (ms).

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetAutoSwpTimeOn****ViStatus \_VI\_FUNC av4957x\_SetAutoSwpTimeOn (ViSession handle, ViBoolean bVal)****Applicable Mode:** Spectrum Analyzer, Network Analyzer, or CAT**Function Usage:**

Set sweep time to auto On / Off. When set to On, the instrument will adopt the sweep speed as high as possible; or you can manually increase the sweep time to meet some specific measurement requirements. Sweep time set manually must be &gt; = automatic sweep time.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

bVal

Variable type: ViBoolean

0 means on, 1 means off.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryAutoSwpTimeOn****ViStatus \_VI\_FUNC av4957x\_QueryAutoSwpTimeOn (ViSession handle, ViBoolean bVal[])****Applicable Mode:** Spectrum Analyzer, Network Analyzer, or CAT**Function Usage:**

Query sweep time auto On / Off status.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

bVal

Variable type: ViBoolean[]

0 means on, 1 means off.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**IASetSwitch**

**ViStatus \_VI\_FUNC av4957x\_IASetSwitch (ViSession handle, ViBoolean bVal)**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Set IA mode to On / Off.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

bVal

Variable type: ViBoolean

0 means on, 1 means off.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**IAQuerySwitch**

**ViStatus \_VI\_FUNC av4957x\_IAQuerySwitch (ViSession handle, ViBoolean bVal[])**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Query IA mode On / Off status.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

bVal

Variable type: ViBoolean[]

0 means on, 1 means off.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**IASetRedLimit**

**ViStatus \_VI\_FUNC av4957x\_IASetRedLimit (ViSession handle, ViReal64 fVal)**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Set spectrum mode, with IA red limit.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64

Red limit value (dBm), value range: -500 ~ 500.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**IAQueryRedLimit**

**ViStatus \_VI\_FUNC av4957x\_IAQueryRedLimit (ViSession handle, ViReal64 fVal[])**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Query spectrum mode, with IA red limit.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64[]

Red limit value (dBm)

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**IASetBlueLimit**

**ViStatus \_VI\_FUNC av4957x\_IASetBlueLimit (ViSession handle, ViReal64 fVal)**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Set spectrum mode, with IA blue limit.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64

Blue limit value (dBm), value range: -500 ~ 500.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**IAQueryBlueLimit**

**ViStatus \_VI\_FUNC av4957x\_IAQueryBlueLimit (ViSession handle, ViReal64 fVal[])**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Query spectrum mode, with IA blue limit.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64[]

Blue limit value (dBm)

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**IASetMode**

**ViStatus \_VI\_FUNC av4957x\_IASetMode (ViSession handle, ViUInt32 nVal)**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Set IA mode, with two modes available, namely, Spec and Wtf.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nVal

Variable type: ViUInt32

nVal	IA mode
0	Spec
1	Wtf

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**IAQueryMode**

**ViStatus \_VI\_FUNC av4957x\_IAQueryMode (ViSession handle, ViUInt32 nVal[])**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Query IA mode, with two modes available, namely, Spec and Wtf.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nVal

Variable type: ViUInt32[]



IA mode:

0 means Spec mode

1 means Wtf mode

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**IAClear**

**ViStatus \_VI\_FUNC av4957x\_IAClear (ViSession handle)**

**Applicable Mode:** Spectrum Analyzer

**Function Usage:**

Clear diagrams in the IA mode.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryMeasData**

**ViStatus \_VI\_FUNC av4957x\_QueryMeasData (ViSession handle, ViReal64 fVal[], ViInt32 nSize[]);**

**Applicable Mode:** USB Power Meter, Power Meter

**Function Usage:**

Query power value in USB Power Meter and Power Monitor modes.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64[]

Power value (dBm).

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**X Commands of the Source Sub-system**

**SetPortOutputSweepType**

**ViStatus \_VI\_FUNC av4957x\_SetPortOutputSweepType (ViSession handle, ViUInt32 nVal)**

**Applicable Mode:** Signal Source

**Function Usage:**

Set sweep type.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nVal

Variable type: ViUInt32

Sweep type:

0 means CW freq

1 means Swp freq

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryPortOutputSweepType**

**ViStatus \_VI\_FUNC av4957x\_QueryPortOutputSweepType (ViSession handle, ViUInt32 nVal[])**

**Applicable Mode:** Signal Source

**Function Usage:**

Query sweep type.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nVal

Variable type: ViUInt32[]

nVal	Sweep type
0	CW Ferq
1	Swp Freq

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetPortOutputMode**

**ViStatus \_VI\_FUNC av4957x\_SetPortOutputMode (ViSession handle, ViUInt32 nVal)**

**Applicable Mode:** Network Analyzer, CAT, or Signal Source

**Function Usage:**

Set output power type.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nVal

Variable type: ViUInt32

nVal	Output power type
0	High power

1	Low power
2	Manual power

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryPortOutputMode**

**ViStatus \_VI\_FUNC av4957x\_QueryPortOutputMode (ViSession handle, ViUInt32 nVal[])**

**Applicable Mode:** Network Analyzer, CAT, or Signal Source

**Function Usage:**

Query output power type.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nVal

Variable type: ViUInt32[]

nVal	Output power type
0	High power
1	Low power
2	Manual power

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetPortOutputManualPwr**

**ViStatus \_VI\_FUNC av4957x\_SetPortOutputManualPwr (ViSession handle, ViReal64 fVal)**

**Applicable Mode:** Network Analyzer, CAT, or Signal Source

**Function Usage:**

Set manual output power value.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64

Manual output power value (dBm), value range: -50 ~ 10.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryPortOutputManualPwr**

**ViStatus \_VI\_FUNC av4957x\_QueryPortOutputManualPwr (ViSession handle, ViReal64 fVal[])**

**Applicable Mode:** Network Analyzer, CAT, or Signal Source

**Function Usage:**

Query manual output power value.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

fVal

Variable type: ViReal64

Manual output power value (dBm).

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**XI Commands of the Trace Sub-system****ReadSATrace**

**ViStatus \_VI\_FUNC av4957x\_ReadSATrace (ViSession handle, ViUInt32 nIdx, float pData[], ViInt32 nSize[])**

**Applicable Mode:** Spectrum mode

**Function Usage:**

Set sweep type.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

pData

Variable type: float[]

Trace data

nSize

Variable type: ViInt32[]

Trace data count

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**SetSATraceStatus**

**ViStatus \_VI\_FUNC av4957x\_SetSATraceStatus (ViSession handle, ViUInt32 nIdx, ViUInt32 nVal)**

**Applicable Mode:** Spectrum mode

**Function Usage:**

Set trace state in the Spectrum mode.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nIdx

Variable type: ViUInt32

Trace subscribe

nVal

Variable type: ViUInt32

nVal	Trace state type
0	Max Hold
1	Min Hold
2	Refresh
3	Current trace
4	Hide Trace

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QuerySATraceStatus**

**ViStatus \_VI\_FUNC av4957x\_QuerySATraceStatus (ViSession handle, ViUInt32 nIdx, ViUInt32 nVal[])**

**Applicable Mode:** Spectrum mode

**Function Usage:**

Set trace state in the Spectrum mode.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nIdx

Variable type: ViUInt32

Trace subscribe

nVal

Variable type: ViUInt32[]

nVal	Trace state type
0	Max Hold
1	Min Hold
2	Refresh
3	Current trace
4	Hide Trace

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**XII Sub-system Commands of the System**

**SetGPSOn**

**ViStatus \_VI\_FUNC av4957x\_SetGPSOn (ViSession handle, ViBoolean bVal)**

**Applicable Mode:** All modes

**Function Usage:**

Set GPS to On / Off. When set to on, the data collected by the GPS chip, such as longitude, latitude, and altitude, will be displayed on the screen.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

bVal

Variable type: ViBoolean

0 means on, 1 means off.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryGPSOn**

**ViStatus \_VI\_FUNC av4957x\_QueryGPSOn (ViSession handle, ViBoolean bVal[])**

**Applicable Mode:** All modes

**Function Usage:**

Query GPS On / Off status.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

bVal

Variable type: ViBoolean[]

0 means on, 1 means off.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryGPSState**

**ViStatus \_VI\_FUNC av4957x\_QueryGPSState (ViSession handle, ViUInt32 nVal[])**

**Applicable Mode:** All modes

**Function Usage:**

Query GPS state.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

nVal

Variable type: ViUInt32[]

nVal	GPS state
------	-----------

0	Positioning not performed
1	Differential positioning not performed
2	Differential positioning
3	Invalid PPS
4	Estimating

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryGPSReceiveState**

**ViStatus\_VI\_FUNC av4957x\_QueryGPSReceiveState (ViSession handle, ViBoolean bVal[])**

**Applicable Mode:** All modes

**Function Usage:**

Query GPS receiver state.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

bVal

Variable type: ViBoolean[]

GPS receiver state. 0 means that there're no data in the receive, and 1 means that there're data in the receiver.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**GPS Cold Start**

**ViStatus\_VI\_FUNC av4957x\_GPSReset (ViSession handle)**

**Applicable Mode:** All modes

**Function Usage:**

It is for GPS cold start. Sometimes, for example, when in places where the signal is extremely poor, there is no way to receive the GPS signal for a long time, and sometimes there is no way to change the location. At this time, choose cold start to run the new star search positioning will make the starting speed faster. In this case, you can choose cold start to let the module search for galaxy positioning again.

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

**QueryGPSData**

**ViStatus\_VI\_FUNC av4957x\_QueryGPSData (ViSession handle, ViChar strVal[])**

**Applicable Mode:** All modes

**Function Usage:**

Query data collected by the GPS, and return current GPS data in the following format: "<Longitude>, <Latitude>, <Altitude>, <Time UTC>".

**Parameter list:**

handle

Variable type: ViSession

Instrument handle returned by the function, communicating with the instrument.

strVal

Variable type: ViChar[]

Returned GPS data.

Example: Return "38 28'11.22" N,122 42'13.23" W,152,06/28/2010 23:35:38\n" where there are data

Return "--,--,--,--\n" when there are no data.

**Returned Value:**

Returned value indicates the execution result of the function: 0 means success, and minus means failure.