



S3302 Series Handheld Spectrum Analyzer

Programming Manual



Saluki Technology Inc.

The document applies to the handheld spectrum analyzers of the following models:

- S3302SA handheld spectrum analyzer (9kHz-4GHz).
- S3302SB handheld spectrum analyzer (9kHz-6.5GHz).
- S3302SC handheld spectrum analyzer (9kHz-9GHz).
- S3302A handheld spectrum analyzer (9kHz-20GHz).
- S3302B handheld spectrum analyzer (9kHz-26.5GHz).
- S3302C handheld spectrum analyzer (9kHz-32GHz).
- S3302D handheld spectrum analyzer (9kHz-44GHz).
- S3302E handheld spectrum analyzer (9kHz-50GHz).
- S3302F handheld spectrum analyzer (9kHz-67GHz).

Standard pack and accessories:

No.	Item
1	Main Machine
2	Power cord
3	Power adapter
4	U disk (manual)
5	USB cable
6	Battery

Options of the S3302 series in addition to standard accessories:

Model No.	Description	Note
S3302-05	Programming manual	/
S3302-06	Power adapter	/
S3302-07	Rechargeable lithium-ion battery	/
S3302-08	CAT5 LAN cable	Point to point, 2m
S3302-09	Micro SD card	Capacity: 8GB
S3302-10	GPS option	GPS Exposed Antenna(BNC), Built-In GPS Module and Software
S3302-11	USB power meter option	Provide USB Power Measurement function (Option 12-15 needed)
S3302-12	S87230 USB power continuous wave power sensor (9kHz - 6GHz)	Need option 11
S3302-13	S87231 USB power continuous wave power sensor (10MHz - 18GHz)	Need option 11
S3302-14	S87232 USB power continuous wave power sensor (50MHz - 26.5GHz)	Need option 11
S3302-15	S87233 USB power continuous wave power sensor (50MHz - 40GHz)	Need option 11
S3302-16	Interference analyzer option	Waterfall, RSSI

Model No.	Description	Note
S3302-17	AM/FM/PM analyzer option	To Realize Modulation Characteristics Analysis of AM/FM/PM Signals.
S3302-18	Channel scanner option	To Realize Signal Power Measurement of Multiple Channels and Frequency.
S3302-19	List sweep option	To Realize Continuous Sweep Measurement of Various Frequency Bands .
S3302-20	Zero span IF output	Output the third(3rd.) IF(140.25MHz) or fourth(4th.) IF(31.25MHz) signal
S3302-21	ZE9080A Directional antenna	9kHz-20MHz, N (f) (need option 25)
S3302-22	ZE9080B Directional antenna	20MHz-200MHz, N (f) (need option 25)
S3302-23	ZE9080C Directional antenna	200MHz-500MHz, N (f) (need option 25)
S3302-24	ZE9080D Directional antenna	500MHz-8GHz, N (f) (need option 25)
S3302-25	ZE9080 Antenna amplifier	9kHz-8GHz, N (f), for option 21-24
S3302-26	S89901 Horn antenna	1GHz - 18GHz, N(f) (do not need amplifier)
S3302-27	S89902 Horn antenna	18GHz - 40GHz, 2.92mm(f) (do not need amplifier)
S3302-28	Functional bag	/
S3302-29	Backpack	/
S3302-30	Carrying case	For safety carrying
S3302-31	S89901 antenna handle	Need option 26
S3302-32	S89902 antenna handle	Need option 27
S3302-33	Signal analyzer	To realize the rapid analysis of interference signal, and provide the audio output and IQ Capture.
S3302-34	Field strength option	Provide Pscan, Fscan, MScan etc. Functions
S3302-35	Tracking generator (100kHz - 4GHz)	For S3302SA only
S3302-36	Tracking generator (100kHz - 6.5GHz)	For S3302SB only
S3302-37	Tracking generator (100kHz - 9GHz)	For S3302SC only
S3302-38	Orientation analysis option	Built-In software(need option10, 50 and directional Antenna)
S3302-39	Coverage map option	Built-In software(need option10)
S3302-41	Portable omnidirectional antenna	Frequency range :700MHz - 2.7GHz (for communication bands)
S3302-42	700MHz - 4GHz directional antenna	Active log-periodic antenna
S3302-43	700MHz - 6GHz directional antenna	Active log-periodic antenna
S3302-44	680MHz - 10GHz directional antenna	Active log-periodic antenna
S3302-45	680MHz - 20GHz directional antenna	Active log-periodic antenna
S3302-46	400MHz - 4GHz directional antenna	Active log-periodic antenna
S3302-47	400MHz - 6GHz directional antenna	Active log-periodic antenna

Model No.	Description	Note
S3302-48	380MHz - 10GHz directional antenna	Active log-periodic antenna
S3302-49	380MHz - 20GHz directional antenna	Active log-periodic antenna
S3302-50	External electric compass	External USB electric compass (need option38)
S3302-51	6GHz omnidirectional antenna (680MHz - 6GHz)	Portable omnidirectional antenna
S3302-52	8GHz omnidirectional antenna (300MHz - 8GHz)	Portable omnidirectional antenna
S3302-53	VHF/UHF portable antenna	Frequency range :140MHz/430MHz
S3302-54	Passive directional antenna (700MHz - 4GHz)	Passive log-periodic antenna
S3302-55	Passive directional antenna (700MHz - 6GHz)	Passive log-periodic antenna
S3302-56	Passive directional antenna (680MHz - 10GHz)	Passive log-periodic antenna
S3302-57	Passive directional antenna (680MHz - 18GHz)	Passive log-periodic antenna
S3302-58	Passive directional antenna (680MHz - 25GHz)	Passive log-periodic antenna
S3302-59	Passive directional antenna (680MHz - 35GHz)	Passive log-periodic antenna
S3302-60	N/SMA-JJ RF cable (2m)	N male to SMA male RF coaxial cable, DC-18GHz, Length 2m
S3302-61	N/SMA-JJ RF cable (1m)	N male to SMA male RF coaxial cable, DC-18GHz, Length 1m
S3302-67	ZE9080 Antenna transport box	Used for antenna and amplifier, including option 21, 22, 23, 24, 25

Preface

Thanks for choosing S3302 series spectrum analyzer produced by Saluki Technology Inc. Please read this maintenance 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.

S3302-03-02

Version

Rev03 2019.11

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 document 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: www.salukitec.com

Email: sales@salukitec.com

Address: No. 367 Fuxing N Road, Taipei 105, Taiwan (R.O.C.)

Contents

Chapter I Introduction to SCPI Commands.....	1
Section I Introduction of SCPI command operation.....	1
Section II IEEE 488.2 commands.....	2
*CLS – Clear the state.....	2
*IDN? - Identification.....	2
*OPC - Operation complete command.....	2
*OPC? - Operation complete query.....	2
*RST - Reset.....	2
*WAI - Wait.....	2
Section III Measurement commands.....	3
:CALCulate[:SElected]:LIMit:BEEP <bool>.....	3
:CALCulate[:SElected]:LIMit:LOWer:DISPlay <bool>.....	3
:CALCulate[:SElected]:LIMit:LOWer:MARGin <num>.....	3
:CALCulate[:SElected]:LIMit:LOWer:TEST <bool>.....	3
:CALCulate[:SElected]:LIMit:LOWer:EDIT:ADD.....	3
:CALCulate[:SElected]:LIMit:LOWer:EDIT:DELeTe.....	4
:CALCulate[:SElected]:LIMit:LOWer:EDIT:CLear.....	4
:CALCulate[:SElected]:LIMit:LOWer:EDIT:DATA.....	4
:CALCulate[:SElected]:LIMit:UPPer:DISPlay <bool>.....	4
:CALCulate[:SElected]:LIMit:UPPer:MARGin <num>.....	4
:CALCulate[:SElected]:LIMit:UPPer:TEST <bool>.....	5
:CALCulate[:SElected]:LIMit:UPPer:EDIT:ADD.....	5
:CALCulate[:SElected]:LIMit:UPPer:EDIT:CLear.....	5
:CALCulate[:SElected]:LIMit:UPPer:EDIT:DELeTe.....	5
:CALCulate[:SElected]:LIMit:UPPer:EDIT:DATA.....	5
:CALCulate[:SElected]:LIST:EDIT:ADD(Options).....	6
:CALCulate[:SElected]:LIST:EDIT:ADD:SEGment.....	6
:CALCulate[:SElected]:LIST:EDIT:CLear.....	6
:CALCulate[:SElected]:LIST:EDIT:DELeTe.....	7
:CALCulate[:SElected]:LIST:EDIT:SEGment.....	7
:CALCulate[:SElected]:MARKer<n>[:STATe] <string>.....	7
:CALCulate[:SElected]:MARKer<n>:ACTivate.....	7
:CALCulate[:SElected]:MARKer<n>:SET <string>.....	8
:CALCulate[:SElected]:MARKer:AOFF.....	8
:CALCulate[:SElected]:MARKer<n>:X <num>.....	8
:CALCulate[:SElected]:MARKer<n>:Y?.....	9
:CALCulate[:SElected]:MARKer<n>:FCOunt[:STATe] <bool>.....	10
:CALCulate[:SElected]:MARKer<n>:FCOunt:X?.....	10
:CALCulate[:SElected]:MARKer<n>:FUNCTion:MAXimum.....	10
:CALCulate[:SElected]:MARKer<n>:FUNCTion:MINimum.....	10
:CALCulate[:SElected]:MARKer<n>:FUNCTion:PEAK.....	11
:CALCulate[:SElected]:MARKer<n>:FUNCTion:PLEFt.....	11
:CALCulate[:SElected]:MARKer<n>:FUNCTion:PNEXt.....	11
:CALCulate[:SElected]:MARKer<n>:FUNCTion:PRIGHt.....	11
:CALCulate[:SElected]:MARKer<n>:NOISe[:STATe] <bool>.....	12
:CALCulate[:SElected]:RELative[:MAGNitude]?.....	12
:CALCulate[:SElected]:RELative[:MAGNitude]:AUTO <bool>.....	12
:CALCulate[:SElected]:PEAK:TRAC <bool>.....	12
:CALibration:ZERO.....	12
:CALibration:ZERO:STATe?.....	13
:DISPlay:WINDow:ANALog:LOWer <num>.....	13

:DISPlay:WINDow:ANALog:UPPer <num>.....	13
:DISPlay:WINDow:TRACe:Y[:SCALe]:AUTO.....	13
:DISPlay:WINDow:TRACe:Y[:SCALe]:PDIVision <num>.....	13
:DISPlay:WINDow:TRACe:Y[:SCALe]:RLEVel <num>.....	14
:DISPlay:WINDow:TRACe:Y[:SCALe]:RPOSition <num>.....	14
:DISPlay:TITLe <string>.....	14
:DISPlay:TITLe:STATe <bool>.....	14
:DISPlay:MODE <string>.....	14
:DISPlay:BRIG <int>.....	15
:DISPlay:BRIG:AUTO <bool>.....	15
:DISPlay:TIME:FMT <string>.....	15
:FORM[:DATA] <string>.....	15
:INITiate:CONTInuous <bool>.....	16
:INITiate.....	16
:INSTrument:CATalog?.....	16
:INSTrument[:SElect] <string>.....	16
:MMEMory:DELeTe:ANTenna <string>.....	17
:MMEMory:DELeTe:ANTenna:ALL.....	17
:MMEMory:DELeTe:LIMit <string>.....	17
:MMEMory:DELeTe:LIMit:ALL.....	17
:MMEMory:DELeTe:LIST <string>.....	18
:MMEMory:DELeTe:LIST:ALL.....	18
:MMEMory:DELeTe:STATe <string>.....	18
:MMEMory:DELeTe:STATe:ALL.....	18
:MMEMory:DELeTe:DATA <string>.....	18
:MMEMory:DELeTe:DATA:ALL.....	18
:MMEMory:LOAD:ANTenna <string>.....	19
:MMEMory:LOAD:LIMit <string>.....	19
:MMEMory:LOAD:SEM <string>.....	19
:MMEMory:LOAD:LIST <string>.....	19
:MMEMory:LOAD:STATe <string>.....	20
:MMEMory:LOAD:DATA <string>.....	20
:MMEMory:LOCation <string>.....	20
:MMEMory:STORe:ANTenna <string>.....	20
:MMEMory:STORe:LIMit <string>.....	20
:MMEMory:STORe:LIST <string>.....	21
:MMEMory:STORe:SCReen <string>.....	21
:MMEMory:STORe:STATe <string>.....	21
:MMEMory:STORe:DATA <string>.....	21
[:SENSe]:ACPower:ADJChbw <num>.....	21
[:SENSe]:ACPower:LIMit[:STATe] <bool>.....	22
[:SENSe]:ACPower:MAINChbw <num>.....	22
[:SENSe]:ACPower:OFFSet:LLIMit <num>.....	22
[:SENSe]:ACPower:OFFSet:ULIMit <num>.....	22
[:SENSe]:ACPower:SPACe <num>.....	22
[:SENSe]:ACPower[:STATe] <bool>.....	23
[:SENSe]:ACPower:UPPer?.....	23
[:SENSe]:ACPower:LOWer?.....	23
[:SENSe]:AFPanalyzer:DEMod:TYPE <string>.....	23
[:SENSe]:AFPanalyzer:DEMod:MODE <string>.....	23
[:SENSe]:AFPanalyzer:SPAN <num>.....	24
[:SENSe]:AFPanalyzer:SCALe <num>.....	24
[:SENSe]:AFPanalyzer:SWEep:TIME <num>.....	24
[:SENSe]:AFPanalyzer:IFBW <num>.....	24

[.SENSe]:AFPanalyzer:TRACe <string>.....	24
[.SENSe]:AMPLitude:ALIGNment:NOW.....	25
[.SENSe]:AMPLitude:CORRections:ANTenna:OFF.....	25
[.SENSe]:AMPLitude:CORRections[:STATe] <bool>.....	25
[.SENSe]:AMPLitude:SCALe<string>.....	25
[.SENSe]:AMPLitude:UNIT <string>.....	25
[.SENSe]:AMPLitude:CORRections:ANTenna:EDIT:ADD.....	26
[.SENSe]:AMPLitude:CORRections:ANTenna:EDIT:ADD:DATA.....	26
[.SENSe]:AMPLitude:CORRections:ANTenna:EDIT:DEL <int>.....	26
[.SENSe]:AMPLitude:CORRections:ANTenna:EDIT:DATA.....	26
[.SENSe]:AVERage:COUNt <num>.....	26
[.SENSe]:AVERage:CLear.....	27
[.SENSe]:AVERage:STATe <bool>.....	27
[.SENSe]:AVERage:CURC?.....	27
[.SENSe]:BANDwidth[:RESolution] <num>.....	27
[.SENSe]:BANDwidth[:RESolution]:AUTO <bool>.....	27
[.SENSe]:BANDwidth[:RESolution]:RATio <num>.....	28
[.SENSe]:BANDwidth:VIDeo <num>.....	28
[.SENSe]:BANDwidth:VIDeo:AUTO <bool>.....	28
[.SENSe]:BANDwidth:VIDeo:RATio <num>.....	28
[.SENSe]:BANDwidth:VIDeo:TYPE <BOOL>.....	29
[.SENSe]:CMEasurement:IBW <num>.....	29
[.SENSe]:CMEasurement:PSDR?.....	29
[.SENSe]:CMEasurement[:STATe] <bool>.....	29
[.SENSe]:CMEasurement:TPWR?.....	29
[.SENSe]:CORRection:GAIN <num>.....	30
[.SENSe]:CORRection:GAIN:STATe <bool>.....	30
[.SENSe]:CNRatio[:STATe] <bool>.....	30
[.SENSe]:CNRatio:CBW <num>.....	30
[.SENSe]:CNRatio:NBW <num>.....	30
[.SENSe]:CNRatio:CNSpace <num>.....	31
[.SENSe]:CNRatio:CNRatio?.....	31
[.SENSe]:CS:DISPlay <string>.....	31
[.SENSe]:CS:MAXHold <bool>.....	31
[.SENSe]:CS:UNIT <string>.....	31
[.SENSe]:CS:PWR <string>.....	32
[.SENSe]:CS:COLOur <string>.....	32
[.SENSe]:CS:ORIEntal <string>.....	32
[.SENSe]:CS:MODE <string>.....	32
[.SENSe]:CS:CHANnel:NUMber <num>.....	32
[.SENSe]:CS:CHANnel:STEP <num>.....	33
[.SENSe]:CS:CHANnel:BANDwidth <num>.....	33
[.SENSe]:CS:FREQuency:STARt <num>.....	33
[.SENSe]:CS:FREQuency:STEP <num>.....	33
[.SENSe]:CS:FREQuency:BANDwidth <num>.....	33
[.SENSe]:CS:FREQuency:NUMber <num>.....	34
[.SENSe]:CS:LIST:NUMber <num>.....	34
[.SENSe]:DETEctor:FUNCTion <string>.....	34
[.SENSe]:DETEctor:FUNCTion:AUTO <bool>.....	34
[.SENSe]:EMISSion[:STATe] <bool>.....	35
[.SENSe]:EMISSion:CBW <num>.....	35
[.SENSe]:EMISSion:RTYPE <string>.....	35
[.SENSe]:EMISSion:MARKer <bool>.....	35
[.SENSe]:EMISSion:Fail?.....	35

[SENSe]:FREQuency:CENTer <num>.....	36
[SENSe]:FREQuency:CENTer:STEP <num>.....	36
[SENSe]:FREQuency:CENTer:STEP:AUTO <bool>.....	36
[SENSe]:FREQuency:SPAN <num>.....	36
[SENSe]:FREQuency:SPAN:FULL.....	37
[SENSe]:FREQuency:SPAN:PREVious.....	37
[SENSe]:FREQuency:SPAN:ZERO.....	37
[SENSe]:FREQuency:STARt <num>.....	37
[SENSe]:FREQuency:STOP <num>.....	37
[SENSe]:FREQuency:SIGStandard:NAME <string>.....	37
[SENSe]:FREQuency:SIGStandard:CHANnel <num>.....	38
[SENSe]:FREQuency:SIGnal:SEARch.....	38
[SENSe]:FREQuency:SIGnal:TRAC <BOOL>.....	38
[SENSe]:FREQuency:RESolution <num>.....	38
[SENSe]:IAMeasure:MODE <string>.....	38
[SENSe]:IAMeasure:TRACe:SPAN <num>.....	39
[SENSe]:IAMeasure:TRACe:SAVE <bool>.....	39
[SENSe]:IAMeasure:TRACe:CURS <num>.....	39
[SENSe]:IAMeasure:TRACe:REStart.....	39
[SENSe]:IAMeasure:TRACe:INTERval <num>.....	39
[SENSe]:IF:OUT <bool>.....	40
[SENSe]:IF:SElect <string>.....	40
[SENSe]:IQ:CAPture[:STATe] <bool>.....	40
[SENSe]:IQ:CAPture:STARt.....	40
[SENSe]:IQ:CAPture:STOP.....	40
[SENSe]:IQ:CAPture:TIME <num>.....	41
[SENSe]:IQ:CAPture:MODE <num>.....	41
[SENSe]:IQ:CAPture:SAMPle <num>.....	41
[SENSe]:IQ:CAPture:NAME <string>.....	41
[SENSe]:IQ:CAPture:TRIG <string>.....	41
[SENSe]:IQ:CAPture:TRIG:SLOPe <string>.....	42
[SENSe]:IQ:CAPture:TRIG:DELAy <num>.....	42
[SENSe]:IQ:CAPture:TRIG:AMPlitude <num>.....	42
[SENSe]:MEASurement <string>.....	42
[SENSe]:MEASurement:AOff.....	43
[SENSe]:OBW:METHod <string>.....	43
[SENSe]:OBW:OBW?.....	43
[SENSe]:OBW:PPOW <num>.....	43
[SENSe]:OBW[:STATe] <bool>.....	43
[SENSe]:OBW:XDB <num>.....	44
[SENSe]:PMManager:LIMit:STATe <bool>.....	44
[SENSe]:PMManager:LIMit:UPPer <num>.....	44
[SENSe]:PMManager:LIMit:LOWer <num>.....	44
[SENSe]:PMManager:MAXHold <bool>.....	45
[SENSe]:ROSC:SOUR <string>.....	45
[SENSe]:POWer[:RF]:ATTenuation <num>.....	45
[SENSe]:POWer[:RF]:ATTenuation:AUTO <bool>.....	45
[SENSe]:POWer[:RF]:GAIN[:STATe] <bool>.....	46
[SENSe]:SWEep:MODE <num>.....	46
[SENSe]:SWEep:TIME <num>.....	46
[SENSe]:SWEep:TIME:AUTO <bool>.....	46
[SENSe]:SWEep:TRIG <string>.....	47
[SENSe]:SWEep:TRIG:EXTRa:AMPlitude <num>.....	47
[SENSe]:SWEep:TRIG:EXTRa:SLOP <string>.....	47

[.SENSe]:SWEEp:TRIG:EXTRa:DELAy <num>.....	47
[.SENSe]:SWEEp:TRIG:VIDEo:AMPLitude <num>.....	47
[.SENSe]:SWEEp:POINts <num>.....	48
[.SENSe]:TAListen:AVOLume <num>.....	48
[.SENSe]:TAListen:DMODE <string>.....	48
[.SENSe]:TAListen:DState <bool>.....	48
[.SENSe]:TAListen:DTYPe <string>.....	49
[.SENSe]:TAListen:LTIMe <num>.....	49
[.SENSe]:FREQ:POIN <num>.....	49
[.SENSe]:FREQ:TRAC <BOOL>.....	49
[.SENSe]:FREQ:FSCan:STARt <num>.....	49
[.SENSe]:FREQ:FSCan:STEP <num>.....	50
[.SENSe]:FREQ:FSCan:POINts <num>.....	50
[.SENSe]:SWEEp:DWELl <num>.....	50
[.SENSe]:SWEEp:DWELl:INFinite <bool>.....	50
[.SENSe]:SWEEp:STAYtime<num>.....	50
[.SENSe]:SWEEp:STAYtime:STATe <bool>.....	51
[.SENSe]:POWEr:LIMit:STATe <bool>.....	51
[.SENSe]:POWEr:LIMit <num>.....	51
[.SENSe]:DMODE <string>.....	51
[.SENSe]:DMODE:VOLume <num>.....	51
[.SENSe]:IFBWidth <num>.....	52
[.SENSe]:DATA:POTF:AMPL?.....	52
[.SENSe]:DATA:POTF:FIELD?.....	52
[.SENSe]:DATA:FSCan:AMPL?.....	52
[.SENSe]:DATA:FSCan:FIELD?.....	52
[.SENSe]:DATA:LSCan:AMPL?.....	52
[.SENSe]:DATA:LSCan:FIELD?.....	53
[.SENSe]:FREQuency:FCOunt?.....	53
[.SENSe]:FST:MEASurement <string>.....	53
[.SENSe]:FST:PEAK.....	53
[.SENSe]:FST:MARKer <bool>.....	53
[.SENSe]:FST:INDEx <int>.....	54
:SYSTem:BATTery:STAT?.....	54
:SYSTem:BATTery:VOLume?.....	54
:SYSTem:GPS <bool>(Options).....	54
:SYSTem:GPS:DATA?.....	54
:SYSTem:GPS:RECeive[.STATe]?.....	55
:SYSTem:GPS:RST.....	55
:SYSTem:GPS:STATe?.....	55
:SYSTem:INFO?.....	55
:SYSTem:PWR:SHUTdown <num>.....	56
:SYSTem:PWR:SHUTdown:STATe <bool>.....	56
:SYSTem:PWR:SLEEp <num>.....	56
:SYSTem:PWR:SLEEp:STATe <bool>.....	56
:SYSTem:TIME <num>,<num>,<num>,<num>,<num>.....	56
:TRACe<n>:DATA?.....	57
:TRACe<n>:TYPE <string>.....	57
[.SENSe]:SIG:MARG:STAT <bool>.....	57
[.SENSe]:SIG:MARG <num>.....	58
[.SENSe]:SIG:LIST <string>.....	58
[.SENSe]:GEN:MODE < string >.....	58
[.SENSe]:GEN:FREQ:OFFSet <num>.....	58
[.SENSe]:GEN:FREQ:POTF <num>.....	58

[:SENSe]:GEN:AMPL:OUT <num>.....	59
[:SENSe]:GEN:AMPL:OFFS <num>.....	59
[:SENSe]:GEN:NORMZ:STAT <bool>.....	59
[:SENSe]:GEN:NORMZ:RLEV <num>.....	59
[:SENSe]:GEN:NORM:RPOS <NUM>.....	59
[:SENSe]:GEN:NORM:PDIV <num>.....	60
[:SENSe]:GEN:NORM:RTRA < num >.....	60
Section IV Programing instances.....	61
Chapter II Function Description of Secondary Development Library.....	65
Section I Drive installation.....	65
Section II Function description.....	65
Instrument connection – Open device.....	65
IDQuery.....	66
Instrument connection – close instrument.....	66
IEEE488.2 function.....	66
Clear instrument state.....	66
Query instrument identification number.....	66
IDN.....	67
Operation complete command.....	67
Operation complete query.....	67
Reset.....	68
Wait.....	68
Universal functions for all measurement modes.....	68
Mode – query available instrument mode.....	68
Mode – set instrument mode.....	69
Mode – query instrument mode.....	69
Data – set data type.....	70
Data – query data type.....	70
File – delete state file.....	70
File – delete all state files.....	71
File – recall state file.....	71
File – save state file.....	71
File – set storage location.....	72
File – query storage location.....	72
File – store screen copy.....	73
System – set frequency reference.....	73
System – query frequency reference.....	73
System – GPS –set GPS on/off.....	74
System – GPS – query GPS on/off.....	74
System – GPS – query GPS state.....	75
System – GPS – query GPS receiver state.....	75
System – GPS – GPS reset.....	75
System – GPS – query GPS data.....	76
System – shutdown – set auto shutdown on/off.....	76
System – shutdown – query auto shutdown on/off.....	77
System – shutdown – set shutdown time.....	77
System – shutdown – query shutdown time.....	77
System – sleep – set sleep auto on/off.....	78
System – sleep – query sleep auto on/off.....	78
System – sleep – set sleep time.....	78
System – sleep – query sleep time.....	79
System – set title.....	79
System – set title on/off.....	79

System – query title on/off.....	80
System – set display mode.....	80
System – query display mode.....	81
System – set brightness auto adjustment on/off.....	81
System – query brightness auto adjustment on/off.....	81
System - set brightness level.....	82
System – query brightness level.....	82
Spectrum Analysis Mode Functions.....	82
Frequency – set center frequency.....	82
Frequency – query center frequency.....	83
Frequency – set step frequency.....	83
Frequency – query step frequency.....	84
Frequency – set step frequency auto on/off.....	84
Frequency – query step frequency auto on/off.....	84
Frequency – set span.....	85
Frequency – query span.....	85
Frequency – full span.....	85
Frequency – zero span.....	86
Frequency – previous span.....	86
Frequency – set start frequency.....	86
Frequency – query start frequency.....	87
Frequency – set stop frequency.....	87
Frequency – query stop frequency.....	87
Frequency – set signal standard name.....	88
Frequency – query signal standard name.....	88
Frequency – set signal standard channel number.....	88
Frequency – query signal standard channel number.....	89
Frequency – set zero span IF output on/off.....	89
Frequency – query zero span IF output on/off.....	89
Frequency – set zero span IF output IF selection.....	90
Frequency – query zero span IF output IF selection.....	90
Amplitude – set reference level.....	91
Amplitude – query reference level.....	91
Amplitude – set reference position.....	91
Amplitude – query reference position.....	92
Amplitude – set attenuation.....	92
Amplitude – query attenuation.....	92
Amplitude - set attenuation auto on/off.....	93
Amplitude - query attenuation auto on/off.....	93
Amplitude – set scale/division.....	93
Amplitude – query scale/division.....	94
Amplitude – set scale type.....	94
Amplitude – query scale type.....	94
Amplitude –set unit.....	95
Amplitude – query unit.....	95
Amplitude – set pre-amplifier on/off.....	96
Amplitude – query pre-amplifier on/off.....	96
Bandwidth – set resolution bandwidth.....	96
Bandwidth – query resolution bandwidth.....	97
Bandwidth – set video bandwidth.....	97
Bandwidth – query video bandwidth.....	97
Bandwidth – set resolution bandwidth auto on/off.....	98
Bandwidth – query resolution bandwidth auto on/off.....	98
Bandwidth – set video bandwidth auto on/off.....	99

Bandwidth – query video bandwidth auto on/off.....	99
Bandwidth – set SPAN/RBW.....	99
Bandwidth – query SPAN/RBW.....	100
Bandwidth – set RBW/VBW.....	100
Bandwidth – query RBW/VBW.....	100
Average – set average on/off.....	101
Average – query average on/off.....	101
Average – set average count.....	101
Average – query average count.....	102
Average – clear average count.....	102
Average – query current average count.....	102
Detector – set detector type.....	103
Detector – query detector type.....	103
Detector – set detector auto on/off.....	104
Detector – query detector auto on/off.....	104
Sweep – set sweep type.....	105
Sweep – query sweep type.....	105
Sweep – trigger single sweep.....	105
Sweep – set sweep mode.....	106
Sweep – query sweep mode.....	106
Sweep – set sweep time.....	106
Sweep – query sweep time.....	107
Sweep – set sweep time auto on/off.....	107
Sweep – query sweep time auto on/off.....	108
List edit – list add default segment.....	108
List edit – list delete segment.....	109
List edit – clear list.....	109
List edit – add segment.....	109
List edit – edit segment.....	110
Sweep – set trigger mode.....	111
Sweep – query trigger mode.....	111
Sweep – set video trigger level.....	111
Sweep – query video trigger level.....	112
Sweep – set external trigger level.....	112
Sweep – query external trigger level.....	113
Sweep- set external trigger polarity.....	113
Sweep – query external trigger polarity.....	113
Sweep – set external trigger delay.....	114
Sweep – query external trigger delay.....	114
Data – query trace data.....	114
Trace - set trace state.....	115
Trace – query trace state.....	115
Limit – set alarm on/off.....	116
Limit – query alarm on/off.....	116
Limit – set lower limit display on/off.....	117
Limit – query lower limit display on/off.....	117
Limit – set upper limit display on/off.....	118
Limit – query upper limit display on/off.....	118
Limit – set lower limit test on/off.....	118
Limit – query lower limit test on/off.....	119
Limit – set upper limit test on/off.....	119
Limit – query upper limit test on/off.....	119
Limit – set lower limit margin.....	120
Limit – query lower limit margin.....	120

Limit – set upper limit margin.....	120
Limit – query upper limit margin.....	121
Limit – lower limit add default point.....	121
Limit – lower limit delete default point.....	121
Limit – clear all lower limit points.....	122
Limit – lower limit edit point.....	122
Limit – upper limit add default point.....	122
Limit – upper limit delete current point.....	123
Limit – upper limit point clear.....	123
Limit – lower limit edit point.....	123
Marker – set market state.....	124
Marker – query marker state.....	124
Marker- activate marker.....	125
Marker- marker function (marker ->).....	125
Marker – disable all markers.....	126
Marker –set marker X value.....	126
Marker – query marker X value.....	127
Marker - search.....	127
Marker –set marker count on.....	128
Marker –query marker count on.....	128
Marker – query marker count frequency.....	129
Marker – set noise marker on.....	129
Marker – query noise marker on.....	130
Marker – set peak track on/off.....	130
Marker – query peak track on/off.....	130
Measurement – set function measurement.....	131
Measurement – query function measurement.....	132
Measurement – turn of measurement.....	132
Measurement – field strength – set antenna factor off.....	133
Measurement – field strength – set field strength on/off.....	133
Measurement – field strength – query field strength on/off.....	133
Measurement – field strength – edit antenna factor – add default point.....	134
Measurement – field strength – edit antenna factor – delete point.....	134
Measurement – field strength – edit antenna factor – edit point.....	134
Measurement – field strength – edit antenna factor – add point.....	135
Measurement – channel power – set channel power on/off.....	135
Measurement – channel power – query channel power state.....	135
Measurement – channel power – set channel power bandwidth.....	136
Measurement – channel power – query channel power bandwidth.....	136
Measurement – channel power – query channel power value.....	137
Measurement – channel power – query channel power density.....	137
Measurement – occupied bandwidth – set occupied bandwidth state.....	137
Measurement – occupied bandwidth – query occupied bandwidth state.....	138
Measurement – occupied bandwidth – set measurement method.....	138
Measurement – occupied bandwidth – query measurement method.....	139
Measurement – occupied bandwidth – set percentage.....	139
Measurement – occupied bandwidth – query percentage.....	139
Measurement – occupied bandwidth – set XdB.....	140
Measurement – occupied bandwidth – query XdB.....	140
Measurement – occupied bandwidth – query occupied bandwidth.....	140
Measurement – audio demodulation – set demodulation state.....	141
Measurement – audio demodulation – query demodulation state.....	141
Measurement – audio demodulation – set demodulation mode.....	141
Measurement – audio demodulation – query demodulation mode.....	142

Measurement – audio demodulation – set demodulation type.....	142
Measurement – audio demodulation – query demodulation type.....	143
Measurement – audio demodulation – set demodulation time.....	143
Measurement – audio demodulation – query demodulation time.....	144
Measurement – audio demodulation – set volume.....	144
Measurement – audio demodulation – query volume.....	144
Measurement – ACPR – set ACPR on/off.....	145
Measurement – ACPR – query ACPR on/off.....	145
Measurement – ACPR – set main channel bandwidth.....	145
Measurement – ACPR – query main channel bandwidth.....	146
Measurement – ACPR – set adjacent channel bandwidth.....	146
Measurement – ACPR – query adjacent channel bandwidth.....	147
Measurement – ACPR – set channel space.....	147
Measurement – ACPR – query channel space.....	147
Measurement – ACPR – set limit test on.....	148
Measurement – ACPR – query limit test on.....	148
Measurement – ACPR – set lower adjacent channel limit.....	148
Measurement – ACPR – query lower adjacent channel limit.....	149
Measurement – ACPR – set upper adjacent channel limit.....	149
Measurement – ACPR – query upper adjacent channel limit.....	149
Measurement – ACPR – query upper ACPR.....	150
Measurement – ACPR – query lower ACPR.....	150
Measurement – emission mask – set emission mask on.....	150
Measurement – emission mask – query emission mask on.....	151
Measurement – emission mask – set reference channel bandwidth.....	151
Measurement – emission mask – query reference channel bandwidth.....	152
Measurement – emission mask – set reference power type.....	152
Measurement – emission mask – query reference power type.....	152
Measurement – emission mask – set peak marker on in emission mask.....	153
Measurement – emission mask – query peak marker on of emission mask.....	153
Measurement – emission mask – query if emission mask fails.....	153
Measurement – CNR – set CNR state.....	154
Measurement – CNR – query CNR state.....	154
Measurement – CNR – set CNR carrier bandwidth.....	154
Measurement – CNR – query CNR carrier bandwidth.....	155
Measurement – CNR – set CNR noise bandwidth.....	155
Measurement – CNR – query CNR noise bandwidth.....	156
Measurement – CNR – set CNR frequency offset.....	156
Measurement – CNR – query CNR frequency offset.....	156
Measurement – CNR – query CNR measurement result.....	157
Measurement – IQ capture – set IQ capture state.....	157
Measurement – IQ capture – query IQ capture state.....	157
Measurement – IQ capture – start capture.....	158
Measurement – IQ capture – stop capture.....	158
Measurement – IQ capture – set capture time.....	158
Measurement – IQ capture – query capture time.....	159
Measurement – IQ capture – set IQ capture mode.....	159
Measurement – IQ capture – query IQ capture mode.....	159
Measurement – IQ capture – set sampling rate.....	160
Measurement – IQ capture – query sampling rate.....	160
Measurement – IQ capture – set IQ capture storage name.....	161
Measurement – IQ capture – set trigger mode.....	161
Measurement – IQ capture – query trigger mode.....	161
Measurement – IQ capture – set external trigger polarity.....	162

Measurement – IQ capture – query external trigger polarity.....	162
Measurement – IQ capture – set external trigger delay.....	163
Measurement – IQ capture – query external trigger delay.....	163
Measurement – IQ capture – set external trigger amplitude.....	163
Measurement – IQ capture – query external trigger amplitude.....	164
Measure- Generator- Set the generator ON/OFF.....	164
Measure- Generator- Query the generator ON/OFF.....	164
Measure- Generator- Set the frequency offset.....	165
Measure- Generator- Query the frequency offset.....	165
Measure- Generator- Set the generator frequency.....	165
Measure- Generator- Query the generator frequency.....	166
Measure- Generator- Set the generator power.....	166
Measure- Generator- Query the generator power.....	166
Measure- Generator- Set the generator power offset.....	167
Measure- Generator- Query the generator power offset.....	167
Measure- Generator- Set the generator normalization ON/OFF.....	167
Measure- Generator- Query the generator normalization ON/OFF.....	168
Measure- Generator- Set the generator normalization reference level.....	168
Measure- Generator- Query the generator normalization reference level.....	168
Measure- Generator- Set the generator normalization reference position.....	169
Measure- Generator- Query the generator normalization reference position.....	169
Measure- Generator- Set the generator normalization scale/division.....	169
Measure- Generator- Query the generator normalization scale/division.....	170
Measure- Generator- Set the generator reference trace ON/OFF.....	170
Measure- Generator- Query the generator reference trace ON/OFF.....	171
Zero calibration – execute.....	171
File – recall antenna factor.....	171
File – store antenna factor.....	172
File – delete antenna factor.....	172
File – delete all antenna factors.....	172
File – emission mask recall limit line.....	173
File – store list to file.....	173
File – recall list file.....	173
File – delete list file.....	174
File – delete all list files.....	174
File – delete limit file.....	174
File – store limit file.....	175
File – recall limit line.....	175
File - delete all limit files.....	175
File – delete data file.....	176
File – delete all data files.....	176
File – recall data file.....	176
File – store data file.....	177
Interference Analysis Mode Functions.....	177
Frequency – set center frequency.....	177
Frequency – query center frequency.....	177
Frequency – set span.....	178
Frequency – query span.....	178
Frequency – full span.....	179
Frequency – zero span.....	179
Frequency – previous span.....	179
Frequency – set start frequency.....	179
Frequency – query start frequency.....	180
Frequency – set stop frequency.....	180

Frequency – query stop frequency.....	181
Frequency – set step frequency.....	181
Frequency – query step frequency.....	181
Frequency – set step frequency auto on/off.....	182
Frequency – query step frequency auto on/off.....	182
Frequency – set signal standard name.....	182
Frequency – query signal standard name.....	183
Frequency – set signal standard channel number.....	183
Frequency – query signal standard channel number.....	183
Amplitude – set reference level.....	184
Amplitude – query reference level.....	184
Amplitude – set reference position.....	184
Amplitude – query reference position.....	185
Amplitude – set attenuation.....	185
Amplitude – query attenuation.....	186
Amplitude set attenuation auto on/off.....	186
Amplitude query attenuation auto on/off.....	186
Amplitude – set scale/division.....	187
Amplitude – query scale/division.....	187
Amplitude – set pre-amplifier on/off.....	187
Amplitude – query pre-amplifier on/off.....	188
Bandwidth – set resolution bandwidth.....	188
Bandwidth – query resolution bandwidth.....	188
Bandwidth – set video bandwidth.....	189
Bandwidth – query video bandwidth.....	189
Bandwidth – set resolution bandwidth auto on/off.....	189
Bandwidth – query resolution bandwidth auto on/off.....	190
Bandwidth – set video bandwidth auto on/off.....	190
Bandwidth – query video bandwidth auto on/off.....	191
Bandwidth – set SPAN/RBW.....	191
Bandwidth – query SPAN/RBW.....	191
Bandwidth – set RBW/VBW.....	192
Bandwidth – query RBW/VBW.....	192
Average – set average on/off.....	192
Average – query average on/off.....	193
Average – set average count.....	193
Average – query average count.....	193
Average – clear average count.....	194
Average – query current average count.....	194
Detector – set detector type.....	194
Detector – query detector type.....	195
Detector – set detector auto on/off.....	195
Detector – query detector auto on/off.....	196
Sweep – set sweep type.....	196
Sweep – query sweep type.....	197
Sweep – trigger single sweep.....	197
Sweep – set sweep time.....	197
Sweep – query sweep time.....	198
Sweep – set sweep time auto on/off.....	198
Sweep – query sweep time auto on/off.....	198
Trace – set trace state.....	199
Trace – query trace state.....	199
Marker – set market state.....	200

Marker – query marker state.....	200
Marker- activate marker.....	201
Marker- marker function (marker ->).....	201
Marker – disable all markers.....	202
Marker –set marker X value.....	202
Marker – query marker X value.....	203
Marker - search.....	204
Marker – set noise marker on.....	204
Marker – query noise marker on.....	205
Measurement – set measurement mode.....	205
Measurement – query measurement mode.....	205
Auto save – set span time.....	206
Auto save – query span time.....	206
Auto save – set auto save state.....	207
Auto save – query auto save state.....	207
Auto save – set time cursor.....	207
Auto save – set sweep interval.....	208
Auto save – query sweep interval.....	208
Auto save – restart measurement.....	208
File – delete data file.....	209
File – delete all data files.....	209
File – recall data file.....	209
File – store data file.....	210
Data – query trace data.....	210
AM-FM-PM Analyzer Mode Functions.....	211
Frequency – set center frequency.....	211
Frequency – query center frequency.....	211
Frequency – set step frequency.....	211
Frequency – query step frequency.....	212
Frequency – set step frequency auto on/off.....	212
Frequency – query step frequency auto on/off.....	212
Frequency – set span.....	213
Frequency – query span.....	213
Frequency – previous span.....	213
Frequency – set start frequency.....	214
Frequency – query start frequency.....	214
Frequency – set stop frequency.....	214
Frequency – query stop frequency.....	215
Frequency – set signal standard name.....	215
Frequency – query signal standard name.....	216
Frequency – set signal standard channel number.....	216
Frequency – query signal standard channel number.....	216
Amplitude – set reference level.....	217
Amplitude – query reference level.....	217
Amplitude – set reference position.....	217
Amplitude – query reference position.....	218
Amplitude – set attenuation.....	218
Amplitude – query attenuation.....	218
Amplitude set attenuation auto on/off.....	219
Amplitude query attenuation auto on/off.....	219
Amplitude – set scale/division.....	219
Amplitude – query scale/division.....	220
Amplitude – set pre-amplifier on/off.....	220
Amplitude – query pre-amplifier on/off.....	221

Average – set average on/off.....	221
Average – query average on/off.....	221
Average – set average count.....	222
Average – query average count.....	222
Average – clear average count.....	222
Average – query current average count.....	223
Sweep – set sweep type.....	223
Sweep – query sweep type.....	223
Sweep – trigger single sweep.....	224
Marker – set market state.....	224
Marker – query marker state.....	225
Marker- activate marker.....	225
Marker – disable all markers.....	225
Marker –set marker X value.....	226
Marker – query marker X value.....	226
Marker - search.....	227
RF spectrum – occupied bandwidth – set occupied bandwidth state.....	227
RF spectrum – occupied bandwidth – query occupied bandwidth state.....	228
RF spectrum – occupied bandwidth – set measurement method.....	228
RF spectrum – occupied bandwidth – query measurement method.....	228
RF spectrum – occupied bandwidth – set percentage.....	229
RF spectrum – occupied bandwidth – query percentage.....	229
RF spectrum – occupied bandwidth – set XdB.....	230
RF spectrum – occupied bandwidth – query XdB.....	230
Measurement – set demodulation type.....	230
Measurement – query demodulation type.....	231
Measurement – set display mode.....	231
Measurement – query display mode.....	232
Audio spectrum – set span.....	232
Audio spectrum – query span.....	232
Audio spectrum – set scale.....	233
Audio spectrum – query scale.....	233
Audio waveform – set sweep time.....	233
Audio waveform – query sweep time.....	234
Bandwidth – set IF bandwidth.....	234
Audio waveform – query IF bandwidth.....	235
Marker – select trace.....	235
File – delete data file.....	235
File – delete all data files.....	236
File – recall data file.....	236
File – store data file.....	236
Data – query trace data.....	237
Power Meter Mode Functions.....	237
Frequency – set center frequency.....	237
Frequency – query center frequency.....	238
Frequency – set frequency resolution.....	238
Frequency – query frequency resolution.....	239
Scale/division – auto scale.....	239
Scale/division – set minimum value.....	239
Scale/division – query minimum value.....	240
Scale/division – set maximum value.....	240
Scale/division – query maximum value.....	240
Scale/division – query relative measurement value.....	241
Scale/division – set relative measurement state.....	241

Scale/division – query relative measurement state.....	241
Scale/division – set offset value.....	242
Scale/division – query offset value.....	242
Scale/division – set offset state.....	242
Scale/division – query offset state.....	243
Scale/division – set maximum hold state.....	243
Scale/division – query maximum hold state.....	243
Average – set average on/off.....	244
Average – query average on/off.....	244
Average – set average count.....	245
Average – query average count.....	245
Average – clear average count.....	245
Average – query current average count.....	246
Calibration - execute.....	246
Calibration – query zero calibration state.....	246
Limit – set limit state.....	247
Limit – query limit state.....	247
Limit – set upper limit.....	247
Limit – query upper limit.....	248
Limit – set lower limit.....	248
Limit – query lower limit.....	248
Limit – set alarm on/off.....	249
Limit – query alarm on/off.....	249
Channel Sweep Mode Functions.....	250
Amplitude – set reference level.....	250
Amplitude – query reference level.....	250
Amplitude – set scale/division.....	250
Amplitude – query scale/division.....	251
Measurement – set graph/table display.....	251
Measurement – query graph/table display.....	251
Measurement – set maximum hold state.....	252
Measurement – query maximum hold state.....	252
Measurement – set unit.....	252
Measurement – query unit.....	253
Measurement – set power display.....	253
Measurement – query power display.....	253
Measurement – set color.....	254
Measurement – query color.....	254
Measurement – set graph/table direction.....	254
Measurement – query graph/table direction.....	255
Sweep – set sweep mode.....	255
Sweep – query sweep mode.....	256
Sweep – channel sweep – set signal standard name.....	256
Sweep – channel sweep – query signal standard name.....	256
Sweep – channel sweep – set signal standard channel number.....	257
Sweep – channel sweep – query signal standard channel number.....	257
Sweep – channel sweep – set channel number.....	257
Sweep – channel sweep – query channel number.....	258
Sweep – channel sweep – set channel step.....	258
Sweep – channel sweep – query channel step.....	258
Sweep – channel sweep – set channel bandwidth.....	259
Sweep – channel sweep – query channel bandwidth.....	259
Sweep – frequency sweep – set start frequency.....	259
Sweep – frequency sweep – query start frequency.....	260

Sweep – frequency sweep – set step frequency.....	260
Sweep – frequency sweep – query step frequency.....	260
Sweep – frequency sweep – set channel bandwidth.....	261
Sweep – frequency sweep – query channel bandwidth.....	261
Sweep – frequency sweep – set channel number.....	262
Sweep – frequency sweep – query channel number.....	262
Sweep – list sweep – set channel number.....	262
Sweep – list sweep – query channel number.....	263
File – delete data file.....	263
File – delete all data files.....	263
File – recall data file.....	264
File – store data file.....	264
Field Strength Mode Functions.....	264
Frequency – set point frequency.....	264
Frequency – query point frequency.....	265
Frequency – set step frequency.....	265
Frequency – query step frequency.....	265
Frequency – set auto step frequency on.....	266
Frequency – Query auto step frequency on.....	266
Frequency – Set frequency track on.....	267
Frequency – Query frequency track on.....	267
Frequency – Set the start frequency for frequency scanning.....	267
Frequency – Query the start frequency for frequency scanning.....	268
Frequency – Set the step frequency for frequency scanning.....	268
Frequency – Query the step frequency for frequency scanning.....	268
Frequency – Set the scan points.....	269
Frequency – Query the scan points.....	269
Amplitude – Set the reference level.....	269
Amplitude – Query the reference level.....	270
Amplitude – set attenuation.....	270
Amplitude – query attenuation.....	270
Amplitude - set attenuation auto on/off.....	271
Amplitude - query attenuation auto on/off.....	271
Amplitude – set scale/division.....	271
Amplitude – query scale/division.....	272
Amplitude – set unit.....	272
Amplitude – query unit.....	273
Amplitude – set pre-amplifier on/off.....	273
Amplitude – query pre-amplifier on/off.....	273
Detector – set detector type.....	274
Detector – query detector type.....	274
Sweep – set sweep type.....	275
Sweep - query sweep type.....	275
Sweep - trigger single sweep.....	275
Sweep – set dwell time.....	276
Sweep - query dwell time.....	276
Sweep – set dwell time auto on.....	276
Sweep - query dwell time auto on.....	277
Sweep - set stay time.....	277
Sweep – query stay time.....	278
Sweep – set stay time auto on.....	278
Sweep - query stay time auto on.....	278
List edit – list add default segment.....	279
List edit – list delete segment.....	279

List edit – clear list.....	279
List edit – add segment.....	280
List edit – edit segment.....	280
Limit – set alarm on/off.....	281
Limit – query alarm on/off.....	282
Limit – set limit.....	282
Limit – query limit.....	282
Limit – set limit on/off.....	283
Limit – query limit on/off.....	283
Demodulation – set demodulation type.....	283
Demodulation – query demodulation type.....	284
Demodulation – set demodulation volume.....	284
demodulation - query demodulation volume.....	285
bandwidth – set bandwidth.....	285
bandwidth – query bandwidth.....	285
Measurement – set measurement type.....	286
Measurement – query measurement type.....	286
Marker - peak.....	286
Marker – set marker on/off.....	287
Marker – query marker on/off.....	287
Marker – set marker index.....	287
Marker – query marker index.....	288
Data – query point scan amplitude value.....	288
Data – query point scan field strength value.....	288
Data – query frequency scan amplitude value.....	289
Data – query frequency scan field strength value.....	289
Data – query list scan amplitude value.....	290
Data – query list scan field strength value.....	290
Data – query offset.....	290
Antenna – set antenna off.....	291
File – recall antenna factor.....	291
File - store antenna factor.....	291
File - delete antenna factor.....	292
File - delete all antenna factors.....	292
File - delete data file.....	292
File - delete all data files.....	293
File - recall data file.....	293
File - store data file.....	293
Edit antenna factor – add default point.....	294
Edit antenna factor – delete point.....	294
Edit antenna factor – edit point.....	294
Edit antenna factor – add point.....	295
Signal analysis mode function.....	295
Frequency- Set the center frequency.....	295
Frequency- Query the center frequency.....	296
Frequency- Set the step frequency.....	296
Frequency- Query the step frequency.....	296
Frequency- Set the step frequency auto ON/OFF.....	297
Frequency- Query the step frequency auto ON/OFF.....	297
Frequency- Set the span.....	298
Frequency- Query the span.....	298
Frequency- Full span.....	298
Frequency- Zero span.....	299
Frequency- Last span.....	299

Frequency- Set the start frequency.....	299
Frequency- Query the start frequency.....	299
Frequency- Set the stop frequency.....	300
Frequency- Query the stop frequency.....	300
Frequency- Set the signal standard name.....	301
Frequency- Query the signal standard name.....	301
Frequency- Set the signal standard channel number.....	301
Frequency- Query the signal standard channel number.....	302
Amplitude - Set the reference level.....	302
Amplitude - Query the reference level.....	302
Amplitude-Set the reference position.....	303
Amplitude-Query the reference position.....	303
Amplitude – Set the attenuation.....	303
Amplitude - Query the attenuation.....	304
Amplitude – Set the attenuation auto ON/OFF.....	304
Amplitude - Query the attenuation auto ON/OFF.....	304
Amplitude - Set the scale/ division.....	305
Amplitude - Query the scale/ division.....	305
Amplitude-Set the pre-amplifier ON/OFF.....	305
Amplitude-Query the pre-amplifier ON/OFF.....	306
Bandwidth- Set the resolution bandwidth.....	306
Bandwidth- Query the resolution bandwidth.....	306
Bandwidth- Set the video bandwidth.....	307
Bandwidth- Query the video bandwidth.....	307
Bandwidth- Set the resolution bandwidth auto ON/OFF.....	307
Bandwidth- Query the resolution bandwidth auto ON/OFF.....	308
Bandwidth- Set the video bandwidth auto ON/OFF.....	308
Bandwidth- Query the video bandwidth auto ON/OFF.....	308
Bandwidth- Set SPAN/RBW.....	309
Bandwidth- Query SPAN/RBW.....	309
Bandwidth- Set RBW/VBW.....	309
Bandwidth- Query RBW/VBW.....	310
Average- Set the average ON/OFF.....	310
Average- Query the average ON/OFF.....	310
Average- Set the average count.....	311
Average- Query the average count.....	311
Average- Clear the average count.....	311
Average- Query the current average count.....	312
Detector - Set the detector type.....	312
Detector - Query the detector type.....	313
Detector- Set the detector auto ON/OFF.....	313
Detector-Query the detector auto ON/OFF.....	314
Sweep- Set the sweep type.....	314
Sweep- Query the sweep type.....	314
Sweep- Trigger the single sweep.....	315
Sweep- Set the sweep time.....	315
Sweep- Query the sweep time.....	315
Sweep- Set the sweep time auto ON/OFF.....	316
Sweep- Query the sweep time auto ON/OFF.....	316
Marker- Set the marker state.....	317
Marker- Query the marker state.....	317
Marker- Activate the marker.....	318
Marker- Marker function (marker->).....	318
Marker- Disable all the markers.....	319

Marker- Set the marker X value.....	319
Marker- Query the X and Y values of marker.....	320
Marker- Search.....	320
Marker- Set the noise marker ON/OFF.....	321
Marker- Query the noise marker ON/OFF.....	321
Demodulator-Set the demodulator ON/OFF.....	322
Demodulator-Query the demodulator ON/OFF.....	322
Demodulator-Set the demodulation mode.....	323
Demodulator-Query the demodulation mode.....	323
Demodulator-Set the demodulation type.....	323
Demodulator-Query the demodulation type.....	324
Demodulator-Set the demodulation time.....	324
Demodulator-Query the demodulation time.....	325
Demodulator-Set the volume.....	325
Demodulator- Query the volume.....	325
IQ capture - Set the IQ capture ON/OFF.....	326
IQ capture - Query the IQ capture ON/OFF.....	326
IQ capture - Start the capture.....	326
IQ capture - Stop the capture.....	327
IQ capture - Set the capture time.....	327
IQ capture - Query the capture time.....	327
IQ capture - Set the IQ capture mode.....	328
IQ capture - Query the IQ capture mode.....	328
IQ capture - Set the sample rate.....	328
IQ capture - Query the sample rate.....	329
IQ capture - Set the IQ capture file name.....	329
IQ capture - Set the trigger mode.....	330
IQ capture - Query the trigger mode.....	330
IQ capture - Set the external trigger slope.....	330
IQ capture - Query the external trigger slope.....	331
IQ capture - Set the external trigger delay.....	331
IQ capture - Query the external trigger delay.....	332
IQ capture - Set the external trigger level.....	332
IQ capture - Query the external trigger level.....	332
Auto save - Set the span time.....	333
Auto save - Query the span time.....	333
Auto save - Set the auto save ON/OFF.....	333
Auto save - Query the auto save ON/OFF.....	334
Auto save - Set the time cursor.....	334
Auto save - Restart the measurement.....	334
Configuration- Set the limit ON/OFF.....	335
Configuration- Query the limit ON/OFF.....	335
Configuration- Set the limit value.....	335
Configuration- Query the limit value.....	335
Configuration- Set the signal list display mode.....	336
Configuration- Query the signal list display mode.....	336
Save/ recall - Delete the data file.....	336
Save/ recall - Delete all the data files.....	336
Save/ recall - Recall the data file.....	336
Save/ recall - Save the data file.....	337

Chapter I Introduction to SCPI Commands

Section I Introduction of SCPI command operation

SCPI (Standard Commands for Programmable Instruments) is a new command language for instrument control developed in accordance with IEEE488.2 standard. The main purpose of SCPI is to ensure that same program control commands are applicable to the same type of instrument, so as to realize standardization of program control commands.

This chapter contains the details of all SCPI commands identified and executed by the S3302 series spectrum analyzer. Introduction and description of IEEE488.2 general commands and measurement commands are also included in this chapter.

Each measurement command has its applicable mode. Under the context when the mode is not applicable, an error identification will be returned for query command.

Section II IEEE 488.2 commands

*CLS – Clear the state

Clear the state of the instrument, i.e.: Clear wrong queue and all event registers. And clear all *OPC commands and query commands to be processed.

*IDN? - Identification

Return the unique identification character string of the device and the identification varies with models.

*OPC - Operation complete command

After completing all overlapping commands to be processed (for example: One sweep or Default command, etc.), set the OPC bit of standard event status register.

*OPC? - Operation complete query

Number “1” will be returned when all overlapping commands to be processed are completed.

*RST - Reset

Perform reset operation and cancel all *OPC commands or query commands to be processed. The contents in the nonvolatile memory of the instrument will not be lost.

*WAI - Wait

The instrument will process all overlapping commands before processing new commands.

Section III Measurement commands

:CALCulate[:SElected]:LIMit:BEEP <bool>

(Read and write) query or set limit beep on/off.

Applicable mode Spectrum Analysis, Power Meter, Field Strength

Parameter Beep on/off.

OFF(0)

Beep off

ON(1)

Beep on

Example :CALC:LIM:BEEP ON

Query syntax :CALC:LIM:BEEP?

Default OFF

Return type Numeric value (bool) or character

:CALCulate[:SElected]:LIMit:LOWer:DISPlay <bool>

(Read and write) query or set lower limit display on/off.

Applicable mode Spectrum Analysis

Parameter Lower limit display on/off.

OFF(0)

Display off

ON(1)

Display on

Example :CALC:LIM:LOW:DISP ON

Query syntax :CALC:LIM:LOW:DISP?

Default OFF

Return type Numeric value (bool) or character

:CALCulate[:SElected]:LIMit:LOWer:MARGin <num>

(Read and write) query or set lower limit margin.

Applicable mode Spectrum Analysis

Parameter Lower limit margin.

Parameter range: 0~40 Unit: dBm

Example :CALC:LIM:LOW:MARG 10

Query syntax :CALC:LIM:LOW:MARG?

Default 0

Return type Numeric value (float) or character

:CALCulate[:SElected]:LIMit:LOWer:TEST <bool>

(Read and write) query or set lower limit test on/off.

Applicable mode Spectrum Analysis

Parameter Lower limit test on/off.

OFF(0)

Test off

ON(1)

Test on

Example :CALC:LIM:LOW:TEST ON

Query syntax :CALC:LIM:LOW:TEST?

Default OFF

Return type Numeric value (bool) or character

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

(Write only) add default point to lower limit, and the default setting of default point is: limit

frequency: 44.1GHz, limit value: 0dBm.

Applicable mode	Spectrum Analysis
Parameter	None
Example	:CALC:LIM:LOW:EDIT:ADD
Query syntax	None
Default	None
Return type	None

:CALCulate[:SElected]:LIMit:LOWer:EDIT:DELeTe

(Write only) delete current point in lower limit.

Applicable mode	Spectrum Analysis
Parameter	None
Example	:CALC:LIM:UPP:EDIT:DEL
Query syntax	None
Default	None
Return type	None

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

(Write only) delete all edit points in lower limit.

Applicable mode	Spectrum Analysis
Parameter	None
Example	:CALC:LIM:LOW:EDIT:CLE
Query syntax	None
Default	None
Return type	None

:CALCulate[:SElected]:LIMit:LOWer:EDIT:DATA

(Write only) lower limit edit point.

Applicable mode	Spectrum Analysis
Parameter	Point index (int, start from 0), limit frequency (0~44.1GHz), limit value (-174dBm to 50dBm)
Example	:CALC:LIM:LOW:EDIT:DATA
Query syntax	None
Default	None
Return type	None

:CALCulate[:SElected]:LIMit:UPPer:DISPlay <bool>

(Read and write) query or set upper limit display on/off.

Applicable mode	Spectrum Analysis
Parameter	Lower limit display on/off. OFF(0) Display off ON(1) Display on
Example	:CALC:LIM:UPP:DISP ON
Query syntax	:CALC:LIM:UPP:DISP?
Default	OFF
Return type	Numeric value (bool) or character

:CALCulate[:SElected]:LIMit:UPPer:MARGin <num>

(Read and write) query or set upper limit margin.

Applicable mode	Spectrum Analysis
Parameter	Upper limit margin.
	Parameter range: -40~0, unit: dBm
Example	:CALC:LIM:UPP:MARG -10
Query syntax	:CALC:LIM:UPP:MARG?
Default	0
Return type	Numeric value (float) or character

:CALCulate[:SElected]:LIMit:UPPer:TEST <bool>

(Read and write) query or set upper limit test on/off.

Applicable mode	Spectrum Analysis
Parameter	Upper limit test on/off.
	OFF(0) Test off
	ON(1) Test on
Example	:CALC:LIM:UPP:TEST ON
Query syntax	:CALC:LIM:UPP:TEST?

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

(Write only) add default point in lower limit, and default setting of default point is: Limit frequency: 44.1GHz, limit value: 0dBm

Applicable mode	Spectrum Analysis
Parameter	None
Example	:CALC:LIM:UPP:EDIT:ADD
Query syntax	None
Default	None
Return type	None

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

(Write only) delete all edit points in upper limit

Applicable mode	Spectrum Analysis
Parameter	None
Example	:CALC:LIM:UPP:EDIT:CLE
Query syntax	None
Default	None
Return type	None

:CALCulate[:SElected]:LIMit:UPPer:EDIT:DELeTe

(Write only) delete current point in upper limit.

Applicable mode	Spectrum Analyser
Parameter	None
Example	:CALC:LIM:UPP:EDIT:DEL
Query syntax	None
Default	None
Return type	None
Default	OFF
Return type	Numeric value (bool) or character

:CALCulate[:SElected]:LIMit:UPPer:EDIT:DATA

(Write only) upper limit edit point.

Applicable mode Spectrum Analysis

Parameter Point index (int, start from 0), limit frequency (0~44.1GHz), limit value (-174dBm to 50dBm)

Example :CALC:LIM:UPP:EDIT:DATA

Query syntax None

Default None

Return type None

:CALCulate[:SELEcted]:LIST:EDIT:ADD(**Options**)

(Write only) edit list to add default segments and set the default of default segments as follows:

Spectrum Analysis		Field Strength	
Starting frequency	1GHz	Frequency	500MHz
Stop frequency	2GHz	Bandwidth	30kHz
Sweep points	51	Detector	Average
Resolution bandwidth	1MHz	Demodulation	Continuous wave
Video bandwidth	30kHz	Limit	-174dBm
On/off	Off	Limit display	Off

Applicable mode Spectrum Analysis, Field Strength

Parameter None

Example :CALC:LIST:EDIT:ADD

Query syntax None

Default None

Return type None

:CALCulate[:SELEcted]:LIST:EDIT:ADD:SEGment

(Write only) edit list to add segments

Applicable mode Spectrum Analysis, Field Strength

Parameter Spectrum Analysis Mode: start frequency (0 - 44.1GHz), stop frequency (0 - 44.1GHz), sweep point (51 - 501), resolution bandwidth (1Hz - 10MHz), video bandwidth (1Hz - 10MHz), state (ON OFF)
Field Strength Mode: frequency (1MHz - 44.1GHz), bandwidth (150Hz - 150kHz), detector (average, real time, peak), demodulation, limit (-174dBm to 50dBm), limit on/off

Example :CALC:LIST:EDIT:ADD:SEG 100000000,200000000,501,3000,1000,ON

Query syntax None

Default None

Return type None

:CALCulate[:SELEcted]:LIST:EDIT:CLEar

(Write only) edit list to clear the list.

Applicable mode Spectrum Analysis, Field Strength

Parameter	None
Example	:CALC:LIST:EDIT:CLE
Query syntax	None
Default	None
Return type	None

:CALCulate[:SElected]:LIST:EDIT:DELeTe

(Write only) edit list to delete segment.

Applicable mode	Spectrum Analysis, Field Strength
Parameter	Segment index (int), start from 0
Example	:CALC:LIST:EDIT:DEL 1
Query syntax	None
Default	None
Return type	None

:CALCulate[:SElected]:LIST:EDIT: SEGment

(Write only) edit list to edit segment

Applicable mode	Spectrum Analysis, Field Strength
Parameter	Segment index (start from 0), start frequency (0~44.1GHz), stop frequency (0~44.1GHz), sweep point (51~501), resolution bandwidth (1Hz~10MHz), video bandwidth (1Hz~10MHz), state (ON OFF) Field Strength Mode: Segment index (start from 0), frequency (1MHz~44.1GHz), bandwidth (150Hz~150kHz), detector (average, real time, peak), demodulation, limit (-174dBm~50dBm), limit on/off
Example	:CALC:LIST:EDIT:SEG 1,100000000,200000000,501,3000,1000,ON
Query syntax	None
Default	None
Return type	None

:CALCulate[:SElected]:MARKer<n>[:STATe] <string>

(Read and write) set or query marker state.

Applicable mode	Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation
Parameter	Marker number, the number can be 1,2,3,4,5,6, indicating Markers 1,2,3,4,5,6. If not marked, then n represents 1.
<n>	Marker state.
<string>	OFF(0) Marker off. NORM(1) Normal marker on. DELTA(2) Delta marker on.
Example	:CALC:MARK1 NORM
Query syntax	:CALC:MARK1?
Default	OFF
Return type	Numeric value (int) or character

:CALCulate[:SElected]:MARKer<n>:ACTivate

(Write only) activate marker n.

Applicable mode Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation

Parameter <n> Marker number, the number can be 1,2,3,4,5,6, indicating Markers 1,2,3,4,5,6.
If not marked, then n represents 1.

Example :CALC:MARK1:ACT

Query syntax None

Default None

Return type None

:CALCulate[:SElected]:MARKer<n>:SET <string>

(Write only) set marker function (marker ->)(if Marker is off, then enable the marker first).

Applicable mode Spectrum Analysis, Interference Analysis

Parameter <n> Marker number, the number can be 1,2,3,4,5,6, indicating Markers 1,2,3,4,5,6.
If not marked, then n represents 1.

<string> Marker function.
Parameters and functions matching with span and non-zero span modes

Instrument mode	Parameter	Function
Non-zero span	START,STOP,CENTER,STEP	Set the start, stop, center and step frequencies as current marker frequency
Zero span	START,STOP,CENTER,STEP	Set marker index as 0, max index, center index; set step frequency as current marker frequency.

Example :CALC:MARK1:SET STAR

Query syntax None

Default None

Return type None

:CALCulate[:SElected]:MARKer:AOFF

(Write only) disable all markers.

Applicable mode Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation

Parameter None.

Example :CALC:MARK:AOFF

Query syntax None

Default None

Return type None

:CALCulate[:SElected]:MARKer<n>:X <num>

(Read and write) set or query the value of marker X (invalid if the marker is off).

Applicable mode	Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation
Parameter	Marker number, the number can be 1,2,3,4,5,6, indicating Markers 1,2,3,4,5,6.
<n>	If not marked, then n represents 1.
<num>	Value of marker X (see table below for unit). If delta marker is selected, the X value can be negative.

Instrument mode	Parameter unit
Spectrum Analysis (non-zero span)	Hz
Spectrum Analysis (zero span)	ms
Interference Analysis (non-zero span)	Hz
Interference Analysis (zero span)	ms
AM-FM-PM Demodulation (RF spectrum)	Hz
AM-FM-PM Demodulation (audio spectrum)	Hz
AM-FM-PM Demodulation (audio waveform)	ms

Time unit: ms. Frequency unit: Hz.

Example :CALC:MARK1:X 10000

Query :CALC:MARK1:X?

syntax

Default When creating the marker, the marker should be set as the center index point.

Return type Numeric value (double) or character

:CALCulate[:SElected]:MARKer<n>:Y?

(Read only) query Y value of marker **(0 will be returned if the marker is off)**.

Applicable mode Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation

Parameter <n> Marker number, the number can be 1,2,3,4,5,6, indicating Markers 1,2,3,4,5,6.

If not marked, then n represents 1.

Example :CALC:MARK1:Y?

Query syntax :CALC:MARK1:Y?

Default None

Return type Numeric value (float) or character

Instrument mode	Return parameter	Unit
Spectrum Analysis (single float value)	Amplitude	dBm
Interference Analysis (single float value)	Amplitude	dBm
AM-FM-PM Demodulation: RF spectrum(single float value)	Amplitude	dBm
AM-FM-PM Demodulation: Audio spectrum (single float value)	Amplitude	dBm
AM-FM-PM Demodulation: Audio waveform (single float value)	Amplitude	AM: Percentage FM:Hz PM:Rad

:CALCulate[:SElected]:MARKer<n>:FCOunt[:STATe] <bool>

(Read and write) set or query counter on/off **(the marker will be set as common marker).**

Applicable mode	Spectrum Analysis	
Parameter <n>	Marker number, the number can be 1,2,3,4,5,6, indicating Markers 1,2,3,4,5,6. If not marked, then n represents 1.	
<bool>	Counter on/off. OFF(0) ON(1)	Counter off. Counter on.

Notes: Only one marker counter can be enabled now.

Example	:CALC:MARK1:FCO ON
Query syntax	:CALC:MARK1:FCO?
Default	OFF
Return type	Numeric value (bool) or character

:CALCulate[:SElected]:MARKer<n>:FCOunt:X?

(Read only) query the frequency count of counter **(0 will be returned if no count is made).**

Applicable mode	Spectrum Analysis
Parameter <n>	Marker number, the number can be 1,2,3,4,5,6, indicating Markers 1,2,3,4,5,6. If not marked, then n represents 1.
Parameter	None.
Example	:CALC:MARK1:FCO:X?
Query syntax	:CALC:MARK1:FCO:X?
Default	None
Return type	Numeric value (double) or character

:CALCulate[:SElected]:MARKer<n>:FUNCtion:MAXimum

(Write only) maximum value of marker search **(enable the marker first if the marker is off).**

Applicable mode	Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation
Parameter <n>	Marker number, the number can be 1,2,3,4,5,6, indicating Markers 1,2,3,4,5,6. If not marked, then n represents 1.
Example	:CALC:MARK1:FUNC:MAX
Query syntax	None
Default	None
Return type	None

:CALCulate[:SElected]:MARKer<n>:FUNCtion:MINimum

(Write only) minimum value of marker search **(enable the marker first if the marker is off).**

Applicable mode	Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation
Parameter <n>	Marker number, the number can be 1,2,3,4,5,6, indicating Markers 1,2,3,4,5,6. If not marked, then n represents 1.
Example	:CALC:MARK1:FUNC:MIN

Query syntax None
Default None
Return type None

:CALCulate[:SElected]:MARKer<n>:FUNCtion:PEAK

(Write only) peak value of marker search **(enable the marker first if the marker is off).**

Applicable mode Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation
Parameter <n> Marker number, the number can be 1,2,3,4,5,6, indicating Markers 1,2,3,4,5,6.
If not marked, then n represents 1.
Example :CALC:MARK1:FUNC:PEAK
Query syntax None
Default None
Return type None

:CALCulate[:SElected]:MARKer<n>:FUNCtion:PLEFt

(Write only) left peak value of marker search **(enable the marker first if the marker is off).**

Applicable mode Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation
Parameter <n> Marker number, the number can be 1,2,3,4,5,6, indicating Markers 1,2,3,4,5,6.
If not marked, then n represents 1.
Example :CALC:MARK1:FUNC:PLEF
Query syntax None
Default None
Return type None

:CALCulate[:SElected]:MARKer<n>:FUNCtion:PNEXt

(Write only) secondary peak value of marker search **(enable the marker first if the marker is off).**

Applicable mode Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation
Parameter <n> Marker number, the number can be 1,2,3,4,5,6, indicating Markers 1,2,3,4,5,6.
If not marked, then n represents 1.
Example :CALC:MARK1:FUNC:PNEX
Query syntax None
Default None
Return type None

:CALCulate[:SElected]:MARKer<n>:FUNCtion:PRIGht

(Write only) right peak value of marker search **(enable the marker first if the marker is off).**

Applicable mode Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation
Parameter <n> Marker number, the number can be 1,2,3,4,5,6, indicating Markers 1,2,3,4,5,6.
If not marked, then n represents 1.
Example :CALC:MARK1:FUNC:PRIG
Query syntax None
Default None
Return type None

:CALCulate[:SElected]:MARKer<n>:NOISe[:STATe] <bool>

(Read and write) set or query noise marker **(enable the marker first if the marker is off)**.

Applicable mode	Spectrum Analysis, Interference Analysis	
Parameter	Marker number, the number can be 1,2,3,4,5,6, indicating Markers 1,2,3,4,5,6.	
<n>	If not marked, then n represents 1.	
<bool>	Noise marker on/off.	
	OFF(0)	Noise marker off.
	ON(1)	Noise marker on.
Example	:CALC:MARK1:NOIS ON	
Query syntax	:CALC:MARK1:NOIS?	
Default	OFF	
Return type	Numeric value (bool) or character	

:CALCulate[:SElected]:RELative[:MAGNitude]?

(Read only) query saved relative magnitude.

Applicable mode	Power meter
Parameter	None
Example	:CALC:REL?
Query syntax	:CALC:REL?
Default	None
Return type	Numeric value (float) or character

:CALCulate[:SElected]:RELative[:MAGNitude]:AUTO <bool>

(Read and write) set or query relative magnitude on/off.

Applicable mode	Power meter	
Parameter	Relative magnitude on/off.	
	OFF(0)	Relative magnitude off.
	ON(1)	Relative magnitude on.
Example	:CALC:REL:AUTO ON	
Query syntax	:CALC:REL:AUTO?	
Default	OFF	
Return type	Numeric value (bool) or character	

:CALCulate[:SElected]:PEAK:TRAC <bool>

(Read and write) set or query peak trace on/off.

Applicable mode	Spectrum Analysis	
Parameter	Peak trace on/off.	
	OFF(0)	Peak trace off.
	ON(1)	Peak trace on.
Example	:CALC:PEAK:TRAC ON	
Query syntax	:CALC:PEAK:TRAC?	
Default	OFF	
Return type	Numeric value (bool) or character	

:CALibration:ZERO

(Write only) Power measurement calibration **(do not repeatedly calibrate during calibration)**. This is an overlapping command. Use ***OPC?** before sending other commands to query if this command is completed.

Applicable mode	Power meter
Parameter	None
Example	:CAL:ZERO;*OPC?
Query syntax	None
Default	None
Return type	None

:CALibration:ZERO:STATe?

(Read only) query if Power measurement calibration is successful **(query is not available during calibration).**

Applicable mode	Power meter
Parameter	None
Example	:CAL:ZERO:STAT?
Query syntax	:CAL:ZERO:STAT?
Default	0
Return type	Numeric value (int) or character 0: no calibration 1: calibration succeed 2: calibration failed

:DISPlay:WINDow:ANALog:LOWer <num>

(Read and write) set or query minimum scale value.

Applicable mode	Power meter
Parameter	Minimum scale value (-70dBm~25dBm).
Example	:DISP:WIND:ANAL:LOW -60
Query syntax	:DISP:WIND:ANAL:LOW?
Default	-70dBm
Return type	Numeric value (float) or character

:DISPlay:WINDow:ANALog:UPPer <num>

(Read and write) set or query maximum scale value.

Applicable mode	Power meter
Parameter	Maximum scale value (-65dBm~30dBm)
Example	:DISP:WIND:ANAL:UPP 20
Query syntax	:DISP:WIND:ANAL:UPP?
Default	30 dBm
Return type	Numeric value (float) or character

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

(Write only) set as auto scale.

Applicable mode	Power meter
Parameter	None
Example	:DISP:WIND:TRAC:Y:AUTO
Query syntax	None
Default	None
Return type	None

:DISPlay:WINDow:TRACe:Y[:SCALe]:PDIVision <num>

(Read and write) query or set scale/division

Applicable mode	Spectrum Analysis (only available for logarithmic scale type), Interference Analysis, AM-FM-PM Demodulation, Channel Scanning
Parameter	Scale/division. Range: 0.1dB - 20dB

Example :DISP:WIND:TRAC:Y:PDIV 0.1
Query :DISP:WIND:TRAC:Y:PDIV?
syntax
Default 10dB
Return type Numeric value (float) or character

:DISPlay:WINDow:TRACe:Y[:SCALe]:RLEVel <num>

(Read and write) query or set reference level.

Applicable mode Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation, Channel Sweep
Parameter Reference level value (reference value). Reference level is related to current amplitude unit, and the setting scope corresponds to dBm. Conversion is required.
Range: -120 dBm to +40 dBm
Example :DISP:WIND:TRAC:Y:RLEV -10
Query :DISP:WIND:TRAC:Y:RLEV?
syntax
Default 0dBm
Return type Numeric value (float) or character

:DISPlay:WINDow:TRACe:Y[:SCALe]:RPOSition <num>

(Read and write) set or query reference position.

Applicable mode Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation
Parameter Reference position (no unit)
Scope -10 to 10.
Example :DISP:WIND:TRAC:Y:RPOS 1
Query syntax :DISP:WIND:TRAC:Y:RPOS?
Default 0
Return type Numeric value (int) or character

:DISPlay:TITLe <string>

(Write only) set the title.

Applicable mode All modes
Parameter Title, 10-digit letter or number at maximum.
Example :DISP:TITL SA_MEASURE
Query syntax None
Default None
Return type None

:DISPlay:TITLe:STATe <bool>

(Read and write) set or query title state.

Applicable mode All modes
Parameter Title state
OFF(0) Title off.
ON(1) Title on.
Example :DISP:TITL:STAT ON
Query syntax :DISP:TITL:STAT?
Default OFF
Return type Numeric value (bool) or character

:DISPlay:MODE <string>

(Read and write) set or query display mode.

Applicable mode	All modes	
Parameter	Display mode. DEFA(0) OUT(1) NIGHT(2)	Default mode Black and white mode Night vision mode
Example	:DISP:MODE DEFA	
Query syntax	:DISP:MODE?	
Default	DEFA	
Return type	Numeric value (int) or character	

:DISPlay:BRIG <int>

(Read and write) set or query brightness level.

Applicable mode	All modes
Parameter	Brightness level. Range: 0 - 4
Example	:DISP:BRIG 1
Query syntax	:DISP:BRIG?
Default	3
Return type	Numeric value (int) or character

:DISPlay:BRIG:AUTO <bool>

(Read and write) set or query brightness auto adjustment on/off.

Applicable mode	All modes	
Parameter	Brightness auto on/off. OFF(0) ON(1)	Brightness auto off. Brightness auto on.
Example	:DISP:BRIG:AUTO ON	
Query syntax	:DISP:BRIG:AUTO?	
Default	OFF	
Return type	Numeric value (bool) or character	

:DISPlay:TIME:FMT <string>

(Write only) set time format.

Applicable mode	All modes	
Parameter	Time format. YMD(0) MDY(1) DMY(2)	Year/month/date. Month/date/year. Date/month/year.
Example	:DISP:TIME:FMT YMD	
Query syntax	None	
Default	YMD	
Return type	Numeric value (int) or character	

:FORM[:DATA] <string>

(Read and write) set or query data format.

Applicable mode	All modes
Parameter	Data format. ASC(0) refers to character format. HEX(1) refers to numerical format. If character format is selected, the data format returned after query will be character format with character as the unit If numerical format is selected, the data format returned after query will

be numerical format with byte as the unit

Example	:FORM ASC
Query syntax	:FORM?
Default	ASC
Return type	Numeric value (bool) or character

:INITiate:CONTinuous <bool>

(Read and write) query or set sweep type. This is an overlapping command. Use ***OPC?** before sending other commands to query if this command is completed.

Applicable mode	Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation
Parameter	Sweep type. OFF or 0 refers to single sweep. ON or 1 refers to continuous sweep.
Example	:INIT:CONT OFF; *OPC?
Query syntax	:INIT:CONT?
Default	ON
Return type	Numeric value (bool) or character

:INITiate

(Write only) trigger one single sweep (only valid for single sweep). This command is an overlapping command. Use ***OPC?** before sending other commands to query if this command is completed.

Applicable mode	Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation
Parameter	None.
Example	:INIT;*OPC?
Query syntax	None
Default	None
Return type	None

:INSTrument:CATalog?

(Read only) query available instrument working mode. Use :INST:CAT?to query available instrument working mode.

Applicable mode	All modes
Parameter	None
Example	:INST:CAT?
Query syntax	:INST:CAT?
Default	0x01
Return type	Numeric value (int) or character The 0 bit is the bit for spectrum analysis test, 1 (mandatory) The 1 bit is the bit for AM-FM-PM Demodulation test, 1 is settable (optional) and 0 is not settable The 2 bit is the bit for interference analysis test, 1 is settable (optional) and 0 is not settable The 3 bit is the bit for Power measurement test, 1 is settable (optional) and 0 is not settable The 4 bit is the bit for channel sweep test, 1 is settable (optional) and 0 is not settable

:INSTrument[:SElect] <string>

(Read and write) query or set current instrument working mode. Use :INST:CAT? to query available instrument working modes. This command is an overlapping command. Use ***OPC?** before sending other commands to query if this command is completed.

Applicable mode	All modes	
Parameter	Instrument mode.	
	SA(1)	Spectrum analysis mode
	IA(2)	Interference analysis mode (option)
	DM(3)	AM-FM-PM Demodulation mode (option)
	PM(4)	Power measurement mode (option, and USB power probe connection required)
	CS(5)	Channel sweep mode (option)
Example	:INST SA;*OPC?	
Query syntax	:INST?	
Default	SA	
Return type	Numeric value (int) or character	

:MMEMory:DELeTe:ANTenna <string>

(Write only) delete antenna file under current mode **(this command will be invalid if the file does not exist and be only valid for current storage location).**

Applicable mode	Spectrum Analysis
Parameter	Name of antenna file.
Example	:MMEM:DEL:ANT set1
Query syntax	None
Default	None
Return type	None

:MMEMory:DELeTe:ANTenna:ALL

(Write only) delete all antenna files under current mode.

Applicable mode	Spectrum Analysis
Parameter	None
Example	:MMEM:DEL:ANT:ALL
Query syntax	None
Default	None
Return type	None

:MMEMory:DELeTe:LIMit <string>

(Write only) delete limit file under current mode **(this command will be invalid if the file does not exist and be only valid for current storage location).**

Applicable mode	Spectrum Analysis
Parameter	Name of limit file
Example	:MMEM:DEL:LIM set1
Query syntax	None
Default	None
Return type	None

:MMEMory:DELeTe:LIMit:ALL

(Write only) delete all limit files under current mode.

Applicable mode	Spectrum Analysis
Parameter	None
Example	:MMEM:DEL:LIM:ALL
Query syntax	None
Default	None
Return type	None

:MMEMory:DELeTe:LIST <string>

(Write only) delete list file under current mode **(this command will be invalid if the file does not exist and be only valid for current storage location).**

Applicable mode Spectrum Analysis
Parameter Name of list file.
Example :MMEM:DEL:LIST set1
Query syntax None
Default None
Return type None

:MMEMory:DELeTe:LIST:ALL

(Write only) delete all list files under current mode.

Applicable mode Spectrum Analysis
Parameter Name of list file.
Example :MMEM:DEL:LIST:ALL
Query syntax None
Default None
Return type None

:MMEMory:DELeTe:STATe <string>

(Write only) delete state file under current mode **(this command will be invalid if the file does not exist and be only valid for current storage location).**

Applicable mode All modes
Parameter Name of state file.
Example :MMEM:DEL:STAT set1
Query syntax None
Default None
Return type None

:MMEMory:DELeTe:STATe:ALL

(Write only) delete all state files under current mode.

Applicable mode All modes
Parameter None
Example :MMEM:DEL:STAT:ALL
Query syntax None
Default None
Return type None

:MMEMory:DELeTe:DATA <string>

(Write only) delete data file under current mode.

Applicable mode Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation, Channel Sweep
Parameter None
Example :MMEM:DEL:DATA set1
Query syntax None
Default None
Return type None

:MMEMory:DELeTe:DATA:ALL

(Write only) delete all data files under current mode.

Applicable Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation,

mode	Channel Sweep
Parameter	None
Example	:MMEM:DEL:DATA:ALL
Query syntax	None
Default	None
Return type	None

:MMEMory:LOAD:ANTenna <string>

(Write only) select antenna factor for field strength function **(this command will be invalid if the file does not exist and be only valid for current storage location).**

Applicable mode	Spectrum Analysis
Parameter	Name of antenna factor file
Example	:MMEM:LOAD:ANT 89101A
Query syntax	None
Default	Antenna factor not selected
Return type	None

:MMEMory:LOAD:LIMit <string>

(Write only) recall limit line **(this command will be invalid if the file does not exist and be only valid for current storage location).**

Applicable mode	Spectrum Analysis
Parameter	Name of limit line file
Example	:MMEM:LOAD:LIM set1
Query syntax	None
Default	None
Return type	None
Default	None
Return type	None

:MMEMory:LOAD:SEM <string>

(Write only) recall Spectrum Emission Mask file **(this command will be invalid if the file does not exist and be only valid for current storage location).**

Applicable mode	Spectrum Analysis
Parameter	Name of SEM file
Example	:MMEM:LOAD:SEM set1
Query syntax	None
Default	None
Return type	None
Default	None
Return type	None

:MMEMory:LOAD:LIST <string>

(Write only) recall list file **(this command will be invalid if the file does not exist and be only valid for current storage location).**

Applicable mode	Spectrum Analysis
Parameter	Name of list file.
Example	:MMEM:LOAD:LIST set1
Query syntax	None
Default	None
Return type	None

:MMEMory:LOAD:STATe <string>

(Write only) recall state file under current mode **(this command will be invalid if the file does not exist and be only valid for current storage location).**

Applicable mode	All modes
Parameter	Name of state file.
Example	:MMEM:LOAD:STAT set1
Query syntax	None
Default	None
Return type	None

:MMEMory:LOAD:DATA <string>

(Write only) recall data file under current mode **(this command will be invalid if the file does not exist and be only valid for current storage location).**

Applicable mode	All modes
Parameter	Name of data file.
Example	:MMEM:LOAD:DATA set1
Query syntax	None
Default	None
Return type	None

:MMEMory:LOCation <string>

(Read and write) query or set storage location.

Applicable mode	All modes
Parameter	Location. INT(0) Internal SD(1) SD card USB(2) USB
Example	:MMEM:LOC USB
Query syntax	:MMEM:LOC?
Default	INT
Return type	Numeric value (int) or character

:MMEMory:STORe:ANTenna <string>

(Write only) store antenna factor file **(this command will be invalid if the file does not exist and be only valid for current storage location).**

Applicable mode	Spectrum Analysis
Parameter	Name of antenna factor file.
Example	:MMEM:STOR:ANT set1
Query syntax	None
Default	None
Return type	None

:MMEMory:STORe:LIMit <string>

(Write only) store current limit line as file **(this command will be invalid if the file does not exist and be only valid for current storage location).**

Applicable mode	Spectrum Analysis
Parameter	Name of limit file
Example	:MMEM:STOR:LIM set1
Query syntax	None
Default	None
Return type	None

:MMEMory:STORe:LIST <string>

(Write only) store current list data in file **(this command will be invalid if the file does not exist and be only valid for current storage location).**

Applicable mode Spectrum Analysis
Parameter Name of list file.
Example :MMEM:STOR:LIST set1
Query syntax None
Default None
Return type None

:MMEMory:STORe:SCReen <string>

(Write only) screen copy, and save current screenshot as file **(this command will be invalid if the file does not exist and be only valid for current storage location).**

Applicable mode All modes
Parameter Name of screen copy file.
Example :MMEM:STOR:SCR pic1
Query syntax None
Default None
Return type None

:MMEMory:STORe:STATe <string>

(Write only) store state under current mode as a file **(this command will be invalid if the file does not exist and be only valid for current storage location).**

Applicable mode All modes
Parameter Name of state file.
Example :MMEM:STOR:STAT set1
Query syntax None
Default None
Return type None

:MMEMory:STORe:DATA <string>

(Write only) store data file under current mode **(this command will be invalid if the file does not exist and be only valid for current storage location).**

Applicable mode Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation, Channel Sweep
Parameter Name of data file.
Example :MMEM:STOR:DATA set1
Query syntax None
Default None
Return type None

[:SENSe]:ACPower:ADJChbw <num>

(Read and write) set or query ACPR adjacent channel bandwidth.

Applicable mode Spectrum Analysis
Parameter ACPR adjacent channel bandwidth (Hz).
 Scope: 300Hz - 20MHz
Example :ACP:ADJC 3000000
Query syntax :ACP:ADJC?
Default 3MHz
Return type Numeric value (int) or character

[\[:SENSe\]:ACPower:LIMit\[:STATe\] <bool>](#)

(Read and write) set or query ACPR door limit state.

Applicable mode	Spectrum Analysis
Parameter	Door limit test state.
	OFF(0) Door limit test off
	ON(1) Door limit test on
Example	:ACP:LIM OFF
Query syntax	:ACP:LIM?
Default	OFF
Return type	Numeric value (bool) or character

[\[:SENSe\]:ACPower:MAINchbw <num>](#)

(Read and write) set or query ACPR main channel bandwidth.

Applicable mode	Spectrum Analysis
Parameter	ACPR main channel bandwidth (Hz).
	Scope: 300Hz - 20MHz
Example	:ACP:MAIN 3000000
Query syntax	:ACP:MAIN?
Default	3MHz
Return type	Numeric value (int) or character

[\[:SENSe\]:ACPower:OFFSet:LLIMit <num>](#)

(Read and write) set or query ACPR lower adjacent channel limit.

Applicable mode	Spectrum Analysis
Parameter	ACPR lower adjacent channel limit (dB).
	Scope: -200dB to 200dB
Example	:ACP:OFFS:LLIM 0
Query syntax	:ACP:OFFS:LLIM?
Default	0
Return type	Numeric value (float) or character

[\[:SENSe\]:ACPower:OFFSet:ULIMit <num>](#)

(Read and write) set or query ACPR upper adjacent channel limit.

Applicable mode	Spectrum Analysis
Parameter	ACPR upper adjacent channel limit (dB).
	Scope: -200dB to 200dB
Example	:ACP:OFFS:ULIM 0
Query syntax	:ACP:OFFS:ULIM?
Default	0
Return type	Numeric value (float) or character

[\[:SENSe\]:ACPower:SPACe <num>](#)

(Read and write) set or query ACPR channel space.

Applicable mode	Spectrum Analysis
Parameter	ACPR channel space (Hz).
	Scope: 0Hz - 45MHz
Example	:ACP:SPAC 3000000
Query syntax	:ACP:SPAC?
Default	3MHz
Return type	Numeric value (int) or character

[[:SENSe]:ACPower[:STATe] <bool>

(Read and write) set or query ACPR state **(Other functional measurements will be disabled after this function is enabled)**, or command [[:SENSe]:MEASurement can be used.

Applicable mode	Spectrum Analysis
Parameter	ACPR state.
	OFF(0) ACPR off
	ON(1) ACPR on
Example	:ACP ON
Query syntax	:ACP?
Default	OFF
Return type	Numeric value (bool) or character

[[:SENSe]:ACPower:UPPer?

(Read only) query ACPR upper adjacent channel power ratio.

Applicable mode	Spectrum Analysis
Parameter	None
Query syntax	:ACP:UPP?
Default	0
Return type	Numeric value (float) or character

[[:SENSe]:ACPower:LOWer?

(Read only) query ACPR lower adjacent channel power ratio.

Applicable mode	Spectrum Analysis
Parameter	None
Query syntax	:ACP:LOW?
Default	0
Return type	Numeric value (float) or character

[[:SENSe]:AFPanalyzer:DEMod:TYPE <string>

(Read and write) set or query demodulation type of AM-FM-PM Demodulation.

Applicable mode	AM-FM-PM Demodulation
Parameter	Demod type.
	AM(0)
	FM(1)
	PM(2)
Example	:AFP:DEM:TYPE AM
Query syntax	:AFP:DEM:TYPE?
Default	AM
Return type	Numeric value (int) or character

[[:SENSe]:AFPanalyzer:DEMod:MODE <string>

(Read and write) set or query mode of AM-FM-PM Demodulation.

Applicable mode	AM-FM-PM Demodulation
Parameter	Demodulation mode.
	RF(0) RF spectrum
	AF(1) audio spectrum
	AW(2) audio waveform
	ALL(3) all spectrums
Example	AFP:DEM:MODE RF
Query syntax	:AFP:DEM:MODE?
Default	ALL
Return type	Numeric value (int) or character

[\[:SENSe\]:AFPanalyzer:SPAN <num>](#)

(Read and write) set or query demodulation audio spectrum span of AM-FM-PM Demodulation.

Applicable mode	AM-FM-PM Demodulation
Parameter	Audio spectrum span.
Example	:AFP:SPAN 10000
Query syntax	:AFP:SPAN?
Default	100000
Return type	Numeric value (double) or character

[\[:SENSe\]:AFPanalyzer:SCALe <num>](#)

(Read and write) set or query audio spectrum scale/division of AM-FM-PM Demodulation.

Applicable mode	AM-FM-PM Demodulation
Parameter	Audio spectrum scale/division
Example	:AFP:SCAL 10
Query syntax	:AFP:SCAL?
Default	10
Return type	Numeric value (double) or character

[\[:SENSe\]:AFPanalyzer:SWEp:TIME <num>](#)

(Read and write) set or query audio wave sweep time of AM-FM-PM Demodulation.

Applicable mode	AM-FM-PM Demodulation
Parameter	Audio waveform sweep time (us).
Example	:AFP:SWE:TIME 1000
Query syntax	:AFP:SWE:TIME?
Default	None
Return type	Numeric value (double) or character

[\[:SENSe\]:AFPanalyzer:IFBW <num>](#)

(Read and write) set or query IF bandwidth of AM-FM-PM Demodulation.

Applicable mode	AM-FM-PM Demodulation
Parameter	intermediate-frequency bandwidth. Range: 10kHz - 300kHz
Example	:AFP:IFBW 100000
Query syntax	:AFP:IFBW?
Default	300000
Return type	Numeric value (double) or character

[\[:SENSe\]:AFPanalyzer:TRACe <string>](#)

(Write only) set the trace of marker selection.

Applicable mode	AM-FM-PM Demodulation
Parameter	Type of marker selection trace. RF(0) RF spectrum AF(1) Audio spectrum AW(2) Audio waveform
Example	:AFP:TRAC RF
Query syntax	None
Default	RF
Return type	Numeric value (int) or character

[:SENSe]:AMPLitude:ALIGNment:NOW

(Read only) zero frequency calibration **(do not repeatedly carry out zero frequency calibration during zero frequency calibration)**. This command is an overlapping command. Use ***OPC?** before sending other commands to query if this command is completed.

Applicable mode	Spectrum Analysis
Parameter	None
Example	:AMPL:ALIG:NOW;*OPC?
Query syntax	None
Default	None
Return type	None

[:SENSe]:AMPLitude:CORRections:ANTenna:OFF

(Write only) set antenna factor loading off and set antenna factor free state.

Applicable mode	Spectrum Analysis, Field Strength
Parameter	None
Example	:AMPL:CORR:ANT:OFF
Query syntax	None
Default	OFF
Return type	None

[:SENSe]:AMPLitude:CORRections[:STATe] <bool>

(Read and write) set or query field strength function measurement state **(Other functional measurements will be disabled after this function is enabled)**, or command [:SENSe]:MEASurement can be used.

Applicable mode	Spectrum Analysis
Parameter	field strength state. OFF(0) field strength off ON(1) field strength on
Example	:AMPL:CORR:STAT ON
Query syntax	:AMPL:CORR?
Default	OFF
Return type	Numeric value (bool) or character

[:SENSe]:AMPLitude:SCALe<string>

(Read and write) query or set scale type.

Applicable mode	Spectrum Analysis
Parameter	Scale Type. LOG(0) Logarithm LIN(1) Linear
Example	:AMPL:SCAL LOG
Query syntax	:AMPL:SCAL?
Default	LOG
Return type	Numeric value (int) or character

[:SENSe]:AMPLitude:UNIT <string>

(Read and write) query or set amplitude unit.

Applicable mode	Spectrum Analysis, Field Strength
Parameter	Amplitude unit.

Spectrum Analysis	DBM(0)	In dBm.
	DBMV(1)	Unit: dBmV
	DBUV(2)	Unit: dBuV
	V(3)	Unit: Volts
	W(4)	Unit: Walts

Example :AMPL:UNIT DBM
Query syntax :AMPL:UNIT?
Default BM
Return type Numeric value (int) or character

[\[:SENSe\]:AMPLitude:CORRections:ANTenna:EDIT:ADD](#)

(Write only) edit antenna factor to add default point

Applicable mode Spectrum Analysis, Field Strength
Parameter None
Example :AMPL:CORR:ANT:EDIT:ADD
Query syntax None
Default None
Return type None

[\[:SENSe\]:AMPLitude:CORRections:ANTenna:EDIT:ADD:DATA](#)

(Write only) edit antenna factor to add point

Applicable mode Spectrum Analysis, Field Strength
Parameter Frequency (0 - 44.1GHz), antenna factor value (-200dB to 200dB)
Example :AMPL:CORR:ANT:EDIT:ADD:DATA 1000000000,5.0
Query syntax None
Default None
Return type None

[\[:SENSe\]:AMPLitude:CORRections:ANTenna:EDIT:DEL <int>](#)

(Write only) edit antenna factor to delete point

Applicable mode Spectrum Analysis, Field Strength
Parameter Point index (start from 0)
Example :AMPL:CORR:ANT:EDIT:DEL 1
Query syntax None
Default None
Return type None

[\[:SENSe\]:AMPLitude:CORRections:ANTenna:EDIT:DATA](#)

(Write only) edit antenna factor to edit point

Applicable mode Spectrum Analysis, Field Strength
Parameter Point index (start from 0), frequency value (0 - 44.1GHz), antenna factor value (-200dB to 200dB)
Example :AMPL:CORR:ANT:EDIT:DATA 1,1000000000,5.0
Query syntax None
Default None
Return type None

[\[:SENSe\]:AVERage:COUNt <num>](#)

(Read and write) query or set average count.

Applicable mode	Spectrum analysis, AM-FM-PM Demodulation, interference analysis and power meter
Parameter	Average. Scope: 1 - 1000.
Example	:AVER:COUN 16
Query syntax	:AVER:COUN?
Default	16
Return type	Numeric value (int) or character

[\[:SENSe\]:AVERage:CLEar](#)

(Write only) start current average count from 0.

Applicable mode	Spectrum analysis, AM-FM-PM Demodulation, interference analysis and power meter
Parameter	None.
Example	:AVER:CLE
Query syntax	None
Return type	None

[\[:SENSe\]:AVERage:STATe <bool>](#)

(Read and write) query or set average state.

Applicable mode	Spectrum analysis, AM-FM-PM Demodulation, interference analysis and power meter
Parameter	Average ON/OFF. OFF or 0 refers to off. ON or 1 refers to on.
Example	:AVER:STAT OFF
Query syntax	:AVER:STAT?
Default	Off
Return type	Numeric value (bool) or character

[\[:SENSe\]:AVERage:CURC?](#)

(Read only) query current average count.

Applicable mode	Spectrum analysis, AM-FM-PM Demodulation, interference analysis and power meter
Parameter	None.
Example	:AVER:CURC?
Query syntax	:AVER:CURC?
Return type	None

[\[:SENSe\]:BANDwidth\[:RESolution\] <num>](#)

(Read and write) query or set resolution bandwidth.

Applicable mode	Spectrum Analysis, Interference Analysis
Parameter	Resolution bandwidth (Hz). Scope: 1Hz~10MHz
Example	:BAND 300000
Query syntax	:BAND?
Default	3MHz
Return type	Numeric value (double) or character

[\[:SENSe\]:BANDwidth\[:RESolution\]:AUTO <bool>](#)

(Read and write) query or set resolution bandwidth auto on/off.

Applicable mode	Spectrum Analysis, Interference Analysis
Parameter	Resolution bandwidth auto on/off. When set as auto, the resolution bandwidth will OFF or 0 refers to manual. ON or 1 refers to auto.
Example	:BAND:AUTO ON
Query syntax	:BAND:AUTO?
Default	ON
Return type	Numeric value (bool) or character

[\[:SENSe\]:BANDwidth\[:RESolution\]:RATio <num>](#)

(Read and write) query or set span/resolution bandwidth ratio.

Applicable mode	Spectrum Analysis, Interference Analysis
Parameter	SPAN/RBW value. Scope: 1 - 500.
Example	:BAND:RAT 100
Query syntax	:BAND:RAT?
Default	100
Return type	Numeric value (int) or character

[\[:SENSe\]:BANDwidth:VIDeo <num>](#)

(Read and write) query or set video bandwidth.

Applicable mode	Spectrum Analysis, Interference Analysis
Parameter	Video bandwidth value (Hz). Scope: 1Hz - 10MHz
Example	:BAND:VID 300000
Query syntax	:BAND:VID?
Default	3MHz
Return type	Numeric value (int) or character

[\[:SENSe\]:BANDwidth:VIDeo:AUTO <bool>](#)

(Read and write) query or set video bandwidth auto on/off.

Applicable mode	Spectrum Analysis, Interference Analysis
Parameter	Video bandwidth auto on/off. When set as auto, the video bandwidth will be adaptable to the resolution bandwidth based on the ratio of SPAN/VBW. OFF or 0 refers to manual. ON or 1 refers to auto.
Example	:BAND:VID:AUTO ON
Query syntax	:BAND:VID:AUTO?
Default	ON
Return type	Numeric value (bool) or character

[\[:SENSe\]:BANDwidth:VIDeo:RATio <num>](#)

(Read and write) set or query resolution bandwidth/video bandwidth ratio.

Applicable mode	Spectrum Analysis, Interference Analysis
Parameter	RBW/VBW value. Scope: 1 - 100.
Example	:BAND:VID:RAT 1
Query syntax	:BAND:VID:RAT?
Default	3
Return type	Numeric value (int) or character

[[:SENSe]:BANDwidth:VIDeo:TYPE <BOOL>

(Read and write) set or query video type.

Applicable mode Spectrum Analysis, Interference Analysis
Parameter Video type LIN(0), line, LOG(1), logarithm
Example : BAND:VID:TYPE LOG
Query syntax : BAND:VID:TYPE?
Default LIN
Return type Numeric value (int) or character

[[:SENSe]:CMEasurement:IBW <num>

(Read and write) set or query channel power bandwidth.

Applicable mode Spectrum Analysis
Parameter Channel power bandwidth (Hz), scope: 100Hz~44.1GHz.
Example :CME:IBW 1000000
Query syntax :CME:IBW?
Default 2MHz
Return type Numeric value (double) or character

[[:SENSe]:CMEasurement:PSDR?

(Read only) query channel power density under channel power function measurement **(valid when the channel power is on and after one sweep).**

Applicable mode Spectrum Analysis
Parameter None
Example :CME:PSDR?
Query syntax :CME:PSDR?
Default None
Return type Numeric value (float) or character

[[:SENSe]:CMEasurement[:STATe] <bool>

(Read and write) set or query channel power function measurement state **(Other functional measurements will be disabled after this function is enabled)**, or use command

[[:SENSe]:MEASurement.

Applicable mode Spectrum Analysis
Parameter Channel power state.
OFF(0) Channel power off
ON(1) Channel power on
Example :CME ON
Query syntax :CME?
Default OFF
Return type Numeric value (bool) or character

[[:SENSe]:CMEasurement:TPWR?

(Read only) query channel power value under channel power function measurement **(valid when the channel power is on and after one sweep).**

Applicable mode Spectrum Analysis
Parameter None
Example :CME:TPWR?
Query syntax :CME:TPWR?
Default None
Return type Numeric value (float) or character

[[:SENSe]:CORRection:GAIN <num>

(Read and write) set or query offset value.

Applicable mode Power meter
Parameter Offset value (-50dB to 30dB).
Example :CORR:GAIN -5
Query syntax :CORR:GAIN?
Default 0dB
Return type Numeric value (float) or character

[[:SENSe]:CORRection:GAIN:STATe <bool>

(Read and write) set or query offset state.

Applicable mode Power meter
Parameter Offset state.
OFF(0) Offset off.
ON(1) Offset on.
Example :CORR:GAIN:STAT ON
Query syntax :CORR:GAIN:STAT?
Default OFF
Return type Numeric value (bool) or character

[[:SENSe]:CNRatio[:STATe] <bool>

(Read and write) set or query CNR ratio measurement state **(Other functional measurements will be disabled after this function is enabled)**, or use command
[:SENSe]:MEASurement.

Applicable mode Spectrum Analysis
Parameter CNR measurement state.
OFF(0) Off.
ON(1) On.
Example :CNR ON
Query syntax :CNR?
Default OFF
Return type Numeric value (bool) or character

[[:SENSe]:CNRatio:CBW <num>

(Read and write) set or query CNR carrier bandwidth.

Applicable mode Spectrum Analysis
Parameter CNR carrier bandwidth.
Range: 300Hz - 20MHz
Example :CNR:CBW 1000000
Query syntax :CNR:CBW?
Default 3000000
Return type Numeric value (double) or character

[[:SENSe]:CNRatio:NBW <num>

(Read and write) set or query CNR noise bandwidth.

Applicable mode Spectrum Analysis
Parameter CNR noise bandwidth.
Range: 300Hz - 20MHz
Example :CNR:NBW 1000000
Query syntax :CNR:NBW?
Default 3000000
Return type Numeric value (double) or character

[[:SENSe]:CNRatio:CNSpace <num>

(Read and write) set or query CNR frequency offset.

Applicable mode Spectrum Analysis
Parameter CNR frequency offset.
Range: 0Hz - 100MHz
Example :CNR:CNSP 1000000
Query syntax :CNR:CNSP?
Default 3000000
Return type Numeric value (double) or character

[[:SENSe]:CNRatio:CNRatio?

(Read only) query CNR under CNR measurement **(valid when CNR is open and after one sweep).**

Applicable mode Spectrum Analysis
Parameter None
Example :CNR:CNR?
Query syntax :CNR:CNR?
Default None
Return type Numeric value (float) or character

[[:SENSe]:CS:DISPlay <string>

(Read and write) set or query channel sweep graph/table display.

Applicable mode Channel sweep
Parameter Channel sweep graph/table display.
GRAPH(0) graph display
TABLE(1) table display
Example :CS:DISP TABLE
Query syntax :CS:DISP?
Default GRAPH
Return type Numeric value (int) or character

[[:SENSe]:CS:MAXHold <bool>

(Read and write) set or query channel sweep maximum hold state.

Applicable mode Channel sweep
Parameter maximum hold state.
OFF(0) maximum hold off
ON(1) maximum hold on
Example :CS:MAXH ON
Query syntax :CS:MAXH?
Default OFF
Return type Numeric value (bool) or character

[[:SENSe]:CS:UNIT <string>

(Read and write) set or query channel sweep unit.

Applicable mode Channel sweep
Parameter channel sweep unit.
CHAN(0) channel
FREQ(1) frequency
Example :CS:UNIT CHAN
Query syntax :CS:UNIT?
Default CHAN
Return type Numeric value (int) or character

[\[:SENSe\]:CS:PWR <string>](#)

(Read and write) set or query channel sweep power display mode.

Applicable mode	Channel sweep
Parameter	channel sweep power display mode. NOW(0) real-time MAX(1) maximum
Example	:CS:PWR NOW
Query syntax	:CS:PWR?
Default	NOW
Return type	Numeric value (int) or character

[\[:SENSe\]:CS:COLOur <string>](#)

(Read and write) set or query channel sweep graph color code display mode.

Applicable mode	Channel sweep
Parameter	Color display mode. SING(0) single color DUAL(1) dual color
Example	:CS:COLO SING
Query syntax	:CS:COLO?
Default	SING
Return type	Numeric value (int) or character

[\[:SENSe\]:CS:ORIEntal <string>](#)

(Read and write) set or query vertical or horizontal display of channel sweep graph.

Applicable mode	Channel sweep
Parameter	Vertical or horizontal display of graph. VERT(0) vertical display HORI(1) horizontal display
Example	:CS:ORIE VERT
Query syntax	:CS:ORIE?
Default	VERT
Return type	Numeric value (int) or character

[\[:SENSe\]:CS:MODE <string>](#)

(Read and write) set or query scan mode of channel sweep.

Applicable mode	Channel sweep
Parameter	Sweep mode. CHAN(0) channel sweep FREQ(1) frequency sweep LIST(2) list sweep
Example	:CS:MODE CHAN
Query syntax	:CS:MODE?
Default	CHAN
Return type	Numeric value (int) or character

[\[:SENSe\]:CS:CHANnel:NUMber <num>](#)

(Read and write) set or query number of channels swept by channel sweep

Applicable mode	Channel sweep
Parameter	Channel number (1 - 20).
Example	:CS:CHAN:NUM 10
Query syntax	:CS:CHAN:NUM?
Default	10

Return type Numeric value (int) or character

[\[:SENSe\]:CS:CHANnel:STEP <num>](#)

(Read and write) set or query channel steps swept by channel sweep

Applicable mode Channel sweep
Parameter Channel step (1 - 25).
Example :CS:CHAN:STEP 10
Query syntax :CS:CHAN:STEP?
Default 1
Return type Numeric value (int) or character

[\[:SENSe\]:CS:CHANnel:BANDwidth <num>](#)

(Read and write) set or query channel bandwidth swept by channel sweep

Applicable mode Channel sweep
Parameter channel bandwidth (Hz).
Range: 1kHz - 20MHz
Example :CS:CHAN:BAND 100000
Query syntax :CS:CHAN:BAND?
Default 200000
Return type Numeric value (double) or character

[\[:SENSe\]:CS:FREQuency:STARt <num>](#)

(Read and write) set or query start frequency of frequency sweep.

Applicable mode Channel sweep
Parameter start frequency (Hz)
Range: 0Hz - 44.1GHz
Example :CS:FREQ:STAR 1000000
Query syntax :CS:FREQ:STAR?
Default 890000000
Return type Numeric value (double) or character

[\[:SENSe\]:CS:FREQuency:STEP <num>](#)

(Read and write) set or query step frequency of frequency sweep in channel sweep.

Applicable mode Channel sweep
Parameter step frequency (Hz)
Range: 1Hz - 5GHz
Example :CS:FREQ:STEP 1000000
Query syntax S:FREQ:STEP?
Default 200000
Return type Numeric value (double) or character

[\[:SENSe\]:CS:FREQuency:BANDwidth <num>](#)

(Read and write) set or query channel bandwidth of frequency sweep in channel sweep.

Applicable mode Channel sweep
Parameter channel bandwidth (Hz).
Range: 1kHz - 20MHz
Example :CS:FREQ:BAND 1000000
Query syntax :CS:FREQ:BAND?
Default 200000
Return type Numeric value (double) or character

[[:SENSe]:CS:FREQuency:NUMber <num>

(Read and write) set or query number of channels swept by frequency sweep in channel sweep.

Applicable mode Channel sweep
Parameter Channel number (1 - 20).
Example :CS:FREQ:NUM 10
Query syntax :CS:FREQ:NUM?
Default 10
Return type Numeric value (int) or character

[[:SENSe]:CS:LIST:NUMber <num>

(Read and write) set or query number of channels swept by list sweep in channel sweep

Applicable mode Channel sweep
Parameter Channel number (1 - 20).
Example :CS:LIST:NUM 10
Query syntax :CS:LIST:NUM?
Default 10
Return type Numeric value (int) or character

[[:SENSe]:DETector:FUNCTion <string>

(Read and write) set or query detector type.

Applicable mode Spectrum Analysis, Interference Analysis, Field Strength
Parameter Detector type

Spectrum Analysis, Interference Analysis		Field Strength	
POSitive (0)	Positive peak	AVERage (0)	Average
NEGative (1)	Negative peak		
SAMPlE (2)	Sample	POSitive (1)	Peak
NORMal (3)	Standard (rosenfeld)		
AVERage (4)	Average	SAMPlE (2)	Real time
RMS (5)	Root mean square		

Example :DET:FUNC NORM
Query syntax :DET:FUNC?
Default NORM
Return type Numeric value (int) or character

[[:SENSe]:DETector:FUNCTion:AUTO <bool>

(Read and write) set or query detector auto on/off.

Applicable mode Spectrum Analysis, Interference Analysis
Parameter Detector auto on/off. When the detector is set as auto, the instrument will automatically select detector type based on different measurement.
OFF(0) refers to manual.
ON(1) refers to auto.
Example :DET:FUNC:AUTO OFF
Query syntax :DET:FUNC:AUTO?

Default ON
Return type Numeric value (bool) or character

[[:SENSe]:EMISSION[:STATe] <bool>

(Read and write) set or query SEM state **(Other functional measurements will be disabled after this function is enabled)**, or use command [[:SENSe]:MEASurement.

Applicable mode Spectrum Analysis
Parameter SEM measurement state
OFF(0), off.
ON(1), on.
Example :EMIS OFF
Query syntax :EMIS?
Default OFF
Return type Numeric value (bool) or character

[[:SENSe]:EMISSION:CBW <num>

(Read and write) set or query SEM reference channel bandwidth.

Applicable mode Spectrum Analysis
Parameter reference channel bandwidth (Hz).
Range: 1kHz - 44.1GHz
Example :EMIS:CBW 1000000
Query syntax :EMIS:CBW?
Default 1000000
Return type Numeric value (double) or character

[[:SENSe]:EMISSION:RTYPE <string>

(Read and write) set or query SEM reference power type.

Applicable mode Spectrum Analysis
Parameter reference power type.
PEAK(0) peak
CHANnel(1) channel
Example :EMIS:RTYP PEAK
Query syntax :EMIS:RTYP?
Default PEAK
Return type Numeric value (int) or character

[[:SENSe]:EMISSION:MARKer <bool>

(Read and write) set or query SEM peak marker state.

Applicable mode Spectrum Analysis
Parameter Peak marker on/off.
OFF(0) off
ON(1) on
Example :EMIS:MARK ON
Query syntax :EMIS:MARK?
Default OFF
Return type Numeric value (bool) or character

[[:SENSe]:EMISSION:Fail?

(Read only) query if SEM test fails. The return value 1 means failed and 0 succeed.

Applicable mode Spectrum Analysis
Parameter None

Example :EMIS:Fail?
Query syntax :EMIS:Fail?
Default None
Return type Numeric value (int) or character

[\[:SENSe\]:FREQuency:CENTer <num>](#)

(Read and write) set or query center frequency.

Applicable mode Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation,
Power Meter
Parameter Center frequency value (Hz).
Frequency range of spectrum analysis, 0Hz - 44.1GHz.
Frequency range of interference analysis, 0Hz - 44.1GHz.
Frequency range of AM-FM-PM analysis, 500Hz - 44.1GHz.
Frequency range of power meter, 10MHz - 40GHz.
Example :FREQ:CENT 10000
Query syntax :FREQ:CENT?
Default 22.05GHz (Spectrum Analysis, Interference Analysis, AM-FM-PM
Demodulation)
10MHz (power meter)
Return type Numeric value (double) or character

[\[:SENSe\]:FREQuency:CENTer:STEP <num>](#)

(Read and write) set or query step frequency.

Applicable mode Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation
Parameter Step frequency (Hz).
Scope of step frequency, 1Hz - 5GHz.
Example :FREQ:CENT:STEP 10000
Query syntax :FREQ:CENT:STEP?
Default 1MHz
Return type Numeric value (double) or character

[\[:SENSe\]:FREQuency:CENTer:STEP:AUTO <bool>](#)

(Read and write) set or query step frequency state.

Applicable mode Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation
Parameter Step frequency auto state.
OFF(0) refers to auto off.
ON(1) refers to auto on.
Example :FREQ:CENT:STEP:AUTO OFF
Query syntax :FREQ:CENT:STEP:AUTO?
Default ON
Return type Numeric value (bool) or character

[\[:SENSe\]:FREQuency:SPAN <num>](#)

(Read and write) set or query span.

Applicable mode Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation
Parameter Span value (Hz).
Frequency range of spectrum analysis span, 0Hz - 44.1GHz.
Frequency range of interference analysis span, 0Hz - 44.1GHz.
Frequency range of AM-FM-PM analysis span, 0Hz - 44.1GHz.
Example :FREQ:SPAN 10000
Query syntax :FREQ:SPAN?
Default 44.1GHz (Spectrum Analysis, Interference Analysis)
3MHz (AM-FM-PM Demodulation)

Return type Numeric value (double) or character

[\[:SENSe\]:FREQuency:SPAN:FULL](#)

(Write only) set as full span.

Applicable mode Spectrum Analysis, Interference Analysis

Parameter None

Example :FREQ:SPAN:FULL

Query syntax None

Return type None

[\[:SENSe\]:FREQuency:SPAN:PREVious](#)

(Write only) set as previous span.

Applicable mode Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation

Parameter None

Example :FREQ:SPAN:PREV

Query syntax None

Return type None

[\[:SENSe\]:FREQuency:SPAN:ZERO](#)

(Write only) set as zero span.

Applicable mode Spectrum Analysis, Interference Analysis

Parameter None

Example :FREQ:SPAN:ZERO

Query syntax None

Return type None

[\[:SENSe\]:FREQuency:STARt <num>](#)

(Read and write) query or set start frequency.

Applicable mode Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation

Parameter Start frequency value (Hz).

Start frequency range of spectrum analysis, 0Hz - 44.1GHz.

Start frequency range of interference analysis, 0Hz - 44.1GHz.

Example :FREQ:STAR 10000

Query syntax :FREQ:STAR?

Default 0Hz (spectrum analysis, interference analysis)

Return type Numeric value (double) or character

[\[:SENSe\]:FREQuency:STOP <num>](#)

(Read and write) query or set stop frequency.

Applicable mode Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation

Parameter Stop frequency value (Hz).

Stop frequency range of spectrum analysis, 0Hz - 44.1GHz.

Stop frequency range of interference analysis, 0Hz - 44.1GHz.

Example :FREQ:STOP 10000

Query syntax :FREQ:STOP?

Default 44.1GHz (Spectrum Analysis, Interference Analysis)

Return type Numeric value (double) or character

[\[:SENSe\]:FREQuency:SIGStandard:NAME <string>](#)

(Read and write) query or set signal standard.

Applicable Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation,

mode	Channel Sweep
Parameter	Name of signal standard.
Example	:FREQ:SIGS:NAME P-GSM UL
Query syntax	:FREQ:SIGS:NAME?
Default	None
Return type	Character string

[\[:SENSe\]:FREQuency:SIGStandard:CHANnel <num>](#)

(Read and write) query or set channel number.

Applicable mode	Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation,
mode	Channel Sweep
Parameter	Channel number.
Example	:FREQ:SIGS:CHAN 1
Query syntax	:FREQ:SIGS:CHAN?
Default	Initial channel of signal standard
Return type	Numeric value (int) or character

[\[:SENSe\]:FREQuency:SIGnal:SEARch](#)

(Write only) signal search

Applicable mode	Spectrum Analysis
Parameter	None
Example	:FREQ:SIG:SEAR
Query syntax	:None
Default	None
Return type	None

[\[:SENSe\]:FREQuency:SIGnal:TRAC <BOOL>](#)

(Read and write) Signal track on/off

Applicable mode	Spectrum Analysis
Parameter	ON(1) OFF(0)
Example	: FREQ:SIG:TRAC ON
Query syntax	: FREQ:SIG:TRAC?
Default	OFF
Return type	Numeric value (int) or character

[\[:SENSe\]:FREQuency:RESolution <num>](#)

(Read and write) query or set Power measurement frequency resolution.

Applicable mode	Power meter
Parameter	Resolution (0 - 3).
Example	:FREQ:RES 1
Query syntax	:FREQ:RES?
Default	2
Return type	Numeric value (int) or character

[\[:SENSe\]:IAMeasure:MODE <string>](#)

(Read and write) query or set interference analysis measurement mode.

Applicable mode	Interference Analysis
Parameter	NORMAL(0) spectrum measurement
	G(1) waterfall graph
	RSSI(2) RSSI

Example :IAM:MODE SG
Query syntax :IAM:MODE?
Default SG
Return type Numeric value (int) or character

[\[:SENSe\]:IAMeasure:TRACe:SPAN <num>](#)

(Read and write) query or set interference analysis span time.

Applicable mode Interference Analysis
Parameter Span time (min)
 Range: 0 - 1440 min
Example :IAM:TRAC:SPAN 1
Query syntax :IAM:TRAC:SPAN?
Default 0
Return type Numeric value (int) or character

[\[:SENSe\]:IAMeasure:TRACe:SAVE <bool>](#)

(Read and write) query or set interference analysis auto save on/off.

Applicable mode Interference Analysis
Parameter Auto save on/off
 OFF(0) off
 ON(1) on
Example :IAM:TRAC:SAVE ON
Query syntax :IAM:TRAC:SAVE?
Default OFF
Return type Numeric value (bool) or character

[\[:SENSe\]:IAMeasure:TRACe:CURS <num>](#)

(Write only) set time cursor.

Applicable mode Interference Analysis
Parameter Time cursor (int)
 Range: 0 - 288
Example :IAM:TRAC:CURS 1
Query syntax None
Default 0
Return type None

[\[:SENSe\]:IAMeasure:TRACe:REStart](#)

(Write only) set restart measurement.

Applicable mode Interference Analysis
Parameter None
Example :IAM:TRAC:REST
Query syntax None
Default None
Return type None

[\[:SENSe\]:IAMeasure:TRACe:INTERval <num>](#)

(Read and write) query or set sweep interval of interference analysis.

Applicable mode Interference Analysis
Parameter Sweep interval (ms)
Example :IAM:TRAC:INTE 1000
Query syntax :IAM:TRAC:INTE?
Default 0

Return type Numeric value (double) or character

[\[:SENSe\]:IF:OUT <bool>](#)

(Read and write) query or set IF output state.

Applicable mode Spectrum Analysis

Parameter IF output state

OFF(0) off

ON(1) on

Example :IF:OUT ON

Query syntax :IF:OUT?

Default OFF

Return type Numeric value (bool) or character

[\[:SENSe\]:IF:SELEct <string>](#)

(Read and write) query or set IF selection.

Applicable mode Spectrum Analysis

Parameter 3IF(0) 3IF output

4IF(1) 4IF output

Example :IF:SEL 3IF

Query syntax :IF:SEL?

Default 3IF

Return type Numeric value (int) or character

[\[:SENSe\]:IQ:CAPture\[:STATe\] <bool>](#)

(Read and write) query or set IQ capture state **(Other functional measurements will be disabled after this function is enabled)**, or use command [:SENSe]:MEASurement.

Applicable mode Spectrum Analysis

Parameter IQ capture measurement state

OFF(0), off.

ON(1), on.

Example :IQ:CAP OFF

Query syntax :IQ:CAP?

Default OFF

Return type Numeric value (bool) or character

[\[:SENSe\]:IQ:CAPture:STARt](#)

(Write only) start IQ capture.

Applicable mode Spectrum Analysis

Parameter None

Example :IQ:CAP:STAR

Query syntax None

Default None

Return type None

[\[:SENSe\]:IQ:CAPture:STOP](#)

(Write only) stop IQ capture.

Applicable mode Spectrum Analysis

Parameter None

Example :IQ:CAP:STOP

Query syntax None

Default None

Return type None

[[:SENSe]:IQ:CAPture:TIME <num>

(Read and write) query or set IQ capture time.

Applicable mode	Spectrum Analysis
Parameter	IQ capture time (us)
Example	:IQ:CAPture:TIME 10
Query syntax	:IQ:CAPture:TIME?
Default	40us
Return type	Numeric value (double) or character

[[:SENSe]:IQ:CAPture:MODE <num>

(Read and write) query or set IQ capture mode.

Applicable mode	Spectrum Analysis
Parameter	IQ capture mode SING(0) single capture CONT(1) continuous capture
Example	:IQ:CAPture:MODE SING
Query syntax	:IQ:CAPture:MODE?
Default	SING
Return type	Numeric value (int) or character

[[:SENSe]:IQ:CAPture:SAMPle <num>

(Read and write) query or set IQ capture sample rate.

Applicable mode	Spectrum Analysis
Parameter	IQ capture sample rate (Hz) Scope:
Example	:IQ:CAP:SAMP 500000
Query syntax	:IQ:CAP:SAMP?
Default	5MHz
Return type	Numeric value (double) or character

[[:SENSe]:IQ:CAPture:NAME <string>

(Read and write) query or set IQ capture save name.

Applicable mode	Spectrum Analysis
Parameter	IQ capture save name
Example	:IQ:CAP:NAME ABCD
Query syntax	:IQ:CAP:NAME?
Default	IQCapture
Return type	Character string

[[:SENSe]:IQ:CAPture:TRIG <string>

(Read and write) query or set IQ capture trigger type.

Applicable mode	Spectrum Analysis
Parameter	Trigger Type FREE(0) free trigger EXTR(1) external trigger
Example	:IQ:CAP:TRIG FREE
Query syntax	:IQ:CAP:TRIG?
Default	FREE
Return type	Numeric value (int) or character

[\[:SENSe\]:IQ:CAPture:TRIG:SLOPe <string>](#)

(Read and write) query or set IQ capture trigger polarity.

Applicable mode Spectrum Analysis
Parameter Trigger Polarity
 POS(0) positive
 NEG(1) negative
Example :IQ:CAP:TRIG:SLOP POS
Query syntax :IQ:CAP:TRIG:SLOP?
Default NEG
Return type Numeric value (int) or character

[\[:SENSe\]:IQ:CAPture:TRIG:DELAy <num>](#)

(Read and write) query or set IQ capture trigger delay.

Applicable mode Spectrum Analysis
Parameter Trigger delay (us)
 Scope: 1us - 500ms
Example :IQ:CAP:TRIG:DELA 10
Query syntax :IQ:CAP:TRIG:DELA?
Default 1us
Return type Numeric value (double) or character

[\[:SENSe\]:IQ:CAPture:TRIG:AMPLitude <num>](#)

(Read and write) query or set IQ capture trigger level.

Applicable mode Spectrum Analysis
Parameter trigger level (mv)
 Scope: 0 - 5V
Example :IQ:CAP:TRIG:AMP 10
Query syntax :IQ:CAP:TRIG:AMP?
Default 1us
Return type Numeric value (double) or character

[\[:SENSe\]:MEASurement <string>](#)

(Read and write) query or set function measurement type, or set through Power measurement on/off. Only one function measurement is available at a time.

Applicable mode Spectrum Analysis
Parameter Function measurement type.

Parameter Setting	Measuring type
NONE(0)	Normal spectrum measurement
FST(1)	Field strength measurement
CHP(2)	Channel power meter
OBW(3)	Occupied bandwidth measurement
ACPR(4)	Adjacent channel power ratio measurement
DEMODO(5)	Audio demodulation measurement
EM(6)	Emission mask measurement
CNR(7)	Carrier to noise ratio measurement
IQ(8)	IQ capture measurement

Example :MEAS NONE
Query syntax :MEAS?
Default NONE
Return type Numeric value (int) or character

[[:SENSe]:MEASurement:AOff

(Write only) disable function measurement and switch to normal spectrum measurement.

Applicable mode Spectrum Analysis
Parameter None
Example :MEAS:AOff
Query syntax None
Default None
Return type None

[[:SENSe]:OBW:MEthod <string>

(Read and write) set or query method for occupied bandwidth function measurement.

Applicable mode Spectrum Analysis
Parameter Occupied bandwidth measurement method.
 PPOW(0) Percentage
 XDB(1) XdB
Example :OBW:MEth XDB
Query syntax :OBW:MEth?
Default PPOW
Return type Numeric value (int) or character

[[:SENSe]:OBW:OBW?

(Read only) query occupied bandwidth value **(valid after the occupied bandwidth is on and after one sweep).**

Applicable mode Spectrum Analysis
Parameter None
Example :OBW:OBW?
Query syntax :OBW:OBW?
Default None
Return type Number (double) or character (Hz)

[[:SENSe]:OBW:PPOW <num>

(Read and write) set or query occupied bandwidth percentage.

Applicable mode Spectrum Analysis
Parameter occupied bandwidth percentage (no unit).
 Scope: 10.00%~99.99%
Example :OBW:PPOW 90
Query syntax :OBW:PPOW?
Default 99%
Return type Numeric value (float) or character

[[:SENSe]:OBW[:STATe] <bool>

(Read and write) set or query occupied bandwidth function measurement state **(Other functional measurements will be disabled after this function is enabled)**, or use command [[:SENSe]:MEASurement.

Applicable mode	Spectrum Analysis
Parameter	Occupied bandwidth state.
	OFF(0) Occupied bandwidth off
	ON(1) Occupied bandwidth on
Example	:OBW ON
Query syntax	:OBW?
Default	OFF
Return type	Numeric value (bool) or character

[\[:SENSe\]:OBW:XDB <num>](#)

(Read and write) set or query occupied bandwidth XdB value.

Applicable mode	Spectrum Analysis
Parameter	occupied bandwidth XdB (dB).
	Scope: -100.0dB to -0.1dB
Example	:OBW:XDB -3
Query syntax	:OBW:XDB?
Default	-3dB
Return type	Numeric value (float) or character

[\[:SENSe\]:PMMManager:LIMit:STATe <bool>](#)

(Read and write) set or query Power measurement limit state.

Applicable mode	Power meter
Parameter	Limit state.
	OFF(0) Off
	ON(1) On
Example	:PMM:LIM:STAT ON
Query syntax	:PMM:LIM:STAT?
Default	OFF
Return type	Numeric value (bool) or character

[\[:SENSe\]:PMMManager:LIMit:UPPer <num>](#)

(Read and write) set or query Power measurement upper limit value.

Applicable mode	Power meter
Parameter	Upper limit value.
	Scope: -65dBm to 30dBm
Example	:PMM:LIM:UPP 10
Query syntax	:PMM:LIM:UPP?
Default	30 dBm
Return type	Numeric value (float) or character

[\[:SENSe\]:PMMManager:LIMit:LOWer <num>](#)

(Read and write) set or query Power measurement lower limit value.

Applicable mode	Power meter
Parameter	Lower limit value.
	Scope: -70dBm to 25dBm
Example	:PMM:LIM:LOW 10
Query syntax	:PMM:LIM:LOW?
Default	-70dBm
Return type	Numeric value (float) or character

`[[:SENSe]:PMManger:MAXHold <bool>`

(Read and write) set or query Power measurement maximum hold state.

Applicable mode	Power meter
Parameter	maximum hold state.
	OFF(0) Off
	ON(1) On
Example	:PMM:MAXH ON
Query syntax	:PMM:MAXH?
Default	OFF
Return type	Numeric value (bool) or character

`[[:SENSe]:ROSC:SOUR <string>`

(Read and write) set or query 10MHz frequency reference source mode.

Applicable mode	All modes
Parameter	Frequency reference type.
	OFF(0) Frequency reference is internal and off.
	INTernal(0) Frequency reference is internal and on.
	EXTernal(1) Frequency reference is external.
Example	:ROSC:SOUR EXTernal
Query syntax	:ROSC:SOUR?
Default	INTernal
Return type	Numeric value (int) or character

`[[:SENSe]:POWer[:RF]:ATTenuation <num>`

(Read and write) set or query attenuation value.

Applicable mode	Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation, Field Strength
Parameter	Attenuation value
	Seven scales covering 0, 10, 20, 30, 40, 50, 60
Example	:POW:ATT 20
Query syntax	:POW:ATT?
Default	10
Return type	Numeric value (int) or character

`[[:SENSe]:POWer[:RF]:ATTenuation:AUTO <bool>`

(Read and write) set or query attenuation auto on/off.

Applicable mode	Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation, Field Strength
Parameter	Attenuation auto on/off. When the attenuation auto is on, the instrument will automatically set relevant attenuation value based on reference value.
	OFF(0) refers to manual.
	ON(1) refers to auto.
Example	:POW:ATT:AUTO ON
Query syntax	:POW:ATT:AUTO?
Default	ON
Return type	Numeric value (bool) or character

[[:SENSe]:POWER[:RF]:GAIN[:STATe] <bool>

(Read and write) query or set pre-amplifier state.

Applicable mode Spectrum Analysis, Field Strength
Parameter Pre-amplifier ON OFF.
OFF(0) refers to off.
ON(1) refers to on.
Example :POW:GAIN OFF
Query syntax :POW:GAIN?
Default Off
Return type Numeric value (bool) or character

[[:SENSe]:SWEep:MODE <num>

(Read and write) query or set sweep mode, including linear sweep and list sweep.

Applicable mode Spectrum Analysis
Parameter Sweep mode.
LIN(0) Linear sweep
LIST(1) List Sweep
Example :SWE:MODE LIN
Query syntax :SWE:MODE?
Default LIN
Return type Numeric value (int) or character

[[:SENSe]:SWEep:TIME <num>

(Read and write) query or set sweep time of linear sweep. The sweep time is the time required for selected frequency interval for local oscillator tuning. The sweep time directly affects the time for completing one test, excluding the dead time between one sweep and the next sweep. The sweep time generally changes with the span, resolution bandwidth and video bandwidth. The sweep time is not available when the resolution bandwidth is ≤1kHz under spectrum analysis mode.

Applicable mode Spectrum Analysis, Interference Analysis
Parameter Sweep time under linear sweep mode (unit: ms).
Example :SWE:TIME 100
Query syntax :SWE:TIME?
Default None
Return type Numeric value (double) or character

[[:SENSe]:SWEep:TIME:AUTO <bool>

(Read and write) set or query sweep time auto on/off of linear sweep. When auto is on, the instrument will use quick sweep speed as possible, or the manual mode is available to increase the sweep time to meet certain measurement needs. The sweep time for manual setting must be equal to or greater than auto sweep time.

Applicable mode Spectrum Analysis, Interference Analysis
Parameter Sweep time auto on/off under linear sweep mode.
OFF(0) Sweep time manual.
ON(1) Sweep time auto.
Example :SWE:TIME:AUTO ON
Query syntax :SWE:TIME:AUTO?
Default ON
Return type Numeric value (bool) or character

[:SENSe]:SWEep:TRIG <string>

(Read and write) set or query trigger type, including free trigger, video trigger or external trigger.

Applicable mode	Spectrum Analysis	
Parameter	Trigger Type.	
	FREE(0)	Free Trigger
	VIDEO(1)	Video Trigger
	EXTRA(2)	External triggering
Example	:SWE:TRIG FREE	
Query syntax	:SWE:TRIG?	
Default	FREE	
Return type	Numeric value (int) or character	

[:SENSe]:SWEep:TRIG:EXTRa:AMPlitude <num>

(Read and write) set or query external trigger level.

Applicable mode	Spectrum Analysis
Parameter	External trigger level (mv).
	Scope: 0 - 5V
Example	:SWE:TRIG:EXTR:AMP 1
Query syntax	:SWE:TRIG:EXTR:AMP?
Default	1.5V
Return type	Numeric value (float) or character

[:SENSe]:SWEep:TRIG:EXTRa:SLOP <string>

(Read and write) set or query external trigger polarity.

Applicable mode	Spectrum Analysis	
Parameter	External trigger polarity.	
	POS(0)	Positive
	NEG(1)	Negative
Example	:SWE:TRIG:EXTR:SLOP POS	
Query syntax	:SWE:TRIG:EXTR:SLOP?	
Default	POS	
Return type	Number (int) or character	

[:SENSe]:SWEep:TRIG:EXTRa:DELAy <num>

(Read and write) set or query external trigger delay.

Applicable mode	Spectrum Analysis
Parameter	External trigger delay (us).
	Scope: 1us - 500ms
Example	:SWE:TRIG:EXTR:DELA 1
Query syntax	:SWE:TRIG:EXTR:DELA?
Default	1us
Return type	Numeric value (float) or character

[:SENSe]:SWEep:TRIG:VIDEo:AMPlitude <num>

(Read and write) set or query video trigger level.

Applicable mode	Spectrum Analysis
Parameter	Video trigger level (dBm).
	Scope: -120dBm to 40dBm
Example	:SWE:TRIG:VIDE:AMP 10
Query syntax	:SWE:TRIG:VIDE:AMP?
Default	-25dBm

Return type Numeric value (float) or character

[[:SENSe]:SWEep:POINts <num>

(Read and write) set or query sweep points.

Applicable mode Spectrum Analysis, Intereference Analysis

Parameter Sweep points.

Scope: 201, 501, 1001, 2001 and 4001

Example : SWE:POIN 501

Query syntax : SWE:POIN?

Default 1001

Return type Numeric value (int) or character

[[:SENSe]:TAListen:AVOLume <num>

(Read and write) set or query video demodulation measurement volume.

Applicable mode Spectrum Analysis

Parameter Demodulation volume (no unit)

Scope: 0 - 100

Example :TAL:AVOL 80

Query syntax :TAL:AVOL?

Default 60

Return type Numeric value (int) or character

[[:SENSe]:TAListen:DMODE <string>

(Read and write) set or query demodulation mode for audio demodulation measurement. The intermittent mode refers to the mode which demodulation will be stopped as per the demodulation time after sweeping one screen of data before sweeping the next screen of data, and will continue this process over and over again; continuous mode refers to the mode that the instrument will continuously demodulate after sweeping one screen of data and will not sweep data again.

Applicable mode Spectrum Analysis

Parameter Demodulation mode.

INTer(0)

Continuous OFF

CONT(1)

Continuous

Example :TAL:DMOD CONT

Query syntax :TAL:DMOD?

Default CONT

Return type Numeric value (int) or character

[[:SENSe]:TAListen:DSTate <bool>

(Read and write) set or query audio demodulation measurement state **(Other functional measurements will be disabled after this function is enabled)**, or use command [[:SENSe]:MEASurement.

Applicable mode Spectrum Analysis

Parameter Demodulation state.

OFF(0)

Demodulation off

ON(1)

Demodulation on

Example :TAL:DST ON

Query syntax :TAL:DST?

Default OFF

Return type Numeric value (bool) or character

[[:SENSe]:TAListen:DTYPe <string>

(Read and write) set or query audio demodulation measurement demodulation type.

Applicable mode	Spectrum Analysis
Parameter	Demod type.
	FM(0) Frequency modulation
	AM(1) Amplitude modulation
	USB(2) Upper-sideband
	LSB(3) Lower-sideband
Example	:TAL:DTYP FM
Query syntax	:TAL:DTYP?
Default	FM
Return type	Numeric value (int) or character

[[:SENSe]:TAListen:LTIMe <num>

(Read and write) set or query demodulation time for audio demodulation measurement. This parameter is valid when the demodulation mode is intermittent mode and the demodulation time refers to the time for staying in demodulation state after one sweep.

Applicable mode	Spectrum Analysis
Parameter	Demodulation time (ms).
	Scope: 1us - 400s
Example	:TAL:LTIM 100
Query syntax	:TAL:LTIM?
Default	100ms
Return type	Numeric value (double) or character

[[:SENSe]:FREQ:POIN <num>

(Read and write) set or query point frequency.

Applicable mode	Field Strength
Parameter	point frequency(Hz).
	range:1MHz - 44.1GHz
Example	:FREQ:POIN 1000000
Query syntax	:FREQ:POIN?
Default	500MHz
Return type	Numeric value (double) or character

[[:SENSe]:FREQ:TRAC <BOOL>

(Read and write) set or query frequency track on.

Applicable mode	Field Strength
Parameter	frequency track on (Hz).
Example	:FREQ:TRAC ON
Query syntax	:FREQ:TRAC?
Default	OFF
Return type	Numeric value (BOOL) or character

[[:SENSe]:FREQ:FSCan:STARt <num>

(Read and write) set or query start frequency of frequency scanner.

Applicable mode	Field Strength
Parameter	start frequency (Hz).
	range:1MHz - 44.1GHz
Example	:FREQ:FSC:STAR 1000000
Query syntax	:FREQ:FSC:STAR?
Default	500MHz

Return type Numeric value (double) or character

[\[:SENSe\]:FREQ:FSCan:STEP <num>](#)

(Read and write) set or query step frequency of frequency scanner.

Applicable mode Field Strength

Parameter step frequency (Hz).

range:0Hz - 5GHz

Example :FREQ:FSC:STEP 1000000

Query syntax :FREQ:FSC:STEP?

Default 10MHz

Return type Numeric value (double) or character

[\[:SENSe\]:FREQ:FSCan:POINTs <num>](#)

(Read and write) set or query scan points of frequency scanner.

Applicable mode Field Strength

Parameter scan points.

range:2 - 58

Example :FREQ:FSC:POIN 10

Query syntax :FREQ:FSC:POIN?

Default 58

Return type Numeric value (int) or character

[\[:SENSe\]:SWEep:DWELl <num>](#)

(Read and write) set or query dwell time.

Applicable mode Field Strength

Parameter dwell time (us).

range:1ms - 40s

Example :SWE:DWEL 10000

Query syntax :SWE:DWEL?

Default 100ms

Return type Numeric value (double) or character

[\[:SENSe\]:SWEep:DWELl:INFinite <bool>](#)

(Read and write) set or query dwell time auto on.

Applicable mode Field Strength

Parameter dwell time auto on.

Example :SWE:DWEL:INF ON

Query syntax :SWE:DWEL:INF?

Default ON

Return type Numeric value (BOOL) or character

[\[:SENSe\]:SWEep:STAYtime<num>](#)

(Read and write) set or query stay time.

Applicable mode Field Strength

Parameter stay time (us).

range:1ms - 40s

Example :SWE:STAY 10000

Query syntax :SWE:STAY?

Default 100ms

Return type Numeric value (double) or character

[\[:SENSe\]:SWEep:STAYtime:STATe <bool>](#)

(Read and write) set or query stay time auto on.
Applicable mode Field Strength
Parameter stay time auto on.
Example :SWE:STAY:STAT OFF
Query syntax :SWE:STAY:STAT?
Default OFF
Return type Numeric value (BOOL) or character

[\[:SENSe\]:POWer:LIMit:STATe <bool>](#)

(Read and write) set or query limit state.
Applicable mode Field Strength
Parameter limit state.
Example :POW:LIM:STAT OFF
Query syntax :POW:LIM:STAT?
Default OFF
Return type Numeric value (BOOL) or character

[\[:SENSe\]:POWer:LIMit <num>](#)

(Read and write) set or query limit.
Applicable mode Field Strength
Parameter limit (dBm).
range: -174dBm~50dBm
Example :POW:LIM10000
Query syntax :POW:LIM?
Default 0dBm
Return type Numeric value (double) or character

[\[:SENSe\]:DMODE <string>](#)

(Read and write) set or query demodulation mode.
Applicable mode Field Strength
Parameter demodulation mode.
CW(0) continuous wave
FM(1) frequency modulation
AM(2) amplitude modulation
USB(3) upper sideband
LSB(4) lower sideband
SPEAK(5) speak
Example :DMOD CW
Query syntax :DMOD?
Default CW
Return type Numeric value (int) or character

[\[:SENSe\]:DMODE:VOLume <num>](#)

(Read and write) set or query demodulation volume.
Applicable mode Field Strength
Parameter demodulation volume (0 - 100).
Example :DMOD:VOLume 50
Query syntax :DMOD:VOLume?
Default 95
Return type Numeric value (int) or character

[\[:SENSe\]:IFBWidth <num>](#)

(Read and write) set or query bandwidth.
Applicable mode Field Strength
Parameter bandwidth (150Hz~150kHz).
Example :IFBW 1000
Query syntax :IFBW?
Default 30kHz
Return type Numeric value (double) or character

[\[:SENSe\]:DATA:POTF:AMPL?](#)

(Read only) query amplitude of point frequency measurement.
Applicable mode Field Strength
Parameter amplitude.
Example :DATA:POTF:AMPL?
Query syntax :DATA:POTF:AMPL?
Default None
Return type Numeric value (double) or character

[\[:SENSe\]:DATA:POTF:FIELD?](#)

(Read only) query field strength of point frequency measurement.
Applicable mode Field Strength
Parameter field strength.
Example :DATA:POTF:FIEL?
Query syntax :DATA:POTF:FIEL?
Default None
Return type Numeric value (double) or character

[\[:SENSe\]:DATA:FSCan:AMPL?](#)

(Read only) query amplitude of frequency scanner
Applicable mode Field Strength
Parameter amplitude.
Example :DATA:FSC:AMPL?
Query syntax :DATA:FSC:AMPL?
Default None
Return type Numeric value or character
The character format is xx.xx,xx.xx,...xx.xx\n, in which xx.xx is float data, numbers are separated by “,” to indicate interval end and “\n” to indicate end.

[\[:SENSe\]:DATA:FSCan:FIELd?](#)

(Read only) query field strength of frequency scanner.
Applicable mode Field Strength
Parameter field strength.
Example :DATA:FSC:FIEL?
Query syntax :DATA:FSC:FIEL?
Default None
Return type Numeric value or character
The character format is xx.xx,xx.xx,...xx.xx\n, in which xx.xx is float data, numbers are separated by “,” to indicate interval end and “\n” to indicate end.

[\[:SENSe\]:DATA:LSCan:AMPL?](#)

(Read only) query amplitude of list scanner.

Applicable mode Field Strength
Parameter amplitude.
Example :DATA:LSC:AMPL?
Query syntax :DATA:LSC:AMPL?
Default None
Return type Numeric value or character
 The character format is xx.xx,xx.xx,...xx.xx\n, in which xx.xx is float data, numbers are separated by “,” to indicate interval end and “\n” to indicate end.

[\[:SENSe\]:DATA:LSCan:FIELD?](#)

(Read only) query field strength of list scanner.

Applicable mode Field Strength
Parameter field strength.
Example :DATA:LSC:FIEL?
Query syntax :DATA:LSC:FIEL?
Default None
Return type Numeric value or character
 The character format is xx.xx,xx.xx,...xx.xx\n, in which xx.xx is float data, numbers are separated by “,” to indicate interval end and “\n” to indicate end.

[\[:SENSe\]:FREQuency:FCOut?](#)

(Read only) query offset of point frequency measurement.

Applicable mode Field Strength
Parameter offset.
Example :FREQ:FCO?
Query syntax :FREQ:FCO?
Default None
Return type Numeric value (double) or character

[\[:SENSe\]:FST:MEASurement <string>](#)

(Read and write) set or query measurement type.

Applicable mode Field Strength
Parameter measurement type.
 POTF(0) point frequency mode
 FREQ(1) frequency scanning mode
 LIST(2) list scanning mode
Example :FST:MEAS POTF
Query syntax :FST:MEAS?
Default POTF
Return type Numeric value (int) or character

[\[:SENSe\]:FST:PEAK](#)

(Write only) set the peak value of marker.

Applicable mode Field Strength
Parameter None.
Example :FST:PEAK
Query syntax None
Default None
Return type None

[\[:SENSe\]:FST:MARKer <bool>](#)

(Read and write) set or query marker state.

Applicable mode Field Strength
Parameter marker state.
Example :FST:MARK ON
Query syntax :FST:MARK?
Default OFF
Return type Numeric value (BOOL) or character

[\[:SENSe\]:FST:INDEx <int>](#)

(Read and write) set or query marker index.

Applicable mode Field Strength
Parameter marker index (0~57).
Example :FST:INDEx 20
Query syntax :FST:INDEx?
Default 29
Return type Numeric value (BOOL) or character

[:SYSTem:BATTery:STAT?](#)

(Read only) query battery state.

Applicable mode all modes
Parameter None
Example :SYST:BATT:STAT?
Query syntax :SYST:BATT:STAT?
Default None
Return type Numeric value (int) 1 battery and external power source 2 external power source only 3 battery only

[:SYSTem:BATTery:VOLume?](#)

(Read only) query battery state.

Applicable mode all modes
Parameter None
Example :SYST:BATT:VOL?
Query syntax :SYST:BATT:VOL?
Default None
Return type Numeric value (int) or character

[:SYSTem:GPS <bool>\(Options\)](#)

(Read and write) set or query GPS on/off. When on, the screen will display the longitude and latitude and sea level collected by GPS. This command is an overlapping command. Use ***OPC?** before sending other commands to query if this command is completed.

Applicable mode	All modes	
Parameter	GPS on/off	
	OFF(0)	GPS off
	ON(1)	GPS on
Example	:SYST:GPS ON;	
Query syntax	:SYST:GPS?	
Default	OFF	
Return type	Numeric value (bool) or character	

[:SYSTem:GPS:DATA?](#)

(Read only) return current GPS data in the format below: "<longitude>,<latitude>,<sea level>,<time UTC>"

Applicable mode All modes

Parameter	None
Example	:SYST:GPS:DATA? Return 38 28'11.22" N,122 42'13.23" W,152,06/28/2010 23:35:38\n Return --,--,--,-- n if no data
Query syntax	:SYST:GPS:DATA?
Default	None
Return type	Character

:SYSTem:GPS:RECeive[:STATe]?

(Read only) query GPS receiver state.

Applicable mode	All modes
Parameter	None
Example	:SYST:GPS:REC?
Query syntax	:SYST:GPS:REC?
Default	None
Return type	Numeric value (int) or character 0: receiver off 1: receiver on

:SYSTem:GPS:RST

(Write only) GPS reset. In areas with poor reception where GPS signal cannot be received after a long time even when moving to another place. Under this condition, reset can run new GPS positioning to quickly start search. In this case, reset can be selected to enable the module to find galaxy positioning again.

Applicable mode	All modes
Parameter	None
Example	:SYST:GPS:RST
Query syntax	None
Default	None
Return type	None

:SYSTem:GPS:STATe?

(Read only) query GPS state.

Applicable mode	All modes
Parameter	None
Example	:SYST:GPS:STAT?
Query syntax	:SYST:GPS:STAT?
Default	None
Return type	Numeric value (int) or character 0: Invalid 1: No Differential Fix 2: Differential Fix 3: Invalid PPS 4: Estimated Reckoning

:SYSTem:INFO?

(Read only) query system information.

Applicable mode	all modes
Parameter	None
Example	:SYST:INFO?
Query syntax	:SYST:INFO?
Default	None

Return type character string

:SYSTem:PWR:SHUTdown <num>

(Read and write) query or set shutdown time.

Applicable mode	All modes
Parameter	Shutdown time. Scope: [1,240] min
Example	:SYST:PWR:SHUT 20
Query syntax	:SYST:PWR:SHUT?
Default	20 minutes
Return type	Numeric value (int) or character

:SYSTem:PWR:SHUTdown:STATe <bool>

(Read and write) query or set shutdown time state. When on, the instrument will automatically shut down when reaching the set shutdown time **(any operation will result in re-timing of shutdown time)**.

Applicable mode	All modes
Parameter	Auto shutdown on/off. OFF(0) refers to off. ON(1) refers to on.
Example	:SYST:PWR:SHUT:STAT OFF
Query syntax	:SYST:PWR:SHUT:STAT?
Default	Off
Return type	Numeric value (bool) or character

:SYSTem:PWR:SLEEp <num>

(Read and write) query or set sleep time.

Applicable mode	All modes
Parameter	Sleep time Scope: [1,240] min
Example	:SYST:PWR:SLE 20
Query syntax	:SYST:PWR:SLE?
Default	240 minutes
Return type	Numeric value (int) or character

:SYSTem:PWR:SLEEp:STATe <bool>

(Read and write) query or set sleep time state. When on, the instrument will automatically in sleep state when reaching the set sleep time **(any operation will result in re-timing of sleep time)**.

Applicable mode	All modes
Parameter	Auto sleep on/off. OFF(0) refers to off. ON(1) refers to on.
Example	:SYST:PWR:SLEE:STAT OFF
Query syntax	:SYST:PWR:SLEE:STAT?
Default	Off
Return type	Numeric value (bool) or character

:SYSTem:TIME <num>,<num>,<num>,<num>,<num>

(Read and write) query or set time.

Applicable mode	All modes
Parameter	Time

	Date & time
Example	:SYST:TIME 2015,12,30,10,30
Query syntax	:SYST:TIME?
Default	None
Return type	Character

:TRACe<n>:DATA?

(Read only) query trace data.

Applicable mode	Spectrum Analysis, Intereference Analysis, AM-FM-PM Analysis
Parameter	Trace number can be set as 1,2 or 3, indicating trace 1, 2, or 3.
<n>	

If not marked, then n represents 1. Trace number is not necessary under Intereference Analysis and AM-FM-PM Analyzer modes.

Example	:TRAC1:DATA?
Query syntax	:TRAC1:DATA?
Default	None
Return type	Numeric value or character

Number format: “#NXXXX data” XXXX is the size of binary data. N is the no. n digit of XXXX. For example: #3512.... means that the size of binary data is 3 and the 3 digits after 3 is 512, indicating this data is followed by a binary data with 512 bytes. The data type of each point of the trace is float Tye, occupying 4 digits.

The character format is xx.xx,xx.xx,...xx.xx/n, in which xx.xx is float data, numbers are separated by “,” to indicate interval end and “/n” to indicate end.

Only one trace data is sent each time under Spectrum Analysis and Interference Analysis modes. Data containing three traces can be sent each time under AM-FM-PM Analyzer mode, totaling 3003 points. Each trace contains 1001 points. The three traces are RF Spectrum, Audio Spectrum and Audio Waveform.

:TRACe<n>:TYPE <string>

(Read and write) query or set trace state.

Applicable mode	Spectrum Analysis
Parameter <n>	Trace number can be set as 1,2 or 3, indicating trace 1, 2, or 3.
<string>	If not marked, then n represents 1.

Trace types.
CLRW(0), Clear Write
MAXH(1), maximum hold
MINH(2), minimum hold
VIEW(3), view
BLANk(4), conceal

Example	:TRAC2:TYPE CLRW
Query syntax	:TRAC2:TYPE?
Default	CLRW
Return type	Numeric value (int) or character

[[:SENSe]:SIG:MARG:STAT <bool>

(Read-write)Query or set the limit ON/OFF.

Applicable mode	Signal analysis
Parameter	ON(1) ON OFF(0) OFF

Example :SIG:MARG:STAT ON
Query syntax :SIG:MARG:STAT?
Default ON
Return type Number (bool) or character

[\[:SENSe\]:SIG:MARG <num>](#)

(Read-write) Query or set the limit value.

Applicable mode Signal analysis
Parameter Limit value
Example: :SIG:MARG 10 **Query**
syntax: :SIG:MARG? **Default**
15
Return type Number (double) or character

[\[:SENSe\]:SIG:LIST <string>](#)

(Read-write) Query or list the signal list.

Applicable mode Signal analysis
Parameter DETA(0) Detail
BRI(1) Brief
Example :SIG:LIST DETA
Query syntax :SIG:LIST?
Default BRI
Return type Number (int) or character

[\[:SENSe\]:GEN:MODE < string >](#)

(Read-write) Query or set the generator mode ON/OFF

Applicable mode Spectrum analysis
Type Read-write
Parameter CW(0) Mode
OFF TRACK(1) Mode
ON
Example :GEN:MODE CW
Query syntax :GEN:MODE?
Default CW
Return type Number (int) or character

[\[:SENSe\]:GEN:FREQ:OFFSet <num>](#)

(Read-write) Query or set the generator frequency offset

Applicable mode Spectrum analysis
Type Read-write
Parameter No
Example :GEN:FREQ:OFFS 1000000
Query syntax :GEN:FREQ:OFFS?
Default 0
Return type Number (double) or character

[\[:SENSe\]:GEN:FREQ:POTF <num>](#)

(Read-write) Query or set the generator frequency

Applicable mode Spectrum analysis
Type Read-write
Parameter No
Example :GEN:FREQ:POTF 1000000000

Query syntax :GEN:FREQ:POTF?
Default 1000000000
Return type Number (double) or character

[\[:SENSe\]:GEN:AMPL:OUT <num>](#)

(Read-write)Query or set the generator output power

Applicable mode Spectrum analysis

Type Read-write

Parameter No

Example :GEN:AMPL:OUT -2

Query syntax :GEN:AMPL:OUT?

Default 0

Return type Number (float) or character

[\[:SENSe\]:GEN:AMPL:OFFS <num>](#)

(Read-write)Query or set the generator power offset

Applicable mode Spectrum analysis

Type Read-write

Parameter No

Example :GEN:AMPL:OFFS 10

Query syntax :GEN:AMPL:OFFS?

Default 0

Return type Number (float) or character

[\[:SENSe\]:GEN:NORMZ:STAT <bool>](#)

(Read-write)Query or set the generator normalization ON/OFF

Applicable mode Spectrum analysis

Type Read-write

Parameter ON(1) Normalization ON
OFF(0) Normalization OFF

Example :GEN:NORM:STAT ON

Query syntax :GEN:NORM:STAT?

Default OFF(0)

Return type Number (bool) or character

[\[:SENSe\]:GEN:NORMZ:RLEV <num>](#)

(Read-write)Query or set the reference level of generator normalization

Applicable mode Spectrum analysis

Type Read-write

Parameter No

Example :GEN:NORM:RLEV 10

Query syntax :GEN:NORM:RLEV?

Default 0

Return type Number (float) or character

[\[:SENSe\]:GEN:NORM:RPOS <NUM>](#)

(Read-write)Query or set the reference position of generator normalization

Applicable mode Spectrum analysis

Type Read-write

Parameter No

Example :GEN:NORM:RPOS 5
Query syntax :GEN:NORM:RPOS?
Default 0
Return type Number (int) or character

[\[:SENSe\]:GEN:NORM:PDIV <num>](#)

(Read-write) Query or set the scale/division of generator normalization

Applicable mode Spectrum analysis

Type Read-write

Parameter No

Example :GEN:NORM:PDIV 10

Query syntax :GEN:NORM:PDIV?

Default 0

Return type Number (float) or character

[\[:SENSe\]:GEN:NORM:RTRA < num >](#)

(Read-write) Query or set the generator reference trace ON/OFF

Applicable mode Spectrum analysis

Type Read-write

Parameter SHOW(1) Reference trace
ON HIDE(0) Reference trace
OFF

Example :GEN:NORM:RTRA SHOW

Query syntax :GEN:NORM:RTRA?

Default HIDE(0)

Return type Number (int) or character

Section IV Programing instances

This chapter describes how to use different I/O libraries and programming design languages to explain the control of spectrum analyzer. Instrument control is realized through LAN for communication (when using USB port communication, first the USB drive should e installed. Refer to Section I of Chapter II for the instructions of USB drive installation. After USB drive is successfully installed, specific implementation step will be the same as LAN port communication).

1. C/C++ instance

PC should at least have the following configuration:

windows XP operating system

VC6.0 integrated development environment

VISA library of NI

Network card

2. Running of C/C++ design program

To run the C/C++ program, VC6.0 should include the corresponding library file.

The following steps should be followed to use VISA library:

Add the visa.h file to the header file.

Add visatype.h to header file

Add visa32.lib to the project

3. Network design example

To correctly use the instances below, please first confirm the IP address of S3302 Spectrum Analyzer.

1) Use socket and C++ to realize frequency setting and query

Specific realization codes are as follows (the code can be removed, and this section provides only an implementation instance):

Establish dialog box-based MFC project and add the code below in the program:

```
void CSocketTestDlg::Test()
{
    CSocket sockClient;
    bool flag;
    char buff[100];
    if(!AfxSocketInit())
    {
        AfxMessageBox(_T("Initialization failed!"));
    }
    else
    {
        flag = sockClient.Create();
        if(flag)
        {
            AfxMessageBox(_T("socket creation succeeded!"));
        }
        else
        {
            AfxMessageBox(_T("socket creation failed!"));
            sockClient.Close();
        }
    }
    flag = sockClient.Connect(name,5000); /* name is the IP address of spectrum analyzer
    flag = sockClient.Send(":FREQ:STAR 1000000\n",100,0);
    if(!flag)
    {
        AfxMessageBox(_T("send failed!"));
        exit(0);
    }
}
```

```

    }
    flag = sockClient.Send("FREQ:STAR?\n",12,0);
    if(!flag)
    {
        AfxMessageBox(_T("send failed!"));
        exit(0);
    }
    flag = sockClient.Receive(buff,100,0);
    if(!flag)
    {
        AfxMessageBox(_T("receive failed!"));
        exit(0);
    }
    sockClient.Close();
}

```

2) Use VISA library and C++ to realize setting and query instructions

The following files should be include

```

#include<visa.h>
#include<afxsock.h>
#include<visa.h>
extern char ResourceStr[50];
ViSession DftRM;
ViSession vi;
/* open the device
ViStatus S3302_OpenDevice(BOOL bUsb)
{
    char ResourceStr[50];
    if(bUsb)
    {
        strcpy(ResourceStr, "USB0::0x8086::0xA6CD::NI-VISA-0::RAW");
    }
    else
    {
        strcpy(ResourceStr, "TCPIP::x.x.x.x::5000::SOCKET"); /* x.x.x.x is the IP address of
spectrum analyzer.
    }
    ViStatus nReturnStatus = 0;
    nReturnStatus = viOpenDefaultRM(&DftRM);
    nReturnStatus = viOpen(DftRM, ResourceStr, VI_NULL, VI_NULL, &vi);
    if(!bUsb)
    {
        viSetAttribute(vi, VI_ATTR_SUPPRESS_END_EN, FALSE);
    }
    return nReturnStatus;
}
/*Set center frequency
ViStatus S3302_SetFqCent(double Fq)
{
    ViChar Buf[64];
    sprintf(Buf,"%s %.3Lf;", ":FREQ:CENT", Fq);
    ViUInt32 returnCount = 0;
    if( strlen(Buf)!= 0)
        return viWrite(vi, (ViBuf)Buf, strlen(Buf), &returnCount);
    else
        return -1;
}
/*Query center frequency
ViStatus S3302_QueryFqCent(double& Fq)
{
    ViChar CmdBuf[64];
    ViChar RcvBuf[64];
    ViUInt32 returnCount = 0;
    ViUInt32 actualCount = 0;
    ViStatus nStatus = TRUE;

```

```
sprintf(CmdBuf,"%s?\n;", ":FREQ:CENT");
nStatus |= viWrite(vi, (ViBuf)CmdBuf, strlen(CmdBuf), &returnCount);
nStatus |= viRead(vi, (ViBuf)RcvBuf, 100, &actualCount);
Fq = *(reinterpret_cast<double*>(RcvBuf));
return TRUE;
}
```

Chapter II Function Description of Secondary Development Library

For users' convenience, we encapsulate SCPI commands and make it into a dynamic linking library. Users can conveniently set or query S3302 by calling this dynamic linking library, which is suitable for them to create auto test system. (Note: This dynamic library is generated under LabWindows/CVI 2010 programming environment and the communication interface is VISA library of NI.

Section I Drive installation

If using cross-over cable for connection, only IO library above NI-VISA Runtime4.4.1 and above will be installed (**not compatible to VISA library of Agilent**). If using USB cable for connection, then ,in addition to install IO library above NI-VISA Runtime4.4.1 and above, a USB drive should also be installed. S3302USBSETUP.inf should be installed for xp operating system and S3302USBSETUP_vista.inf for win7 operating system. Specific installation process is as follows:

1. click Setup.exe under Volume directory in the disk, and click Add/Remove to start install NI-VISA library, click next until installation is completed.
2. for USB drive, right click S3302USBSETUP.inf installation file and select install until completion.
3. the computer will realize communication by connecting to S3302 through a cross-over cable or USB cable.

Section II Function description

Operating Instructions of Dynamic Linking Library

This dynamic linking library includes three files which are S3302.h, S3302.dll and S3302.lib Under LabWindows/CVI programing environment, by adding these three files into the project, the user can realize control of the instrument by using functions in S3302.h.

Instrument connection – Open device

ViStatus _VI_FUNC S3302_init(ViRsrc resourceName, ViBoolean IDQuery, ViBoolean resetDevice, ViSession* instrumentHandle)

Function purpose:

Open device

This function is the first function to be called for accessing the instrument as this function completes the initialization operation below:

Open the handle of the module based on the interface designated by **parameter** resourceName and logical address information to establish data channel with the spectrum analyzer.

The returned instrumentHandle is for identifying the module in the future recall of instrument drive functions.

Parameter list:

resourceName

Instrument resource character

S3302 resource character string under USB cable connection state can be obtained by the following mode:

ViChar ResourceName[256];

ViSession defaultRM;

ViFindList fList;

ViUInt32 num;

viOpenDefaultRM(&defaultRM);

viFindRsrc(defaultRM,"USB?::0x045E::0x00CE::?*::RAW",&fList,&num,ResourceName);

The resource character string stored in the ResourceName is the string queried.

TCP connection resource character string is "TCPIP::172.141.11.202::5000::SOCKET", in which the underlined part is the default IP address of the instrument. If the IP of the instrument is changed, the underlined part shall be the actual IP of the instrument.

IDQuery

ID query parameter. If set as VI_TRUE, the function will query instrument ID and check compliance with the drive.

resetDevice

If this parameter is set as VI_TRUE, the function will reset the device, same as sending *RST command.

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Instrument connection – close instrument

ViStatus _VI_FUNC S3302_close (ViSession instrumentHandle) Function

purpose:

Close the instrument. This function is called to close the instrument after the control is done.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

IEEE488.2 function

Clear instrument state

ViStatus _VI_FUNC S3302_CLS (ViSession instrumentHandle) Function

purpose:

Clear the state of the instrument, i.e.: Clear error queue and all event registers and cancel *OPC command and query command to be processed.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Query instrument identification number

ViStatus _VI_FUNC S3302_QueryIDN (ViSession instrumentHandle, char IDN[])**Function purpose:**

Inquire the character string of the instrument identification number.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

IDN

Instrument identification character string sent from the instrument, in the form of "SALUKI,S3302,XXXXXX,X.X.X" under normal state. XXXXXX is the serial number of the instrument, X.X.X is the version number of current host program.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Operation complete command.](#)

ViStatus _VI_FUNC S3302_OPC (ViSession instrumentHandle) Function**purpose:**

After completing all overlapping commands to be processed (**for example: trigger one sweep command**), set the OPC bit of standard event state register.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Operation complete query](#)

ViStatus _VI_FUNC S3302_QueryOPC (ViSession instrumentHandle, ViInt32 nVal[])**Function purpose:**

1 will be returned when all overlapping commands to be processed are completed.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Query value after operation. If the value is 1, this means that overlapping commands are completed.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means

failed.

Reset

ViStatus _VI_FUNC S3302_Reset (ViSession instrumentHandle) Function

purpose:

Restore current work mode of the instrument to known default state known as default state.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Wait

ViStatus _VI_FUNC S3302_WAI (ViSession instrumentHandle) Function

purpose:

Wait for processing of all overlapping commands before processing of new commands.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Universal functions for all measurement modes

Mode – query available instrument mode

ViStatus _VI_FUNC S3302_QueryInstCatalog (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

query available instrument mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Available instrument mode.

The 0 bit is the bit for spectrum analysis test, 1 (mandatory)

The 1 bit is the bit for AM-FM-PM demodulation test, 1 is settable (optional) and 0 is not settable

The 2 bit is the bit for interference analysis test, 1 is settable (optional) and 0 is not settable

The 3 bit is the bit for power measurement test, 1 is settable (optional) and 0 is not settable

The 4 bit is the bit for channel sweep test, 1 is settable (optional) and 0 is not settable

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Mode – set instrument mode](#)

ViStatus _VI_FUNC S3302_SetInstSel (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set instrument mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Instrument mode.

1: spectrum analysis mode

2: interference analysis mode

3: AM-FM-PM analysis mode

4: Power measurement mode

5: channel sweep mode

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Mode – query instrument mode](#)

ViStatus _VI_FUNC S3302_QueryInstSel (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

query instrument mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Instrument mode.

1: spectrum analysis mode

2: interference analysis mode

3: AM-FM-PM analysis mode

4: Power measurement mode

5: channel sweep mode

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Data – set data type](#)

ViStatus _VI_FUNC S3302_SetFormat (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set data type.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Data type.

0: character type

1: binary type

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Data – query data type](#)

ViStatus _VI_FUNC S3302_QueryFormat (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

query data type.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Data type.

0: character type

1: binary type

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[File – delete state file](#)

ViStatus _VI_FUNC S3302_DeleteStateFile (ViSession instrumentHandle, char chStr[])

Function purpose:

Delete state file under current mode **(if the file does not exist, the command will be invalid).**

The command be only valid for current storage location).

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[File – delete all state files](#)

ViStatus _VI_FUNC S3302_DeleteAllStateFile (ViSession instrumentHandle) Function

purpose:

delete all state files under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[File – recall state file](#)

ViStatus _VI_FUNC S3302_LoadStateFile (ViSession instrumentHandle, char chStr[])

Function purpose:

Recall state file under current mode **(this command will be invalid if the file does not exist and be only valid for current storage location).**

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[File – save state file](#)

ViStatus _VI_FUNC S3302_StoreStateFile (ViSession instrumentHandle, char chStr[])

Function purpose:

Store state file under current mode **(if the file does not exist, the command will be invalid.**

The command be only valid for current storage location).

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[File – set storage location](#)

ViStatus _VI_FUNC S3302_SetLocation (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

set storage location.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Location.

0: internal.

1: SD card.

2: USB.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[File – query storage location](#)

ViStatus _VI_FUNC S3302_QueryLocation (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query current storage location

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Location.

0: internal.

1: SD card.

2: USB.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[File – store screen copy](#)

ViStatus _VI_FUNC S3302_StoreScreen (ViSession instrumentHandle, char chStr[])

Function purpose:

Screen copy, save current screenshot as file **(file will be overwritten if exist and only valid to current storage location).**

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[System – set frequency reference](#)

ViStatus _VI_FUNC S3302_SetRoscSource (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set 10MHz frequency reference source mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Frequency reference type.

0: frequency reference is internal

1: frequency reference is external.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[System – query frequency reference](#)

ViStatus _VI_FUNC S3302_QueryRoscSource (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query 10MHz frequency reference source mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Frequency reference type.

0: frequency reference is internal

1: frequency reference is external.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[System – GPS –set GPS on/off](#)

ViStatus _VI_FUNC S3302_SetGPSOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

set GPS on/off, when on, the screen will display the longitude and latitude and sea level collected by GPS chip. This command is an overlapping command. Before sending other commands, use **S3302_QueryOPC()** to query if this command is completed.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[System – GPS – query GPS on/off](#)

ViStatus _VI_FUNC S3302_QueryGPSOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query GPS on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

System – GPS – query GPS state

ViStatus _VI_FUNC S3302_QueryGPSState (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

query GPS state.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

GPS state.

0: Invalid

1: No Differential Fix

2: Differential Fix

3: Invalid PPS

4: Estimated Reckoning

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

System – GPS – query GPS receiver state

ViStatus _VI_FUNC S3302_QueryGPSReceiveState (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query GPS receiver state.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

GPS receiver state.

0: no receiver data

1: receiver data

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

System – GPS – GPS reset

ViStatus _VI_FUNC S3302_GPSCReset (ViSession instrumentHandle)

Function purpose:

GPS reset. In areas with poor reception where GPS signal cannot be received after a long time even when moving to another place. Under this condition, reset can run new GPS positioning to quickly start search. In this case, reset can be selected to enable the module to find galaxy positioning again.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

System – GPS – query GPS data

ViStatus _VI_FUNC S3302_QueryGPSData (ViSession instrumentHandle, ViChar chStr[])

Function purpose:

Query data collected by GPS chip, and return current GPS data in the format below: "<longitude>,<latitude>,<sea level>,<timeUTC>".

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Returned GPS data.

Example: Return "38 28'11.22" N,122 42'13.23" W,152,06/28/2010 23:35:38 n" when there is data

Return "--,--,--,-- n" if no data

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

System – shutdown – set auto shutdown on/off

ViStatus _VI_FUNC S3302_SetAutoShutdownOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set auto shutdown on/off. When on, the instrument will shutdown automatically once reaching the set shutdown time.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[System – shutdown – query auto shutdown on/off](#)

ViStatus _VI_FUNC S3302_QueryAutoShutdownOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

Query shutdown auto on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[System – shutdown – set shutdown time](#)

ViStatus _VI_FUNC S3302_SetShutdown (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set shutdown time. When shutdown auto is on, the instrument will automatically shutdown once reaching the set shutdown time.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Shutdown time (minute), scope: 1~240 min.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[System – shutdown – query shutdown time](#)

ViStatus _VI_FUNC S3302_QueryShutdown (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query shutdown time

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Shutdown time (min)

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[System – sleep – set sleep auto on/off](#)

ViStatus _VI_FUNC S3302_SetAutoSleepOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set sleep auto on/off. When auto sleep is on, when reaching the set sleep time, the instrument will automatically go to sleep and turn off screen display.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[System – sleep – query sleep auto on/off](#)

ViStatus _VI_FUNC S3302_QueryAutoSleepOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query sleep auto on/off

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[System – sleep – set sleep time](#)

ViStatus _VI_FUNC S3302_SetSleep (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set sleep time. When auto sleep is on, when reaching the set sleep time, the instrument will automatically go to sleep and turn off screen display.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Sleep time (minute), scope: 1~240 min.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[System – sleep – query sleep time](#)

ViStatus _VI_FUNC S3302_QuerySleep (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

query sleep time.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Sleep time (min)

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[System – set title](#)

ViStatus _VI_FUNC S3302_SetTitle (ViSession instrumentHandle, char chStr[]) Function

purpose:

set title.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Title name

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[System – set title on/off](#)

ViStatus _VI_FUNC S3302_SetTitleOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

set title on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[System – query title on/off](#)

ViStatus _VI_FUNC S3302_QueryTitleOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query title on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[System – set display mode](#)

ViStatus _VI_FUNC S3302_SetShowMode (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Setting display mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Location.

0: default mode.

1: black and white mode.

2: night vision mode.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

System – query display mode

ViStatus _VI_FUNC S3302_QueryShowMode (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

query display mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Location.

0: default mode.

1: black and white mode.

2: night vision mode.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

System – set brightness auto adjustment on/off

ViStatus _VI_FUNC S3302_SetBrightOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

set brightness auto adjustment on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

System – query brightness auto adjustment on/off

ViStatus _VI_FUNC S3302_QueryBrightOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query brightness auto adjustment on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[System - set brightness level.](#)

ViStatus _VI_FUNC S3302_SetBright (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set brightness level.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Brightness level, scope: 0~4.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[System – query brightness level](#)

ViStatus _VI_FUNC S3302_QueryBright (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query brightness level.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Brightness level.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Spectrum Analysis Mode Functions](#)

[Frequency – set center frequency](#)

ViStatus _VI_FUNC S3302_SetCntFreq (ViSession instrumentHandle, ViReal64 dbVal)

Function purpose:

Set center frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency.

Frequency range of spectrum analysis, 0Hz~44.1GHz.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Frequency – query center frequency](#)

ViStatus _VI_FUNC S3302_QueryCntFreq (ViSession instrumentHandle, ViReal64 dbVal[])

Function purpose:

Query center frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Frequency – set step frequency](#)

ViStatus _VI_FUNC S3302_SetStepFreq (ViSession instrumentHandle, ViReal64 dbVal)

Function purpose:

Set step value of center frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency, scope: 1Hz~5GHz

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – query step frequency

ViStatus _VI_FUNC S3302_QueryStepFreq (ViSession instrumentHandle, ViReal64 dbVal[])

Function purpose:

Query step value of center frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – set step frequency auto on/off

ViStatus _VI_FUNC S3302_SetAutoStepFreqOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

set step frequency auto on/off. When auto is on, the step frequency is 1MHz. when off, the step frequency can be 1Hz~5GHz.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – query step frequency auto on/off

ViStatus _VI_FUNC S3302_QueryAutoStepFreqOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query step frequency auto on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Frequency – set span](#)

ViStatus _VI_FUNC S3302_SetSpan (ViSession instrumentHandle, ViReal64 dbVal)

Function purpose:

Set span under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency.

Frequency range of spectrum analysis, 0Hz~44.1GHz.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Frequency – query span](#)

ViStatus _VI_FUNC S3302_QuerySpan (ViSession instrumentHandle, ViReal64 dbVal[])

Function purpose:

Query span under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Frequency – full span](#)

ViStatus _VI_FUNC S3302_SetFullSpan (ViSession instrumentHandle)

Function purpose:

Set as full span.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Frequency – zero span](#)**ViStatus _VI_FUNC S3302_SetZeroSpan (ViSession instrumentHandle) Function****purpose:**

Set as zero span.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Frequency – previous span](#)**ViStatus _VI_FUNC S3302_SetLastSpan (ViSession instrumentHandle)****Function purpose:**

Set as previous span.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Frequency – set start frequency](#)**ViStatus _VI_FUNC S3302_SetSttFreq (ViSession instrumentHandle, ViReal64 dbVal)****Function purpose:**

Set start frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency.

Frequency range of spectrum analysis, 0Hz~44.1GHz.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – query start frequency

ViStatus _VI_FUNC S3302_QuerySttFreq (ViSession instrumentHandle, ViReal64 dbVal[])

Function purpose:

Query start frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – set stop frequency

ViStatus _VI_FUNC S3302_SetStpFreq (ViSession instrumentHandle, ViReal64 dbVal)

Function purpose:

Set stop frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency.

Frequency range of spectrum analysis, 0Hz~44.1GHz.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – query stop frequency

ViStatus _VI_FUNC S3302_QueryStpFreq (ViSession instrumentHandle, ViReal64 dbVal[])

Function purpose:

Query stop frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Frequency – set signal standard name](#)

ViStatus _VI_FUNC S3302_SetSIGStandard (ViSession instrumentHandle, char* standard)

Function purpose:

set signal standard name under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Standard

Name of signal standard.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Frequency – query signal standard name](#)

ViStatus _VI_FUNC S3302_QuerySIGstandard (ViSession instrumentHandle, char standard[])

Function purpose:

Query signal standard name under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Standard

Name of signal standard.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Frequency – set signal standard channel number](#)

ViStatus _VI_FUNC S3302_SetChannelNum (ViSession instrumentHandle, ViInt32 channelNum)

Function purpose:

Set channel number under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

channelNum

Channel number.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – query signal standard channel number

ViStatus _VI_FUNC S3302_QueryChannelNum (ViSession instrumentHandle, ViInt32 channelNum[])

Function purpose:

Set channel number under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

channelNum

Channel number.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – set zero span IF output on/off

ViStatus _VI_FUNC S3302_SetIFOutOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

set zero span IF output on/off

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – query zero span IF output on/off

ViStatus _VI_FUNC S3302_QueryIFOutOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query zero span IF output on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Frequency – set zero span IF output IF selection](#)

ViStatus _VI_FUNC S3302_SetIFOutSelect (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

set zero span IF output IF selection.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Intermediate frequency selection

0: 3IF, 1: 4IF.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Frequency – query zero span IF output IF selection](#)

ViStatus _VI_FUNC S3302_QueryIFOutSelect (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

query zero span IF output IF selection.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Intermediate frequency selection

0: 3IF, 1: 4IF.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – set reference level

ViStatus _VI_FUNC S3302_SetRef (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

set reference level. Reference level is related to current amplitude unit, and the setting scope corresponds to dBm. Conversion is required.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

reference level (-120dBm~40dBm).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – query reference level

ViStatus _VI_FUNC S3302_QueryRef (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

query reference level (reference value). Reference level value is related to current amplitude unit.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Reference level value (reference value).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – set reference position

ViStatus _VI_FUNC S3302_SetRefPos (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Setting of reference position.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Reference position, scope: -10~10.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – query reference position

ViStatus _VI_FUNC S3302_QueryRefPos (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

query reference position.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

reference position value.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – set attenuation

ViStatus _VI_FUNC S3302_SetAtt (ViSession instrumentHandle, ViInt32 nVal) Function

purpose:

Set attenuation, only seven scales are available, which are 0, 10, 20, 30, 40, 50, 60. Other values set will be set as the attenuation for adjacent channel.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Attenuation, seven scales 0, 10, 20, 30, 40, 50, 60

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – query attenuation

ViStatus _VI_FUNC S3302_QueryAtt (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

query attenuation.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Attenuation, seven scales 0, 10, 20, 30, 40, 50, 60

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Amplitude - set attenuation auto on/off](#)

ViStatus _VI_FUNC S3302_SetAutoAttOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set attenuation auto on/off. When on, the instrument will set relevant attenuation value automatically based on the reference value.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Amplitude - query attenuation auto on/off](#)

ViStatus _VI_FUNC S3302_QueryAutoAttOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query attenuation auto on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Amplitude – set scale/division](#)

ViStatus _VI_FUNC S3302_SetScalePDiv (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set scale/division. Not available when the spectrum analysis mode is linear scale type.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Scale/division (0.1dB~20dB)

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Amplitude – query scale/division](#)

ViStatus _VI_FUNC S3302_QueryScalePDiv (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query scale/division.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Scale/division.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Amplitude – set scale type](#)

ViStatus _VI_FUNC S3302_SetScaleType (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set scale type.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Scale Type

0: logarithmic scale, 1: linear scale.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Amplitude – query scale type](#)

ViStatus _VI_FUNC S3302_QueryScaleType (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query scale type.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Scale Type

0: logarithmic scale, 1: linear scale.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude –set unit

ViStatus _VI_FUNC S3302_SetAmpUnit (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set amplitude unit.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Amplitude unit

DBM(0)	In dBm.
DBMV(1)	Unit: dBmV
DBUV(2)	Unit: dBuV
V(3)	Unit: Volts
W(4)	Unit: Walts

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – query unit

ViStatus _VI_FUNC S3302_QueryAmpUnit (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query amplitude unit.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Amplitude unit

DBM(0)	In dBm.
DBMV(1)	Unit: dBmV
DBUV(2)	Unit: dBuV
V(3)	Unit: Volts
W(4)	Unit: Watts

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – set pre-amplifier on/off

ViStatus _VI_FUNC S3302_SetPreAmpOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set pre-amplifier on/off. When on, the measurement accuracy of small signal can be improved. However, when measuring large power signal, the pre-amplifier is better be off, or measurement AD overflow may occur.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – query pre-amplifier on/off

ViStatus _VI_FUNC S3302_QueryPreAmpOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

Query pre-amplifier on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Bandwidth – set resolution bandwidth

ViStatus _VI_FUNC S3302_SetRBW (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set resolution bandwidth of linear sweep under spectrum analysis mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Frequency (Hz), scope: 1Hz~10MHz, step: 1-3-10.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Bandwidth – query resolution bandwidth](#)

ViStatus _VI_FUNC S3302_QueryRBW (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query resolution bandwidth of linear sweep under spectrum analysis mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Bandwidth – set video bandwidth](#)

ViStatus _VI_FUNC S3302_SetVBW (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set video bandwidth of linear sweep under spectrum analysis mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Frequency (Hz), scope: 1Hz~10MHz, step: 1-3-10.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Bandwidth – query video bandwidth](#)

ViStatus _VI_FUNC S3302_QueryVBW (ViSession instrumentHandle, ViReal64

dVal[])**Function purpose:**

Query video bandwidth of linear sweep under spectrum analysis mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Bandwidth – set resolution bandwidth auto on/off](#)

ViStatus _VI_FUNC S3302_SetAutoRBWOn (ViSession instrumentHandle, ViBoolean bOn)**Function purpose:**

set resolution bandwidth auto on/off. When on, the resolution bandwidth will automatically adapter the resolution bandwidth as per SPAN/RBW based on span.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Bandwidth – query resolution bandwidth auto on/off](#)

ViStatus _VI_FUNC S3302_QueryAutoRBWOn (ViSession instrumentHandle, ViBoolean bOn[])**Function purpose:**

query resolution bandwidth auto on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means

failed.

Bandwidth – set video bandwidth auto on/off

ViStatus _VI_FUNC S3302_SetAutoVBWOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set video bandwidth auto on/off. When on, the video bandwidth will automatically adapter the resolution bandwidth as per SPAN/RBW based on span.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Bandwidth –query video bandwidth auto on/off

ViStatus _VI_FUNC S3302_QueryAutoVBWOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query video bandwidth auto on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Bandwidth – set SPAN/RBW

ViStatus _VI_FUNC S3302_SetSR100 (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set SPAN/RBW under spectrum analysis mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

SPAN/RBW, scope: 1~500.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Bandwidth – query SPAN/RBW

ViStatus _VI_FUNC S3302_QuerySR100 (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query SPAN/RBW under spectrum analysis mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

SPAN/RBW value.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Bandwidth – set RBW/VBW

ViStatus _VI_FUNC S3302_SetRV300 (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set RBW/VBW under spectrum analysis mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

RBW/VBW, scope: 1~100.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Bandwidth – query RBW/VBW

ViStatus _VI_FUNC S3302_QueryRV300 (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query RBW/VBW under spectrum analysis mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

SPAN/RBW value.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Average – set average on/off](#)

ViStatus _VI_FUNC S3302_SetAvgOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set up average switch.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Average – query average on/off](#)

ViStatus _VI_FUNC S3302_QueryAvgOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query average on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Average – set average count](#)

ViStatus _VI_FUNC S3302_SetAvgCount (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set up average frequency.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

average count, scope: 1~1000

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Average – query average count](#)

ViStatus _VI_FUNC S3302_QueryAvgCount (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query average count.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Average.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Average – clear average count](#)

ViStatus _VI_FUNC S3302_ClearAvgCount (ViSession instrumentHandle)

Function purpose:

Clear average count to start from 0.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Average – query current average count](#)

ViStatus _VI_FUNC S3302_QueryCurrentCount (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

query current average count.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

average count.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Detector – set detector type

ViStatus _VI_FUNC S3302_SetDetectorType (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

set detector type.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Detector type

POSitive(0)	Positive Peak
NEGative(1)	Negative Peak
SAMPlE(2)	Sample
NORMal(3)	Standard (Rosenfeld)
AVERage(4)	Average
RMS(5)	Root mean square

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Detector – query detector type

ViStatus _VI_FUNC S3302_QueryDetectorType (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

query detector type.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Detector type

POSitive(0)	Positive
-------------	----------

	Peak
NEGative(1)	Negative Peak
SAMPlE(2)	Sample
NORMal(3)	Standard (Rosenfeld)
AVERage(4)	Average
RMS(5)	Root mean square

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Detector – set detector auto on/off

ViStatus _VI_FUNC S3302_SetAutoDetectorOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

set detector auto on/off. When on, the instrument will automatically select detector type based on different measurement.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Detector – query detector auto on/off

ViStatus _VI_FUNC S3302_QueryAutoDetectorOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query detector auto on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – set sweep type

ViStatus _VI_FUNC S3302_SetSwpType (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set the scanning type. This is an overlapping command. Use **S3302_QueryOPC()** to query if this command is completed before sending other command.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Sweep type.

0: single sweep.

1: continuous sweep.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – query sweep type

ViStatus _VI_FUNC S3302_QuerySwpType (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query sweep type.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Sweep type.

0: single sweep.

0: single sweep.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – trigger single sweep

ViStatus _VI_FUNC S3302_TrigSingleSwp (ViSession instrumentHandle)

Function purpose:

Trigger one single sweep (only valid for single sweep). This command function is an overlapping command function. Use **S3302_QueryOPC()** to query if this command is completed before sending other command.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Sweep – set sweep mode](#)

ViStatus _VI_FUNC S3302_SetSwpMode (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set sweep mode, including linear sweep and list sweep modes.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Sweep mode.

0: linear sweep mode

1: list sweep mode

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Sweep – query sweep mode](#)

ViStatus _VI_FUNC S3302_QuerySwpMode (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query sweep mode, including linear sweep and list sweep modes. The user can edit the list segment to observe the signal of several sweep segments.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Sweep mode.

0: linear sweep mode

1: list sweep mode

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Sweep – set sweep time](#)

ViStatus _VI_FUNC S3302_SetSwpTime (ViSession instrumentHandle, ViReal64

dbVal)**Function purpose:**

Set sweep time under current mode. The sweep time is the time required for selected frequency interval for local oscillator tuning. The sweep time directly affects the time for completing one test, excluding the dead time between one sweep and the next sweep. The sweep time generally changes with the span, resolution bandwidth and video bandwidth. The sweep time is not available when the resolution bandwidth is $\leq 1\text{kHz}$ under spectrum analysis mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Time (ms).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Sweep – query sweep time](#)

ViStatus _VI_FUNC S3302_QuerySwpTime (ViSession instrumentHandle, ViReal64 dbVal[])

Function purpose:

Query sweep time under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Time (ms).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Sweep – set sweep time auto on/off](#)

ViStatus _VI_FUNC S3302_SetAutoSwpTimeOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set sweep time auto on/off. When auto is on, the instrument will use quick sweep speed as possible, or the manual mode is available to increase the sweep time to meet certain measurement needs. The sweep time for manual setting must be equal to or greater than auto sweep time.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Sweep – query sweep time auto on/off](#)

ViStatus _VI_FUNC S3302_QueryAutoSwpTimeOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query sweep time auto on/off

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[List edit – list add default segment](#)

ViStatus _VI_FUNC S3302_ListAddSeg (ViSession instrumentHandle)

Function purpose:

Add default sweep segment to the list edit under current mode.

Starting frequency	1GHz
Stop frequency	2GHz
Sweep Points	51
Resolution Bandwidth	1MHz
Video Bandwidth	30kHz
On/off	Off

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[List edit – list delete segment](#)

ViStatus _VI_FUNC S3302_ListDelSeg (ViSession instrumentHandle, ViInt32 nval)

Function purpose:

Delete segment from list edit under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nval

Segment index

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[List edit – clear list](#)

ViStatus _VI_FUNC S3302_ListClear (ViSession instrumentHandle)

Function purpose:

Delete all list edit segments under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[List edit – add segment](#)

ViStatus _VI_FUNC S3302_ListAdd (ViSession instrumentHandle, ViReal64 startfrequency, ViReal64 stopfrequency, ViInt32 rbw, ViInt32 vbw, ViInt32 sweepPoints, ViInt32 on)

Function purpose:

Add segment in list edit under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

startfrequency

Start frequency (0~44.1GHz)

stopfrequency

Stop frequency (0~44.1GHz)

rbw

Resolution bandwidth (0~10MHz)

vbw

Video bandwidth (0~10MHz)

sweepPoints

Sweep points (51~501)

on

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[List edit – edit segment](#)

ViStatus _VI_FUNC S3302 ListEdit (ViSession instrumentHandle, ViInt32 index, ViReal64 startfrequency, ViReal64 stopfrequency, ViInt32 rbw, ViInt32 vbw, ViInt32 sweepPoints, ViInt32 on)

Function purpose:

Add segment in list edit under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Index

Segment index

startfrequency

Start frequency (0~44.1GHz)

stopfrequency

Stop frequency (0~44.1GHz)

rbw

Resolution bandwidth (0~10MHz)

vbw

Video bandwidth (0~10MHz)

sweepPoints

Sweep points (51~501)

on

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Sweep – set trigger mode](#)

ViStatus _VI_FUNC S3302_SetTrigType (ViSession instrumentHandle,ViInt32 nval)

Function purpose:

Set trigger mode, including free trigger, video trigger and external trigger modes.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Trigger Type.

0: free trigger

1: video trigger

2: external trigger

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Sweep – query trigger mode](#)

ViStatus _VI_FUNC S3302_QueryTrigType (ViSession instrumentHandle,ViInt32 nval[])

Function purpose:

Query trigger mode, including free trigger, video trigger and external trigger modes.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Trigger Type.

0: free trigger

1: video trigger

2: external trigger

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Sweep – set video trigger level](#)

ViStatus _VI_FUNC S3302_SetTrigVideoAMP (ViSession instrumentHandle,

ViReal64 dVal)**Function purpose:**

Set video trigger level.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Trigger level (dBm).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Sweep – query video trigger level](#)

ViStatus _VI_FUNC S3302_QueryTrigVideoAMP (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query video trigger level.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Trigger level (dBm).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Sweep – set external trigger level](#)

ViStatus _VI_FUNC S3302_SetTrigExtraAMP (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

set external trigger level.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Trigger level (mv).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – query external trigger level

ViStatus _VI_FUNC S3302_QueryTrigExtraAMP (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

query external trigger level.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Trigger level (mv).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep- set external trigger polarity

ViStatus _VI_FUNC S3302_SetTrigExtraSlop (ViSession instrumentHandle, ViInt32 nval)

Function purpose:

set external trigger polarity.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Trigger Type.

0: positive

1: negative

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – query external trigger polarity

ViStatus _VI_FUNC S3302_QueryTrigExtraSlop (ViSession instrumentHandle, ViInt32 nval[])

Function purpose:

query external trigger polarity

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Trigger Type.

0: positive

1: negative

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Sweep – set external trigger delay](#)

ViStatus _VI_FUNC S3302_SetTrigExtraDelay (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set external trigger delay.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

trigger delay (us).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Sweep – query external trigger delay](#)

ViStatus _VI_FUNC S3302_QueryTrigExtraDelay (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

query external trigger delay.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

trigger delay (us).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Data – query trace data](#)

ViStatus _VI_FUNC S3302_QueryTraceData (ViSession instrumentHandle, ViInt32 size[], ViReal64 data[], ViInt32 index)

Function purpose:

Query trace data under spectrum analysis mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

size

Size of trace data received.

data

Trace data storage array point. The array should meet the size of trace data received.

index

Trace number can be set as 1, 2 or 3, indicating query trace data of trace 1, 2, or 3.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Trace - set trace state](#)

ViStatus _VI_FUNC S3302_SetTraceType (ViSession instrumentHandle, ViInt32 nVal, ViInt32 nTrace)

Function purpose:

Set trace state

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Trace types.

0: refresh trace

1: maximum hold

2: minimum hold

3: hold trace

4: conceal trace

nTrace

Trace number can be set as 1, 2 or 3, indicating query trace data of trace 1, 2, or 3.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Trace – query trace state](#)

ViStatus _VI_FUNC S3302_QueryTraceType (ViSession instrumentHandle, ViInt32

nVal[], ViInt32 nTrace)**Function purpose:**

Query trace state.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Trace types.

0: refresh trace

1: maximum hold

2: minimum hold

3: hold trace

4: conceal trace

nTrace

Trace number can be set as 1, 2 or 3, indicating query trace data of trace 1, 2, or 3.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Limit – set alarm on/off](#)

ViStatus _VI_FUNC S3302_SetAlarmOn (ViSession instrumentHandle, ViBoolean bOn)**Function purpose:**

Set limit alarm on/off. If the alarm is on, when the limit test on/off is on and the test fails, the alarm will give “beep” tone pip after each sweep.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Limit – query alarm on/off](#)

ViStatus _VI_FUNC S3302_QueryAlarmOn (ViSession instrumentHandle, ViBoolean bOn[])**Function purpose:**

Query limit alarm on/off. If the alarm is on, when the limit test on/off is on and the test fails,

the alarm will give “beep” tone pip after each sweep.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Limit – set lower limit display on/off](#)

ViStatus _VI_FUNC S3302_SetLowLmtDispOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

set lower limit display on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Limit – query lower limit display on/off](#)

ViStatus _VI_FUNC S3302_QueryLowLmtDispOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query lower limit display on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Limit – set upper limit display on/off

ViStatus _VI_FUNC S3302_SetUppLmtDispOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

set upper limit display on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Limit – query upper limit display on/off

ViStatus _VI_FUNC S3302_QueryUppLmtDispOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query upper limit display on/off

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Limit – set lower limit test on/off

ViStatus _VI_FUNC S3302_SetLowLmtTestOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

set lower limit test on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Limit – query lower limit test on/off](#)

ViStatus _VI_FUNC S3302_QueryLowLmtTestOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query lower limit test on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Limit – set upper limit test on/off](#)

ViStatus _VI_FUNC S3302_SetUppLmtTestOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

set upper limit test on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Limit – query upper limit test on/off](#)

ViStatus _VI_FUNC S3302_QueryUppLmtTestOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query upper limit test on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Limit – set lower limit margin](#)

ViStatus _VI_FUNC S3302_SetLowLmtMargin (ViSession instrumentHandle, ViReal64 dbVal)

Function purpose:

set lower limit margin.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Margin (0dB~40dB).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Limit – query lower limit margin](#)

ViStatus _VI_FUNC S3302_QueryLowLmtMargin (ViSession instrumentHandle, ViReal64 dbVal[])

Function purpose:

Query lower limit margin.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Margin

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Limit – set upper limit margin](#)

ViStatus _VI_FUNC S3302_SetUppLmtMargin (ViSession instrumentHandle, ViReal64 dbVal)

Function purpose:

set upper limit margin.

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Margin (-40dB~0dB).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Limit – query upper limit margin](#)

ViStatus _VI_FUNC S3302_QueryUppLmtMargin (ViSession instrumentHandle, ViReal64 dbVal[])

Function purpose:

query upper limit margin.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Margin

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Limit – lower limit add default point](#)

ViStatus _VI_FUNC S3302_LowLmtAddPt (ViSession instrumentHandle)

Function purpose:

lower limit add default point.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Limit – lower limit delete default point](#)

ViStatus _VI_FUNC S3302_LowLmtDelPt (ViSession instrumentHandle)

Function purpose:

lower limit delete default point.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Limit – clear all lower limit points](#)

ViStatus _VI_FUNC S3302_LowLmtClear (ViSession instrumentHandle) Function

purpose:

Clear all edit points in lower limit.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Limit – lower limit edit point](#)

ViStatus _VI_FUNC S3302_LowerLimitEdit (ViSession instrumentHandle, ViInt32 index, ViReal64 frequency, ViReal64 amplitude)

Function purpose:

Set lower limit edit point.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

index

Limit point index.

frequency

Frequency (Hz)(0~44.1GHz).

amplitude

Amplitude (dBm)(-174~50dBm).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Limit – upper limit add default point](#)

ViStatus _VI_FUNC S3302_UppLmtAddPt (ViSession instrumentHandle)

Function purpose:

upper limit add default point.

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Limit – upper limit delete current point](#)

ViStatus _VI_FUNC S3302_UppLmtDelPt (ViSession instrumentHandle) Function

purpose:

Upper limit delete current point.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Limit – upper limit point clear](#)

ViStatus _VI_FUNC S3302_UppLmtClear (ViSession instrumentHandle) Function

purpose:

Clear all edit points in the upper limit.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Limit – lower limit edit point](#)

ViStatus _VI_FUNC S3302_UpperLimitEdit (ViSession instrumentHandle, ViInt32 index, ViReal64 frequency, ViReal64 amplitude)

Function purpose:

Set upper limit edit point.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

index

Limit point index.

frequency

Frequency (Hz)(0~44.1GHz).

amplitude

Amplitude (dBm)(-174~50dBm).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Marker – set market state

ViStatus _VI_FUNC S3302_SetMkrState (ViSession instrumentHandle, ViInt32 nVal, ViInt32 nState)

Function purpose:

set marker state under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nState

Marker state.

0: marker off.

1: normal marker on.

2: offset marker on.

nVal

Marker index, 1~6 available.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Marker – query marker state

ViStatus _VI_FUNC S3302_QueryMkrState (ViSession instrumentHandle, ViInt32 nVal, ViInt32 nState[])

Function purpose:

Query marker state under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nState

Marker state.

0: marker off.

1: normal marker on.

2: offset marker on.

nVal

Marker index, 1~6 available.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Marker- activate marker](#)

ViStatus _VI_FUNC S3302_SetMkrActive (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Activate marker under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Marker index, 1~6 available.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Marker- marker function \(marker ->\)](#)

ViStatus _VI_FUNC S3302_SetMkrTo (ViSession instrumentHandle, ViInt32 nVal, ViInt32 nSetIdx)

Function purpose:

Set marker function under current mode (marker -> for spectrum analysis mode).

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nSetIdx

Instrument mode	nSetIdx	Function
Spectrum Analysis (non-zero span)	0	Marker -> start frequency (set marker frequency as start frequency)
	1	Marker -> stop frequency (set marker frequency as stop frequency)
	2	Marker -> center frequency (set marker frequency as center frequency)
	3	Marker -> step frequency (set marker frequency as

		step frequency)
Spectrum Analysis (zero span)	0	Marker -> start frequency (set marker index as minimum index)
	1	Marker -> stop frequency (set marker index as maximum index)
	2	Marker -> center frequency (set marker index as center index)
	3	Marker -> step frequency (set marker frequency as step frequency)

nVal

Marker index, 1~6 available.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Marker – disable all markers](#)

ViStatus _VI_FUNC S3302_DisableAllMarkers (ViSession instrumentHandle)

Function purpose:

Disable all markers under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Marker –set marker X value](#)

ViStatus _VI_FUNC S3302_MoveMarker (ViSession instrumentHandle, ViInt32 nVal, ViReal64 dbVal,)

Function purpose:

set marker X value under current mode, when marker is an offset marker, the X value can be negative.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Marker X value, time in ms and frequency in Hz.

Instrument mode	Parameter unit
-----------------	----------------

Spectrum Analysis (non-zero span)	Hz
Spectrum Analysis (zero span)	us

nVal

Marker index, 1~6 available.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Marker – query marker X value](#)

ViStatus _VI_FUNC S3302_QueryMarker (ViSession instrumentHandle, ViInt32 markerIndex, ViReal64 markerPosition[], ViReal64 markerAmplitude[])

Function purpose:

Query marker X value and Y value under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

markerPosition

Marker X value, time in ms and frequency in Hz.

Instrument mode	Parameter unit
Spectrum Analysis (non-zero span)	Hz
Spectrum Analysis (zero span)	us

markerIndex

Marker index, 1~6 available.

markerAmplitude

Marker Y value.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Marker - search](#)

ViStatus _VI_FUNC S3302_SetMkrSearch (ViSession instrumentHandle, ViInt32 nVal, ViInt32 type)

Function purpose:

Move marker to maximum, minimum, peak, secondary peak, left adjacent peak and right adjacent peak.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Marker index can be set from 1 to 6 1~6 available, indicating marker 1, 2, 3 and 4.

type

Search type.

1 Maximum value

2 Minimum value

3 Peak

4 Secondary peak 5

left adjacent peak

6 right adjacent peak

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Marker –set marker count on](#)

ViStatus _VI_FUNC S3302_SetMkrCountOn (ViSession instrumentHandle, ViInt32 nVal, ViBoolean bOn)

Function purpose:

Set the marker count state under current mode and set marker will be switched to normal marker state.

Notes: Only one marker counter can be enabled now.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

State, 0 is off and 1 is on.

nVal

Marker index, 1~6 available.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Marker –query marker count on](#)

ViStatus _VI_FUNC S3302_QueryMkrCountOn (ViSession instrumentHandle, ViInt32 nVal, ViBoolean bOn[])

Query the marker count state under current mode

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

State, 0 is off and 1 is on.

nVal

Marker index, 1~6 available.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Marker – query marker count frequency

ViStatus _VI_FUNC S3302_QueryMkrCountFreq (ViSession
instrumentHandle, ViInt32 nVal,
ViReal64 dbVal[])

Function purpose:

Query marker count frequency (invalid if the count is disabled or does not start count).

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Marker index, 1~6 available.

dbVal

Returned counter frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Marker – set noise marker on

ViStatus _VI_FUNC S3302_SetMkrNoiseOn (ViSession instrumentHandle, ViInt32
nVal, ViBoolean bOn)

Function purpose:

Set noise marker on under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

State, 0 is off and 1 is on.

nVal

Marker index, 1~6 available.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Marker – query noise marker on](#)

ViStatus _VI_FUNC S3302_QueryMkrNoiseOn (ViSession instrumentHandle, ViInt32 nVal, ViBoolean bOn[])

Function purpose:

Query noise marker on under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

State, 0 is off and 1 is on.

nVal

Marker index, 1~6 available.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Marker – set peak track on/off](#)

ViStatus _VI_FUNC S3302_SetPeakTrack (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set peak track on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

State, 0 is off and 1 is on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Marker – query peak track on/off](#)

ViStatus _VI_FUNC S3302_QueryPeakTrack (ViSession instrumentHandle, ViBoolean bOn[])

Query peak track on/off under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

State, 0 is off and 1 is on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – set function measurement

ViStatus _VI_FUNC S3302_SetMeasFunc (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

set function measurement type, or the function measurement type can be directly set by the function measurement state. Only one function measurement is allowed at a time.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Function measurement type.

Parameter Setting	Measuring type
NONE(0)	Normal spectrum measurement
FST(1)	Field strength measurement
CHP(2)	Channel power meter
OBW(3)	Occupied bandwidth measurement
ACPR(4)	Adjacent channel power ratio measurement
DEMODO(5)	Audio demodulation measurement
EM(6)	Emission mask measurement
CNR(7)	Carrier to noise ratio measurement
IQ(8)	IQ capture measurement

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – query function measurement.

ViStatus _VI_FUNC S3302_QueryMeasFunc (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

query function measurement type.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Function measurement type.

Parameter Setting	Measuring type
NONE(0)	Normal spectrum measurement
FST(1)	Field strength measurement
CHP(2)	Channel power meter
OBW(3)	Occupied bandwidth measurement
ACPR(4)	Adjacent channel power ratio measurement
DEMOD(5)	Audio demodulation measurement
EM(6)	Emission mask measurement
CNR(7)	Carrier to noise ratio measurement
IQ(8)	IQ capture measurement

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – turn of measurement

ViStatus _VI_FUNC S3302_SetMeasFuncAOff (ViSession instrumentHandle)

Function purpose:

Turn off current function measurement and switch to normal spectrum measurement.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means

failed. [Measurement – field strength – set antenna factor off](#)

ViStatus _VI_FUNC S3302_SetAntOff (ViSession instrumentHandle)

Function purpose:

Set antenna factor loading off and set as antenna factor free state.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – field strength – set field strength on/off](#)

ViStatus _VI_FUNC S3302_SetFstOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

set field strength function measurement on/off, or use function **S3302_SetMeasFunc()** to enable **(by enabling this function measurement, other function measurement will be disabled).**

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – field strength – query field strength on/off](#)

ViStatus _VI_FUNC S3302_QueryFstOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query field strength on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – field strength – edit antenna factor – add default point](#)

ViStatus _VI_FUNC S3302_AntennaAddDefault (ViSession instrumentHandle) Function

purpose:

edit antenna factor to add default point. Frequency: 1GHz antenna factor value: 0dB

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – field strength – edit antenna factor – delete point](#)

ViStatus _VI_FUNC S3302_AntennaDelete (ViSession instrumentHandle, ViInt32 index)

Function purpose:

edit antenna factor to delete point.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

index

Point index.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – field strength – edit antenna factor – edit point](#)

ViStatus _VI_FUNC S3302_AntennaEdit (ViSession instrumentHandle, ViInt32 index, ViReal64 frequency, ViReal64 factor)

Function purpose:

edit antenna factor to edit point.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

index

Point index.

frequency

Frequency (0~44.1GHz)

factor

Antenna factor value (-200~200dB)

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – field strength – edit antenna factor – add point](#)

ViStatus _VI_FUNC S3302_AntennaAdd (ViSession instrumentHandle, ViReal64 frequency, ViReal64 factor)

Function purpose:

edit antenna factor to add point.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

frequency

Frequency (0~44.1GHz)

factor

Antenna factor value (-200~200dB)

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – channel power – set channel power on/off](#)

ViStatus _VI_FUNC S3302_SetCHPOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

set channel power on/off, or use function **S3302_SetMeasFunc()** to enable **(by enabling this function measurement, other function measurement will be disabled)**.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – channel power – query channel power state](#)

ViStatus _VI_FUNC S3302_QueryCHPOn (ViSession instrumentHandle, ViBoolean

bOn[])**Function purpose:**

Query channel power state.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – channel power – set channel power bandwidth](#)

ViStatus _VI_FUNC S3302_SetCHPIBW (ViSession instrumentHandle, ViReal64 dbVal)**Function purpose:**

set channel power bandwidth for channel power function measurement under spectrum analysis mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency (Hz), scope: 100Hz~44.1GHz

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – channel power – query channel power bandwidth](#)

ViStatus _VI_FUNC S3302_QueryCHPIBW (ViSession instrumentHandle, ViReal64 dbVal[])**Function purpose:**

query channel power bandwidth for channel power function measurement under spectrum analysis mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – channel power – query channel power value

ViStatus _VI_FUNC S3302_QueryTPWR (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query channel power value for channel power function measurement under spectrum analysis mode (**valid when channel power is on and after swept**).

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Power value (dBm).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – channel power – query channel power density

ViStatus _VI_FUNC S3302_QueryPSDR (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query channel power density for channel power function measurement under spectrum analysis mode (**valid when channel power is on and after swept**).

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Power density (dBm/Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – occupied bandwidth – set occupied bandwidth state

ViStatus _VI_FUNC S3302_SetOBWOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set occupied bandwidth function measurement state, or use function **S3302_SetMeasFunc()** (**Other functional measurements will be disabled after this function is enabled**).

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – occupied bandwidth – query occupied bandwidth state](#)

ViStatus _VI_FUNC S3302_QueryOBWOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query occupied bandwidth state.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – occupied bandwidth – set measurement method](#)

ViStatus _VI_FUNC S3302_SetOBWMethod (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set occupied bandwidth measurement method. The percentage measurement method is to obtain the x% bandwidth of total power of all span. The XdB measurement method is to obtain the xdB bandwidth less than maximum power value at both sides.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Measurement Method.

0: percentage measurement method

1: XdB measurement method

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – occupied bandwidth – query measurement method

ViStatus _VI_FUNC S3302_QueryOBWMMethod (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query occupied bandwidth measurement method.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Measurement Method.

0: percentage measurement method

1: XdB measurement method

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – occupied bandwidth – set percentage

ViStatus _VI_FUNC S3302_SetOBWPpow (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set occupied bandwidth percentage.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Percentage, scope: 10.00%~99.99%.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – occupied bandwidth – query percentage

ViStatus _VI_FUNC S3302_QueryOBWPpow (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query occupied bandwidth percentage.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

fVal

Percentage.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – occupied bandwidth – set XdB

ViStatus _VI_FUNC S3302_SetOBWXdB (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set occupied bandwidth XdB, valid when the measurement method is XdB.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

XdB value (dB), scope: -100dB~-0.1dB

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – occupied bandwidth – query XdB

ViStatus _VI_FUNC S3302_QueryOBWXdB (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query occupied bandwidth XdB.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

XdB value (dB).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – occupied bandwidth – query occupied bandwidth

ViStatus _VI_FUNC S3302_QueryOBWOBW (ViSession instrumentHandle, ViReal64 dbVal[])

Function purpose:

query occupied bandwidth

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

occupied bandwidth (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – audio demodulation – set demodulation state](#)

ViStatus _VI_FUNC S3302_SetDemodOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set AF/FM function measurement state, or use function **S3302_SetMeasFunc()**(**by enabling this function measurement, other functional measurement will be disabled**).

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – audio demodulation – query demodulation state](#)

ViStatus _VI_FUNC S3302_QueryDemodOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

Query AM/FM demodulation state.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – audio demodulation – set demodulation mode](#)

ViStatus _VI_FUNC S3302_SetDMMode (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set demodulation mode. The intermittent mode refers to the mode which demodulation will

be stopped as per the demodulation time after sweeping one screen of data before sweeping the next screen of data, and will continue this process over and over again; continuous mode refers to the mode that the instrument will continuously demodulate after sweeping one screen of data and will not sweep data again.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Demodulation mode.

0: intermittent mode

1: continuous mode

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – audio demodulation – query demodulation mode](#)

ViStatus _VI_FUNC S3302_QueryDMode (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query demodulation mode. The intermittent mode refers to the mode which demodulation will be stopped as per the demodulation time after sweeping one screen of data before sweeping the next screen of data, and will continue this process over and over again; continuous mode refers to the mode that the instrument will continuously demodulate after sweeping one screen of data and will not sweep data again.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Demodulation mode.

0: intermittent mode

1: continuous mode

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – audio demodulation – set demodulation type](#)

ViStatus _VI_FUNC S3302_SetDType (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set demodulation type, frequency modulation, amplitude demodulation, upper sideband and lower sideband available.

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Demod types.

0: frequency modulation

1: amplitude demodulation

2: upper sideband

3: lower sideband

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – audio demodulation – query demodulation type](#)

ViStatus _VI_FUNC S3302_QueryDType (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query demodulation type.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Demod type.

0: frequency modulation

1: amplitude demodulation

2: upper sideband

3: lower sideband

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – audio demodulation – set demodulation time](#)

ViStatus _VI_FUNC S3302_SetDTime (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set demodulation time. This **parameter** is valid when the demodulation mode is intermittent mode and the demodulation time refers to the time for staying in demodulation state after one sweep.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Demodulation time (ms), scope: 1us~400s

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – audio demodulation – query demodulation time](#)

ViStatus _VI_FUNC S3302_QueryDTime (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query demodulation time.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Demodulation time (ms).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – audio demodulation – set volume](#)

ViStatus _VI_FUNC S3302_SetVolume (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set loudspeaker volume of demodulation.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Demodulation volume, scope: 0~100.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – audio demodulation – query volume](#)

ViStatus _VI_FUNC S3302_QueryVolume (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query loudspeaker volume of demodulation.

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Demodulation volume.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – ACPR – set ACPR on/off](#)

ViStatus _VI_FUNC S3302_SetACPROn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

set ACP on, or use function **S3302_SetMeasFunc()**(Other functional measurements will be disabled after this function is enabled).

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – ACPR – query ACPR on/off](#)

ViStatus _VI_FUNC S3302_QueryACPROn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

Query ACP on.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – ACPR – set main channel bandwidth](#)

ViStatus _VI_FUNC S3302_SetACPRMainChBW (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set main channel bandwidth for ACP function measurement under spectrum analysis mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Frequency (Hz), scope: 300Hz~20MHz.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – ACPR – query main channel bandwidth](#)

ViStatus _VI_FUNC S3302_QueryACPRMainChBW (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query main channel bandwidth for ACP function measurement under spectrum analysis mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – ACPR – set adjacent channel bandwidth](#)

ViStatus _VI_FUNC S3302_SetACPRAdjChBW (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set adjacent channel bandwidth for ACPR function measurement under spectrum analysis mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Frequency (Hz), scope: 300Hz~20MHz.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means

failed.

Measurement – ACPR – query adjacent channel bandwidth

ViStatus _VI_FUNC S3302_QueryACPRAdjChBW (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query adjacent channel bandwidth for ACP function measurement under spectrum analysis mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – ACPR – set channel space

ViStatus _VI_FUNC S3302_SetACPRSpace (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set channel space for ACPR function measurement under spectrum analysis mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Frequency (Hz), scope: 0Hz~45MHz

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – ACPR – query channel space

ViStatus _VI_FUNC S3302_QueryACPRSpace (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query channel space for ACP function measurement under spectrum analysis mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – ACPR – set limit test on](#)

ViStatus _VI_FUNC S3302_SetACPRLmtTestOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

set adjacent channel power limit test on.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – ACPR – query limit test on](#)

ViStatus _VI_FUNC S3302_QueryACPRLmtTestOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

Query adjacent channel power limit test on.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – ACPR – set lower adjacent channel limit](#)

ViStatus _VI_FUNC S3302_SetACPRLowLmt (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

set lower adjacent channel limit of ACPR.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Adjacent channel limit (dB), scope: -200dB~200dB.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – ACPR – query lower adjacent channel limit](#)

ViStatus _VI_FUNC S3302_QueryACPRLowLmt (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query lower adjacent channel limit of ACPR.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Adjacent channel limit (dB), scope: -200dB~200dB.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – ACPR – set upper adjacent channel limit](#)

ViStatus _VI_FUNC S3302_SetACPRUppLmt (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set upper adjacent channel limit of ACPR.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Adjacent channel limit (dB), scope: -200dB~200dB.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – ACPR – query upper adjacent channel limit](#)

ViStatus _VI_FUNC S3302_QueryACPRUppLmt (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query upper adjacent channel limit of ACPR.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Adjacent channel limit (dB), scope: -200dB~200dB.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – ACPR – query upper ACPR](#)

ViStatus _VI_FUNC S3302_QueryACPRRateUpper (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

query upper ACPR of ACPR.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

ACPR (dBc).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – ACPR – query lower ACPR](#)

ViStatus _VI_FUNC S3302_QueryACPRRateLower (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query lower ACPR of ACPR.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

ACPR (dBc).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – emission mask – set emission mask on](#)

ViStatus _VI_FUNC S3302_SetEMOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

set emission mask on, or use function **S3302_SetMeasFunc()** to open this option (**other functional measurements will be disabled after this measurement is enabled**).

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – emission mask – query emission mask on

ViStatus _VI_FUNC S3302_QueryEmOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

Query emission mask on

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – emission mask – set reference channel bandwidth

ViStatus _VI_FUNC S3302_SetEMRefCHBW (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

set reference channel bandwidth of emission mask.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Frequency (Hz), scope: 1kHz~44.1GHz.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – emission mask – query reference channel bandwidth

ViStatus _VI_FUNC S3302_QueryEMRefCHBW (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query reference channel bandwidth of emission mask.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – emission mask – set reference power type

ViStatus _VI_FUNC S3302_SetEMRefPowerType (ViSession instrumentHandle, ViInt32 type)

Function purpose:

set reference power type for emission mask.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

type

Reference power type, 0 is peak and 1 is channel.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – emission mask – query reference power type

ViStatus _VI_FUNC S3302_QueryEMRefPowerType (ViSession instrumentHandle, ViInt32 type[])

Function purpose:

Query reference power type for emission mask.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

type

Reference power type, 0 is peak and 1 is channel.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – emission mask – set peak marker on in emission mask](#)

ViStatus _VI_FUNC S3302_SetEMPeakMarkerOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

set peak marker on in emission mask.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – emission mask – query peak marker on of emission mask](#)

ViStatus _VI_FUNC S3302_QueryEMPeakMarkerOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

Query peak marker on of emission mask.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – emission mask – query if emission mask fails](#)

ViStatus _VI_FUNC S3302_QueryEMFail (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query if emission mask fails.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: pass, 1: fail.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement –CNR – set CNR state

ViStatus _VI_FUNC S3302_SetCNROn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

set CNR state, or use function **S3302_SetMeasFunc()** to open this item **(by opening this function measurement, other function measurement will be turned off).**

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement –CNR – query CNR state

ViStatus _VI_FUNC S3302_QueryCNROn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

Query CNR state

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement –CNR – set CNR carrier bandwidth

ViStatus _VI_FUNC S3302_SetCNRCBW (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set CNR carrier bandwidth.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Frequency (Hz), scope: 300Hz~20MHz.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement –CNR – query CNR carrier bandwidth](#)

ViStatus _VI_FUNC S3302_QueryCNRCBW (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query CNR carrier bandwidth.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement –CNR – set CNR noise bandwidth](#)

ViStatus _VI_FUNC S3302_SetCNRNBW (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set CNR noise bandwidth.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Frequency (Hz), scope: 300Hz~20MHz.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement –CNR – query CNR noise bandwidth

ViStatus _VI_FUNC S3302_QueryCNRNBW (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query CNR noise bandwidth.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement –CNR – set CNR frequency offset

ViStatus _VI_FUNC S3302_SetCNRSpace (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set CNR frequency offset.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Frequency value (unit: Hz), scope: 0Hz~100MHz.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement –CNR – query CNR frequency offset

ViStatus _VI_FUNC S3302_QueryCNRSpace (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query CNR frequency offset.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement –CNR – query CNR measurement result](#)

ViStatus _VI_FUNC S3302_QueryCNRValue (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query CNR measurement result.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

measurement result (dB).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – IQ capture – set IQ capture state](#)

ViStatus _VI_FUNC S3302_SetIQcaptureOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

set IQ capture state.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – IQ capture – query IQ capture state](#)

ViStatus _VI_FUNC S3302_QueryIQCaptureOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query IQ capture state.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – IQ capture – start capture](#)

ViStatus _VI_FUNC S3302_SetIQCaptureStart (ViSession instrumentHandle)

Function purpose:

Set IQ capture to start capture.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – IQ capture – stop capture](#)

ViStatus _VI_FUNC S3302_SetIQCaptureStop (ViSession instrumentHandle) Function

purpose:

Set IQ capture to stop capture.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – IQ capture – set capture time](#)

ViStatus _VI_FUNC S3302_SetIQCaptureTime (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

set capture time for IQ capture.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Capture time (us)

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – IQ capture – query capture time

ViStatus _VI_FUNC S3302_QueryIQCaptureTime (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

query capture time of IQ capture.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Capture time (us)

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – IQ capture – set IQ capture mode

ViStatus _VI_FUNC S3302_SetIQCaptureMode (ViSession instrumentHandle, ViInt32 mode)

Function purpose:

Set IQ capture mode, including single capture and continuous capture.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

mode

0: single capture, 1: continuous capture.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – IQ capture – query IQ capture mode

ViStatus _VI_FUNC S3302_QueryIQCaptureMode (ViSession instrumentHandle, ViInt32 mode[])

Function purpose:

Query IQ capture mode, including single capture and continuous capture.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

mode

0: single capture, 1: continuous capture.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – IQ capture – set sampling rate

ViStatus _VI_FUNC S3302_SetIQCaptureSample (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set sampling rate of IQ capture.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Sample rate.

Capture sampling rate	Capture bandwidth
12.5MHz	10MHz
5MHz	4MHz
1.25MHz	1MHz
500kHz	400kHz
125kHz	100kHz
50kHz	40kHz

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – IQ capture – query sampling rate

ViStatus _VI_FUNC S3302_QueryIQCaptureSample (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query IQ capture sampling rate.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Sample rate.

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – IQ capture – set IQ capture storage name

ViStatus **_VI_FUNC** **S3302_SetIQCaptureFilename(ViSession instrumentHandle, char* name)**

Function purpose:

Set IQ capture file storage name.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

name

File storage name

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – IQ capture – set trigger mode

ViStatus **_VI_FUNC** **S3302_SetIQCaptureTrigMode (ViSession instrumentHandle, ViInt32 nval)**

Function purpose:

Set trigger mode, including free trigger and external trigger.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Trigger Type.

0: free trigger

1: external trigger

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – IQ capture – query trigger mode

ViStatus **_VI_FUNC** **S3302_QueryIQCaptureTrigMode (ViSession instrumentHandle, ViInt32 nval[])**

Function purpose:

Query trigger mode, including free trigger and external trigger.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Trigger Type.

0: free trigger

1: external trigger

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – IQ capture – set external trigger polarity

ViStatus **_VI_FUNC** **S3302_SetIQCaptureTrigSlop** **(ViSession**
instrumentHandle, ViInt32 nval)

Function purpose:

set external trigger polarity.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Trigger Type.

0: positive

1: negative

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – IQ capture – query external trigger polarity

ViStatus **_VI_FUNC** **S3302_QueryIQCaptureTrigSlop** **(ViSession**
instrumentHandle, ViInt32 nval[])

Function purpose:

query external trigger polarity

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Trigger Type.

0: positive

1: negative

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – IQ capture – set external trigger delay

ViStatus _VI_FUNC S3302_SetIQCaptureTrigDelay (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set external trigger delay for IQ capture.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

trigger delay (us).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – IQ capture – query external trigger delay

ViStatus _VI_FUNC S3302_QueryIQCaptureTrigDelay (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query external trigger delay for IQ capture.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

trigger delay (us).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – IQ capture – set external trigger amplitude

ViStatus _VI_FUNC S3302_SetIQCaptureTrigAMP (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set external trigger amplitude for IQ capture.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Trigger level (volt).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – IQ capture – query external trigger amplitude](#)

ViStatus _VI_FUNC S3302_QueryIQCaptureTrigAMP (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

query external trigger amplitude for IQ capture.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Trigger level (volt).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measure- Generator- Set the generator ON/OFF](#)

ViStatus _VI_FUNC S3302_SetGenMode (ViSession instrumentHandle, ViInt32 bOn)

Purpose of function:

Set the generator ON/OFF.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

bOn

0: OFF; 1: ON.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Measure- Generator- Query the generator ON/OFF](#)

ViStatus _VI_FUNC S3302_QueryGenMode (ViSession instrumentHandle, ViInt32 bOn[]) **Purpose of function:**

Query the generator ON/OFF.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

bOn

0: OFF; 1: ON.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Measure- Generator- Set the frequency offset](#)

**ViStatus _VI_FUNC S3302_SetGenFreqOffset (ViSession instrumentHandle,
ViReal64 dVal)**

Purpose of function:

Set the generator frequency offset.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

dVal

Frequency offset.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Measure- Generator- Query the frequency offset](#)

**ViStatus _VI_FUNC S3302_QueryGenFreqOffset (ViSession instrumentHandle,
ViReal64 dVal[])**

Purpose of function:

Query the generator frequency offset.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

dVal

Frequency offset.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Measure- Generator- Set the generator frequency](#)

ViStatus _VI_FUNC S3302_SetGenFreq (ViSession instrumentHandle, ViReal64 dVal)

Purpose of function:

Set the generator frequency.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

dVal

Frequency.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Measure- Generator- Query the generator frequency](#)

ViStatus _VI_FUNC S3302_QueryGenFreq (ViSession instrumentHandle, ViReal64

dVal[]) **Purpose of function:**

Query the generator frequency.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

dVal

Frequency.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Measure- Generator- Set the generator power](#)

ViStatus _VI_FUNC S3302_SetGenPower (ViSession instrumentHandle, ViReal64 dVal)

Purpose of function:

Set the generator power.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

dVal

Power.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Measure- Generator- Query the generator power](#)

ViStatus _VI_FUNC S3302_QueryGenPower (ViSession instrumentHandle, ViReal64 dVal[])

Purpose of function:

Query the generator power.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

dVal

Power.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Measure- Generator- Set the generator power offset](#)

ViStatus _VI_FUNC S3302_SetGenPowerOffset (ViSession instrumentHandle, ViReal64 dVal)

Purpose of function:

Set the generator power offset.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

dVal

Power offset.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Measure- Generator- Query the generator power offset](#)

ViStatus _VI_FUNC S3302_QueryGenPowerOffset (ViSession instrumentHandle, ViReal64 dVal[])

Purpose of function:

Query the generator power offset.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

dVal

Power offset.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Measure- Generator- Set the generator normalization ON/OFF](#)

ViStatus _VI_FUNC S3302_SetGenNormzStat (ViSession instrumentHandle, ViInt32

bOn) Purpose of function:

Set the generator normalization ON/OFF.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

bOn

0: OFF; 1: ON.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Measure- Generator- Query the generator normalization ON/OFF](#)

**ViStatus _VI_FUNC S3302_QueryGenNormzStat (ViSession
instrumentHandle, ViInt32 bOn[])**

Purpose of function:

Query the generator normalization ON/OFF.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

bOn

0: OFF; 1: ON.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Measure- Generator- Set the generator normalization reference level](#)

**ViStatus _VI_FUNC S3302_SetGenNormzRef (ViSession instrumentHandle,
ViReal64 dVal)**

Purpose of function:

Set the generator normalization reference level.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

dVal

Reference level.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Measure- Generator- Query the generator normalization reference level](#)

**ViStatus _VI_FUNC S3302_QueryGenNormzRef (ViSession
instrumentHandle, ViReal64 dVal[])**

Purpose of function:

Query the generator normalization reference level.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

dVal

Reference level.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Measure- Generator- Set the generator normalization reference position](#)

ViStatus _VI_FUNC S3302_SetGenNormzRefPos (ViSession instrumentHandle, ViInt32 nVal)

Purpose of function:

Set the generator normalization reference position.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

nVal

Reference position.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Measure- Generator- Query the generator normalization reference position](#)

ViStatus _VI_FUNC S3302_QueryGenNormzRefPos (ViSession instrumentHandle, ViInt32 nVal[])

Purpose of function:

Query the generator normalization reference position.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

nVal

Reference position.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Measure- Generator- Set the generator normalization scale/division](#)

ViStatus _VI_FUNC S3302_SetGenNormzPdiv (ViSession instrumentHandle, ViReal64

dVal)

Purpose of function:

Set the generator normalization scale/division.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

dVal

Scale/division.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Measure- Generator- Query the generator normalization scale/division](#)

**ViStatus _VI_FUNC S3302_QueryGenNormzPdiv (ViSession
instrumentHandle, ViReal64 dVal[])**

Purpose of function:

Query the generator normalization scale/division.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

dVal

Scale/division.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Measure- Generator- Set the generator reference trace ON/OFF](#)

**ViStatus _VI_FUNC S3302_SetGenRefTraceStat (ViSession instrumentHandle,
ViInt32 bOn)**

Purpose of function:

Set the generator reference trace ON/OFF.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

bOn

0: OFF; 1: ON.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

Measure- Generator- Query the generator reference trace ON/OFF

ViStatus _VI_FUNC S3302_QueryGenRefTraceStat (ViSession instrumentHandle, ViInt32 bOn[])

Purpose of function:

Query the generator reference trace ON/OFF.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

bOn

0: OFF; 1: ON.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Zero calibration – execute](#)

ViStatus _VI_FUNC S3302_SetAligNow (ViSession instrumentHandle)

Function purpose:

Zero calibration (**please do not repeatedly calibrate during calibration**). This command is an overlapping command. Use **S3302_QueryOPC()** to query if this command is completed before sending other commands.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[File – recall antenna factor](#)

ViStatus _VI_FUNC S3302_LoadAntFile (ViSession instrumentHandle, char chStr[])

Function purpose:

Recall antenna factor in field strength function measurement under spectrum analysis mode (**the command will be invalid if the file does not exist and be only valid for current storage location**) so that this antenna factor can be weighted when opening relevant measurement function.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

File – store antenna factor

ViStatus _VI_FUNC S3302_StoreAntFile (ViSession instrumentHandle, char chStr[])

Function purpose:

Store antenna factor edited in field strength function measurement under spectrum analysis mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

File – delete antenna factor

ViStatus _VI_FUNC S3302_DelAntFile (ViSession instrumentHandle, char chStr[])

Function purpose:

delete antenna factor file.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

File – delete all antenna factors

ViStatus _VI_FUNC S3302_DelAllAntFile (ViSession instrumentHandle) Function

purpose:

delete all antenna factor files.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means

failed.

File – emission mask recall limit line

ViStatus _VI_FUNC S3302_LoadLmtFile (ViSession instrumentHandle, char chStr[])

Function purpose:

Recall limit line as the mask for emission mask measurement.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

File – store list to file

ViStatus _VI_FUNC S3302_StoreList (ViSession instrumentHandle, char chStr[])

Function purpose:

Store the list as list file (the command will be invalid if the file does not exist and will only be valid for current storage location).

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

File – recall list file

ViStatus _VI_FUNC S3302_LoadList (ViSession instrumentHandle, char chStr[])

Function purpose:

Recall list file **(the command will be invalid if the file does not exist and be only valid for current storage location).**

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[File – delete list file](#)

ViStatus _VI_FUNC S3302_DelList (ViSession instrumentHandle, char chStr[]) Function purpose:

Delete list file **(the command will be invalid if the file does not exist and be only valid for current storage location).**

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[File – delete all list files](#)

ViStatus _VI_FUNC S3302_DelAllList (ViSession instrumentHandle)**Function purpose:**

Delete all list files.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[File – delete limit file](#)

ViStatus _VI_FUNC S3302_DelLimit (ViSession instrumentHandle, char chStr[])**Function purpose:**

Delete limit file **(the command will be invalid if the file does not exist and be only valid for current storage location).**

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[File – store limit file](#)

**ViStatus _VI_FUNC S3302_StoreLimit (ViSession instrumentHandle,
char chStr[])**

Function purpose:

Store limit line file as limit file **(the command will be invalid if the file does not exist and will only be valid for current storage location).**

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[File – recall limit line](#)

**ViStatus _VI_FUNC S3302_LoadLimit (ViSession instrumentHandle,
char chStr[])**

Function purpose:

Recall limit file to limit line **(the command will be invalid if the file does not exist and be only valid for current storage location).**

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[File - delete all limit files](#)

ViStatus _VI_FUNC S3302_DelAllLimit (ViSession instrumentHandle)

Function purpose:

delete all limit files

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[File – delete data file](#)

ViStatus _VI_FUNC S3302_DeleteDataFile (ViSession instrumentHandle, char chStr[])

Function purpose:

Delete data file under current mode **(the command will be invalid if the file does not exist and be only valid for current storage location).**

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[File – delete all data files](#)

ViStatus _VI_FUNC S3302_DeleteAllDataFile (ViSession instrumentHandle) Function

purpose:

delete all data files under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[File – recall data file](#)

ViStatus _VI_FUNC S3302_LoadDataFile (ViSession instrumentHandle, char chStr[])

Function purpose:

Recall data file under current mode **(the command will be invalid if the file does not exist and be only valid for current storage location).**

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[File – store data file](#)

**ViStatus _VI_FUNC S3302_StoreDataFile (ViSession instrumentHandle,
char chStr[])**

Function purpose:

Store current measurement data as data file **(the file will be overwritten if the file exists and this command be only valid for current storage location).**

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Interference Analysis Mode Functions](#)

[Frequency – set center frequency](#)

ViStatus _VI_FUNC S3302_SetCntFreq (ViSession instrumentHandle, ViReal64 dbVal)

Function purpose:

Set center frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency.

Scope of interference analysis frequency 0Hz~44.1GHz.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Frequency – query center frequency](#)

ViStatus _VI_FUNC S3302_QueryCntFreq (ViSession instrumentHandle,

ViReal64 dbVal[])**Function purpose:**

Query center frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Frequency – set span](#)

ViStatus _VI_FUNC S3302_SetSpan (ViSession instrumentHandle, ViReal64 dbVal)**Function purpose:**

Set span under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency.

Scope of interference analysis frequency 0Hz~44.1GHz.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Frequency – query span](#)

ViStatus _VI_FUNC S3302_QuerySpan (ViSession instrumentHandle, ViReal64 dbVal[])**Function purpose:**

Query span under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – full span

ViStatus _VI_FUNC S3302_SetFullSpan (ViSession instrumentHandle)

Function purpose:

Set as full span.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – zero span

ViStatus _VI_FUNC S3302_SetZeroSpan (ViSession instrumentHandle) Function

purpose:

Set as zero span.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – previous span

ViStatus _VI_FUNC S3302_SetLastSpan (ViSession instrumentHandle)

Function purpose:

Set as previous span.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – set start frequency

ViStatus _VI_FUNC S3302_SetSttFreq (ViSession instrumentHandle, ViReal64 dbVal)

Function purpose:

Set start frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency.

Scope of interference analysis frequency 0Hz~44.1GHz.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Frequency – query start frequency](#)

**ViStatus _VI_FUNC S3302_QuerySttFreq (ViSession instrumentHandle,
ViReal64 dbVal[])**

Function purpose:

Query start frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Frequency – set stop frequency](#)

**ViStatus _VI_FUNC S3302_SetStpFreq (ViSession instrumentHandle, ViReal64
dbVal)**

Function purpose:

Set stop frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency.

Scope of interference analysis frequency 0Hz~44.1GHz.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – query stop frequency

**ViStatus _VI_FUNC S3302_QueryStpFreq (ViSession instrumentHandle,
ViReal64 dbVal[])**

Function purpose:

Query stop frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – set step frequency

**ViStatus _VI_FUNC S3302_SetStepFreq (ViSession instrumentHandle, ViReal64
dbVal)**

Function purpose:

Set step value of center frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency, scope: 1Hz~5GHz

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – query step frequency

**ViStatus _VI_FUNC S3302_QueryStepFreq (ViSession instrumentHandle,
ViReal64 dbVal[])**

Function purpose:

Query step value of center frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – set step frequency auto on/off

ViStatus _VI_FUNC S3302_SetAutoStepFreqOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

set step frequency auto on/off. When auto is on, the step frequency is 1MHz. when off, the step frequency can be 1Hz~5GHz.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – query step frequency auto on/off

ViStatus _VI_FUNC S3302_QueryAutoStepFreqOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query step frequency auto on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – set signal standard name

ViStatus _VI_FUNC S3302_SetSIGStandard (ViSession instrumentHandle, char* standard)

Function purpose:

set signal standard name under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Standard

Name of signal standard.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Frequency – query signal standard name](#)

ViStatus _VI_FUNC S3302_QuerySIGstandard (ViSession instrumentHandle, char standard[])

Function purpose:

Query signal standard name under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Standard

Name of signal standard.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Frequency – set signal standard channel number](#)

ViStatus _VI_FUNC S3302_SetChannelNum (ViSession instrumentHandle, ViInt32 channelNum)

Function purpose:

Set channel number under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

channelNum

Channel number.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Frequency – query signal standard channel number](#)

ViStatus _VI_FUNC S3302_QueryChannelNum (ViSession instrumentHandle, ViInt32 channelNum[])

Function purpose:

Set channel number under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

channelNum

Channel number.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Amplitude – set reference level](#)

ViStatus _VI_FUNC S3302_SetRef (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

set reference level. Reference level is related to current amplitude unit, and the setting scope corresponds to dBm. Conversion is required.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

reference level (-120dBm~40dBm).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Amplitude – query reference level](#)

ViStatus _VI_FUNC S3302_QueryRef (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

query reference level (reference value). Reference level value is related to current amplitude unit.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Reference level value (reference value).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Amplitude – set reference position](#)

ViStatus _VI_FUNC S3302_SetRefPos (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Setting of reference position.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Reference position, scope: -10~10.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Amplitude – query reference position](#)

ViStatus _VI_FUNC S3302_QueryRefPos (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

query reference position.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

reference position value.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Amplitude – set attenuation](#)

ViStatus _VI_FUNC S3302_SetAtt (ViSession instrumentHandle, ViInt32 nVal) Function

purpose:

Set attenuation, only seven scales are available, which are 0, 10, 20, 30, 40, 50, 60. Other values set will be set as the attenuation for adjacent channel.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Attenuation, seven scales 0, 10, 20, 30, 40, 50, 60

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – query attenuation

ViStatus _VI_FUNC S3302_QueryAtt (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

query attenuation.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Attenuation, seven scales 0, 10, 20, 30, 40, 50, 60

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude set attenuation auto on/off

ViStatus _VI_FUNC S3302_SetAutoAttOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set attenuation auto on/off. When on, the instrument will set relevant attenuation value automatically based on the reference value.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude query attenuation auto on/off

ViStatus _VI_FUNC S3302_QueryAutoAttOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query attenuation auto on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Amplitude – set scale/division](#)

ViStatus _VI_FUNC S3302_SetScalePDiv (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set scale/division.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Scale (0.1dB~20dB)

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Amplitude – query scale/division](#)

ViStatus _VI_FUNC S3302_QueryScalePDiv (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query scale/division.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Scale/division.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Amplitude – set pre-amplifier on/off](#)

ViStatus _VI_FUNC S3302_SetPreAmpOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

set pre-amplifier on/off. When on, the measurement accuracy of small signal can be improved. However, when measuring large power signal, the pre-amplifier is better be off, or measurement AD overflow may occur.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Amplitude – query pre-amplifier on/off](#)

ViStatus _VI_FUNC S3302_QueryPreAmpOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

Query pre-amplifier on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Bandwidth – set resolution bandwidth](#)

ViStatus _VI_FUNC S3302_SetRBW (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set resolution bandwidth.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Frequency (Hz), scope: 1Hz~10MHz, step: 1-3-10.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Bandwidth – query resolution bandwidth](#)

ViStatus _VI_FUNC S3302_QueryRBW (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query resolution bandwidth.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Bandwidth – set video bandwidth](#)

ViStatus _VI_FUNC S3302_SetVBW (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set video bandwidth.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Frequency (Hz), scope: 1Hz~10MHz, step: 1-3-10.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Bandwidth – query video bandwidth](#)

ViStatus _VI_FUNC S3302_QueryVBW (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query video bandwidth.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Bandwidth – set resolution bandwidth auto on/off](#)

ViStatus _VI_FUNC S3302_SetAutoRBWOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

set resolution bandwidth auto on/off. When on, the resolution bandwidth will automatically adapter the resolution bandwidth as per SPAN/RBW based on span.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Bandwidth – query resolution bandwidth auto on/off](#)

ViStatus _VI_FUNC S3302_QueryAutoRBWOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query resolution bandwidth auto on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Bandwidth – set video bandwidth auto on/off](#)

ViStatus _VI_FUNC S3302_SetAutoVBWOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set video bandwidth auto on/off. When on, the video bandwidth will automatically adapter the resolution bandwidth as per SPAN/RBW based on span.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means

failed.

Bandwidth –query video bandwidth auto on/off

ViStatus _VI_FUNC S3302_QueryAutoVBWOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query video bandwidth auto on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Bandwidth – set SPAN/RBW

ViStatus _VI_FUNC S3302_SetSR100 (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set SPAN/RBW value.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

SPAN/RBW, scope: 1~500.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Bandwidth – query SPAN/RBW

ViStatus _VI_FUNC S3302_QuerySR100 (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query SPAN/RBW value.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

SPAN/RBW value.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Bandwidth – set RBW/VBW

ViStatus _VI_FUNC S3302_SetRV300 (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set RBW/VBW value.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

RBW/VBW, scope: 1~100.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Bandwidth – query RBW/VBW

ViStatus _VI_FUNC S3302_QueryRV300 (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query RBW/VBW value.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

SPAN/RBW value.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Average – set average on/off

ViStatus _VI_FUNC S3302_SetAvgOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set up average switch.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Average – query average on/off](#)

ViStatus _VI_FUNC S3302_QueryAvgOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query average on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Average – set average count](#)

ViStatus _VI_FUNC S3302_SetAvgCount (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set up average frequency.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

average count, scope: 1~1000

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Average – query average count](#)

ViStatus _VI_FUNC S3302_QueryAvgCount (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query average count.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Average.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Average – clear average count](#)

ViStatus _VI_FUNC S3302_ClearAvgCount (ViSession instrumentHandle)

Function purpose:

Clear average count to start from 0.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Average – query current average count](#)

ViStatus _VI_FUNC S3302_QueryCurrentCount (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

query current average count.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

average count.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Detector – set detector type](#)

ViStatus _VI_FUNC S3302_SetDetectorType (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

set detector type.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Detector type

POSitive(0)	Positive Peak
NEGative(1)	Negative Peak
SAMPlE(2)	Sample
NORMal(3)	Standard (Rosenfeld)
AVERage(4)	Average
RMS(5)	Root mean square

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Detector – query detector type](#)

**ViStatus _VI_FUNC S3302_QueryDetectorType (ViSession instrumentHandle,
ViInt32 nVal [])**

Function purpose:

query detector type.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Detector type

POSitive(0)	Positive Peak
NEGative(1)	Negative Peak
SAMPlE(2)	Sample
NORMal(3)	Standard (Rosenfeld)
AVERage(4)	Average
RMS(5)	Root mean square

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Detector – set detector auto on/off](#)

**ViStatus _VI_FUNC S3302_SetAutoDetectorOn (ViSession instrumentHandle,
ViBoolean bOn)**

Function purpose:

set detector auto on/off. When on, the instrument will automatically select detector type

based on different measurement.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Detector – query detector auto on/off](#)

**ViStatus _VI_FUNC S3302_QueryAutoDetectorOn (ViSession instrumentHandle,
ViBoolean bOn[])**

Function purpose:

query detector auto on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Sweep – set sweep type](#)

**ViStatus _VI_FUNC S3302_SetSwpType (ViSession instrumentHandle, ViBoolean
nVal)**

Function purpose:

Set the scanning type. This is an overlapping command. Use **S3302_QueryOPC()** to query if this command is completed before sending other command.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Sweep type.

0: single sweep.

1: continuous sweep.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means

failed.

Sweep – query sweep type

ViStatus _VI_FUNC S3302_QuerySwpType (ViSession instrumentHandle, ViBoolean nVal[])

Function purpose:

Query sweep type.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Sweep type.

0: single sweep.

1: continuous sweep.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – trigger single sweep

ViStatus _VI_FUNC S3302_TrigSingleSwp (ViSession instrumentHandle)

Function purpose:

Trigger one single sweep (only valid for single sweep). This command function is an overlapping command function. Use **S3302_QueryOPC()** to query if this command is completed before sending other command.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – set sweep time

ViStatus _VI_FUNC S3302_SetSwpTime (ViSession instrumentHandle, ViReal64 dbVal)

Function purpose:

Set sweep time under current mode. The sweep time is the time required for selected frequency interval for local oscillator tuning. The sweep time directly affects the time for completing one test, excluding the dead time between one sweep and the next sweep. The sweep time generally changes with the span, resolution bandwidth and video bandwidth. The sweep time is not available when the resolution bandwidth is $\leq 1\text{kHz}$ under spectrum analysis mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Time (ms).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Sweep – query sweep time](#)

**ViStatus _VI_FUNC S3302_QuerySwpTime (ViSession instrumentHandle,
ViReal64 dbVal[])**

Function purpose:

Query sweep time under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Time (ms).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Sweep – set sweep time auto on/off](#)

**ViStatus _VI_FUNC S3302_SetAutoSwpTimeOn (ViSession instrumentHandle,
ViBoolean bOn)**

Function purpose:

Set sweep time auto on/off. When auto is on, the instrument will use quick sweep speed as possible, or the manual mode is available to increase the sweep time to meet certain measurement needs. The sweep time for manual setting must be equal to or greater than auto sweep time.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Sweep – query sweep time auto on/off](#)

ViStatus _VI_FUNC S3302_QueryAutoSwpTimeOn (ViSession instrumentHandle,

ViBoolean bOn[])

Function purpose:

query sweep time auto on/off

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Trace - set trace state

**ViStatus _VI_FUNC S3302_SetTraceType (ViSession instrumentHandle, ViInt32 nVal,
ViInt32 nTrace)**

Function purpose:

Set trace state

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Trace types.

0: refresh trace

1: maximum hold

2: minimum hold

Set nTrace value as 1

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Trace – query trace state

**ViStatus _VI_FUNC S3302_QueryTraceType (ViSession instrumentHandle, ViInt32
nVal[],
ViInt32 nTrace)**

Function purpose:

Query trace state.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Trace types.

0: refresh trace

1: maximum hold

2: minimum hold

Set nTrace value as 1

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Marker – set market state

**ViStatus _VI_FUNC S3302_SetMkrState (ViSession instrumentHandle,
ViInt32 nVal, ViInt32 nState)**

Function purpose:

set marker state under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nState

Marker state.

0: marker off.

1: normal marker on.

2: offset marker on.

nVal

Marker index, 1~6 available.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Marker – query marker state

**ViStatus _VI_FUNC S3302_QueryMkrState (ViSession instrumentHandle, ViInt32
nVal,
ViInt32 nState[])**

Function purpose:

Query marker state under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nState

Marker state.

0: marker off.

1: normal marker on.

2: offset marker on.

nVal

Marker index, 1~6 available.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Marker- activate marker](#)

ViStatus _VI_FUNC S3302_SetMkrActive (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Activate marker under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Marker index, 1~6 available.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Marker- marker function \(marker ->\)](#)

ViStatus _VI_FUNC S3302_SetMkrTo (ViSession instrumentHandle, ViInt32 nVal , ViInt32 nSetIdx)

Function purpose:

Set marker function under current mode (marker -> for spectrum analysis mode).

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nSetIdx

Instrument mode	nSetIdx	Function
Interference Analysis (non-zero span)	0	Marker -> start frequency (set marker frequency as start frequency)
	1	Marker -> stop frequency (set marker frequency as

		stop frequency)
	2	Marker -> center frequency (set marker frequency as center frequency)
	3	Marker -> step frequency (set marker frequency as step frequency)
Interference Analysis (zero span)	0	Marker -> start frequency (set marker index as minimum index)
	1	Marker -> stop frequency (set marker index as maximum index)
	2	Marker -> center frequency (set marker index as center index)
	3	Marker -> step frequency (set marker frequency as step frequency)

nVal

Marker index, 1~6 available.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Marker – disable all markers

ViStatus _VI_FUNC S3302_DisableAllMarkers (ViSession instrumentHandle)

Function purpose:

Disable all markers under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Marker –set marker X value

ViStatus _VI_FUNC S3302_MoveMarker (ViSession instrumentHandle, ViInt32 nVal, ViReal64 dbVal,)

Function purpose:

set marker X value under current mode, when marker is an offset marker, the X value can be negative.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Marker X value, time in ms and frequency in Hz.

Instrument mode	Parameter unit
Interference Analysis (non-zero span)	Hz
Interference Analysis (zero span)	ms

nVal

Marker index, 1~6 available.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Marker – query marker X value](#)

**ViStatus _VI_FUNC S3302_QueryMarker (ViSession instrumentHandle,
ViInt32 markerIndex,
ViReal64 markerPosition[],
ViReal64 markerAmplitude[])**

Function purpose:

Query marker X value and Y value under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

markerPosition

Marker X value, time in ms and frequency in Hz.

Instrument mode	Parameter unit
Interference Analysis (non-zero span)	Hz
Interference Analysis (zero span)	ms

markerIndex

Marker index, 1~6 available.

markerAmplitude

Marker Y value.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Marker - search

ViStatus _VI_FUNC S3302_SetMkrSearch (ViSession instrumentHandle, ViInt32 nVal, ViInt32 type)

Function purpose:

Move marker to maximum, minimum, peak, secondary peak, left adjacent peak and right adjacent peak.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Marker index can be set from 1 to 6 1~6 available, indicating marker 1, 2, 3 and 4.

type

Search type.

1 Maximum value

2 Minimum value

3 Peak

4 Secondary peak 5

left adjacent peak

6 right adjacent peak

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Marker – set noise marker on

ViStatus _VI_FUNC S3302_SetMkrNoiseOn (ViSession instrumentHandle, ViInt32 nVal, ViBoolean bOn)

Function purpose:

Set noise marker on under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

State, 0 is off and 1 is on.

nVal

Marker index, 1~6 available.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means

failed.

[Marker – query noise marker on](#)

ViStatus _VI_FUNC S3302_QueryMkrNoiseOn (ViSession instrumentHandle, ViInt32 nVal,

ViBoolean bOn[])

Function purpose:

Query noise marker on under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

State, 0 is off and 1 is on.

nVal

Marker index, 1~6 available.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – set measurement mode.](#)

ViStatus _VI_FUNC S3302_SetIAMode (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set the measurement mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Measurement mode.

0: spectrum measurement.

1: waterfall graph.

2: RSSI

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – query measurement mode.](#)

ViStatus _VI_FUNC S3302_QueryIAMode (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query the measurement mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Measurement mode.

0: spectrum measurement.

1: waterfall graph.

2: RSSI

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Auto save – set span time](#)

ViStatus _VI_FUNC S3302_SetTraceTimeSpan (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set span time.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Span time (0~1440 min)

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Auto save – query span time](#)

ViStatus _VI_FUNC S3302_QueryTraceTimeSpan (ViSession instrumentHandle, ViInt32 nVal [])

Function purpose:

Query span time.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Span time.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Auto save – set auto save state](#)

ViStatus _VI_FUNC S3302_SetIASaveOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set auto save state.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Auto save – query auto save state](#)

ViStatus _VI_FUNC S3302_QueryIASaveOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

Query auto save state.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Auto save – set time cursor](#)

ViStatus _VI_FUNC S3302_SetIACursor (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set time cursor.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Time cursor (0-288)

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Auto save – set sweep interval](#)

ViStatus _VI_FUNC S3302_SetIAInterval (ViSession instrumentHandle, ViReal64 dbVal)

Function purpose:

Set sweep interval.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

sweep interval (ms).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Auto save – query sweep interval](#)

ViStatus _VI_FUNC S3302_QueryIAInterval (ViSession instrumentHandle, ViReal64 dbVal[])

Function purpose:

Query sweep interval

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

sweep interval (ms).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Auto save – restart measurement](#)

ViStatus _VI_FUNC S3302_SetIARestart (ViSession instrumentHandle) Function purpose:

restart measurement.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[File – delete data file](#)

ViStatus _VI_FUNC S3302_DeleteDataFile (ViSession instrumentHandle, char chStr[])

Function purpose:

Delete data file under current mode **(the command will be invalid if the file does not exist and be only valid for current storage location).**

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[File – delete all data files](#)

ViStatus _VI_FUNC S3302_DeleteAllDataFile (ViSession instrumentHandle) Function purpose:

delete all data files under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[File – recall data file](#)

ViStatus _VI_FUNC S3302_LoadDataFile (ViSession instrumentHandle, char chStr[])

Function purpose:

Recall data file under current mode **(the command will be invalid if the file does not exist and be only valid for current storage location).**

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[File – store data file](#)

**ViStatus _VI_FUNC S3302_StoreDataFile (ViSession instrumentHandle,
char chStr[])**

Function purpose:

Store current measurement data as data file **(the file will be overwritten if the file exists and this command be only valid for current storage location).**

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Data – query trace data](#)

**ViStatus _VI_FUNC S3302_QueryTraceData (ViSession instrumentHandle,ViInt32
size[], ViReal64 data[],
ViInt32 index)**

Function purpose:

Query trace data under Interference Analysis mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

size

Number of trace data received.

data

Trace data stores array pointer and the array should match with the size of trace data received.

index

Trace number, and the default value is 1 under Interference Analysis mode.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

AM-FM-PM Analyzer Mode Functions

Frequency – set center frequency

ViStatus _VI_FUNC S3302_SetCntFreq (ViSession instrumentHandle, ViReal64 dbVal)

Function purpose:

Set center frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency.

Frequency range of spectrum analysis, 0Hz~44.1GHz.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – query center frequency

ViStatus _VI_FUNC S3302_QueryCntFreq (ViSession instrumentHandle, ViReal64 dbVal[])

Function purpose:

Query center frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – set step frequency

ViStatus _VI_FUNC S3302_SetStepFreq (ViSession instrumentHandle, ViReal64 dbVal)

Function purpose:

Set step value of center frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency, scope: 1Hz~5GHz

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Frequency – query step frequency](#)

**ViStatus _VI_FUNC S3302_QueryStepFreq (ViSession instrumentHandle,
ViReal64 dbVal[])**

Function purpose:

Query step value of center frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Frequency – set step frequency auto on/off](#)

**ViStatus _VI_FUNC S3302_SetAutoStepFreqOn (ViSession instrumentHandle,
ViBoolean bOn)**

Function purpose:

set step frequency auto on/off. When auto is on, the step frequency is 1MHz. when off, the step frequency can be 1Hz~5GHz.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Frequency – query step frequency auto on/off](#)

**ViStatus _VI_FUNC S3302_QueryAutoStepFreqOn (ViSession instrumentHandle,
ViBoolean bOn[])**

Function purpose:

query step frequency auto on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Frequency – set span](#)

ViStatus _VI_FUNC S3302_SetSpan (ViSession instrumentHandle, ViReal64 dbVal)

Function purpose:

Set span under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency.

Scope: 1kHz~10MHz.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Frequency – query span](#)

ViStatus _VI_FUNC S3302_QuerySpan (ViSession instrumentHandle, ViReal64 dbVal[])

Function purpose:

Query span under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Frequency – previous span](#)

ViStatus _VI_FUNC S3302_SetLastSpan (ViSession instrumentHandle)

Function purpose:

Set as previous span.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – set start frequency

ViStatus _VI_FUNC S3302_SetSttFreq (ViSession instrumentHandle, ViReal64 dbVal)

Function purpose:

Set start frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – query start frequency

**ViStatus _VI_FUNC S3302_QuerySttFreq (ViSession instrumentHandle,
ViReal64 dbVal[])**

Function purpose:

Query start frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – set stop frequency

**ViStatus _VI_FUNC S3302_SetStpFreq (ViSession instrumentHandle, ViReal64
dbVal)**

Function purpose:

Set stop frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Frequency – query stop frequency](#)

**ViStatus _VI_FUNC S3302_QueryStpFreq (ViSession instrumentHandle,
ViReal64 dbVal[])**

Function purpose:

Query stop frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Frequency – set signal standard name](#)

**ViStatus _VI_FUNC S3302_SetSIGStandard (ViSession instrumentHandle, char*
standard)**

Function purpose:

set signal standard name under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Standard

Name of signal standard.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – query signal standard name

ViStatus _VI_FUNC S3302_QuerySIGstandard (ViSession instrumentHandle, char standard[])

Function purpose:

Query signal standard name under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Standard

Name of signal standard.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – set signal standard channel number

ViStatus _VI_FUNC S3302_SetChannelNum (ViSession instrumentHandle, ViInt32 channelNum)

Function purpose:

Set channel number under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

channelNum

Channel number.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – query signal standard channel number

ViStatus _VI_FUNC S3302_QueryChannelNum (ViSession instrumentHandle, ViInt32 channelNum[])

Function purpose:

Set channel number under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

channelNum

Channel number.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – set reference level

ViStatus _VI_FUNC S3302_SetRef (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

set reference level. Reference level is related to current amplitude unit, and the setting scope corresponds to dBm. Conversion is required.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

reference level (-120dBm~40dBm).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – query reference level

ViStatus _VI_FUNC S3302_QueryRef (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

query reference level (reference value). Reference level value is related to current amplitude unit.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Reference level value (reference value).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – set reference position

ViStatus _VI_FUNC S3302_SetRefPos (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Setting of reference position.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Reference position, scope: -10~10.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Amplitude – query reference position](#)

ViStatus _VI_FUNC S3302_QueryRefPos (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

query reference position.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

reference position value.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Amplitude – set attenuation](#)

ViStatus _VI_FUNC S3302_SetAtt (ViSession instrumentHandle, ViInt32 nVal) Function purpose:

Set attenuation, only seven scales are available, which are 0, 10, 20, 30, 40, 50, 60. Other values set will be set as the attenuation for adjacent channel.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Attenuation, seven scales 0, 10, 20, 30, 40, 50, 60

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Amplitude – query attenuation](#)

ViStatus _VI_FUNC S3302_QueryAtt (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

query attenuation.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Attenuation, seven scales 0, 10, 20, 30, 40, 50, 60

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Amplitude set attenuation auto on/off](#)

ViStatus _VI_FUNC S3302_SetAutoAttOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set attenuation auto on/off. When on, the instrument will set relevant attenuation value automatically based on the reference value.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Amplitude query attenuation auto on/off](#)

ViStatus _VI_FUNC S3302_QueryAutoAttOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query attenuation auto on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Amplitude – set scale/division](#)

ViStatus _VI_FUNC S3302_SetScalePDiv (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set scale/division.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Scale (0.1dB~20dB)

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Amplitude – query scale/division](#)

ViStatus _VI_FUNC S3302_QueryScalePDiv (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query scale/division.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Scale/division.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Amplitude – set pre-amplifier on/off](#)

ViStatus _VI_FUNC S3302_SetPreAmpOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

set pre-amplifier on/off. When on, the measurement accuracy of small signal can be improved. However, when measuring large power signal, the pre-amplifier is better be off, or measurement AD overflow may occur.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – query pre-amplifier on/off

ViStatus _VI_FUNC S3302_QueryPreAmpOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

Query pre-amplifier on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Average – set average on/off

ViStatus _VI_FUNC S3302_SetAvgOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set up average switch.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Average – query average on/off

ViStatus _VI_FUNC S3302_QueryAvgOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query average on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Average – set average count

ViStatus _VI_FUNC S3302_SetAvgCount (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set up average frequency.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

average count, scope: 1-1000

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Average – query average count

ViStatus _VI_FUNC S3302_QueryAvgCount (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query average count.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Average.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Average – clear average count

ViStatus _VI_FUNC S3302_ClearAvgCount (ViSession instrumentHandle)

Function purpose:

Clear average count to start from 0.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

failed.

Average – query current average count

ViStatus _VI_FUNC S3302_QueryCurrentCount (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

query current average count.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

average count.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Sweep – set sweep type](#)

ViStatus _VI_FUNC S3302_SetSwpType (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set the scanning type. This is an overlapping command. Use **S3302_QueryOPC()** to query if this command is completed before sending other command.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Sweep type.

0: single sweep.

1: continuous sweep.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Sweep – query sweep type](#)

ViStatus _VI_FUNC S3302_QuerySwpType (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query sweep type.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Sweep type.

0: single sweep.

1: continuous sweep.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Sweep – trigger single sweep](#)

ViStatus _VI_FUNC S3302_TrigSingleSwp (ViSession instrumentHandle)

Function purpose:

Trigger one single sweep (only valid for single sweep). This command function is an overlapping command function. Use **S3302_QueryOPC()** to query if this command is completed before sending other command.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Marker – set market state](#)

**ViStatus _VI_FUNC S3302_SetMkrState (ViSession instrumentHandle,
ViInt32 nVal, ViInt32 nState)**

Function purpose:

set marker state under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nState

Marker state.

0: marker off.

1: normal marker on.

2: offset marker on.

nVal

Marker index, 1-6 available.

failed.

Marker – query marker state

ViStatus _VI_FUNC S3302_QueryMkrState (ViSession instrumentHandle, ViInt32 nVal, ViInt32 nState[])

Function purpose:

Query marker state under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nState

Marker state.

0: marker off.

1: normal marker on.

2: offset marker on.

nVal

Marker index, 1~6 available.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Marker- activate marker

ViStatus _VI_FUNC S3302_SetMkrActive (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Activate marker under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Marker index, 1~6 available.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Marker – disable all markers

ViStatus _VI_FUNC S3302_SetMkrAOff (ViSession instrumentHandle)

Function purpose:

Disable all markers under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Marker –set marker X value](#)

**ViStatus _VI_FUNC S3302_MoveMarker (ViSession instrumentHandle,
ViInt32 nVal, ViReal64 dbVal,)**

Function purpose:

set marker X value under current mode, when marker is an offset marker, the X value can be negative.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Marker X value, time in ms and frequency in Hz.

nVal

Marker index, 1~6 available.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Marker – query marker X value](#)

**ViStatus _VI_FUNC S3302_QueryMarker (ViSession instrumentHandle,
ViInt32 markerIndex,
ViReal64 markerPosition[],
ViReal64 markerAmplitude[])**

Function purpose:

Query marker X value and Y value under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

markerPosition

Marker X value, time in ms and frequency in Hz.

markerIndex

Marker index, 1~6 available.

markerAmplitude

Marker Y value.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Marker - search](#)

ViStatus _VI_FUNC S3302_SetMkrSearch (ViSession instrumentHandle, ViInt32 nVal, ViInt32 type)

Function purpose:

Move marker to maximum, minimum, peak, secondary peak, left adjacent peak and right adjacent peak.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Marker index can be set from 1 to 6 1~6 available, indicating marker 1, 2, 3 and 4.

type

Search type.

1 Maximum value

2 Minimum value

3 Peak

4 Secondary peak 5

left adjacent peak

6 right adjacent peak

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[RF spectrum – occupied bandwidth – set occupied bandwidth state](#)

ViStatus _VI_FUNC S3302_SetOBWOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set occupied bandwidth function measurement state, or use function **S3302_SetMeasFunc()** **(Other functional measurements will be disabled after this function is enabled).**

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[RF spectrum – occupied bandwidth – query occupied bandwidth state](#)

ViStatus _VI_FUNC S3302_QueryOBWOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query occupied bandwidth state.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[RF spectrum – occupied bandwidth – set measurement method](#)

ViStatus _VI_FUNC S3302_SetOBWMethod (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set occupied bandwidth measurement method. The percentage measurement method is to obtain the x% bandwidth of total power of all span. The XdB measurement method is to obtain the xdB bandwidth less than maximum power value at both sides.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Measurement Method.

0: percentage measurement method

1: XdB measurement method

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[RF spectrum – occupied bandwidth – query measurement method](#)

ViStatus _VI_FUNC S3302_QueryOBWMethod (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query occupied bandwidth measurement method.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Measurement Method.

0: percentage measurement method

1: XdB measurement method

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[RF spectrum – occupied bandwidth – set percentage](#)

ViStatus _VI_FUNC S3302_SetOBWPpow (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set occupied bandwidth percentage.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Percentage, scope: 10.00%~99.99%.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[RF spectrum – occupied bandwidth – query percentage](#)

ViStatus _VI_FUNC S3302_QueryOBWPpow (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query occupied bandwidth percentage.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

fVal

Percentage.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

RF spectrum – occupied bandwidth – set XdB

ViStatus _VI_FUNC S3302_SetOBWXdB (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set occupied bandwidth XdB, valid when the measurement method is XdB.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

XdB value (dB), scope: -100dB~-0.1dB

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

RF spectrum – occupied bandwidth – query XdB

ViStatus _VI_FUNC S3302_QueryOBWXdB (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query occupied bandwidth XdB.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

XdB value (dB).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – set demodulation type

ViStatus _VI_FUNC S3302_SetDMType (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set demodulation type, including AM, F and PM.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Demod type.

0: AM demodulation

1: FM demodulation

2: PM demodulation

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – query demodulation type

ViStatus _VI_FUNC S3302_QueryDMType (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query demodulation type.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Demod type.

0: AM demodulation

1: FM demodulation

2: PM demodulation

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – set display mode

ViStatus _VI_FUNC S3302_SetDMMMode (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set demonstration mode, including RF, AF, AW and ALL

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Display mode.

0: RF spectrum.

1: audio spectrum.

2: audio waveform.

3: display all.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means

failed. [Measurement – query display mode](#)

ViStatus _VI_FUNC S3302_QueryDMMMode (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

query display mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Display mode.

0: RF spectrum.

1: audio spectrum.

2: audio waveform.

3: display all.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Audio spectrum – set span](#)

ViStatus _VI_FUNC S3302_SetDMAFSpan (ViSession instrumentHandle, ViReal64 dbVal)

Function purpose:

Set span for audio spectrum.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Span (500Hz~150KHz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Audio spectrum – query span](#)

ViStatus _VI_FUNC S3302_QueryDMAFSpan (ViSession instrumentHandle, ViReal64 dbVal[])

Function purpose:

Query span of audio spectrum.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Span.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Audio spectrum – set scale](#)

ViStatus _VI_FUNC S3302_SetDMScale (ViSession instrumentHandle, ViReal64 dbVal)

Function purpose:

Set scale of audio spectrum.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Scale/division.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Audio spectrum – query scale](#)

ViStatus _VI_FUNC S3302_QueryDMScale (ViSession instrumentHandle, ViReal64 dbVal[])

Function purpose:

Query scale of audio spectrum.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Scale/division.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Audio waveform – set sweep time](#)

ViStatus _VI_FUNC S3302_SetDMTime (ViSession instrumentHandle, ViReal64 dbVal)

Function purpose:

Set the sweep time for audio waveform.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

sweep time (us).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Audio waveform – query sweep time](#)

ViStatus _VI_FUNC S3302_QueryDMTime (ViSession instrumentHandle, ViReal64 dbVal[])

Function purpose:

Query the sweep time for audio waveform.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

sweep time (us).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Bandwidth – set IF bandwidth](#)

ViStatus _VI_FUNC S3302_SetDMIBW (ViSession instrumentHandle, ViReal64 dbVal)

Function purpose:

Set IF bandwidth.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Bandwidth (Hz), scope: 10KHz~300KHz

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Audio waveform – query IF bandwidth

ViStatus _VI_FUNC S3302_QueryDMIBW (ViSession instrumentHandle, ViReal64 dbVal[])

Function purpose:

query IF bandwidth.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Bandwidth (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Marker – select trace

ViStatus _VI_FUNC S3302_SetDMTrace (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set the trace selected for the marker.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Display mode.

0: RF spectrum.

1: audio spectrum.

2: audio waveform.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

File – delete data file

ViStatus _VI_FUNC S3302_DeleteDataFile (ViSession instrumentHandle, char chStr[])

Function purpose:

Delete data file under current mode **(the command will be invalid if the file does not exist and be only valid for current storage location).**

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[File – delete all data files](#)

ViStatus _VI_FUNC S3302_DeleteAllDataFile (ViSession instrumentHandle) Function

purpose:

delete all data files under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[File – recall data file](#)

ViStatus _VI_FUNC S3302_LoadDateFile (ViSession instrumentHandle, char chStr[])

Function purpose:

Recall data file under current mode **(the command will be invalid if the file does not exist and be only valid for current storage location).**

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[File – store data file](#)

**ViStatus _VI_FUNC S3302_StoreDataFile (ViSession instrumentHandle,
char chStr[])**

Function purpose:

Store current measurement data as data file **(the file will be overwritten if the file exists and this command be only valid for current storage location).**

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Data – query trace data](#)

ViStatus _VI_FUNC S3302_QueryTraceData (ViSession instrumentHandle, ViInt32 size[], ViReal64 data[], ViInt32 index)

Function purpose:

Query trace data under demodulation analysis mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

size

Number of trace data received.

data

Trace data stores array pointer and the array should match with the size of trace data received. The returned data of 3003 points are RF spectrum trace data, audio waveform trace data and audio spectrum trace data and each trace contains data of 1001 points.

index

Trace number, and the default value is 1 under demodulation analysis mode.

Return value:

The return value represents the function execution: 0 means succeeded, and less than 0 means failed.

[Power Meter Mode Functions](#)

[Frequency – set center frequency](#)

ViStatus _VI_FUNC S3302_SetCntFreq (ViSession instrumentHandle, ViReal64 dbVal)

Function purpose:

Set center frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency.

The frequency range of USB Power measurement is decided by the USB power probe model.

Power probe model	Frequency range
87231	10MHz~18GHz
87232	50MHz~26.5GHz
87233	50MHz~40GHz

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Frequency – query center frequency](#)

ViStatus _VI_FUNC S3302_QueryCntFreq (ViSession instrumentHandle, ViReal64 dbVal[])

Function purpose:

Query center frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Frequency – set frequency resolution](#)

ViStatus _VI_FUNC S3302_SetPMResolution (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

set frequency resolution.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Frequency resolution, scope: 0~3.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – query frequency resolution

ViStatus _VI_FUNC S3302_QueryPMResolution (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

query frequency resolution.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

frequency resolution

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Scale/division – auto scale

ViStatus _VI_FUNC S3302_SetPMAutoscale (ViSession instrumentHandle)

Function purpose:

Set current display scale as a scale suitable for observation.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Scale/division – set minimum value

ViStatus _VI_FUNC S3302_SetPMLower (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

set minimum value of scale.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Minimum scale (dBm), scope: -70dBm~25dBm.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Scale/division – query minimum value](#)

ViStatus _VI_FUNC S3302_QueryPMLower (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Query minimum scale.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Minimum scale (dBm), scope: -70dBm~25dBm.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Scale/division – set maximum value](#)

ViStatus _VI_FUNC S3302_SetPMUpper (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set maximum value.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Maximum scale (dBm), scope: -65dBm~30dBm.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Scale/division – query maximum value](#)

ViStatus _VI_FUNC S3302_QueryPMUpper (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

query maximum scale value.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Maximum scale (dBm), scope: -65dBm~30dBm.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Scale/division – query relative measurement value](#)

ViStatus _VI_FUNC S3302_QueryPMRelative (ViSession instrumentHandle, ViReal64 dVal [])

Function purpose:

Query relative measurement value

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

relative measurement value.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Scale/division – set relative measurement state](#)

ViStatus _VI_FUNC S3302_SetPMRelativeOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

set relative measurement state

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Scale/division – query relative measurement state](#)

ViStatus _VI_FUNC S3302_QueryPMRelativeOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query relative measurement state.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Scale/division – set offset value](#)

ViStatus _VI_FUNC S3302_SetPMCORRection (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set offset value.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Offset value (dB), scope -50dB~30dB.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Scale/division – query offset value](#)

ViStatus _VI_FUNC S3302_QueryPMCORRection (ViSession instrumentHandle, ViReal64 dVal [])

Function purpose:

Query offset value.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Offset value (dB).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Scale/division – set offset state](#)

ViStatus _VI_FUNC S3302_SetPMCORRectionOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set offset state.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Scale/division – query offset state](#)

ViStatus _VI_FUNC S3302_QueryPMCORrectionOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

Query offset state.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Scale/division – set maximum hold state](#)

ViStatus _VI_FUNC S3302_SetPMMMaxholdOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set maximum hold state

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Scale/division – query maximum hold state](#)

ViStatus _VI_FUNC S3302_QueryPMMMaxholdOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

Query maximum hold state.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Average – set average on/off](#)

ViStatus _VI_FUNC S3302_SetAvgOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set up average switch.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Average – query average on/off](#)

ViStatus _VI_FUNC S3302_QueryAvgOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query average on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Average – set average count

ViStatus _VI_FUNC S3302_SetAvgCount (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set up average frequency.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

average count, scope: 1~1000

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Average – query average count

ViStatus _VI_FUNC S3302_QueryAvgCount (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query average count.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Average.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Average – clear average count

ViStatus _VI_FUNC S3302_ClearAvgCount (ViSession instrumentHandle)

Function purpose:

Clear average count to start from 0.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Average – query current average count

ViStatus _VI_FUNC S3302_QueryCurrentCount (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

query current average count.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

average count.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Calibration - execute

ViStatus _VI_FUNC S3302_SetPMCalibZero (ViSession instrumentHandle)

Function purpose:

USB Power measurement zero calibration **(do not repeat zero calibration during calibration)**. This command function is an overlapping command function. Use **S3302_QueryOPC()** before sending other commands function to query if this command is completed.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Calibration – query zero calibration state

ViStatus _VI_FUNC S3302_QueryPMCalibZero (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query USB Power measurement calibration fails (cannot be queried during zero calibration).

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Calibration state.

0: no calibration

1: calibration succeed

2: calibration failed

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Limit – set limit state](#)

ViStatus _VI_FUNC S3302_SetPMLimitOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set limit state.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Limit – query limit state](#)

ViStatus _VI_FUNC S3302_QueryPMLimitOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

Query limit state

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Limit – set upper limit](#)

ViStatus _VI_FUNC S3302_SetPMLimitUpper (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set upper limit

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

upper limit (-65~30dBm).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Limit – query upper limit](#)

ViStatus _VI_FUNC S3302_QueryPMLimitUpper (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Query upper limit

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Upper limit value.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Limit – set lower limit](#)

ViStatus _VI_FUNC S3302_SetPMLimitLower (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set lower limit.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Lower limit (-70~25dBm).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Limit – query lower limit](#)

ViStatus _VI_FUNC S3302_QueryPMLimitLower (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Query lower limit.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Lower limit value.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Limit – set alarm on/off](#)

ViStatus _VI_FUNC S3302_SetAlarmOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set limit audio alarm state.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Limit – query alarm on/off](#)

ViStatus _VI_FUNC S3302_QueryAlarmOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

Query limit audio alarm state.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Channel Sweep Mode Functions

Amplitude – set reference level

ViStatus _VI_FUNC S3302_SetRef (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

set reference level.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

reference level (-120dBm~40dBm).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – query reference level

ViStatus _VI_FUNC S3302_QueryRef (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

query reference level (reference value).

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Reference level value (reference value).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – set scale/division

ViStatus _VI_FUNC S3302_SetScalePDiv (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set scale/division.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Scale (0.1dB - 20dB)

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Amplitude – query scale/division](#)

ViStatus _VI_FUNC S3302_QueryScalePDiv (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query scale/division.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Scale/division.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – set graph/table display](#)

ViStatus _VI_FUNC S3302_SetCSDisplay (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set graph/table display.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

0 is graph display and 1 is table display.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – query graph/table display](#)

ViStatus _VI_FUNC S3302_QueryCSDisplay (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query graph/table display.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

0 is graph display and 1 is table display.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – set maximum hold state](#)

ViStatus _VI_FUNC S3302_SetCSMaxhold (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set maximum hold state

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – query maximum hold state](#)

ViStatus _VI_FUNC S3302_QueryCSMaxhold (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

Query maximum hold state.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – set unit](#)

ViStatus _VI_FUNC S3302_SetCSUnit (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set unit

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

0 indicates channel, and 1 frequency.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – query unit](#)

ViStatus _VI_FUNC S3302_QueryCSUnit (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query unit.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

0 indicates channel, and 1 frequency.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – set power display](#)

ViStatus _VI_FUNC S3302_SetCSPower (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set power display mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

0 indicates now, and 1 maximum.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – query power display](#)

ViStatus _VI_FUNC S3302_QueryCSPower (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query power display mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

0 indicates now, and 1 maximum.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – set color](#)

ViStatus _VI_FUNC S3302_SetCSColour (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set color.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

0 indicates single color, 1 dual color.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – query color](#)

ViStatus _VI_FUNC S3302_QueryCSColour (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query color.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

0 indicates single color, 1 dual color.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – set graph/table direction](#)

ViStatus _VI_FUNC S3302_SetCSOriental (ViSession instrumentHandle, ViInt32

nVal)

Function purpose:

Set graph/table direction

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

0 indicates vertical display, 1 horizontal display.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Measurement – query graph/table direction](#)

ViStatus _VI_FUNC S3302_QueryCSOriental (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query graph/table direction

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

0 indicates vertical display, 1 horizontal display.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Sweep – set sweep mode](#)

ViStatus _VI_FUNC S3302_SetCSMode (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set sweep mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

0 indicates channel sweep, 1 frequency sweep and 2 list sweep.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – query sweep mode

ViStatus _VI_FUNC S3302_QueryCSMode (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query sweep mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

0 indicates channel sweep, 1 frequency sweep and 2 list sweep.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – channel sweep – set signal standard name

ViStatus _VI_FUNC S3302_SetSIGStandard (ViSession instrumentHandle, char* standard)

Function purpose:

set signal standard name under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Standard

Name of signal standard.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – channel sweep – query signal standard name

ViStatus _VI_FUNC S3302_QuerySIGstandard (ViSession instrumentHandle, char standard[])

Function purpose:

Query signal standard name under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Standard

Name of signal standard.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – channel sweep – set signal standard channel number

ViStatus _VI_FUNC S3302_SetChannelNum (ViSession instrumentHandle, ViInt32 channelNum)

Function purpose:

Set channel number under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

channelNum

Channel number.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – channel sweep – query signal standard channel number

ViStatus _VI_FUNC S3302_QueryChannelNum (ViSession instrumentHandle, ViInt32 channelNum[])

Function purpose:

Set channel number under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

channelNum

Channel number.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – channel sweep – set channel number

ViStatus _VI_FUNC S3302_SetCSNum (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set Channel number

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Channel number (1-20).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Sweep – channel sweep – query channel number](#)

ViStatus _VI_FUNC S3302_QueryCNum (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query channel number

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Channel number (1-20)

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Sweep – channel sweep – set channel step](#)

ViStatus _VI_FUNC S3302_SetCSStep (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set channel step.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

channel step (1-25).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Sweep – channel sweep – query channel step](#)

ViStatus _VI_FUNC S3302_QueryCSStep (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

query channel step.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

channel step (1-25)

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Sweep – channel sweep – set channel bandwidth](#)

ViStatus _VI_FUNC S3302_SetCSBandwidth (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

set channel bandwidth

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

channel bandwidth (1kHz - 20MHz)

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Sweep – channel sweep – query channel bandwidth](#)

ViStatus _VI_FUNC S3302_QueryCSBandwidth (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

query channel bandwidth.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

channel bandwidth (1kHz - 20MHz)

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Sweep – frequency sweep – set start frequency](#)

ViStatus _VI_FUNC S3302_SetCSStartFreq (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

set start frequency for frequency sweep.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Start frequency.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Sweep – frequency sweep – query start frequency](#)

ViStatus _VI_FUNC S3302_QueryCSStartFreq (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query start frequency for frequency sweep.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Start frequency.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Sweep – frequency sweep – set step frequency](#)

ViStatus _VI_FUNC S3302_SetCSStepFreq (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

set step frequency for frequency sweep.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Step frequency.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Sweep – frequency sweep – query step frequency](#)

ViStatus _VI_FUNC S3302_QueryCSStepFreq (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query step frequency for frequency sweep.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Step frequency.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Sweep – frequency sweep – set channel bandwidth](#)

ViStatus _VI_FUNC S3302_SetCSFreqScanBandwidth (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

set channel bandwidth

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

channel bandwidth (1kHz - 20MHz)

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Sweep – frequency sweep – query channel bandwidth](#)

ViStatus _VI_FUNC S3302_QueryCSFreqScanBandwidth (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

query channel bandwidth.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

channel bandwidth (1kHz - 20MHz)

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – frequency sweep – set channel number

ViStatus _VI_FUNC S3302_SetCSFreqScanNum (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set channel number for frequency sweep.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Channel number (1-20).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – frequency sweep – query channel number

ViStatus _VI_FUNC S3302_QueryCSFreqScanNum (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query channel number for frequency sweep.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Channel number (1-20)

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – list sweep – set channel number

ViStatus _VI_FUNC S3302_SetCSListScanNum (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set channel number for list sweep.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Channel number (1-20).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Sweep – list sweep – query channel number](#)

ViStatus _VI_FUNC S3302_QueryCSListScanNum (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query channel number for list sweep.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Channel number (1-20)

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[File – delete data file](#)

ViStatus _VI_FUNC S3302_DeleteDataFile (ViSession instrumentHandle, char chStr[])

Function purpose:

Delete data file under current mode **(the command will be invalid if the file does not exist and be only valid for current storage location).**

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[File – delete all data files](#)

ViStatus _VI_FUNC S3302_DeleteAllDataFile (ViSession instrumentHandle) Function purpose:

delete all data files under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[File – recall data file](#)

ViStatus _VI_FUNC S3302_LoadDateFile (ViSession instrumentHandle, char chStr[])

Function purpose:

Recall data file under current mode **(the command will be invalid if the file does not exist and be only valid for current storage location).**

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[File – store data file](#)

**ViStatus _VI_FUNC S3302_StoreDataFile (ViSession instrumentHandle,
char chStr[])**

Function purpose:

Store current measurement data as data file **(the file will be overwritten if the file exists and this command be only valid for current storage location).**

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Field Strength Mode Functions

[Frequency – set point frequency](#)

ViStatus _VI_FUNC S3302_SetPointFreq (ViSession instrumentHandle, ViReal64 dbVal)

Function purpose:

Set the point frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency, range: 1Hz - 5GHz.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Frequency – query point frequency](#)

**ViStatus _VI_FUNC S3302_QueryPointFreq (ViSession instrumentHandle,
ViReal64 dbVal[])**

Function purpose:

Query the point frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Frequency – set step frequency](#)

**ViStatus _VI_FUNC S3302_SetStepFreq (ViSession instrumentHandle, ViReal64
dbVal)**

Function purpose:

Set the step value of point frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency, range: 1Hz - 5GHz.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Frequency – query step frequency](#)

**ViStatus _VI_FUNC S3302_QueryStepFreq (ViSession instrumentHandle,
ViReal64 dbVal[])**

Function purpose:

Query the step value of point frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Frequency – set auto step frequency on](#)

ViStatus _VI_FUNC S3302_SetAutoStepFreqOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set auto step frequency on. When auto is on, the step frequency of the instrument will be 1MHz, and when off, the frequency can be set as 1Hz - 5GHz.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0 is off and 1 is on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Frequency – Query auto step frequency on](#)

ViStatus _VI_FUNC S3302_QueryAutoStepFreqOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

Query auto step frequency on.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0 is off and 1 is on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – Set frequency track on

ViStatus _VI_FUNC S3302_SetFreqTrac (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set frequency track on.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0 is off and 1 is on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – Query frequency track on

ViStatus _VI_FUNC S3302_QueryAutoStepFreqOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

Query frequency track on.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0 is off and 1 is on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – Set the start frequency for frequency scanning

ViStatus _VI_FUNC S3302_SetStepFreq (ViSession instrumentHandle, ViReal64 dbVal)

Function purpose:

Set the start frequency for frequency scanning under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency, range: 1MHz - 44.1GHz.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Frequency – Query the start frequency for frequency scanning](#)

**ViStatus _VI_FUNC S3302_QueryStepFreq (ViSession instrumentHandle,
ViReal64 dbVal[])**

Function purpose:

Query the start frequency for frequency scanning under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Frequency – Set the step frequency for frequency scanning](#)

**ViStatus _VI_FUNC S3302_SetStepFreq (ViSession instrumentHandle, ViReal64
dbVal)**

Function purpose:

Set the step frequency for frequency scanning under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency, range: 1Hz - 5GHz.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Frequency – Query the step frequency for frequency scanning](#)

**ViStatus _VI_FUNC S3302_QueryStepFreq (ViSession instrumentHandle,
ViReal64 dbVal[])**

Function purpose:

Query the step frequency for frequency scanning under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Frequency – Set the scan points](#)

ViStatus _VI_FUNC S3302_SetFscanPoints (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set the scan points.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Scan points (2-58).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Frequency – Query the scan points](#)

ViStatus _VI_FUNC S3302_QueryFscanPoints(ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query the scan points.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Scan points (2-58)

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Amplitude – Set the reference level](#)

ViStatus _VI_FUNC S3302_SetRef (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set the reference level. The reference level depends on current amplitude unit. The setting range corresponds to dBm and conversion of units is required.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Reference level (-150dBm to 40dBm).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Amplitude – Query the reference level](#)

ViStatus _VI_FUNC S3302_QueryRef (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query the reference level. The reference level depends on current amplitude unit.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Reference level (reference value).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Amplitude – set attenuation](#)

ViStatus _VI_FUNC S3302_SetAtt (ViSession instrumentHandle, ViInt32 nVal) Function

purpose:

Set attenuation, only seven scales are available, which are 0, 10, 20, 30, 40, 50, 60. Other values set will be set as the attenuation for adjacent channel.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Attenuation, seven scales 0, 10, 20, 30, 40, 50, 60.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Amplitude – query attenuation](#)

ViStatus _VI_FUNC S3302_QueryAtt (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

query attenuation.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Attenuation, seven scales 0, 10, 20, 30, 40, 50, 60.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Amplitude - set attenuation auto on/off](#)

ViStatus _VI_FUNC S3302_SetAutoAttOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set attenuation auto on/off. When on, the instrument will set relevant attenuation value automatically based on the reference value.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0 is off and 1 is on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Amplitude - query attenuation auto on/off](#)

ViStatus _VI_FUNC S3302_QueryAutoAttOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query attenuation auto on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0 is off and 1 is on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Amplitude – set scale/division](#)

ViStatus _VI_FUNC S3302_SetScalePDiv (ViSession instrumentHandle, ViReal64

dVal)

Function purpose:

Set scale/division.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Set scale/division (1dB - 40dB).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Amplitude – query scale/division](#)

ViStatus _VI_FUNC S3302_QueryScalePDiv (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query scale/division.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Scale/division.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Amplitude – set unit](#)

ViStatus _VI_FUNC S3302_SetAmpUnit (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set the amplitude unit.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Amplitude unit

DBM(0)	unit:dBm
DBMV(1)	unit:dBmV
DBUV(2)	unit:dBuV
V(3)	unit:Volts
W(4)	unit:Watts

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – query unit

ViStatus _VI_FUNC S3302_QueryAmpUnit (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query amplitude unit.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Amplitude unit

DBM(0)	unit:dBm
DBMV(1)	unit:dBmV
DBUV(2)	unit:dBuV
V(3)	unit:Volts
W(4)	unit:Watts

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – set pre-amplifier on/off

ViStatus _VI_FUNC S3302_SetPreAmpOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set pre-amplifier on/off. When on, the measurement accuracy of small signal can be improved. However, when measuring large power signal, the pre-amplifier is better be off, or measurement AD overflow may occur.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0 is off and 1 is on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – query pre-amplifier on/off

ViStatus _VI_FUNC S3302_QueryPreAmpOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

Query pre-amplifier on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0 is off and 1 is on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Detector – set detector type](#)

ViStatus _VI_FUNC S3302_SetDetectorType (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set detector type.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Detector type.

POSitive(0)	Peak
SAMPlE(2)	Real time
AVERAge(4)	Average

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Detector – query detector type](#)

ViStatus _VI_FUNC S3302_QueryDetectorType (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

query detector type.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Detector type

POSitive(1)	Peak
SAMPlE(2)	Real time

AVERage(0)	Average
------------	---------

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Sweep – set sweep type](#)

ViStatus _VI_FUNC S3302_SetSwpType (ViSession instrumentHandle, ViBoolean nVal)

Function purpose:

Set the scanning type. This is an overlapping command. Use S3302_QueryOPC() to query if this command is completed before sending other command.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Sweep type.

0: single sweep.

1: continuous sweep.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Sweep - query sweep type](#)

ViStatus _VI_FUNC S3302_QuerySwpType (ViSession instrumentHandle, ViBoolean nVal[])

Function purpose:

Query sweep type.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Sweep type.

0: single sweep.

0: single sweep.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Sweep - trigger single sweep](#)

ViStatus _VI_FUNC S3302_TrigSingleSwp (ViSession instrumentHandle)

Function purpose:

Trigger one single sweep (only valid for single sweep). This command function is an overlapping command function. Use S3302_QueryOPC() to query if this command is completed before sending other command.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Sweep – set dwell time](#)

ViStatus _VI_FUNC S3302_SetTDwell (ViSession instrumentHandle, ViReal64 dbVal)

Function purpose:

Set dwell time **under current mode** (1ms~40s).

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Time (us).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Sweep - query dwell time](#)

**ViStatus _VI_FUNC S3302_QueryTDwell (ViSession instrumentHandle,
ViReal64 dbVal[])**

Function purpose:

Query dwell time under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Time (us).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Sweep – set dwell time auto on.](#)

ViStatus _VI_FUNC S3302_SetAutoSwpTimeOn (ViSession instrumentHandle,

ViBoolean bOn)**Function purpose:**

set dwell time auto on.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0 is off and 1 is on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Sweep - query dwell time auto on.](#)

**ViStatus _VI_FUNC S3302_QueryTDwellOn (ViSession instrumentHandle,
ViBoolean bOn[])**

Function purpose:

query dwell time auto on.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0 is off and 1 is on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Sweep - set stay time](#)

ViStatus _VI_FUNC S3302_SetTStay (ViSession instrumentHandle, ViReal64 dbVal)

Function purpose:

Set stay time under current mode (1ms~40s).

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

time (us).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – query stay time

**ViStatus _VI_FUNC S3302_QueryTStay (ViSession instrumentHandle,
ViReal64 dbVal[])**

Function purpose:

Query stay time under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

time (us).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – set stay time auto on

**ViStatus _VI_FUNC S3302_SetTStayOn (ViSession instrumentHandle, ViBoolean
bOn)**

Function purpose:

set stay time auto on.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0 is off and 1 is on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep - query stay time auto on

**ViStatus _VI_FUNC S3302_QueryTStayOn (ViSession instrumentHandle,
ViBoolean bOn[])**

Function purpose:

query stay time auto on.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0 is off and 1 is on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[List edit – list add default segment](#)

ViStatus _VI_FUNC S3302_ListAddSeg (ViSession instrumentHandle)

Function purpose:

Add default sweep segment to the list edit under current mode.

Point frequency	500MHz
Bandwidth	30kHz
Detector	Average
Demodulation	Continuous wave
Limit	-174dBm
Limit on/off	Off

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[List edit – list delete segment](#)

ViStatus _VI_FUNC S3302_ListDelSeg (ViSession instrumentHandle, ViInt32 nval)

Function purpose:

Delete segment from list edit under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nval

Segment index

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[List edit – clear list](#)

ViStatus _VI_FUNC S3302_ListClear (ViSession instrumentHandle)

Function purpose:

Delete all list edit segments under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[List edit – add segment](#)

**ViStatus _VI_FUNC S3302_ListAdd (ViSession instrumentHandle,
ViReal64 startfrequency,
ViReal64 stopfrequency, ViInt32 rbw,
ViInt32 vbw, ViInt32 sweepPoints,
ViInt32 on)**

Function purpose:

Add segment in list edit under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

startfrequency

point frequency(1MHz - 44.1GHz)

stopfrequency

Bandwidth (150Hz - 150kHz)

rbw

Demodulation

vbw

Limit (-174dBm to 50dBm)

sweepPoints

Detector

on

Limit on/off, 0 is off and 1 is on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[List edit – edit segment](#)

**ViStatus _VI_FUNC S3302_ListEdit (ViSession instrumentHandle, ViInt32 index,
ViReal64 startfrequency,
ViReal64 stopfrequency, ViInt32 rbw,**

**ViInt32 vbw, ViInt32 sweepPoints,
ViInt32 on)****Function purpose:**

Add segment in list edit under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Index

Segment index

startfrequency

point frequency(1MHz~44.1GHz)

stopfrequency

bandwidth(150Hz~150kHz)

rbw

demodulation

vbw

limit(-174dBm~50dBm)

sweepPoints

detector

on

limit on/off, 0 is off and 1 is on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Limit – set alarm on/off](#)

**ViStatus _VI_FUNC S3302_SetAlarmOn (ViSession instrumentHandle, ViBoolean
bOn)****Function purpose:**

Set limit alarm on/off. If the alarm is on, when the limit test on/off is on and the test fails, the alarm will give “beep” tone pip after each sweep.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0 is off and 1 is on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Limit – query alarm on/off

ViStatus _VI_FUNC S3302_QueryAlarmOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

Query limit alarm on/off. If the alarm is on, when the limit test on/off is on and the test fails, the alarm will give “beep” tone pip after each sweep.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0 is off and 1 is on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Limit – set limit

ViStatus _VI_FUNC S3302_SetLimit (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set limit.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

limit(-174~50dBm).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Limit – query limit

ViStatus _VI_FUNC S3302_QueryLimit (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Query limit.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

limit.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Limit – set limit on/off](#)

ViStatus _VI_FUNC S3302_SetLimitOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set limit on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0 is off and 1 is on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Limit – query limit on/off](#)

ViStatus _VI_FUNC S3302_QueryLimitOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query limit on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0 is off and 1 is on.

Return value:

The return value represents the function execution: 0 means succeeded, and less than 0 means failed.

[Demodulation – set demodulation type](#)

ViStatus _VI_FUNC S3302_SetFSTDmode (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

set demodulation type.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Demodulation type.

CW(0)	continuous wave
FM(1)	frequency modulation
AM(2)	amplitude modulation
USB(3)	upper sideband
	LSB(4)
	lower sideband
SPEAK(5)	speak

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Demodulation – query demodulation type](#)

**ViStatus _VI_FUNC S3302_QueryFSTDmode (ViSession instrumentHandle,
ViInt32 nVal[])**

Function purpose:

Query demodulation type.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

demodulation type.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Demodulation – set demodulation volume](#)

**ViStatus _VI_FUNC S3302_SetDmodeVolume (ViSession instrumentHandle, ViInt32
nVal)**

Function purpose:

Set the speaker volumne for demodulation.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

deemodulation volume, range:0 - 100.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

demodulation - query demodulation volume

ViStatus _VI_FUNC S3302_QueryDmodeVolume (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query the speaker volume for demodulation.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

demodulation volume.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

bandwidth – set bandwidth

ViStatus _VI_FUNC S3302_SetFSTBw (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set bandwidth.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Frequency (unit:Hz), range:150Hz - 150kHz.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

bandwidth – query bandwidth

ViStatus _VI_FUNC S3302_QueryFSTBw (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query bandwidth.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – set measurement type

ViStatus _VI_FUNC S3302_SetFstMeasureType (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set measure type.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Measurement types.

POTF (0) point frequency mode

FREQ(1) frequency scanning mode

LIST(2) list scanning mode

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – query measurement type

ViStatus _VI_FUNC S3302_QueryFstMeasureType (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query measurement type.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Measurement type.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Marker - peak

ViStatus _VI_FUNC S3302_SetFstPeak (ViSession instrumentHandle)

Function purpose:

set the peak value of marker.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Marker – set marker on/off](#)

ViStatus _VI_FUNC S3302_SetFstMarkerOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

set marker on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0 is off and 1 is on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Marker – query marker on/off](#)

ViStatus _VI_FUNC S3302_QueryFstMarkerOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

Query marker on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0 is off and 1 is on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Marker – set marker index](#)

ViStatus _VI_FUNC S3302_SetFstMarkerIndex (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

set marker index.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

marker index (0-57).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Marker – query marker index](#)

ViStatus _VI_FUNC S3302_QueryFstMarkerIndex (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

query marker index.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

marker index.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Data – query point scan amplitude value](#)

ViStatus _VI_FUNC S3302_QueryFstPscanAmpValue (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

query point scan amplitude value.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

amplitude.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Data – query point scan field strength value](#)

ViStatus _VI_FUNC S3302_QueryFstPscanFstValue (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

query point scan field strength value.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Field strength value.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Data – query frequency scan amplitude value](#)

ViStatus _VI_FUNC S3302_QueryFscanAmpValue (ViSession vi, ViReal64 data[], ViInt32 *size)

Function purpose:

query frequency scan amplitude value.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

data

amplitude.

Size

Number of measured value.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Data – query frequency scan field strength value](#)

ViStatus _VI_FUNC S3302_QueryFscanFstValue (ViSession vi, ViReal64 data[], ViInt32 *size)

Function purpose:

query frequency scan field strength value.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

data

field strength value.

Size

number of measured value.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Data – query list scan amplitude value

ViStatus _VI_FUNC S3302_QueryLscanAmpValue (ViSession vi, ViReal64 data[], ViInt32 *size)

Function purpose:

query list scan amplitude value.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

data

amplitude.

Size

number of measured value.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Data – query list scan field strength value

ViStatus _VI_FUNC S3302_QueryLscanFstValue (ViSession vi, ViReal64 data[], ViInt32 *size)

Function purpose:

query list scan field strength value.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

data

field strength value.

Size

number of measured value.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Data – query offset

ViStatus _VI_FUNC S3302_QueryFstOffset (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

query offset.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

offset.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Antenna – set antenna off](#)

ViStatus _VI_FUNC S3302_SetAntOff (ViSession instrumentHandle)**Function purpose:**

Set antenna factor loading off and set antenna factor free state.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[File – recall antenna factor](#)

ViStatus _VI_FUNC S3302_LoadAntFile (ViSession instrumentHandle, char chStr[])**Function purpose:**

Recall antenna factor in field strength function measurement under spectrum analysis mode (the command will be invalid if the file does not exist and be only valid for current storage location).

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[File - store antenna factor](#)

ViStatus _VI_FUNC S3302_StoreAntFile (ViSession instrumentHandle, char chStr[])**Function purpose:**

Store antenna factor under field strength mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[File - delete antenna factor](#)

ViStatus _VI_FUNC S3302_DelAntFile (ViSession instrumentHandle, char chStr[])

Function purpose:

delete antenna factor file.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[File - delete all antenna factors](#)

ViStatus _VI_FUNC S3302_DelAllAntFile (ViSession instrumentHandle) Function

purpose:

delete all antenna factor files.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[File - delete data file](#)

ViStatus _VI_FUNC S3302_DeleteDataFile (ViSession instrumentHandle, char chStr[])

Function purpose:

Delete data file under current mode (the command will be invalid if the file does not exist and be only valid for current storage location)

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[File - delete all data files](#)

ViStatus _VI_FUNC S3302_DeleteAllDataFile (ViSession instrumentHandle) Function**purpose:**

delete all data files under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[File - recall data file](#)

ViStatus _VI_FUNC S3302_LoadDataFile (ViSession instrumentHandle, char chStr[])**Function purpose:**

Recall data file under current mode (the command will be invalid if the file does not exist and be only valid for current storage location).

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[File - store data file](#)

**ViStatus _VI_FUNC S3302_StoreDataFile (ViSession instrumentHandle,
char chStr[])****Function purpose:**

Store current measurement data as data file (the file will be overwritten if the file exists and

this command be only valid for current storage location).

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Edit antenna factor – add default point](#)

ViStatus _VI_FUNC S3302_AntennaAddDefault (ViSession instrumentHandle) Function

purpose:

edit antenna factor to add default point. Frequency: 1GHz antenna factor value: 0dB

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Edit antenna factor – delete point](#)

ViStatus _VI_FUNC S3302_AntennaDelete (ViSession instrumentHandle, ViInt32 index)

Function purpose:

edit antenna factor to delete point.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

index

Point index.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Edit antenna factor – edit point](#)

ViStatus _VI_FUNC S3302_AntennaEdit (ViSession instrumentHandle, ViInt32 index, ViReal64 frequency, ViReal64 factor)

Function purpose:

edit antenna factor to edit point.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

index

Point index.

frequency

frequency (0 - 44.1GHz).

factor

Antenna factor value (-200 to 200dB).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Edit antenna factor – add point](#)

**ViStatus _VI_FUNC S3302_AntennaAdd (ViSession instrumentHandle,
ViReal64 frequency, ViReal64 factor)**

Function purpose:

edit antenna factor to add point.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

frequency

Frequency (0 - 44.1GHz).

factor

Antenna factor value (-200 to 200dB).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

[Signal analysis mode function](#)

[Frequency- Set the center frequency](#)

ViStatus _VI_FUNC S3302_SetCntFreq (ViSession instrumentHandle, ViReal64 dbVal)

Purpose of function:

Set the center frequency value in the current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

dbVal

Frequency value .

The allowable setting scope of spectrum analysis frequency is 0Hz~ allowable maximum frequency of spectrometer of current model.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Frequency- Query the center frequency](#)

**ViStatus _VI_FUNC S3302_QueryCntFreq (ViSession instrumentHandle,
ViReal64* dbVal)**

Purpose of function:

Query the center frequency value in the current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

dbVal

Frequency value (Hz).

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Frequency- Set the step frequency](#)

ViStatus _VI_FUNC S3302_SetStepFreq (ViSession instrumentHandle, ViReal64 dbVal)

Purpose of function:

Set the step value of center frequency in the current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

dbVal

Frequency value, setting range: 1Hz~5GHz.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Frequency- Query the step frequency](#)

**ViStatus _VI_FUNC S3302_QueryStepFreq (ViSession instrumentHandle,
ViReal64* dbVal)**

Purpose of function:

Query the step value of center frequency in the current mode .

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

dbVal

Frequency value (Hz).

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Frequency- Set the step frequency auto ON/OFF](#)

**ViStatus _VI_FUNC S3302_SetAutoStepFreqOn (ViSession instrumentHandle,
ViBoolean bOn)**

Purpose of function:

Set the step frequency auto ON/OFF; if auto, the step frequency of instrument is 1MHz; if case of manual, the setting range is 1Hz~5GHz.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

bOn

0: OFF; 1: ON.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Frequency- Query the step frequency auto ON/OFF](#)

**ViStatus _VI_FUNC S3302_QueryAutoStepFreqOn (ViSession instrumentHandle,
ViBoolean* bOn)**

Purpose of function:

Query the step frequency auto ON/OFF .

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

bOn

0: OFF; 1: ON.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

Frequency- Set the span

ViStatus _VI_FUNC S3302_SetSpan (ViSession instrumentHandle, ViReal64 dbVal) **Purpose of function:**

Set the span in the current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

dbVal

Frequency value .

The allowable setting scope of spectrum analysis frequency is 0Hz~ allowable maximum frequency of spectrometer of current model.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

Frequency- Query the span

ViStatus _VI_FUNC S3302_QuerySpan (ViSession instrumentHandle, ViReal64* dbVal)

Purpose of function:

Query the span in the current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

dbVal

Frequency value (Hz).

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

Frequency- Full span

ViStatus _VI_FUNC S3302_SetFullSpan (ViSession instrumentHandle)

Purpose of function:

Set as full span.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

Frequency- Zero span

ViStatus _VI_FUNC S3302_SetZeroSpan (ViSession instrumentHandle)

Purpose of function:

Set as zero span.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

Frequency- Last span

ViStatus _VI_FUNC S3302_SetLastSpan (ViSession instrumentHandle)

Purpose of function:

Set as last span.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

Frequency- Set the start frequency

ViStatus _VI_FUNC S3302_SetSttFreq (ViSession instrumentHandle, ViReal64 dbVal)

Purpose of function:

Set the start frequency in the current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

dbVal

Frequency value .

The allowable setting scope of spectrum analysis frequency is 0Hz~ allowable maximum frequency of spectrometer of current model.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

Frequency- Query the start frequency

ViStatus _VI_FUNC S3302_QuerySttFreq (ViSession instrumentHandle, ViReal64* dbVal)

Purpose of function:

Query the start frequency in the current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

dbVal

Frequency value (Hz).

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Frequency- Set the stop frequency](#)

ViStatus _VI_FUNC S3302_SetStpFreq (ViSession instrumentHandle, ViReal64 dbVal)

Purpose of function:

Set the stop frequency in the current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

dbVal

Frequency value .

The allowable setting scope of spectrum analysis frequency is 0Hz~ allowable maximum frequency of spectrometer of current model.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Frequency- Query the stop frequency](#)

**ViStatus _VI_FUNC S3302_QueryStpFreq (ViSession instrumentHandle,
ViReal64* dbVal)**

Purpose of function:

Query the stop frequency in the current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

dbVal

Frequency value (Hz).

Frequency- Set the signal standard name

ViStatus _VI_FUNC S3302_SetSIGStandard (ViSession instrumentHandle, char* standard) Purpose of function:

Set the signal standard name in the current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

Standard

Signal standard name.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

Frequency- Query the signal standard name

ViStatus _VI_FUNC S3302_QuerySIGStandard (ViSession instrumentHandle, char standard[])

Purpose of function:

Query the signal standard name in the current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

Standard

Signal standard name.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

Frequency- Set the signal standard channel number

ViStatus _VI_FUNC S3302_SetChannelNum (ViSession instrumentHandle, ViInt32 channelNum)

Purpose of function:

Set the channel number in the current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

channelNum

Channel number.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

Frequency- Query the signal standard channel number

ViStatus _VI_FUNC S3302_QueryChannelNum (ViSession instrumentHandle, ViInt32* channelNum)

Purpose of function:

Set the channel number in the current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

channelNum

Channel number.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Amplitude - Set the reference level](#)

ViStatus _VI_FUNC S3302_SetRef (ViSession instrumentHandle, ViReal64 dVal)

Purpose of function:

Set the reference level value. The reference level value is related to the current amplitude unit, and the setting range corresponds to dBm, and needs conversion.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

dVal

Reference level value(-120dBm~40dBm).

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Amplitude - Query the reference level](#)

ViStatus _VI_FUNC S3302_QueryRef (ViSession instrumentHandle, ViReal64* dVal)

Purpose of function:

Query the reference level value (reference value). The reference level value is related to the current amplitude unit.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

dVal

Reference level value (reference value).

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Amplitude-Set the reference position](#)

ViStatus _VI_FUNC S3302_SetRefPos (ViSession instrumentHandle, ViInt32

nVal) Purpose of function:

Set the reference position.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

nVal

Reference position, range: -10~10.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Amplitude-Query the reference position](#)

ViStatus _VI_FUNC S3302_QueryRefPos (ViSession instrumentHandle, ViInt32* nVal)

Purpose of function:

Query the reference position.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

nVal

Reference position value.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Amplitude – Set the attenuation](#)

ViStatus _VI_FUNC S3302_SetAtt (ViSession instrumentHandle, ViInt32 nVal)

Purpose of function:

Set the attenuation, only seven options: 0, 10, 20, 30, 40, 50, 60; if set to other value, the adjacent attenuation value will be set.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

nVal

Attenuation value, only seven options: 0, 10, 20, 30, 40, 50, 60

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Amplitude - Query the attenuation](#)

ViStatus _VI_FUNC S3302_QueryAtt (ViSession instrumentHandle, ViInt32* nVal) Purpose of function:

Query the attenuation.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

nVal

Attenuation value, only seven options: 0, 10, 20, 30, 40, 50, 60

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Amplitude – Set the attenuation auto ON/OFF](#)

ViStatus _VI_FUNC S3302_SetAutoAttOn (ViSession instrumentHandle, ViBoolean bOn) Purpose of function:

Set the attenuation auto ON/OFF; if auto, the instrument will set the corresponding attenuation value automatically in accordance with the reference value.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

bOn

0: OFF; 1: ON.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Amplitude - Query the attenuation auto ON/OFF](#)

ViStatus _VI_FUNC S3302_QueryAutoAttOn (ViSession instrumentHandle, ViBoolean* bOn)

Purpose of function:

Query the attenuation auto ON/OFF .

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

bOn

0: OFF; 1: ON.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Amplitude - Set the scale/ division](#)

ViStatus _VI_FUNC S3302_SetScalePDiv (ViSession instrumentHandle, ViReal64 dVal)

Purpose of function:

Set the scale/ division. The spectrum analysis mode with a type of linear scale cannot be set.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

dVal

Scale/ division (0.1dB~20dB)。

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Amplitude - Query the scale/ division](#)

ViStatus _VI_FUNC S3302_QueryScalePDiv (ViSession instrumentHandle, ViReal64*

dVal) Purpose of function:

Query the scale/ division.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

dVal

Scale/ division.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Amplitude-Set the pre-amplifier ON/OFF](#)

ViStatus _VI_FUNC S3302_SetPreAmpOn (ViSession instrumentHandle, ViBoolean

bOn) Purpose of function:

Set the pre-amplifier ON/OFF,

Set the pre-amplifier ON/OFF, being ON can improve the measurement accuracy of small signals, but it is better to be OFF when measuring high power signals, otherwise it may cause the overload when measuring AD.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

bOn

0: OFF; 1: ON.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Amplitude-Query the pre-amplifier ON/OFF](#)

ViStatus _VI_FUNC S3302_QueryPreAmpOn (ViSession instrumentHandle, ViBoolean* bOn)

Purpose of function:

Query the pre-amplifier ON/OFF.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

bOn

0: OFF; 1: ON.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Bandwidth- Set the resolution bandwidth](#)

ViStatus _VI_FUNC S3302_SetRBW (ViSession instrumentHandle, ViReal64 dVal) **Purpose of function:**

Set the resolution bandwidth of linear sweep in the signal analysis mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

dVal

Frequency value (Unit: Hz), range: 1Hz~10MHz, with a step of 1-3-10.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Bandwidth- Query the resolution bandwidth](#)

ViStatus _VI_FUNC S3302_QueryRBW (ViSession instrumentHandle, ViReal64* dVal)

Purpose of function:

Query the resolution bandwidth of linear sweep in the signal analysis mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

dVal

Frequency value (Hz).

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Bandwidth- Set the video bandwidth](#)

ViStatus _VI_FUNC S3302_SetVBW (ViSession instrumentHandle, ViReal64 dVal)

Purpose of function:

Set the video bandwidth of linear sweep in the signal analysis mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

dVal

Frequency value (Unit: Hz), range: 1Hz~10MHz, with a step of 1-3-10.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Bandwidth- Query the video bandwidth](#)

ViStatus _VI_FUNC S3302_QueryVBW (ViSession instrumentHandle, ViReal64* dVal)

Purpose of function:

Query the video bandwidth of linear sweep in the signal analysis mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

dVal

Frequency value (Hz).

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Bandwidth- Set the resolution bandwidth auto ON/OFF](#)

ViStatus _VI_FUNC S3302_SetAutoRBWOn (ViSession instrumentHandle, ViBoolean bOn)

Purpose of function:

Set the resolution bandwidth auto ON/OFF; if auto, the resolution bandwidth will adjust the resolution bandwidth automatically in accordance with the SPAN/RBW ratio and the span.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

bOn

0: OFF; 1: ON.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Bandwidth- Query the resolution bandwidth auto ON/OFF](#)

ViStatus _VI_FUNC S3302_QueryAutoRBWOn (ViSession instrumentHandle, ViBoolean* bOn)

Purpose of function:

Query the resolution bandwidth auto ON/OFF .

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

bOn

0: OFF; 1: ON.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Bandwidth- Set the video bandwidth auto ON/OFF](#)

ViStatus _VI_FUNC S3302_SetAutoVBWOn (ViSession instrumentHandle, ViBoolean bOn)

Purpose of function:

Set the video bandwidth auto ON/OFF; if auto, the video bandwidth will adjust the video bandwidth automatically in accordance with the RBW/VBW ratio and the resolution bandwidth.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

bOn

0: OFF; 1: ON.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Bandwidth- Query the video bandwidth auto ON/OFF](#)

ViStatus _VI_FUNC S3302_QueryAutoVBWOn (ViSession instrumentHandle, ViBoolean* bOn)

Purpose of function:

Query the video bandwidth auto ON/OFF .

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

bOn

0: OFF; 1: ON.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Bandwidth- Set SPAN/RBW](#)

ViStatus _VI_FUNC S3302_SetSR100 (ViSession instrumentHandle, ViInt32

nVal) Purpose of function:

Set the SPAN/RBW value in the spectrum analysis mode .

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

nVal

SPAN/RBW value, range: 1~500.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Bandwidth- Query SPAN/RBW](#)

ViStatus _VI_FUNC S3302_QuerySR100 (ViSession instrumentHandle, ViInt32* nVal)

Purpose of function:

Query the SPAN/RBW value in the spectrum analysis mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

nVal

SPAN/RBW value.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Bandwidth- Set RBW/VBW](#)

ViStatus _VI_FUNC S3302_SetRV300 (ViSession instrumentHandle, ViInt32 nVal)

Purpose of function:

Set the RBW/VBW value in the spectrum analysis mode .

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

nVal

RBW/VBW value, range: 1~100.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Bandwidth- Query RBW/VBW](#)

ViStatus _VI_FUNC S3302_QueryRV300 (ViSession instrumentHandle, ViInt32* nVal)

Purpose of function:

Query the RBW/VBW value in the spectrum analysis mode .

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

nVal

SPAN/RBW value.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Average- Set the average ON/OFF](#)

ViStatus _VI_FUNC S3302_SetAvgOn (ViSession instrumentHandle, ViBoolean bOn)

Purpose of function:

Set the average ON/OFF.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

bOn

0: OFF; 1: ON.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Average- Query the average ON/OFF](#)

ViStatus _VI_FUNC S3302_QueryAvgOn (ViSession instrumentHandle, ViBoolean* bOn)

Purpose of function:

Query the average ON/OFF.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

bOn

0: OFF; 1: ON.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Average- Set the average count](#)

ViStatus _VI_FUNC S3302_SetAvgCount (ViSession instrumentHandle, ViInt32 nVal)

Purpose of function:

Set the average count.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

nVal

Average count, range: 1~1000.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Average- Query the average count](#)

ViStatus _VI_FUNC S3302_QueryAvgCount (ViSession instrumentHandle, ViInt32* nVal) Purpose of function:

Query the average count.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

nVal

Average count.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Average- Clear the average count](#)

ViStatus _VI_FUNC S3302_ClearAvgCount (ViSession instrumentHandle)

Purpose of function:

Calculate the current average from 0.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Average- Query the current average count](#)

ViStatus _VI_FUNC S3302_QueryCurrentCount (ViSession instrumentHandle, ViInt32* nVal)

Purpose of function:

Query the count that has been averaged now.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

nVal

Averaged current.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Detector - Set the detector type](#)

ViStatus _VI_FUNC S3302_SetDetectorType (ViSession instrumentHandle, ViInt32 nVal) **Purpose of function:**

Set the detector type.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

nVal

Detector type.

POSitive (0)	Positive peak
NEGative (1)	Negative peak
SAMPLE (2)	Sampling
NORMAL (3)	Standard (Rosenfield)
AVERage	Average

e(4)	
RMS(5)	Root mean square

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Detector - Query the detector type](#)

ViStatus _VI_FUNC S3302_QueryDetectorType (ViSession instrumentHandle, ViInt32* nVal)

Purpose of function:

Query the detector type.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

nVal

Detector type.

POSitive (0)	Positive peak
NEGative (1)	Negative peak
SAMPLE (2)	Sampling
NORMAL (3)	Standard (Rosenfield)
AVERAGE (4)	Average
RMS(5)	Root mean square

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Detector- Set the detector auto ON/OFF](#)

ViStatus _VI_FUNC S3302_SetAutoDetectorOn (ViSession instrumentHandle, ViBoolean bOn)

Purpose of function:

Set the detector auto ON/OFF; if auto, the instrument can select the detector type automatically in accordance with the different measurement .

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

bOn

0: OFF; 1: ON.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Detector-Query the detector auto ON/OFF](#)

**ViStatus _VI_FUNC S3302_QueryAutoDetectorOn (ViSession instrumentHandle,
ViBoolean* bOn)**

Purpose of function:

Query the detector auto ON/OFF.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

bOn

0: OFF; 1: ON.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Sweep- Set the sweep type](#)

ViStatus _VI_FUNC S3302_SetSwpType (ViSession instrumentHandle, ViInt32 nVal)

Purpose of function:

Set the sweep type. Set the single sweep command as overlap command; before other command is sent, use **S3302_QueryOPC()** to query if the command is completed.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

nVal

Sweep type.

0: Single sweep.

1: Continuous sweep.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Sweep- Query the sweep type](#)

**ViStatus _VI_FUNC S3302_QuerySwpType (ViSession instrumentHandle, ViInt32*
nVal) Purpose of function:**

Query the sweep type.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

nVal

Sweep type.

0: Single sweep.

1: Continuous sweep.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Sweep- Trigger the single sweep](#)

ViStatus _VI_FUNC S3302_TrigSingleSwp (ViSession instrumentHandle)

Purpose of function:

Trigger one single sweep (only valid upon single sweep); this command function is an overlap command function; before other command is sent, use the function **S3302_QueryOPC()** to query if the command is completed.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Sweep- Set the sweep time](#)

ViStatus _VI_FUNC S3302_SetSwpTime (ViSession instrumentHandle, ViReal64 dbVal)

Purpose of function:

Set the sweep time in the current mode. The sweep time is the time needed by the local oscillation tuning to pass through the selected frequency interval. The sweep time affects the time for one test directly, and does not include the dead time between two successive sweeps; the sweep time varies with the span, resolution bandwidth and video bandwidth; in the spectrum analysis mode, when the resolution bandwidth is not over 1kHz, the sweep time cannot be set .

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

dbVal

Time value (ms).

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Sweep- Query the sweep time](#)

ViStatus _VI_FUNC S3302_QuerySwpTime (ViSession instrumentHandle,

ViReal64* dbVal)**Purpose of function:**

Query the sweep time in the current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

dbVal

Time value (ms).

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Sweep-](#) [Set the sweep time auto ON/OFF](#)

**ViStatus _VI_FUNC S3302_SetAutoSwpTimeOn (ViSession instrumentHandle,
ViBoolean bOn)****Purpose of function:**

Set the sweep time auto ON/OFF; if auto, the instrument can increase the sweep time at a sweep speed as fast as possible or manually to meet the specific measuring demands; the sweep time step manually must not be shorter than the auto sweep time.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

bOn

0: OFF; 1: ON.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Sweep-](#) [Query the sweep time auto ON/OFF](#)

**ViStatus _VI_FUNC S3302_QueryAutoSwpTimeOn (ViSession instrumentHandle,
ViBoolean* bOn)****Purpose of function:**

Query the sweep time auto ON/OFF .

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

bOn

0: OFF; 1: ON.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Marker- Set the marker state](#)

ViStatus _VI_FUNC S3302_SetMkrState (ViSession instrumentHandle, ViInt32 nVal, ViInt32 nState)

Purpose of function:

Set the marker state in the current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

nState

Marker state. 0:

Marker OFF.

1: Ordinary marker ON. 2:

Differential marker ON.

nVal

Marker index, range: 1-6.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Marker- Query the marker state](#)

ViStatus _VI_FUNC S3302_QueryMkrState (ViSession instrumentHandle, ViInt32 nVal, ViInt32* nState)

Purpose of function:

Query the marker state in the current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

nState

Marker state. 0:

Marker OFF.

1: Ordinary marker ON. 2:

Differential marker ON.

nVal

Marker index, range: 1-6.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Marker- Activate the marker](#)

ViStatus _VI_FUNC S3302_SetMkrActive (ViSession instrumentHandle, ViInt32 nVal)

Purpose of function:

Activate the marker in the current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

nVal

Marker index, range: 1-6.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Marker- Marker function \(marker->\)](#)

ViStatus _VI_FUNC S3302_SetMkrTo (ViSession instrumentHandle, ViInt32 nVal, ViInt32 nSetIdx)

Purpose of function:

Set the marker function in the current mode (the spectrum analysis mode is marker->).

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

nSetIdx

Instrument mode	n	Function
Spectrum analysis (non-zero span)	0	Marker->Start frequency (set the marker frequency as start frequency)
	1	Marker->Stop frequency (set the marker frequency as stop frequency)
	2	Marker->Center frequency (set the marker frequency as center frequency)
	3	Marker->Step frequency (set the marker frequency as step frequency)
Spectrum analysis (zero span)	0	Marker->Start frequency (set the marker index as minimum index)
	1	Marker->Stop frequency (set the marker index

		as maximum index)
	2	Marker->Center frequency (set the marker index as center index)
	3	Marker->Step frequency (set the marker frequency as step frequency)

nVal

Marker index, range: 1~6.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Marker- Disable all the markers](#)

ViStatus _VI_FUNC S3302_DisableAllMarkers (ViSession instrumentHandle)

Purpose of function:

Disable all the markers in the current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Marker- Set the marker X value](#)

ViStatus _VI_FUNC S3302_MoveMarker (ViSession instrumentHandle, ViInt32 nVal, ViReal64 dbVal,)

Purpose of function:

Set the marker X value in the current mode; in case of differential marker, the X value can be negative.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

dbVal

Marker X value, with time expressed in ms and frequency expressed in Hz.

Instrument mode	Reference unit
Spectrum analysis (non-zero span)	Hz
Spectrum	us

analysis (zero span)	
----------------------	--

nVal

Marker index, range: 1~6.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Marker- Query the X and Y values of marker](#)

**ViStatus _VI_FUNC S3302_QueryMarker (ViSession instrumentHandle,
ViInt32 markerIndex,
ViReal64* markerPosition,
ViReal64* markerAmplitude)**

Purpose of function:

Query the X and Y values of marker in the current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

markerPosition

Marker X value, with time expressed in ms and frequency expressed in Hz.

Instrument mode	Reference unit
Spectrum analysis (non-zero span)	Hz
Spectrum analysis (zero span)	us

markerIndex

Marker index, range: 1~6.

markerAmplitude

Marker Y value

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Marker- Search](#)

**ViStatus _VI_FUNC S3302_SetMkrSearch (ViSession instrumentHandle,
ViInt32 nVal,ViInt32 type)**

Purpose of function:

Move the marker in the current mode to maximum, minimum, peak, secondary peak, next peak left, and next peak right.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

nVal

4. Marker index, range: 1-6, respectively indicating the marker 1, marker 2, marker 3 and marker

type

Search type.

1 : Maximum

2: Minimum

3: Peak

4: Secondary peak

5: Next peak left

6: Next peak right

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Marker- Set the noise marker ON/OFF](#)

ViStatus _VI_FUNC S3302_SetMkrNoiseOn (ViSession instrumentHandle, ViInt32 nVal, ViBoolean bOn)

Purpose of function:

Set the noise marker ON/OFF in the current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

bOn

ON/OFF: 0: OFF; 1: ON.

nVal

Marker index, range: 1-6.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Marker- Query the noise marker ON/OFF](#)

ViStatus _VI_FUNC S3302_QueryMkrNoiseOn (ViSession instrumentHandle, ViInt32 nVal,

ViBoolean* bOn)

Purpose of function:

Query the noise marker ON/OFF in the current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

bOn

ON/OFF: 0: OFF; 1: ON.

nVal

Marker index, range: 1-6.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Demodulator-Set the demodulator ON/OFF](#)

ViStatus _VI_FUNC S3302_SetDemodOn (ViSession instrumentHandle, ViBoolean bOn)

Purpose of function:

Set the AF/FM function measure switch, or use the function **S3302_SetMeasFunc()** to activate **(the activation of this function measure will deactivate other function measure)**.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

bOn

0: OFF; 1: ON.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Demodulator-Query the demodulator ON/OFF](#)

ViStatus _VI_FUNC S3302_QueryDemodOn (ViSession instrumentHandle, ViBoolean* bOn)

Purpose of function:

Query the AM/FM demodulator ON/OFF.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

bOn

0: OFF; 1: ON.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

Demodulator-Set the demodulation mode

ViStatus _VI_FUNC S3302_SetDMode (ViSession instrumentHandle, ViInt32

nVal) Purpose of function:

Set the demodulation mode. Intermittent mode: after the data of one page is swept, do one demodulation as per the demodulation interval, then sweep the data of another page, and next do another demodulation as per the demodulation interval, and so on; continuous mode: after the data of one page is swept, do the demodulation continuously without data sweep.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

nVal

Demodulation mode.

0: Intermittent

1: Continuous

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

Demodulator-Query the demodulation mode

ViStatus _VI_FUNC S3302_QueryDMode (ViSession instrumentHandle, ViInt32* nVal)

Purpose of function:

Query the demodulation mode: after the data of one page is swept, do one demodulation as per the demodulation interval, then sweep the data of another page, and next do another demodulation as per the demodulation interval, and so on; continuous mode: after the data of one page is swept, do the demodulation continuously without data sweep.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

nVal

Demodulation mode.

0: Intermittent

1: Continuous

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

Demodulator-Set the demodulation type

ViStatus _VI_FUNC S3302_SetDType (ViSession instrumentHandle, ViInt32

nVal) Purpose of function:

Set the demodulation type, which can be FM, AM, upper sideband or lower sideband.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

nVal

Demodulation type. 0:

FM

1: AM

2: Upper sideband

3: Lower sideband

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Demodulator-Query the demodulation type](#)

ViStatus _VI_FUNC S3302_QueryDType (ViSession instrumentHandle, ViInt32* nVal)

Purpose of function:

Query the demodulation type.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

nVal

Demodulation type. 0:

FM

1: AM

2: Upper sideband

3: Lower sideband

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Demodulator-Set the demodulation time](#)

ViStatus _VI_FUNC S3302_SetDTime (ViSession instrumentHandle, ViReal64

dVal) Purpose of function:

Set the demodulation time. It is effective in the intermittent demodulation mode, and is the time entering the demodulation state after one sweep .

Instrument handle returned by the function, communicate with instrument.

dVal

Demodulation time(ms), range: 1us~400s.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Demodulator-Query the demodulation time](#)

ViStatus _VI_FUNC S3302_QueryDTime (ViSession instrumentHandle, ViReal64* dVal)

Purpose of function:

Query the demodulation time.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

dVal

Demodulation time(ms).

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Demodulator-Set the volume](#)

ViStatus _VI_FUNC S3302_SetVolume (ViSession instrumentHandle, ViInt32 nVal) **Purpose of function:**

Set the speaker volume of demodulator.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

nVal

Demodulator volume, scope: 0-100.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Demodulator- Query the volume](#)

ViStatus _VI_FUNC S3302_QueryVolume (ViSession instrumentHandle, ViInt32* nVal)

Purpose of function:

Query the speaker volume of demodulator.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

nVal

Demodulator volume.

Returned value:

[IQ capture - Set the IQ capture ON/OFF](#)

**ViStatus _VI_FUNC S3302_SetIQcaptureOn (ViSession instrumentHandle,
ViBoolean bOn)**

Purpose of function:

Set the IQ capture ON/OFF.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

bOn

0: OFF; 1: ON.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[IQ capture - Query the IQ capture ON/OFF](#)

**ViStatus _VI_FUNC S3302_QueryIQCaptureOn (ViSession instrumentHandle,
ViBoolean* bOn)**

Purpose of function:

Query the IQ capture ON/OFF.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

bOn

0: OFF; 1: ON.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[IQ capture - Start the capture](#)

ViStatus _VI_FUNC S3302_SetIQCaptureStart (ViSession instrumentHandle)

Purpose of function:

Set the IQ capture as Start .

Instrument handle returned by the function, communicate with instrument.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[IQ capture - Stop the capture](#)

ViStatus _VI_FUNC S3302_SetIQCaptureStop (ViSession instrumentHandle)

Purpose of function:

Set the IQ capture as Stop.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[IQ capture - Set the capture time](#)

**ViStatus _VI_FUNC S3302_SetIQCaptureTime (ViSession instrumentHandle,
ViReal64 dVal)**

Purpose of function:

Set the IQ capture time.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

dVal

Capture time (us).

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[IQ capture - Query the capture time](#)

**ViStatus _VI_FUNC S3302_QueryIQCaptureTime (ViSession instrumentHandle,
ViReal64* dVal)**

Purpose of function:

Query the IQ capture time.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

dVal

Capture time (us).

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[IQ capture - Set the IQ capture mode](#)

ViStatus _VI_FUNC S3302_SetIQCaptureMode (ViSession instrumentHandle, ViInt32 mode)

Purpose of function:

Set the IQ capture mode, which can be single or continuous .

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

mode

0: Single; 1: Continuous.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[IQ capture - Query the IQ capture mode](#)

ViStatus _VI_FUNC S3302_QueryIQCaptureMode (ViSession instrumentHandle, ViInt32* mode)

Purpose of function:

Query the IQ capture mode, which can be single or continuous .

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

mode

0: Single; 1: Continuous.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[IQ capture - Set the sample rate](#)

ViStatus _VI_FUNC S3302_SetIQCaptureSample (ViSession instrumentHandle, ViReal64 dVal)

Purpose of function:

Set the IQ capture sample rate.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

dVal

Sample rate.

Capture sample rate	Capture bandwidth
12.5MHz	10MHz
5MHz	4MHz
1.25MHz	1MHz
500kHz	400kHz
125kHz	100kHz
50kHz	40kHz

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[IQ capture - Query the sample rate](#)

ViStatus _VI_FUNC S3302_QueryIQCaptureSample (ViSession instrumentHandle, ViReal64* dVal)

Purpose of function:

Query the IQ capture sample rate.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

dVal

Sample rate.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[IQ capture - Set the IQ capture file name](#)

ViStatus _VI_FUNC S3302_SetIQCaptureFilename(ViSession instrumentHandle, char* name)

Purpose of function:

Set the IQ capture file name.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

name

File name

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[IQ capture - Set the trigger mode](#)

**ViStatus _VI_FUNC S3302_SetIQCaptureTrigMode (ViSession
instrumentHandle, ViInt32 nval)**

Purpose of function:

Set the trigger mode, which can be free or external.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

nVal

Trigger type.

0: Free

1: External

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[IQ capture - Query the trigger mode](#)

**ViStatus _VI_FUNC S3302_QueryIQCaptureTrigMode
(ViSession instrumentHandle, ViInt32* nVal)**

Purpose of function:

Query the trigger mode, which can be free or external .

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

nVal

Trigger type.

0: Free

1: External

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[IQ capture - Set the external trigger slope](#)

**ViStatus _VI_FUNC S3302_SetIQCaptureTrigSlop (ViSession
instrumentHandle, ViInt32 nval)**

Purpose of function:

Set the external trigger slope.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

nVal

Trigger type.

0: Rising

1: Falling

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[IQ capture - Query the external trigger slope](#)

ViStatus **_VI_FUNC** **S3302_QueryIQCaptureTrigSlop**
(ViSession instrumentHandle, ViInt32* nVal)

Purpose of function:

Query the external trigger slope.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

nVal

Trigger type.

0: Rising

1: Falling

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[IQ capture - Set the external trigger delay](#)

ViStatus **_VI_FUNC** **S3302_SetIQCaptureTrigDelay** (**ViSession instrumentHandle,**
ViReal64 dVal)

Purpose of function:

Set the external trigger delay of IQ capture.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

dVal

Trigger delay (us).

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[IQ capture - Query the external trigger delay](#)

**ViStatus _VI_FUNC S3302_QueryIQCaptureTrigDelay (ViSession instrumentHandle,
ViReal64* dVal)**

Purpose of function:

Query the external trigger delay of IQ capture.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

dVal

Trigger delay (us).

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[IQ capture - Set the external trigger level](#)

**ViStatus _VI_FUNC S3302_SetIQCaptureTrigAMP (ViSession
instrumentHandle, ViReal64 dVal)**

Purpose of function:

Set the external trigger level of IQ capture.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

dVal

Trigger level (volt).

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[IQ capture - Query the external trigger level](#)

**ViStatus _VI_FUNC S3302_QueryIQCaptureTrigAMP (ViSession instrumentHandle,
ViReal64* dVal)**

Purpose of function:

Query the external trigger level of IQ capture.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

dVal

Trigger level (volt).

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Auto save - Set the span time](#)

ViStatus _VI_FUNC S3302_SetTraceTimeSpan (ViSession instrumentHandle, ViInt32 nVal)

Purpose of function:

Set the span time.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

nVal

Span time (0~28800min)

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Auto save - Query the span time](#)

ViStatus _VI_FUNC S3302_QueryTraceTimeSpan (ViSession instrumentHandle, ViInt32* nVal)

Purpose of function:

Query the span time .

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

nVal

Span time .

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Auto save - Set the auto save ON/OFF](#)

ViStatus _VI_FUNC S3302_SetIASaveOn (ViSession instrumentHandle, ViBoolean bOn)

Purpose of function:

Set the auto save ON/OFF.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

bOn

0: OFF; 1: ON.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Auto save - Query the auto save ON/OFF](#)

ViStatus _VI_FUNC S3302_QueryIASaveOn (ViSession instrumentHandle, ViBoolean* bOn)

Purpose of function:

Query the auto save ON/OFF.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

bOn

0: OFF; 1: ON.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Auto save - Set the time cursor](#)

ViStatus _VI_FUNC S3302_SetIACursor (ViSession instrumentHandle, ViInt32 nVal)

Purpose of function:

Set the time cursor.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

nVal

Time cursor (0-290).

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Auto save - Restart the measurement](#)

ViStatus _VI_FUNC S3302_SetIARestart (ViSession instrumentHandle)

Purpose of function:

Restart the measurement.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Configuration- Set the limit ON/OFF](#)

ViStatus _VI_FUNC S3302_SetMargStat (ViSession instrumentHandle, ViInt32 bOn)

Purpose of function:

Set the limit ON/OFF.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

bOn

0: OFF; 1: ON.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Configuration- Query the limit ON/OFF](#)

ViStatus _VI_FUNC S3302_QueryMargStat (ViSession instrumentHandle, ViInt32* bOn)

Purpose of function:

Query the limit ON/OFF.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

bOn

0: OFF; 1: ON.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Configuration- Set the limit value](#)

ViStatus _VI_FUNC S3302_SetMarg (ViSession instrumentHandle, ViReal64 dVal)

Purpose of function:

Set the limit value.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

dVal

Limit value (dB).

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Configuration- Query the limit value](#)

ViStatus _VI_FUNC S3302_QueryMarg (ViSession instrumentHandle, ViReal64* dVal)

Purpose of function:

Query the limit value.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

dVal

Limit value (dB).

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Configuration- Set the signal list display mode](#)

ViStatus _VI_FUNC S3302_SetSigList (ViSession instrumentHandle, ViInt32

nVal) Purpose of function:

Set the signal list display mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

nVal

Signal list display mode

0: Detail; 1: Brief.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Configuration- Query the signal list display mode](#)

ViStatus _VI_FUNC S3302_QuerySigList (ViSession instrumentHandle, ViInt32* nVal)

Purpose of function:

Query the signal list display mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

nVal

Signal list display mode

0: Detail; 1: Brief.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Save/ recall - Delete the data file](#)

ViStatus _VI_FUNC S3302_DeleteDataFile (ViSession instrumentHandle, char chStr[])

Purpose of function:

Delete the data file in the current mode(**If the file does not exist, the command is invalid except for the current save position**).

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

chStr

File name (not add the suffix of file type).

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Save/ recall - Delete all the data files](#)

ViStatus _VI_FUNC S3302_DeleteAllDataFile (ViSession instrumentHandle)

Purpose of function:

Delete all the data files in the current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Save/ recall - Recall the data file](#)

ViStatus _VI_FUNC S3302_LoadDataFile (ViSession instrumentHandle, char chStr[])

Purpose of function:

Recall the data file in the current mode(**If the file does not exist, the command is invalid except for the current save position**).

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

chStr

File name (not add the suffix of file type).

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

[Save/ recall - Save the data file](#)

**ViStatus _VI_FUNC S3302_StoreDataFile (ViSession instrumentHandle,
char chStr[])**

Purpose of function:

Save the currently measured data as data file(**if the file has existed, the file will be covered and the command is only valid for the current save position**).

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

chStr

File name (not add the suffix of file type).

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

- End of Document -