

# **RIGOL**

## **Programming Guide**

### **RSA3000 Series Real-time Spectrum Analyzer**

Jan. 29, 2018  
**RIGOL TECHNOLOGIES, INC.**



# Guaranty and Declaration

## Copyright

© 2017 **RIGOL** TECHNOLOGIES, INC. All Rights Reserved.

## Trademark Information

**RIGOL** is a registered trademark of **RIGOL** TECHNOLOGIES, INC.

## Publication Number

PGD21101-1110

## Software Version

00.01.00

Software upgrade might change or add product features. Please acquire the latest version of the manual from **RIGOL** website or contact **RIGOL** to upgrade the software.

## Notices

- **RIGOL** products are covered by P.R.C. and foreign patents, issued and pending.
- **RIGOL** reserves the right to modify or change parts of or all the specifications and pricing policies at the company's sole decision.
- Information in this publication replaces all previously released materials.
- Information in this publication is subject to change without notice.
- **RIGOL** shall not be liable for either incidental or consequential losses in connection with the furnishing, use, or performance of this manual, as well as any information contained.
- Any part of this document is forbidden to be copied, photocopied, or rearranged without prior written approval of **RIGOL**.

## Product Certification

**RIGOL** guarantees that this product conforms to the national and industrial standards in China as well as the ISO9001:2008 standard and the ISO14001:2004 standard. Other international standard conformance certifications are in progress.

## Contact Us

If you have any problem or requirement when using our products or this manual, please contact **RIGOL**.

E-mail: [service@rigol.com](mailto:service@rigol.com)

Website: [www.rigol.com](http://www.rigol.com)

# Document Overview

This manual introduces how to program and control **RIGOL** RSA3000 series spectrum analyzer by using SCPI commands through USB and LAN interface.

## Main Topics in this Manual:

### Chapter 1 Programming Overview

This chapter introduces how to set up remote communication between the spectrum analyzer and the PC, the remote control methods, the syntax, symbols, parameters, and abbreviation rules of the SCPI commands.

### Chapter 2 Command System

This chapter introduces the syntax, function, parameters, and usage of each command.

### Chapter 3 Programming Examples

This chapter illustrates how to control the RSA3000 series by programming in the development environments such as Visual C++ , Visual Basic, and LabVIEW.

#### Tip

For the latest version of this manual, download it from the official website of **RIGOL** ([www.rigol.com](http://www.rigol.com)).

## Format Conventions in this Manual:

### 1. Keys:

The keys on the front panel are usually denoted by the format of "Key Name (Bold) + Text Box". For example, **FREQ** denotes the **FREQ** key.

### 2. Menu keys:

The menu softkeys are usually denoted by the format of "Menu Word (Bold) + Character Shading". For example, **Center Freq** denotes the center frequency menu item under the **FREQ** function key.

### 3. Connectors:

The connectors on the front or rear panel are usually denoted by the format of "Connector Name (Bold) + Square Brackets (Bold)". For example, **[Gen Output 50Ω]**.

### 4. Operation Procedures:

"→" indicates the next step of operation. For example, **FREQ** → **Center Freq** indicates pressing **FREQ** on the front panel and then pressing the menu softkey **Center Freq**.

## Content Conventions in this Manual:

The RSA3000 series spectrum analyzer includes the following models. Unless otherwise specified, this manual takes RSA3045 as an example to illustrate the usage of each command of RSA3000 series spectrum analyzer.

| Model      | Frequency Range  | Tracking Generator |
|------------|------------------|--------------------|
| RSA3045    | 9 kHz to 4.5 GHz | None               |
| RSA3030    | 9 kHz to 3 GHz   | None               |
| RSA3045-TG | 9 kHz to 4.5 GHz | 4.5 GHz            |
| RSA3030-TG | 9 kHz to 3 GHz   | 3 GHz              |

# Contents

|                                                    |            |
|----------------------------------------------------|------------|
| <b>Guaranty and Declaration .....</b>              | <b>I</b>   |
| <b>Document Overview .....</b>                     | <b>II</b>  |
| <b>Chapter 1 Programming Overview.....</b>         | <b>1-1</b> |
| Build Remote Communication .....                   | 1-2        |
| Remote Control Method .....                        | 1-3        |
| SCPI Command Overview .....                        | 1-4        |
| Syntax.....                                        | 1-4        |
| Symbol Description .....                           | 1-4        |
| Parameter Type.....                                | 1-5        |
| Command Abbreviation .....                         | 1-5        |
| <b>Chapter 2 Command System.....</b>               | <b>2-1</b> |
| :CALCulate Commands.....                           | 2-2        |
| :CALCulate:BANDwidth  BWIDth:NDB .....             | 2-4        |
| :CALCulate:BANDwidth  BWIDth:RESult?.....          | 2-4        |
| :CALCulate:BANDwidth  BWIDth:RLEft?.....           | 2-5        |
| :CALCulate:BANDwidth  BWIDth:RRIGht?.....          | 2-5        |
| :CALCulate:BANDwidth  BWIDth [:STATe] .....        | 2-5        |
| :CALCulate:LLINe:ALL:DELeTe.....                   | 2-6        |
| :CALCulate:LLINe:TEST .....                        | 2-6        |
| :CALCulate:LLINe<n>:AMPLitude:CMODE:RELative.....  | 2-7        |
| :CALCulate:LLINe<n>:BUILd .....                    | 2-7        |
| :CALCulate:LLINe<n>:COPY.....                      | 2-8        |
| :CALCulate:LLINe<n>:DATA .....                     | 2-8        |
| :CALCulate:LLINe<n>:DELeTe .....                   | 2-9        |
| :CALCulate:LLINe<n>:DISPlay.....                   | 2-9        |
| :CALCulate:LLINe<n>:FAIL?.....                     | 2-10       |
| :CALCulate:LLINe<n>:FREQuency:CMODE:RELative ..... | 2-10       |
| :CALCulate:LLINe<n>:MARGin .....                   | 2-11       |
| :CALCulate:LLINe<n>:MARGin:STATe .....             | 2-11       |
| :CALCulate:LLINe<n>:TRACe .....                    | 2-12       |
| :CALCulate:LLINe<n>:TYPE .....                     | 2-12       |
| :CALCulate:MARKer:AOff .....                       | 2-13       |
| :CALCulate:MARKer:COUPle[:STATe] .....             | 2-13       |
| :CALCulate:MARKer:PEAK:EXCURsion .....             | 2-13       |
| :CALCulate:MARKer:PEAK:EXCURsion:STATe .....       | 2-14       |
| :CALCulate:MARKer:PEAK:SEARch:MODE .....           | 2-14       |
| :CALCulate:MARKer:PEAK:SORT.....                   | 2-15       |
| :TRACe:MATH:PEAK: SORT.....                        | 2-15       |
| :CALCulate:MARKer:PEAK: TABLE:READout.....         | 2-16       |
| :TRACe:MATH:PEAK: THReshold .....                  | 2-16       |
| :CALCulate:MARKer:PEAK: TABLE[:STATe] .....        | 2-16       |
| :TRACe:MATH:PEAK: TABLE:STATe.....                 | 2-16       |
| :CALCulate:MARKer:PEAK: THReshold .....            | 2-17       |
| :CALCulate:MARKer:PEAK: THReshold:STATe .....      | 2-17       |
| :CALCulate:MARKer: TABLE[:STATe] .....             | 2-18       |
| :CALCulate:MARKer:TRCKing[:STATe] .....            | 2-18       |
| :CALCulate:MARKer<n>:CPSeArch[:STATe] .....        | 2-19       |
| :CALCulate:MARKer<n>:FCOunt:GATetime .....         | 2-20       |
| :CALCulate:MARKer<n>:FCOunt:GATetime:AUTO.....     | 2-20       |
| :CALCulate:MARKer<n>:FCOunt[:STATe] .....          | 2-21       |
| :CALCulate:MARKer<n>:FCOunt:X? .....               | 2-21       |
| :CALCulate:MARKer<n>:FUNCTion .....                | 2-22       |
| :CALCulate:MARKer<n>:FUNCTion:BAND:LEFT .....      | 2-22       |

|                                                      |      |
|------------------------------------------------------|------|
| :CALCulate:MARKer< n> :FUNCTION:BAND:RI GHt .....    | 2-23 |
| :CALCulate:MARKer< n> :FUNCTION:BAND:SPAN .....      | 2-24 |
| :CALCulate:MARKer< n> :FUNCTION:BAND:SPAN:AUTO ..... | 2-24 |
| :CALCulate:MARKer< n> :LI Nes[: STATE] .....         | 2-25 |
| :CALCulate:MARKer< n> :MAXimum:LEFT .....            | 2-25 |
| :CALCulate:MARKer< n> :MAXimum[: MAX] .....          | 2-26 |
| :CALCulate:MARKer< n> :MAXimum:NEXT .....            | 2-26 |
| :CALCulate:MARKer< n> :MAXimum:RI GHt .....          | 2-27 |
| :CALCulate:MARKer< n> :MI Nimum .....                | 2-27 |
| :CALCulate:MARKer< n> :MODE .....                    | 2-28 |
| :CALCulate:MARKer< n> :PTPeak .....                  | 2-28 |
| :CALCulate:MARKer< n> :REference .....               | 2-29 |
| :CALCulate:MARKer< n> [: SET] :CENTer .....          | 2-29 |
| :CALCulate:MARKer< n> [: SET] :DELTA:CENTer .....    | 2-30 |
| :CALCulate:MARKer< n> [: SET] :DELTA:SPAN .....      | 2-30 |
| :CALCulate:MARKer< n> [: SET] :RLEVEL .....          | 2-31 |
| :CALCulate:MARKer< n> [: SET] :START .....           | 2-31 |
| :CALCulate:MARKer< n> [: SET] :STEP .....            | 2-32 |
| :CALCulate:MARKer< n> [: SET] :STOP .....            | 2-32 |
| :CALCulate:MARKer< n> :STATE .....                   | 2-33 |
| :CALCulate:MARKer< n> :TRACE .....                   | 2-33 |
| :CALCulate:MARKer< n> :TRACE:AUTO .....              | 2-34 |
| :CALCulate:MARKer< n> :X .....                       | 2-35 |
| :CALCulate:MARKer< n> :X:READout .....               | 2-35 |
| :CALCulate:MARKer< n> :X:READout:AUTO .....          | 2-36 |
| :CALCulate:MARKer< n> :Y .....                       | 2-37 |
| :CALCulate:MARKer< n> :Z:POSition .....              | 2-37 |
| :CALCulate:MATH .....                                | 2-38 |
| :CALCulate:NTData[: STATE] .....                     | 2-39 |
| :CALibration Commands .....                          | 2-40 |
| :CALibration:[ ALL] .....                            | 2-40 |
| :CALibration:AUTO .....                              | 2-40 |
| :CONFigure Commands .....                            | 2-41 |
| :CONFigure? .....                                    | 2-41 |
| :CONFigure:ACPower .....                             | 2-42 |
| :CONFigure:CNRatio .....                             | 2-42 |
| :CONFigure:DENSity .....                             | 2-42 |
| :CONFigure:DSPEctrogram .....                        | 2-43 |
| :CONFigure:EBWidth .....                             | 2-43 |
| :CONFigure:HDISt .....                               | 2-43 |
| :CONFigure:LPSTep .....                              | 2-43 |
| :CONFigure:MCHPower .....                            | 2-44 |
| :CONFigure:NORMal .....                              | 2-44 |
| :CONFigure:OBWidth .....                             | 2-44 |
| :CONFigure:PSGRam .....                              | 2-45 |
| :CONFigure:PSPEctrum .....                           | 2-45 |
| :CONFigure:PVT .....                                 | 2-45 |
| :CONFigure:SANalyzer .....                           | 2-45 |
| :CONFigure:SPEctrogram .....                         | 2-46 |
| :CONFigure:TOI .....                                 | 2-46 |
| :CONFigure:TPOWER .....                              | 2-46 |
| :COUPle Commands .....                               | 2-47 |
| :COUPle .....                                        | 2-47 |
| :DISPlay Commands .....                              | 2-48 |
| :DISPlay:BACKlight .....                             | 2-49 |
| :DISPlay:ENABLE .....                                | 2-49 |
| :DISPlay:GRATicule[: STATE] .....                    | 2-50 |
| :DISPlay:HDMI[: STATE] .....                         | 2-50 |

|                                                             |      |
|-------------------------------------------------------------|------|
| :DISPlay:PVTime:WINDow:TRACe:X[:SCALE]:COUPle .....         | 2-51 |
| :DISPlay:PVTime:WINDow:TRACe:X[:SCALE]:PDI Vision .....     | 2-51 |
| :DISPlay:PVTime:WINDow:TRACe:X[:SCALE]:RLEVel .....         | 2-52 |
| :DISPlay:PVTime:WINDow:TRACe:X[:SCALE]:RPOStion .....       | 2-52 |
| :DISPlay:PVTime:WINDow:TRACe:Y[:SCALE]:PDI Vision .....     | 2-53 |
| :DISPlay:PVTime:WINDow:TRACe:Y[:SCALE]:RLEVel < real> ..... | 2-53 |
| :DISPlay:VIEW:DENSity:AADJust .....                         | 2-54 |
| :DISPlay:VIEW:DENSity:CNONlinear .....                      | 2-54 |
| :DISPlay:VIEW:DENSity:CPALettes .....                       | 2-55 |
| :DISPlay:VIEW:DENSity:HDHue .....                           | 2-55 |
| :DISPlay:VIEW:DENSity:LDHue .....                           | 2-56 |
| :DISPlay:VIEW:DENSity:PERStence .....                       | 2-56 |
| :DISPlay:VIEW:DENSity:PERStence:INFinite .....              | 2-57 |
| :DISPlay:VIEW[:SElect] .....                                | 2-57 |
| :DISPlay:VIEW:SPECTrogram:AADJust .....                     | 2-58 |
| :DISPlay:VIEW:SPECTrogram:BOTTom .....                      | 2-58 |
| :DISPlay:VIEW:SPECTrogram:HUE .....                         | 2-59 |
| :DISPlay:VIEW:SPECTrogram:POStion .....                     | 2-59 |
| :DISPlay:VIEW:SPECTrogram:REFerence .....                   | 2-60 |
| :DISPlay:VIEW:SPECTrogram:TRACe:COUPle .....                | 2-60 |
| :DISPlay:VIEW:SPECTrogram:TRACe:SELECTION .....             | 2-61 |
| :DISPlay:WINDow:SElect .....                                | 2-62 |
| :DISPlay:WINDow:TRACe:Y:DLINe .....                         | 2-62 |
| :DISPlay:WINDow:TRACe:Y:DLINe:STATe .....                   | 2-63 |
| :DISPlay:WINDow:TRACe:Y[:SCALE]:NRLevel .....               | 2-63 |
| :DISPlay:WINDow:TRACe:Y[:SCALE]:NRPosition .....            | 2-64 |
| :DISPlay:WINDow:TRACe:Y[:SCALE]:PDI Vision .....            | 2-64 |
| :DISPlay:WINDow:TRACe:Y[:SCALE]:RLEVel .....                | 2-65 |
| :DISPlay:WINDow:TRACe:Y[:SCALE]:RLEVel:OFFSet .....         | 2-65 |
| :DISPlay:WINDow:TRACe:Y[:SCALE]:SPACing .....               | 2-66 |
| :FETCh Commands .....                                       | 2-67 |
| :FETCh:ACPower? .....                                       | 2-68 |
| :FETCh:ACPower:LOWer? .....                                 | 2-68 |
| :FETCh:ACPower:MAIN? .....                                  | 2-69 |
| :FETCh:ACPower:UPPer? .....                                 | 2-69 |
| :FETCh:CNRatio? .....                                       | 2-70 |
| :FETCh:CNRatio:CARRier? .....                               | 2-70 |
| :FETCh:CNRatio:CNRatio? .....                               | 2-71 |
| :FETCh:CNRatio:NOISe? .....                                 | 2-71 |
| :FETCh:EBWidth? .....                                       | 2-72 |
| :FETCh:HARMonics:AMPLitude:ALL? .....                       | 2-72 |
| :FETCh:HARMonics:AMPLitude? < n> .....                      | 2-73 |
| :FETCh:HARMonics[:DISTortion]? .....                        | 2-73 |
| :FETCh:HARMonics:FREQuency:ALL? .....                       | 2-74 |
| :FETCh:HARMonics:FREQuency? < n> .....                      | 2-74 |
| :FETCh:HARMonics:FUNDamental? .....                         | 2-75 |
| :FETCh:OBWidth? .....                                       | 2-75 |
| :FETCh:OBWidth:OBWidth? .....                               | 2-76 |
| :FETCh:OBWidth:OBWidth:FERRor? .....                        | 2-76 |
| :FETCh:SANalyzer< n> ? .....                                | 2-77 |
| :FETCh:TOIntercept? .....                                   | 2-77 |
| :FETCh:TOIntercept:IP3? .....                               | 2-78 |
| :FETCh:TPOWer? .....                                        | 2-78 |
| :FORMat Commands .....                                      | 2-79 |
| :FORMat:BORDER .....                                        | 2-79 |
| :FORMat[:TRACe][:DATA] .....                                | 2-80 |
| IEEE 488.2 Common Commands .....                            | 2-81 |
| *CLS .....                                                  | 2-82 |

|                                          |       |
|------------------------------------------|-------|
| * ESE.....                               | 2-82  |
| * ESR?.....                              | 2-83  |
| * IDN?.....                              | 2-83  |
| * OPC.....                               | 2-84  |
| * RCL.....                               | 2-84  |
| * RST.....                               | 2-84  |
| * SAV.....                               | 2-85  |
| * SRE.....                               | 2-85  |
| * STB?.....                              | 2-86  |
| * TRG.....                               | 2-86  |
| * TST?.....                              | 2-86  |
| * WAI.....                               | 2-86  |
| :INITiate Commands.....                  | 2-87  |
| :INITiate:CONTinuous.....                | 2-87  |
| :INITiate[:IMMediate].....               | 2-87  |
| :INSTrument Commands.....                | 2-88  |
| :INSTrument:COUPle:FREQUency:CENTer..... | 2-88  |
| :INSTrument:DEFault.....                 | 2-88  |
| :INSTrument:NSElect.....                 | 2-89  |
| :INSTrument[:SElect].....                | 2-89  |
| :MMEMory Commands.....                   | 2-90  |
| :MMEMory:DELeTe.....                     | 2-90  |
| :MMEMory:LOAD:FMT.....                   | 2-91  |
| :MMEMory:LOAD:LI Mit.....                | 2-91  |
| :MMEMory:LOAD:STATe.....                 | 2-92  |
| :MMEMory:LOAD:TRACe.....                 | 2-92  |
| :MMEMory:LOAD:TRACe:DATA.....            | 2-93  |
| :MMEMory:MOVE.....                       | 2-93  |
| :MMEMory:STORe:LI Mit.....               | 2-94  |
| :MMEMory:STORe:MTABLE.....               | 2-94  |
| :MMEMory:STORe:PTABLE.....               | 2-95  |
| :MMEMory:STORe:RESults.....              | 2-95  |
| :MMEMory:STORe:SCReen.....               | 2-95  |
| :MMEMory:STORe:STATe.....                | 2-96  |
| :MMEMory:STORe:TRACe.....                | 2-96  |
| :MMEMory:STORe:TRACe:DATA.....           | 2-97  |
| :OUTPut Commands.....                    | 2-98  |
| :OUTPut[:EXTeRnal][:STATe].....          | 2-98  |
| :READ Commands.....                      | 2-99  |
| :READ:ACPoweR?.....                      | 2-100 |
| :READ:ACPoweR:LOWer?.....                | 2-100 |
| :READ:ACPoweR:MAIN?.....                 | 2-100 |
| :READ:ACPoweR:UPPeR?.....                | 2-101 |
| :READ:CNRatio?.....                      | 2-101 |
| :READ:CNRatio:CARRier?.....              | 2-101 |
| :READ:CNRatio:CNRatio?.....              | 2-102 |
| :READ:CNRatio:NOISe?.....                | 2-102 |
| :READ:EBWidth?.....                      | 2-102 |
| :READ:HARMonics:AMPLitude:ALL?.....      | 2-103 |
| :READ:HARMonics:AMPLitude? < n>.....     | 2-103 |
| :READ:HARMonics[:DISToRtion]?.....       | 2-104 |
| :READ:HARMonics:FREQUency:ALL?.....      | 2-104 |
| :READ:HARMonics:FREQUency? < n>.....     | 2-104 |
| :READ:HARMonics:FUNDamental?.....        | 2-105 |
| :READ:OBWidth?.....                      | 2-105 |
| :READ:OBWidth:OBWidth?.....              | 2-106 |
| :READ:OBWidth:OBWidth:FERRor?.....       | 2-106 |
| :READ:SANalyzer< n> ?.....               | 2-107 |

|                                                            |       |
|------------------------------------------------------------|-------|
| :READ:TOIntercept?                                         | 2-107 |
| :READ:TOIntercept:IP3?                                     | 2-108 |
| :READ:TPower?                                              | 2-108 |
| [[:SENSe] Commands                                         | 2-109 |
| [:SENSe]:ACPower:AVERage:COUNT                             | 2-112 |
| [:SENSe]:ACPower:AVERage[:STATe]                           | 2-112 |
| [:SENSe]:ACPower:AVERage:TCONtrol                          | 2-113 |
| [:SENSe]:ACPower:BANDwidth:ACHannel                        | 2-113 |
| [:SENSe]:ACPower:BANDwidth:INTegration                     | 2-114 |
| [:SENSe]:ACPower:CSPacing                                  | 2-114 |
| [:SENSe]:ACQuisition:TIME                                  | 2-115 |
| [:SENSe]:ACQuisition:TIME:AUTO                             | 2-115 |
| [:SENSe]:ACQuisition:TIME:PVTime                           | 2-116 |
| [:SENSe]:ACQuisition:TIME:PVTime:AUTO                      | 2-117 |
| [:SENSe]:AVERage:COUNT                                     | 2-117 |
| :TRACe:AVERage:COUNT                                       | 2-117 |
| [:SENSe]:AVERage:COUNT:CURRent?                            | 2-118 |
| :TRACe:AVERage:COUNT:CURRent?                              | 2-118 |
| [:SENSe]:AVERage:TYPE                                      | 2-118 |
| :TRACe<n>:AVERage:TYPE                                     | 2-118 |
| [:SENSe]:AVERage:TYPE:AUTO                                 | 2-119 |
| [:SENSe]:BANDwidth BWIDth:EMI Filter:STATe                 | 2-119 |
| [:SENSe]:BANDwidth BWIDth[:RESolution]                     | 2-120 |
| [:SENSe]:BANDwidth BWIDth[:RESolution]:AUTO                | 2-120 |
| [:SENSe]:BANDwidth BWIDth[:RESolution]:SElect              | 2-121 |
| [:SENSe]:BANDwidth BWIDth[:RESolution]:SElect:AUTO[:STATe] | 2-122 |
| [:SENSe]:BANDwidth BWIDth:SHAPE                            | 2-122 |
| [:SENSe]:BANDwidth BWIDth:VIDeo                            | 2-123 |
| [:SENSe]:BANDwidth BWIDth:VIDeo:AUTO                       | 2-123 |
| [:SENSe]:BANDwidth BWIDth:VIDeo:RATio                      | 2-124 |
| [:SENSe]:BANDwidth BWIDth:VIDeo:RATio:AUTO                 | 2-125 |
| [:SENSe]:CNRatio:AVERage:COUNT                             | 2-125 |
| [:SENSe]:CNRatio:AVERage[:STATe]                           | 2-126 |
| [:SENSe]:CNRatio:AVERage:TCONtrol                          | 2-126 |
| [:SENSe]:CNRatio:BANDwidth:INTegration                     | 2-127 |
| [:SENSe]:CNRatio:BANDwidth:NOISe                           | 2-127 |
| [:SENSe]:CNRatio:OFFSet                                    | 2-128 |
| :INPut:IMPedance                                           | 2-129 |
| [:SENSe]:CORRection:IMPedance[:INPut][:MAGNitude]          | 2-129 |
| [:SENSe]:CORRection:SA[:RF]:GAIN                           | 2-129 |
| [:SENSe]:DEMod                                             | 2-130 |
| [:SENSe]:DEMod:GAIN:AUTO                                   | 2-130 |
| [:SENSe]:DEMod:GAIN:INCRement                              | 2-131 |
| [:SENSe]:DEMod:STATe                                       | 2-131 |
| [:SENSe]:DETector:TRACe:PVTime                             | 2-132 |
| [:SENSe]:DETector[:FUNCTION]                               | 2-133 |
| [:SENSe]:DETector:TRACe<n>                                 | 2-133 |
| [:SENSe]:DETector:TRACe<n>:AUTO                            | 2-134 |
| [:SENSe]:EBWidth:AVERage:COUNT                             | 2-134 |
| [:SENSe]:EBWidth:AVERage[:STATe]                           | 2-135 |
| [:SENSe]:EBWidth:AVERage:TCONtrol                          | 2-135 |
| [:SENSe]:EBWidth:FREQuency:SPAN                            | 2-136 |
| [:SENSe]:EBWidth:MAXHold:STATe                             | 2-136 |
| [:SENSe]:EBWidth:XDB                                       | 2-137 |
| [:SENSe]:FREQuency:CENTer                                  | 2-138 |
| [:SENSe]:FREQuency:CENTer:STEP:AUTO                        | 2-138 |
| [:SENSe]:FREQuency:CENTer:STEP[:INCRement]                 | 2-139 |
| [:SENSe]:FREQuency:OFFSet                                  | 2-139 |

|                                                                  |       |
|------------------------------------------------------------------|-------|
| [ :SENSe]:FREQuency:SPAN.....                                    | 2-140 |
| [ :SENSe]:FREQuency:SPAN:BANDwidth[:RESolution]:RATio .....      | 2-140 |
| [ :SENSe]:FREQuency:SPAN:BANDwidth[:RESolution]:RATio:AUTO ..... | 2-141 |
| [ :SENSe]:FREQuency:SPAN:FULL .....                              | 2-141 |
| [ :SENSe]:FREQuency:SPAN:PREVious .....                          | 2-141 |
| [ :SENSe]:FREQuency:SPAN:ZERO .....                              | 2-142 |
| [ :SENSe]:FREQuency:STARt .....                                  | 2-142 |
| [ :SENSe]:FREQuency:STOP.....                                    | 2-143 |
| [ :SENSe]:FREQuency:TUNE:IMMediate .....                         | 2-143 |
| [ :SENSe]:HDISt:AVERAge:COUNt .....                              | 2-143 |
| [ :SENSe]:HDISt:AVERAge[:STATe] .....                            | 2-144 |
| [ :SENSe]:HDISt:AVERAge:TCONtrol .....                           | 2-145 |
| [ :SENSe]:HDISt:NUMBers .....                                    | 2-145 |
| [ :SENSe]:HDISt:TIME.....                                        | 2-146 |
| [ :SENSe]:MCHPower:AVERAge:COUNt .....                           | 2-146 |
| [ :SENSe]:MCHPower:AVERAge[:STATe] .....                         | 2-147 |
| [ :SENSe]:MCHPower:AVERAge:TCONtrol .....                        | 2-147 |
| [ :SENSe]:OBWidth:AVERAge:COUNt .....                            | 2-148 |
| [ :SENSe]:OBWidth:AVERAge[:STATe] .....                          | 2-148 |
| [ :SENSe]:OBWidth:AVERAge:TCONtrol .....                         | 2-149 |
| [ :SENSe]:OBWidth:FREQuency:SPAN.....                            | 2-150 |
| [ :SENSe]:OBWidth:MAXHold:STATe .....                            | 2-150 |
| [ :SENSe]:OBWidth:PERCent .....                                  | 2-151 |
| [ :SENSe]:POWER[:RF]:ATTenuation .....                           | 2-151 |
| [ :SENSe]:POWER[:RF]:ATTenuation:AUTO .....                      | 2-152 |
| [ :SENSe]:POWER[:RF]:GAIN[:STATe] .....                          | 2-152 |
| [ :SENSe]:POWER[:RF]:MIxer:RANGe[:UPPer] .....                   | 2-153 |
| [ :SENSe]:SIGCapture:2FSK:AMPDown .....                          | 2-153 |
| [ :SENSe]:SIGCapture:2FSK:AMPUp.....                             | 2-154 |
| [ :SENSe]:SIGCapture:2FSK:MARK1:FREQ .....                       | 2-154 |
| [ :SENSe]:SIGCapture:2FSK:MARK1:SWitch[:STATe] .....             | 2-155 |
| [ :SENSe]:SIGCapture:2FSK:MARK2:FREQ .....                       | 2-155 |
| [ :SENSe]:SIGCapture:2FSK:MARK2:SWitch[:STATe] .....             | 2-156 |
| [ :SENSe]:SIGCapture:2FSK:MAXHold[:STATe] .....                  | 2-156 |
| [ :SENSe]:SIGCapture:2FSK:PEAKAmp? .....                         | 2-157 |
| [ :SENSe]:SIGCapture:2FSK:PEAKFreq? .....                        | 2-157 |
| [ :SENSe]:SIGCapture:2FSK:PF?.....                               | 2-158 |
| [ :SENSe]:SIGCapture:2FSK:PFSWitch[:STATe] .....                 | 2-158 |
| [ :SENSe]:SIGCapture:2FSK:RESet .....                            | 2-159 |
| [ :SENSe]:SIGCapture:2FSK:SIgNal .....                           | 2-159 |
| [ :SENSe]:SWEep:POINts.....                                      | 2-160 |
| [ :SENSe]:SWEep:TIME .....                                       | 2-160 |
| [ :SENSe]:SWEep:TIME:AUTO.....                                   | 2-161 |
| [ :SENSe]:SWEep:TIME:AUTO:RULes .....                            | 2-161 |
| [ :SENSe]:TOI:AVERAge:COUNt .....                                | 2-162 |
| [ :SENSe]:TOI:AVERAge[:STATe] .....                              | 2-162 |
| [ :SENSe]:TOI:AVERAge:TCONtrol .....                             | 2-163 |
| [ :SENSe]:TOI:FREQuency:SPAN .....                               | 2-163 |
| [ :SENSe]:TPOWer:AVERAge:COUNt .....                             | 2-164 |
| [ :SENSe]:TPOWer:AVERAge[:STATe] .....                           | 2-164 |
| [ :SENSe]:TPOWer:AVERAge:TCONtrol.....                           | 2-165 |
| [ :SENSe]:TPOWer:LLIMit .....                                    | 2-165 |
| [ :SENSe]:TPOWer:MODE .....                                      | 2-166 |
| [ :SENSe]:TPOWer:RLIMit .....                                    | 2-167 |
| :SOURce Commands .....                                           | 2-168 |
| :SOURce:CORRection:OFFSet .....                                  | 2-168 |
| :SOURce[:EXTernal]:POWER[:LEVel][:IMMediate][:AMPLitude] .....   | 2-168 |
| :SOURce:TRACe:REFerence:STATe.....                               | 2-169 |

|                                                                     |       |
|---------------------------------------------------------------------|-------|
| : SOURce: TRACe: STORref .....                                      | 2-169 |
| : STATus Commands .....                                             | 2-170 |
| : STATus: OPERation: CONDition? .....                               | 2-171 |
| : STATus: OPERation: ENABle .....                                   | 2-171 |
| : STATus: OPERation[ : EVEnt] ? .....                               | 2-172 |
| : STATus: PRESet .....                                              | 2-172 |
| : STATus: QUEStionable: CONDition? .....                            | 2-172 |
| : STATus: QUEStionable: ENABle .....                                | 2-172 |
| : STATus: QUEStionable[ : EVEnt] ? .....                            | 2-173 |
| : SYSTem Commands .....                                             | 2-174 |
| : SYSTem: BEEPer: STATe .....                                       | 2-175 |
| : SYSTem: COMMunicate: LAN[ : SELF] : APPLy .....                   | 2-175 |
| : SYSTem: COMMunicate: LAN[ : SELF] : AUTOip: STATe .....           | 2-175 |
| : SYSTem: COMMunicate: LAN[ : SELF] : DHCP: STATe .....             | 2-176 |
| : SYSTem: COMMunicate: LAN[ : SELF] : IP: ADDRess .....             | 2-176 |
| : SYSTem: COMMunicate: LAN[ : SELF] : IP: DNS: AUTO[ : STATe] ..... | 2-177 |
| : SYSTem: COMMunicate: LAN[ : SELF] : IP: DNSBack .....             | 2-178 |
| : SYSTem: COMMunicate: LAN[ : SELF] : IP: DNSPreferred .....        | 2-178 |
| : SYSTem: COMMunicate: LAN[ : SELF] : IP: DNSServer .....           | 2-178 |
| : SYSTem: COMMunicate: LAN[ : SELF] : IP: GATeway .....             | 2-179 |
| : SYSTem: COMMunicate: LAN[ : SELF] : IP: SUBMask .....             | 2-179 |
| : SYSTem: COMMunicate: LAN[ : SELF] : MANuip: STATe .....           | 2-180 |
| : SYSTem: COMMunicate: LAN[ : SELF] : MDNS: STATe .....             | 2-181 |
| : SYSTem: COMMunicate: LAN[ : SELF] : RESet .....                   | 2-181 |
| : SYSTem: CONFigure: INFormation? .....                             | 2-181 |
| : SYSTem: DATE .....                                                | 2-182 |
| : SYSTem: FSWitch[ : STATe] .....                                   | 2-182 |
| : SYSTem: LANGuage .....                                            | 2-183 |
| : SYSTem: LKEY .....                                                | 2-183 |
| : SYSTem: OPTion: STATe? .....                                      | 2-184 |
| : SYSTem: PON: TYPE .....                                           | 2-184 |
| : SYSTem: PRESet .....                                              | 2-185 |
| : SYSTem: PRESet: SAVE .....                                        | 2-185 |
| : SYSTem: PRESet: TYPE .....                                        | 2-185 |
| : SYSTem: PRESet: USER: SAVE .....                                  | 2-186 |
| : SYSTem: SCPI: DISPlay .....                                       | 2-186 |
| : SYSTem: SHOW .....                                                | 2-186 |
| : SYSTem: TIME .....                                                | 2-187 |
| : TRACe Commands .....                                              | 2-188 |
| : TRACe: CLear: ALL .....                                           | 2-188 |
| : TRACe[ : DATA] .....                                              | 2-188 |
| : TRACe: MATH: A .....                                              | 2-190 |
| : TRACe: MATH: B .....                                              | 2-190 |
| : TRACe: MATH: CONST .....                                          | 2-191 |
| : TRACe: MATH: PEAK[ : DATA] ? .....                                | 2-191 |
| : TRACe: MATH: PEAK: POINts? .....                                  | 2-191 |
| : TRACe: MATH: REFerence .....                                      | 2-192 |
| : TRACe: MATH: STATe .....                                          | 2-192 |
| : TRACe: MATH: TYPE .....                                           | 2-193 |
| : TRACe: PRESet: ALL .....                                          | 2-193 |
| : TRACe< n> : DISPlay: STATe .....                                  | 2-194 |
| : TRACe< n> : MODE .....                                            | 2-194 |
| : TRACe< n> : TYPE .....                                            | 2-194 |
| : TRACe< n> : UPDate: STATe .....                                   | 2-195 |
| : TRIgger Commands .....                                            | 2-196 |
| : TRIgger[ : SEQuence] : ATRigger .....                             | 2-197 |
| : TRIgger[ : SEQuence] : ATRigger: STATe .....                      | 2-197 |
| : TRIgger[ : SEQuence] : EXTernal< n> : DELay .....                 | 2-198 |

|                                                        |            |
|--------------------------------------------------------|------------|
| :TRI Gger[:SEQuence]:EXTeRnal<n>:DELay:STATe .....     | 2-198      |
| :TRI Gger[:SEQuence]:EXTeRnal<n>:SLOPe .....           | 2-199      |
| :TRI Gger[:SEQuence]:FMT:APTRigger .....               | 2-199      |
| :TRI Gger[:SEQuence]:FMT:CRITeria .....                | 2-200      |
| :TRI Gger[:SEQuence]:FMT:DELay .....                   | 2-201      |
| :TRI Gger[:SEQuence]:FMT:DELay:STATe .....             | 2-201      |
| :TRI Gger[:SEQuence]:FMT:MASK .....                    | 2-202      |
| :TRI Gger[:SEQuence]:FMT:MASK:EDIT .....               | 2-202      |
| :TRI Gger[:SEQuence]:FMT:MASK:RELative:AMPLitude ..... | 2-203      |
| :TRI Gger[:SEQuence]:FMT:MASK:RELative:FREQuency ..... | 2-203      |
| :TRI Gger[:SEQuence]:FMT:MASK<n>:BUIld .....           | 2-204      |
| :TRI Gger[:SEQuence]:FMT:MASK<n>:DATA .....            | 2-204      |
| :TRI Gger[:SEQuence]:FMT:MASK<n>:DELeTe .....          | 2-205      |
| :TRI Gger[:SEQuence]:FMT:MASK<n>:NEW .....             | 2-205      |
| :TRI Gger[:SEQuence]:HOLDoff .....                     | 2-206      |
| :TRI Gger[:SEQuence]:HOLDoff:STATe .....               | 2-206      |
| :TRI Gger[:SEQuence]:SOURce .....                      | 2-207      |
| :TRI Gger[:SEQuence]:VIDeo:DELay .....                 | 2-207      |
| :TRI Gger[:SEQuence]:VIDeo:DELay:STATe .....           | 2-208      |
| :TRI Gger[:SEQuence]:VIDeo:LEVel .....                 | 2-208      |
| :TRI Gger[:SEQuence]:VIDeo:SLOPe .....                 | 2-209      |
| :TRI Gger2:MODE .....                                  | 2-210      |
| :UNIT Command .....                                    | 2-211      |
| :UNIT:POWer .....                                      | 2-211      |
| <b>Chapter 3 Programming Examples .....</b>            | <b>3-1</b> |
| Programming Instructions .....                         | 3-2        |
| Programming Preparations .....                         | 3-2        |
| Visual C++ 6.0 Programming Example .....               | 3-3        |
| Visual Basic 6.0 Programming Example .....             | 3-11       |
| LabVIEW 2010 Programming Example .....                 | 3-15       |
| Linux Programming Example .....                        | 3-19       |
| Programming Preparations .....                         | 3-19       |
| Linux Programming Procedures .....                     | 3-22       |
| <b>Chapter 4 Appendix .....</b>                        | <b>4-1</b> |
| Appendix A: Default Settings .....                     | 4-1        |
| Appendix B: Warranty .....                             | 4-6        |

# Chapter 1 Programming Overview

This chapter introduces how to set up remote communication between the spectrum analyzer and the PC, the remote control methods, the syntax, symbols, parameters, and abbreviation rules of the SCPI commands.

## Contents in this chapter:

- ◆ [Build Remote Communication](#)
- ◆ [Remote Control Method](#)
- ◆ [SCPI Command Overview](#)

## Build Remote Communication

RSA3000 supports communication with PC via USB and LAN interface for remote control. This chapter introduces in details how to use Ultra Sigma to send commands through the USB interface to control the analyzer.

### Operation Procedures:

#### 1. Install Ultra Sigma (PC) software.

You can download Ultra Sigma from the official website of **RIGOL** ([www.rigol.com](http://www.rigol.com)) and install the software according to the installation wizard.

#### 2. Connect the analyzer to the PC and configure the interface parameters for the instrument

RSA3000 supports three communication interfaces: USB and LAN.

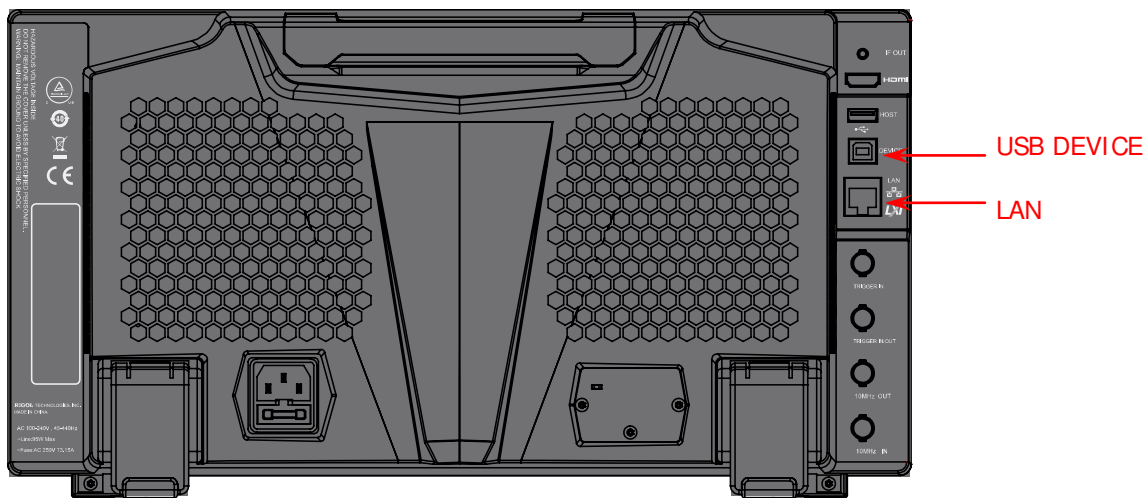


Figure 1-1 RSA3000 Communication Interface

##### (1) Remote communication via USB:

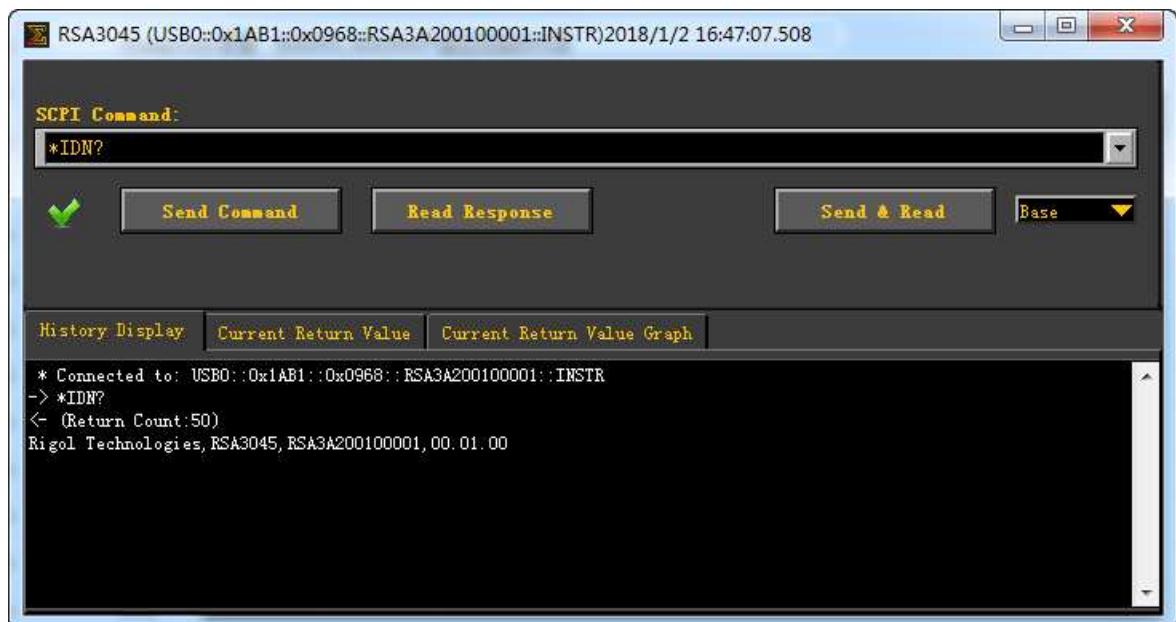
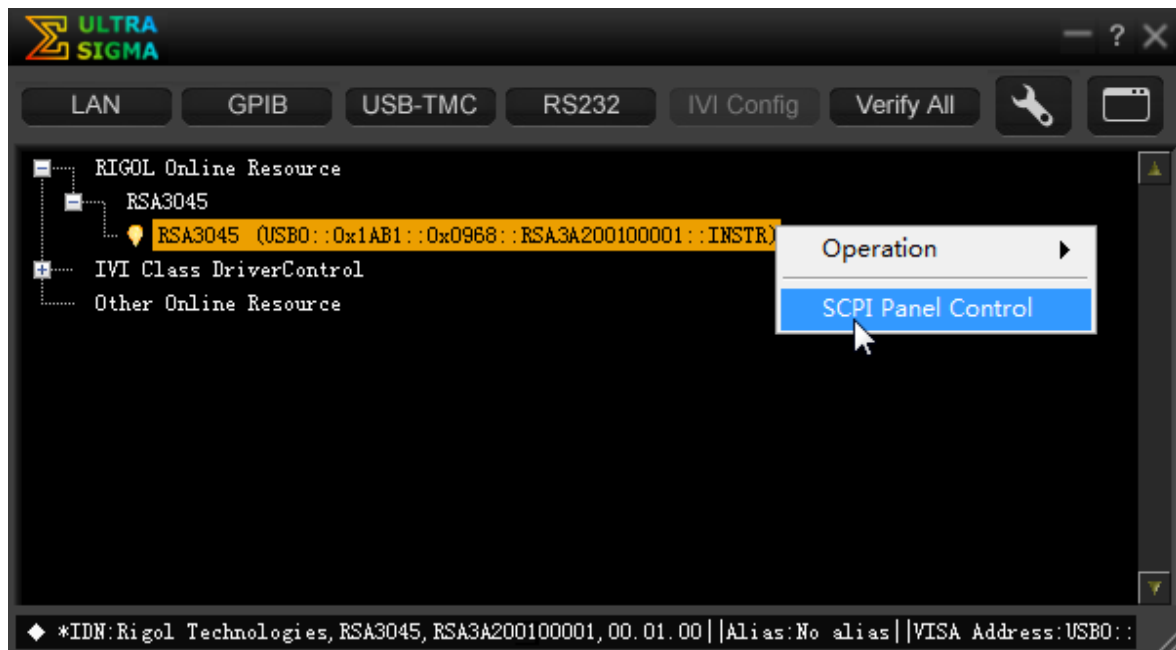
Use the USB cable to connect the USB DEVICE interface on the rear panel of RSA3000 to the USB HOST interface of the PC.

##### (2) Remote communication via LAN:

- Make sure that your PC has been accessed to the local area network.
- Check whether the local area network where your PC resides supports DHCP or auto IP mode. If not, you need to obtain the available network interface parameters, including the IP address, subnet mask, default gateway, and DNS.
- Connect RSA3000 to the local area network by using a network cable.
- Press **System** → **Interface** → **LAN** to configure the IP address, subnet mask, default gateway, and DNS for the instrument.

#### 3. Check whether the connection is successful.

Run Ultra Sigma, and then search for the resources and right-click the resource name. Select "SCPI Panel Control" to open the SCPI command control panel. In the SCPI command control panel, input a correct command. After that, click **Send Command** first and then click **Read Response**. Or you can also directly click **Send & Read** to verify whether the connection works properly. The following figure takes USB interface as an example to illustrate it.



## Remote Control Method

### 1. User-defined programming

You can refer to the SCPI (Standard Commands for Programmable Instruments) commands listed in Chapter 2 in this manual to control RSA3000 by programming in LabVIEW, Visual C++, and other development environments. For details, refer to descriptions in Chapter 4 in this manual.

### 2. Send SCPI commands via the PC software

You can use the PC software to send commands to control RSA3000 remotely. **RIGOL** Ultra Sigma is recommended. You can download the software from **RIGOL** official website ([www.rigol.com](http://www.rigol.com)).

## SCPI Command Overview

SCPI (Standard Commands for Programmable Instruments) is a standardized instrument programming language that is built upon the existing standard IEEE 488.1 and IEEE 488.2 and conforms to various standards, such as the floating point operation rule in IEEE 754 standard, ISO 646 7-bit coded character set for information interchange (equivalent to ASCII programming). This chapter introduces the syntax, symbols, parameters, and abbreviation rules of the SCPI commands.

### Syntax

The SCPI commands provide a hierarchical tree structure, and consist of multiple subsystems. Each command subsystem consists of one root keyword and one or more sub-keywords. The command line usually starts with ":"; the keywords are separated by ":" and are followed by the parameter settings available; "?" is added at the end of the command line to indicate a query; the commands and parameters are separated by spaces.

For example,

```
:CALCulate:BANDwidth:NDB <rel_ampl>
:CALCulate:BANDwidth:NDB?
```

CALCulate is the root keyword of the command. BANDwidth is the second-level keyword, and NDB is the third-level keyword. The command line starts with ":", and a colon is also used to separate the multiple-level keywords. <rel\_ampl> represents the parameters available for setting. "?" represents query. The command :CALCulate:BANDwidth:NDB and the parameter <rel\_ampl> are separated by a space.

In some commands with multiple parameters, "," is often used to separate these parameters. For example, :SYSTem:DATE <year>,<month>,<day>.

### Symbol Description

The following four symbols are not part of the SCPI command, and they are not sent with the commands, but taken as delimiters to better describe the parameters in the command.

#### 1. Braces { }

The parameters enclosed in the braces are optional. You do not have to set it, also you can set for one or more times.

#### 2. Vertical Bar |

The vertical bar is used to separate multiple parameters. When using the command, you must select one of the parameters. For example, in the command :CALCulate:MARKer<n>:LINes[:STATe] OFF|ON|0|1, you can select any of the four available parameters: OFF, ON, 0, or 1.

#### 3. Square Brackets [ ]

The content (command keyword) in the square brackets can be omitted. If a keyword is omitted, it will be set to the default. For example, for the [:SENSe]:POWer[:RF]:ATTenuation? command, sending any of the four commands below can generate the same effect:

```
:POWer:ATTenuation?
:POWer:RF:ATTenuation?
:SENSe:POWer:ATTenuation?
:SENSe:POWer:RF:ATTenuation?
```

#### 4. Angle Brackets < >

The parameter enclosed in the angle brackets must be replaced by an effective value. For example, send the :SENSe:FREQuency:CENTer 1000 command in [:SENSe]:FREQuency:CENTer <freq> format.

## Parameter Type

The command parameters introduced in this manual include 6 types: Bool, Keyword, Integer, Consecutive Real Number, Discrete, and ASCII String.

**1. Bool**

The parameter can be set to OFF (0) or ON (1).

**2. Keyword**

The parameter can be any of the values listed.

**3. Integer**

Unless otherwise specified, the parameter can be any integer within the effective value range. Note: Do not set the parameter to a decimal, otherwise, errors will occur.

**4. Consecutive Real Number**

Unless otherwise specified, the parameter could be any value within the effective value range according to the accuracy requirement (by default, there are 9 digits after the decimal point).

**5. Discrete**

The parameter could only be one of the specified values and these values are discontinuous.

**6. ASCII String**

The parameter can be the combinations of ASCII characters.

## Command Abbreviation

All the commands are case-insensitive. They can all be written in upper case or in lower case. For abbreviations, you only need to input all the upper-case letters in the command. For example, :CALCulate:BANDwidth:NDB? can be abbreviated as :CALC:BAND:NDB? or :calc:band:ndb?



## Chapter 2 Command System

This chapter introduces the commands of the RSA3000 series spectrum analyzer.

### Contents in this chapter:

- ◆ [:CALCulate Commands](#)
- ◆ [:CALibration Commands](#)
- ◆ [:CONFigure Commands](#)
- ◆ [:COUPle Commands](#)
- ◆ [:DISPlay Commands](#)
- ◆ [:FETCh Commands](#)
- ◆ [:FORMat Commands](#)
- ◆ [IEEE 488.2 Common Commands](#)
- ◆ [:INITiate Commands](#)
- ◆ [:INSTrument Commands](#)
- ◆ [:MMEMory Commands](#)
- ◆ [:OUTPut Commands](#)
- ◆ [:READ Commands](#)
- ◆ [\[:SENSe\] Commands](#)
- ◆ [:SOURce Commands](#)
- ◆ [:STATus Commands](#)
- ◆ [:SYSTem Commands](#)
- ◆ [:TRACe Commands](#)
- ◆ [:TRIgger Commands](#)
- ◆ [:UNIT Command](#)

### Remarks:

1. The commands concerning the advanced measurement are only available for the RSA3000 model installed with relevant options. For details, refer to remarks for each command subsystem.
2. The commands related to the tracking generator are only available for RSA3045-TG/ RSA3030-TG. They are only available for GPSA mode.
3. For the command set, unless otherwise specified, the query command returns "N/A" (without quotations in its return format) if no specified option is installed. If the queried function is disabled or improper type match is found, the query command will return "Error" (without quotations in its return format).
4. This manual takes RSA3045 as an example to illustrate the range of the parameters in each command.

## :CALCulate Commands

### Command List:

- ◆ [:CALCulate:BANDwidth|BWIDth:NDB](#)
- ◆ [:CALCulate:BANDwidth|BWIDth:RESult?](#)
- ◆ [:CALCulate:BANDwidth|BWIDth:RLEFt?](#)
- ◆ [:CALCulate:BANDwidth|BWIDth:RRIGht?](#)
- ◆ [:CALCulate:BANDwidth|BWIDth\[:STATe\]](#)
- ◆ [:CALCulate:LLINe:ALL:DELeTe](#)
- ◆ [:CALCulate:LLINe:TEST](#)
- ◆ [:CALCulate:LLINe<n>:AMPLitude:CMODE:RELative](#)
- ◆ [:CALCulate:LLINe<n>:BUILd](#)
- ◆ [:CALCulate:LLINe<n>:COPY](#)
- ◆ [:CALCulate:LLINe<n>:DATA](#)
- ◆ [:CALCulate:LLINe<n>:DELeTe](#)
- ◆ [:CALCulate:LLINe<n>:DISPlay](#)
- ◆ [:CALCulate:LLINe<n>:FAIL?](#)
- ◆ [:CALCulate:LLINe<n>:FREQuency:CMODE:RELative](#)
- ◆ [:CALCulate:LLINe<n>:MARGIn](#)
- ◆ [:CALCulate:LLINe<n>:MARGIn:STATe](#)
- ◆ [:CALCulate:LLINe<n>:TRACe](#)
- ◆ [:CALCulate:LLINe<n>:TYPE](#)
- ◆ [:CALCulate:MARKer:AOff](#)
- ◆ [:CALCulate:MARKer:COUPle\[:STATe\]](#)
- ◆ [:CALCulate:MARKer:PEAK:EXCursion](#)
- ◆ [:CALCulate:MARKer:PEAK:EXCursion:STATe](#)
- ◆ [:CALCulate:MARKer:PEAK:SEARch:MODE](#)
- ◆ [:CALCulate:MARKer:PEAK:SORT](#)
- ◆ [:TRACe:MATH:PEAK:SORT](#)
- ◆ [:CALCulate:MARKer:PEAK:TABLE:READout](#)
- ◆ [:CALCulate:MARKer:PEAK:TABLE\[:STATe\]](#)
- ◆ [:CALCulate:MARKer:PEAK:THReshold](#)
- ◆ [:CALCulate:MARKer:PEAK:THReshold:STATe](#)
- ◆ [:CALCulate:MARKer:TABLE\[:STATe\]](#)
- ◆ [:CALCulate:MARKer:TRCKing\[:STATe\]](#)
- ◆ [:CALCulate:MARKer<n>:CPSearch\[:STATe\]](#)
- ◆ [:CALCulate:MARKer<n>:FCOunt:GATetime](#)
- ◆ [:CALCulate:MARKer<n>:FCOunt:GATetime:AUTO](#)
- ◆ [:CALCulate:MARKer<n>:FCOunt\[:STATe\]](#)

- ◆ [:CALCulate:MARKer<n>:FCOUNT:X?](#)
- ◆ [:CALCulate:MARKer<n>:FUNCTION](#)
- ◆ [:CALCulate:MARKer<n>:FUNCTION:BAND:LEFT](#)
- ◆ [:CALCulate:MARKer<n>:FUNCTION:BAND:RIGHT](#)
- ◆ [:CALCulate:MARKer<n>:FUNCTION:BAND:SPAN](#)
- ◆ [:CALCulate:MARKer<n>:FUNCTION:BAND:SPAN:AUTO](#)
- ◆ [:CALCulate:MARKer<n>:LINES\[:STATE\]](#)
- ◆ [:CALCulate:MARKer<n>:MAXimum:LEFT](#)
- ◆ [:CALCulate:MARKer<n>:MAXimum\[:MAX\]](#)
- ◆ [:CALCulate:MARKer<n>:MAXimum:NEXT](#)
- ◆ [:CALCulate:MARKer<n>:MAXimum:RIGHT](#)
- ◆ [:CALCulate:MARKer<n>:MINimum](#)
- ◆ [:CALCulate:MARKer<n>:MODE](#)
- ◆ [:CALCulate:MARKer<n>:PTPeak](#)
- ◆ [:CALCulate:MARKer<n>:REFERENCE](#)
- ◆ [:CALCulate:MARKer<n>\[:SET\]:CENTER](#)
- ◆ [:CALCulate:MARKer<n>\[:SET\]:DELTA:CENTer](#)
- ◆ [:CALCulate:MARKer<n>\[:SET\]:DELTA:SPAN](#)
- ◆ [:CALCulate:MARKer<n>\[:SET\]:RLEVEL](#)
- ◆ [:CALCulate:MARKer<n>\[:SET\]:START](#)
- ◆ [:CALCulate:MARKer<n>\[:SET\]:STEP](#)
- ◆ [:CALCulate:MARKer<n>\[:SET\]:STOP](#)
- ◆ [:CALCulate:MARKer<n>:STATE](#)
- ◆ [:CALCulate:MARKer<n>:TRACE](#)
- ◆ [:CALCulate:MARKer<n>:TRACE:AUTO](#)
- ◆ [:CALCulate:MARKer<n>:X](#)
- ◆ [:CALCulate:MARKer<n>:X:READout](#)
- ◆ [:CALCulate:MARKer<n>:X:READout:AUTO](#)
- ◆ [:CALCulate:MARKer<n>:Y](#)
- ◆ [:CALCulate:MARKer<n>:Z:POSITION](#)
- ◆ [:CALCulate:MATH](#)
- ◆ [:CALCulate:NTData\[:STATE\]](#)

## :CALCulate:BANDwidth| BWIDth:NDB

### Syntax

:CALCulate:BANDwidth|BWIDth:NDB <rel\_ampl>  
:CALCulate:BANDwidth|BWIDth:NDB?

### Description

Sets the N value in N dB BW measurement.  
Queries the N value in N dB BW measurement.

### Parameter

| Name       | Type                    | Range               | Default  |
|------------|-------------------------|---------------------|----------|
| <rel_ampl> | Consecutive Real Number | -140 dB to -0.01 dB | -3.01 dB |

### Remarks

This command is invalid for the PVT measurement mode in RTSA.

### Return Format

The query returns the N value in scientific notation.

### Example

The following command sets N to -4.

:CALCulate:BANDwidth:NDB -4  
:CALCulate:BWIDth:NDB -4

The following query returns -4.000000e+00.

:CALCulate:BANDwidth:NDB?  
:CALCulate:BWIDth:NDB?

## :CALCulate:BANDwidth| BWIDth:RESult?

### Syntax

:CALCulate:BANDwidth|BWIDth:RESult?

### Description

Queries the measurement results of the N dB band, and its unit is Hz.

### Remarks

This command is invalid for the PVT measurement mode in RTSA.

### Return Format

The query returns the bandwidth value in scientific notation (with Hz as the unit).

If the two points which are located at both sides of the current marker with N dB fall or rise in its peak amplitude are not found, the query returns -1.000000000e+02.

## :CALCulate:BANDwidth| BWIDth:RLEFt?

### Syntax

:CALCulate:BANDwidth| BWIDth:RLEFt?

### Description

Queries the frequency value of the frequency point that is located at the left side of the current marker with N dB fall or rise in its peak amplitude.

### Remarks

This command is invalid for the PvT measurement mode in RTSA.

### Return Format

The query returns the frequency value in scientific notation (with Hz as the unit).

If the frequency point that is located at the left side of the current marker with N dB fall or rise in its peak amplitude is not found, the query returns -1.000000000e+02.

## :CALCulate:BANDwidth| BWIDth:RRI Ght?

### Syntax

:CALCulate:BANDwidth| BWIDth:RRI Ght?

### Description

Queries the frequency value of the frequency point that is located at the right side of the current marker with N dB fall or rise in its peak amplitude.

### Remarks

This command is invalid for the PvT measurement mode in RTSA.

### Return Format

The query returns the frequency value in scientific notation (with Hz as the unit).

If the frequency point that is located at the right side of the current marker with N dB fall or rise in its peak amplitude is not found, the query returns -1.000000000e+02.

## :CALCulate:BANDwidth| BWIDth [:STATe]

### Syntax

:CALCulate:BANDwidth| BWIDth[:STATe] OFF|ON|0|1

:CALCulate:BANDwidth| BWIDth[:STATe]?

### Description

Enables or disables the N dB bandwidth measurement function.

Queries the status of the N dB bandwidth measurement function.

### Parameter

| Name | Type | Range      | Default |
|------|------|------------|---------|
| —    | Bool | OFF ON 0 1 | OFF 0   |

### Remarks

This command is invalid for the PvT measurement mode in RTSA.

**Return Format**

The query returns 0 or 1.

**Example**

The following command enables the N dB BW measurement function.

:CALCulate:BANDwidth:STATe ON or :CALCulate:BANDwidth:STATe 1

:CALCulate:BWIDth:STATe ON or :CALCulate:BWIDth:STATe 1

The following query returns 1.

:CALCulate:BANDwidth:STATe?

:CALCulate:BWIDth:STATe?

**:CALCulate:LLINe:ALL:DELeTe****Syntax**

:CALCulate:LLINe:ALL:DELeTe

**Description**

Deletes all the limit line data.

**:CALCulate:LLINe:TEST****Syntax**

:CALCulate:LLINe:TEST OFF|ON|0|1

:CALCulate:LLINe:TEST?

**Description**

Enables or disables the limit line test function.

Queries the status of the limit line test function.

**Parameter**

| Name | Type | Range      | Default |
|------|------|------------|---------|
| —    | Bool | OFF ON 0 1 | ON 1    |

**Return Format**

The query returns 0 or 1.

**Example**

The following command disables the limit line test function.

:CALCulate:LLINe:TEST OFF or :CALCulate:LLINe:TEST 0

The following query returns 0.

:CALCulate:LLINe:TEST?

## :CALCulate:LLINe< n> :AMPLitude:CMODE:RELative

### Syntax

:CALCulate:LLINe< n> :AMPLitude:CMODE:RELative OFF|ON|0|1  
 :CALCulate:LLINe< n> :AMPLitude:CMODE:RELative?

### Description

Enables or disables the coupling functions between the selected limit line data points and the reference level.

Queries the status of the coupling functions between the selected limit line data points and the reference level.

### Parameter

| Name | Type     | Range       | Default |
|------|----------|-------------|---------|
| < n> | Discrete | 1 2 3 4 5 6 | ——      |
| ——   | Bool     | OFF ON 0 1  | OFF 0   |

### Remarks

OFF|0: disables the coupling function. That is, "Fixed" is selected under **Y to Ref**, and the amplitude of the current limit line editing point is not affected by the reference level.

ON|1: enables the coupling function. That is, "Relative" is selected under **Y to Ref**, and the amplitude of the current limit line editing point is the difference between the current point with the current reference level. At this time, if the reference level changes, the position of the current editing point will move up and down with it.

### Return Format

The query returns 0 or 1.

### Example

The following command disables the coupling functions between the selected limit line data points and the reference level.

:CALCulate:LLINe2:AMPLitude:CMODE:RELative OFF or :CALCulate:LLINe2:AMPLitude:CMODE:RELative 0

The following query returns 0.

:CALCulate:LLINe2:AMPLitude:CMODE:RELative?

## :CALCulate:LLINe< n> :BUI Ld

### Syntax

:CALCulate:LLINe< n> :BUI Ld TRACE1|TRACE2|TRACE3|TRACE4|TRACE5|TRACE6

### Description

Builds the limit line from the selected trace.

### Parameter

| Name | Type     | Range                                     | Default |
|------|----------|-------------------------------------------|---------|
| < n> | Discrete | 1 2 3 4 5 6                               | ——      |
| ——   | Keyword  | TRACE1 TRACE2 TRACE3 TRACE4 TRACE5 TRACE6 | ——      |

### Example

The following command builds Limit Line 2 from Trace2.

:CALCulate:LLINe2:BUI Ld TRACE2

## :CALCulate:LLINe< n> :COPY

### Syntax

:CALCulate:LLINe< n> :COPY LLINE1|LLINE2|LLINE3|LLINE4|LLINE5|LLINE6

### Description

Copies the selected limit line to the current limit line.

### Parameter

| Name | Type     | Range                                     | Default |
|------|----------|-------------------------------------------|---------|
| < n> | Discrete | 1 2 3 4 5 6                               | —       |
| —    | Keyword  | LLINE1 LLINE2 LLINE3 LLINE4 LLINE5 LLINE6 | —       |

### Remarks

If the limit line to be copied that you select is the same as the current limit line, no operation should be performed.

### Example

The following command copies Limit Line 1 to Limit Line 2.

```
:CALCulate:LLINe2:COPY LLINE1
```

## :CALCulate:LLINe< n> :DATA

### Syntax

:CALCulate:LLINe< n> :DATA < x> ,< ampl> ,< connect> { ,< x> ,< ampl> ,< connect> }

:CALCulate:LLINe< n> :DATA?

### Description

Edits one limit line, and marks it with n.

Queries the limit line data that you are editing currently.

### Parameter

| Name       | Type                    | Range                   | Default |
|------------|-------------------------|-------------------------|---------|
| < n>       | Discrete                | 1 2 3 4 5 6             | 1       |
| < x>       | Consecutive Real Number | 0 Hz to 4.5 GHz         | —       |
| < ampl>    | Consecutive Real Number | -1,000 dBm to 1,000 dBm | —       |
| < connect> | Discrete                | 0 1                     | 0       |

### Remarks

< x> : indicates frequency.

< apml> : indicates the amplitude. By default, its unit is dBm. The same X value can be configured with at most two amplitude values.

< connect> : can be configured with 0 or 1. When it is configured with 1, it indicates that the current point connects with the previous point to determine the limit line; when configured with 0, it indicates that the current point is disconnected from the previous point. The < connect> value of the first point can be configured with 0.

### Return Format

Queries the returned limit line data that you are editing currently.

### Example

The following command edits a limit line that contains three points, and marks it Limit Line 2.

```
:CALCulate:LLINe2:DATA 50,100,0,100,150,1,200,200,1
```

The following query returns 50,100,0,100,150,1,200,200,1.  
:CALCulate:LLINe2:DATA?

## :CALCulate:LLINe< n> :DELeTe

### Syntax

:CALCulate:LLINe< n> :DELeTe

### Description

Deletes the selected limit line.

### Parameter

| Name | Type     | Range       | Default |
|------|----------|-------------|---------|
| < n> | Discrete | 1 2 3 4 5 6 | —       |

### Example

The following command deletes Limit Line 2.

:CALCulate:LLINe2:DELeTe

## :CALCulate:LLINe< n> :DI SPlay

### Syntax

:CALCulate:LLINe< n> :DI SPlay OFF|ON|0|1

:CALCulate:LLINe< n> :DI SPlay?

### Description

Turns on or off the selected limit line.

Queries the status of the selected limit line.

### Parameter

| Name | Type     | Range       | Default |
|------|----------|-------------|---------|
| < n> | Discrete | 1 2 3 4 5 6 | 1       |
| —    | Bool     | OFF ON 0 1  | OFF 0   |

### Return Format

The query returns 0 or 1.

### Example

The following command turns on Limit Line 2.

:CALCulate:LLINe2:DI SPlay ON or :CALCulate:LLINe2:DI SPlay 1

The following query returns 1.

:CALCulate:LLINe2:DI SPlay?

## :CALCulate:LLINe< n> :FAIL?

### Syntax

:CALCulate:LLINe< n> :FAIL?

### Description

Queries the measurement results of the selected limit line and its associated trace.

### Parameter

| Name | Type     | Range       | Default |
|------|----------|-------------|---------|
| < n> | Discrete | 1 2 3 4 5 6 | —       |

### Remarks

If you enable the test for the displayed trace and its corresponding limit line, and each displayed trace has its corresponding limit line enabled, then you can run the command to query whether the trace test passes.

### Return Format

The query returns 0 (pass) or 1 (fail).

## :CALCulate:LLINe< n> :FREQUENCY:CMODE:RELative

### Syntax

:CALCulate:LLINe< n> :FREQUENCY:CMODE:RELative OFF|ON|0|1

:CALCulate:LLINe< n> :FREQUENCY:CMODE:RELative?

### Description

Enables or disables the coupling functions between the selected limit line data points and the center frequency.

Queries the status of the coupling functions between the selected limit line data points and the center frequency.

### Parameter

| Name | Type     | Range       | Default |
|------|----------|-------------|---------|
| < n> | Discrete | 1 2 3 4 5 6 | —       |
| —    | Bool     | OFF ON 0 1  | OFF 0   |

### Remarks

OFF|0: disables the coupling function. That is, "Fixed" is selected under **X to CF**, and the frequency of the current limit line editing point is not affected by the center frequency.

ON|1: enables the coupling function. That is, "Relative" is selected under **X to CF**, and the amplitude of the current limit line editing point is the difference between the current point with the current center frequency. At this time, if the center frequency changes, the position of the current editing point will move left and right with it.

### Return Format

The query returns 0 or 1.

### Example

The following command disables the coupling functions between data points of Limit Line 2 and the center frequency.

```
:CALCulate:LLINe2:FREQUENCY:CMODE:RELative OFF or :CALCulate:LLINe2:FREQUENCY:CMODE:RELative 0
```

The following query returns 0.

```
:CALCulate:LLINe2:FREQUENCY:CMODE:RELative?
```

## :CALCulate:LLINe< n> :MARGin

### Syntax

```
:CALCulate:LLINe<n> :MARGin <rel_ampl>
:CALCulate:LLINe<n> :MARGin?
```

### Description

Sets the margin for the selected limit line.  
Queries the margin for the selected limit line.

### Parameter

| Name       | Type                    | Range          | Default |
|------------|-------------------------|----------------|---------|
| <n>        | Discrete                | 1 2 3 4 5 6    | —       |
| <rel_ampl> | Consecutive Real Number | -40 dB to 0 dB | 0 dB    |

### Return Format

The query returns the margin for the limit line in scientific notation.

### Example

The following command sets the margin for Limit Line 2 to 1 dB.

```
:CALCulate:LLINe2:MARGin 1 dB
```

The following query returns 1.000000e+00.

```
:CALCulate:LLINe2:MARGin?
```

## :CALCulate:LLINe< n> :MARGin:STATe

### Syntax

```
:CALCulate:LLINe<n> :MARGin:STATe OFF|ON|0|1
:CALCulate:LLINe<n> :MARGin:STATe?
```

### Description

Enables or disables the margin for the selected limit line.  
Queries the status of the margin for the selected limit line.

### Parameter

| Name | Type     | Range       | Default |
|------|----------|-------------|---------|
| <n>  | Discrete | 1 2 3 4 5 6 | —       |
| —    | Bool     | OFF ON 0 1  | OFF 0   |

### Return Format

The query returns 0 or 1.

### Example

The following command turns on the margin for Limit Line 2.

```
:CALCulate:LLINe2:MARGin:STATe ON or :CALCulate:LLINe2:MARGin:STATe 1
```

The following query returns 1.

```
:CALCulate:LLINe2:MARGin:STATe?
```

## :CALCulate:LLI Ne< n> :TRACe

### Syntax

:CALCulate:LLI Ne< n> :TRACe 1|2|3|4|5|6

:CALCulate:LLI Ne< n> :TRACe?

### Description

Sets the trace to be tested against the current limit line.

Queries the trace to be tested against the current limit line.

### Parameter

| Name | Type     | Range       | Default            |
|------|----------|-------------|--------------------|
| < n> | Discrete | 1 2 3 4 5 6 | ——                 |
| ——   | Discrete | 1 2 3 4 5 6 | Refer to "Remarks" |

### Remarks

Limit Line 1 and 2: The default is Trace 1.

Limit Line 3 and 4: The default is Trace 2.

Limit Line 5 and 6: The default is Trace 3.

### Return Format

The query returns 1, 2, 3, 4, 5, or 6.

### Example

The following command sets that Trace 2 is tested against Limit Line 3.

:CALCulate:LLI Ne3:TRACe 2

The following query returns 2.

:CALCulate:LLI Ne3:TRACe?

## :CALCulate:LLI Ne< n> :TYPE

### Syntax

:CALCulate:LLI Ne< n> :TYPE UPPer|LOWer

:CALCulate:LLI Ne< n> :TYPE?

### Description

Sets the type of the specified limit line.

Queries the type of the specified limit line.

### Parameter

| Name | Type     | Range       | Default            |
|------|----------|-------------|--------------------|
| < n> | Discrete | 1 2 3 4 5 6 | ——                 |
| ——   | Keyword  | UPPer LOWer | Refer to "Remarks" |

### Remarks

Limit Line 1, Limit Line 3, and Limit Line 5 belong to the upper type; Limit Line 2, Limit Line 4, and Limit Line 6 belong to the lower type.

### Return Format

The query returns UPP or LOW.

### Example

The following command sets Limit Line 2 to the upper type.

:CALCulate:LLINe2:TYPE UPPer

The following query returns UPP.

:CALCulate:LLINe2:TYPE?

## :CALCulate:MARKer:AOff

### Syntax

:CALCulate:MARKer:AOff

### Description

Turns off all the enabled markers.

## :CALCulate:MARKer:COUPle[:STATe]

### Syntax

:CALCulate:MARKer:COUPle[:STATe] OFF|ON|0|1

:CALCulate:MARKer:COUPle[:STATe]?

### Description

Enables or disables the couple marker function.

Queries the state of the couple marker function.

### Parameter

| Name | Type | Range      | Default |
|------|------|------------|---------|
| —    | Bool | OFF ON 0 1 | OFF 0   |

### Remarks

When you enable the couple marker function, moving any marker will make other markers (that are not fixed or off) move along with it.

### Return Format

The query returns 0 or 1.

### Example

The following command disables the couple marker function.

:CALCulate:MARKer:COUPle:STATe OFF or :CALCulate:MARKer:COUPle:STATe 0

The following query returns 0.

:CALCulate:MARKer:COUPle:STATe?

## :CALCulate:MARKer:PEAK:EXCursion

### Syntax

:CALCulate:MARKer:PEAK:EXCursion <rel\_ampl>

:CALCulate:MARKer:PEAK:EXCursion?

### Description

Sets the peak offset. Its unit is dB.

Queries the peak offset.

**Parameter**

| Name       | Type                    | Range          | Default |
|------------|-------------------------|----------------|---------|
| <rel_ampl> | Consecutive Real Number | 0 dB to 100 dB | 6 dB    |

**Return Format**

The query returns the peak offset in scientific notation.

**Example**

The following command sets the peak offset to 12 dB.

```
:CALCulate:MARKer:PEAK:EXCursion 12
```

The following query returns 1.200000e+ 01.

```
:CALCulate:MARKer:PEAK:EXCursion?
```

**:CALCulate:MARKer:PEAK:EXCursion:STATe****Syntax**

```
:CALCulate:MARKer:PEAK:EXCursion:STATe OFF|ON|0|1
```

```
:CALCulate:MARKer:PEAK:EXCursion:STATe?
```

**Description**

Enables or disables the peak offset function.

Queries the status of the peak offset function.

**Parameter**

| Name | Type | Range      | Default |
|------|------|------------|---------|
| —    | Bool | OFF ON 0 1 | ON 1    |

**Return Format**

The query returns 0 or 1.

**Example**

The following command enables the peak offset function.

```
:CALCulate:MARKer:PEAK:EXCursion:STATe ON or :CALCulate:MARKer:PEAK:EXCursion:STATe 1
```

The following query returns 1.

```
:CALCulate:MARKer:PEAK:EXCursion:STATe?
```

**:CALCulate:MARKer:PEAK:SEARCh:MODE****Syntax**

```
:CALCulate:MARKer:PEAK:SEARCh:MODE PARameter|MAXimum
```

```
:CALCulate:MARKer:PEAK:SEARCh:MODE?
```

**Description**

Sets the peak search mode.

Queries the peak search mode.

**Parameter**

| Name | Type    | Range             | Default |
|------|---------|-------------------|---------|
| —    | Keyword | PARameter MAXimum | MAXimum |

**Remarks**

- PARAmeter: indicates parameter. If "parameter" is selected under search mode, the system will search for the peak of the specified parameter on the trace.
- MAXimum: indicates maximum. If "maximum" is selected under search mode, the system will search for the maximum value on the trace.

The command is only valid for the peak search executed by sending the `:CALCulate:MARKer<n>:MAXimum[:MAX]` command. Other items under the peak search menu such as Next Peak, Next Peak Right, Next Peak Left, and Minimum Peak are all searched based on "parameter".

**Return Format**

The query returns PAR or MAX.

**Example**

The following command sets the peak search mode to "parameter".

```
:CALCulate:MARKer:PEAK:SEARch:MODE PARAmeter
```

The following query returns PAR.

```
:CALCulate:MARKer:PEAK:SEARch:MODE?
```

## **:CALCulate:MARKer:PEAK:SORT :TRACe:MATH:PEAK:SORT**

**Syntax**

```
:CALCulate:MARKer:PEAK:SORT FREQuency| AMPLitude
```

```
:CALCulate:MARKer:PEAK:SORT?
```

```
:TRACe:MATH:PEAK:SORT FREQuency| AMPLitude
```

```
:TRACe:MATH:PEAK:SORT?
```

**Description**

Sets the sorting order of the data displayed in the peak table.

Queries the sorting order of the data displayed in the peak table.

**Parameter**

| Name | Type    | Range                | Default   |
|------|---------|----------------------|-----------|
| —    | Keyword | FREQuency  AMPLitude | AMPLitude |

**Remarks**

FREQuency: lists the peaks in order of ascending frequency.

AMPLitude: lists the peaks in order of descending amplitude.

**Return Format**

The query returns FREQ or AMPL.

**Example**

The following command sets the data in the peak table to be sorted in descending amplitude.

```
:CALCulate:MARKer:PEAK:SORT AMPLitude
```

```
:TRACe:MATH:PEAK:SORT AMPLitude
```

The following query returns AMPL.

```
:CALCulate:MARKer:PEAK:SORT?
```

```
:TRACe:MATH:PEAK:SORT?
```

## **:CALCulate:MARKer:PEAK:TABLE:READout :TRACe:MATH:PEAK:THReshold**

### **Syntax**

```
:CALCulate:MARKer:PEAK:TABLE:READout ALL|GTDLine|LTDLine
:CALCulate:MARKer:PEAK:TABLE:READout?
:TRACe:MATH:PEAK:THReshold NORMal|DLMore|DLLess
:TRACe:MATH:PEAK:THReshold?
```

### **Description**

Sets the peak criteria that the displayed peak must meet.  
Queries the peak criteria that the displayed peak must meet.

### **Parameter**

| Name | Type    | Range                                       | Default |
|------|---------|---------------------------------------------|---------|
| —    | Keyword | ALL GTDLine LTDLine<br>NORMal DLMore DLLess | ALL     |

### **Remarks**

ALL|NORMal: lists all the peaks defined by the peak criteria, and displays data based on the current sorting order in the peak table.  
GTDLine|DLMore: lists the peaks that are greater than the defined display line and that meet the peak criteria.  
LTDLine|DLLess: lists the peaks that are smaller than the defined display line and that meet the peak criteria.

### **Return Format**

The query returns ALL (NORM), GTDL (DLM), or LTDL (DLL).

### **Example**

The following command lists all the peaks that are greater than the defined display line and that meet the peak criteria.

```
:CALCulate:MARKer:PEAK:TABLE:READout GTDLine
:TRACe:MATH:PEAK:THReshold DLMore
```

The following query returns GTDL.

```
:CALCulate:MARKer:PEAK:TABLE:READout?
```

The following query returns DLM.

```
:TRACe:MATH:PEAK:THReshold?
```

## **:CALCulate:MARKer:PEAK:TABLE[:STATe] :TRACe:MATH:PEAK:TABLE:STATe**

### **Syntax**

```
:CALCulate:MARKer:PEAK:TABLE[:STATe] OFF|ON|0|1
:CALCulate:MARKer:PEAK:TABLE[:STATe]?
:TRACe:MATH:PEAK:TABLE:STATe OFF|ON|0|1
:TRACe:MATH:PEAK:TABLE:STATe?
```

### **Description**

Enables or disables the peak table.  
Queries the status of the peak table.

**Parameter**

| Name | Type | Range      | Default |
|------|------|------------|---------|
| —    | Bool | OFF ON 0 1 | OFF 0   |

**Return Format**

The query returns 0 or 1.

**Example**

The following command enables the peak table.

```
:CALCulate:MARKer:PEAK:TABLE:STATe ON or :CALCulate:MARKer:PEAK:TABLE:STATe 1
```

```
:TRACe:MATH:PEAK:TABLE:STATe ON or :TRACe:MATH:PEAK:TABLE:STATe 1
```

The following query returns 1.

```
:CALCulate:MARKer:PEAK:TABLE:STATe?
```

```
:TRACe:MATH:PEAK:TABLE:STATe?
```

**:CALCulate:MARKer:PEAK:THReshold****Syntax**

```
:CALCulate:MARKer:PEAK:THReshold < ampl>
```

```
:CALCulate:MARKer:PEAK:THReshold?
```

**Description**

Sets the peak threshold, and its default unit is dBm.

Queries the peak threshold.

**Parameter**

| Name    | Type                    | Range             | Default |
|---------|-------------------------|-------------------|---------|
| < ampl> | Consecutive Real Number | -200 dBm to 0 dBm | -90 dBm |

**Return Format**

The query returns the peak threshold in scientific notation.

**Example**

The following command sets the peak threshold to -100 dB.

```
:CALCulate:MARKer:PEAK:THReshold -100
```

The following query returns -1.000000e+02.

```
:CALCulate:MARKer:PEAK:THReshold?
```

**:CALCulate:MARKer:PEAK:THReshold:STATe****Syntax**

```
:CALCulate:MARKer:PEAK:THReshold:STATe OFF|ON|0|1
```

```
:CALCulate:MARKer:PEAK:THReshold:STATe?
```

**Description**

Enables or disables the peak threshold function.

Queries the status of the peak threshold function.

**Parameter**

| Name | Type | Range      | Default |
|------|------|------------|---------|
| —    | Bool | OFF ON 0 1 | ON 1    |

**Return Format**

The query returns 0 or 1.

**Example**

The following command enables the peak threshold function.

:CALCulate:MARKer:PEAK:THReshold:STATe ON or :CALCulate:MARKer:PEAK:THReshold:STATe 1

The following query returns 1.

:CALCulate:MARKer:PEAK:THReshold:STATe?

**:CALCulate:MARKer:TABLE[:STATe]****Syntax**

:CALCulate:MARKer:TABLE[:STATe] OFF|ON|0|1

:CALCulate:MARKer:TABLE[:STATe]?

**Description**

Enables or disables the marker table.

Queries the status of the marker table.

**Parameter**

| Name | Type | Range      | Default |
|------|------|------------|---------|
| —    | Bool | OFF ON 0 1 | OFF 0   |

**Return Format**

The query returns 0 or 1.

**Example**

The following command disables the marker table.

:CALCulate:MARKer:TABLE:STATe OFF or :CALCulate:MARKer:TABLE:STATe 0

The following query returns 0.

:CALCulate:MARKer:TABLE:STATe?

**:CALCulate:MARKer:TRCKing[:STATe]****Syntax**

:CALCulate:MARKer:TRCKing[:STATe] OFF|ON|0|1

:CALCulate:MARKer:TRCKing[:STATe]?

**Description**

Enables or disables the signal track.

Queries the status of the signal track.

**Parameter**

| Name | Type | Range      | Default |
|------|------|------------|---------|
| —    | Bool | OFF ON 0 1 | OFF 0   |

**Remarks**

When the signal track is enabled, the instrument will execute a peak search after each sweep, and set the frequency value at the current peak to the center frequency to keep the signal always displayed at the center of the screen.

This command is only available for GPSA mode.

**Return Format**

The query returns 0 or 1.

**Example**

The following command enables the signal track.

:CALCulate:MARKer:TRCKing:STATe ON or :CALCulate:MARKer:TRCKing:STATe 1

The following query returns 1.

:CALCulate:MARKer:TRCKing:STATe?

**:CALCulate:MARKer< n> :CPSearch[:STATe]****Syntax**

:CALCulate:MARKer< n> :CPSearch[:STATe] OFF|ON|0|1

:CALCulate:MARKer< n> :CPSearch[:STATe]?

**Description**

Enables or disables continuous peak search function, and marks the peak value with n.

Queries the status of continuous peak search function.

**Parameter**

| Name | Type     | Range           | Default |
|------|----------|-----------------|---------|
| < n> | Discrete | 1 2 3 4 5 6 7 8 | —       |
| —    | Bool     | OFF ON 0 1      | OFF 0   |

**Return Format**

The query returns 0 or 1.

**Example**

The following command performs the continuous peak search function, and marks with Marker 1.

:CALCulate:MARKer1:CPSeatch:STATe ON or :CALCulate:MARKer1:CPSeatch:STATe 1

The following query returns 1.

:CALCulate:MARKer1:CPSeatch:STATe?

## :CALCulate:MARKer< n> :FCOunt:GATetime

### Syntax

```
:CALCulate:MARKer< n> :FCOunt:GATetime < time>
:CALCulate:MARKer< n> :FCOunt:GATetime?
```

### Description

Sets the gate time for Marker n. The unit is s.  
Queries the gate time for Marker n.

### Parameter

| Name    | Type                    | Range               | Default |
|---------|-------------------------|---------------------|---------|
| < n>    | Discrete                | 1 2 3 4 5 6 7 8     | —       |
| < time> | Consecutive Real Number | 1 $\mu$ s to 500 ms | 100 ms  |

### Remarks

This command is only available for GPSA mode.

### Return Format

The query returns the gate time for Marker n in scientific notation.

### Example

The following command sets the gate time for Marker 2 to 10 ms.

```
:CALCulate:MARKer2:FCOunt:GATetime 0.01
```

The following query returns 1.000000e-02.

```
:CALCulate:MARKer2:FCOunt:GATetime?
```

## :CALCulate:MARKer< n> :FCOunt:GATetime:AUTO

### Syntax

```
:CALCulate:MARKer< n> :FCOunt:GATetime:AUTO OFF|ON|0|1
:CALCulate:MARKer< n> :FCOunt:GATetime:AUTO?
```

### Description

Turns on or off the auto gate time for Marker n.  
Queries the status of the auto gate time for Marker n.

### Parameter

| Name | Type     | Range           | Default |
|------|----------|-----------------|---------|
| < n> | Discrete | 1 2 3 4 5 6 7 8 | —       |
| —    | Bool     | OFF ON 0 1      | OFF 0   |

### Remarks

This command is only available for GPSA mode.

### Return Format

The query returns 1 or 0.

### Example

The following command enables the auto gate time for Marker 2.

```
:CALCulate:MARKer2:FCOunt:GATetime:AUTO ON or :CALCulate:MARKer2:FCOunt:GATetime:AUTO 1
```

The following query returns 1.

```
:CALCulate:MARKer2:FCOunt:GATetime:AUTO?
```

## :CALCulate:MARKer< n> :FCOunt[:STATe]

### Syntax

```
:CALCulate:MARKer< n> :FCOunt[:STATe] OFF|ON|0|1
:CALCulate:MARKer< n> :FCOunt[:STATe] ?
```

### Description

Enables or disables the frequency counter function of the specified marker.  
Queries the status of the frequency counter function of the specified marker.

### Parameter

| Name | Type     | Range           | Default |
|------|----------|-----------------|---------|
| < n> | Discrete | 1 2 3 4 5 6 7 8 | —       |
| —    | Bool     | OFF ON 0 1      | OFF 0   |

### Remarks

This command is only available for GPSA mode.

### Return Format

The query returns 0 or 1.

### Example

The following command enables the frequency counter function of Marker 2.

```
:CALCulate:MARKer2:FCOunt:STATe ON or :CALCulate:MARKer2:FCOunt:STATe 1
```

The following query returns 1.

```
:CALCulate:MARKer2:FCOunt:STATe?
```

## :CALCulate:MARKer< n> :FCOunt:X?

### Syntax

```
:CALCulate:MARKer< n> :FCOunt:X?
```

### Description

Queries the readout of the frequency counter of the specified marker.

### Parameter

| Name | Type     | Range           | Default |
|------|----------|-----------------|---------|
| < n> | Discrete | 1 2 3 4 5 6 7 8 | —       |

### Remarks

This command is only available for GPSA mode.

### Return Format

The query returns the readout in integer. Its unit is Hz.

When the frequency counter function is disabled, the command returns NAN.

## :CALCulate:MARKer< n> :FUNCTion

### Syntax

:CALCulate:MARKer< n> :FUNCTion NOISe|BPOWer|BDENSity| OFF  
 :CALCulate:MARKer< n> :FUNCTion?

### Description

Selects the special measurement type for the specified marker.  
 Queries the special measurement type of the specified marker.

### Parameter

| Name | Type     | Range                      | Default |
|------|----------|----------------------------|---------|
| < n> | Discrete | 1 2 3 4 5 6 7 8            | —       |
| —    | Keyword  | NOISe BPOWer BDENSity  OFF | OFF     |

### Remarks

NOISe: indicates the noise measurement.  
 BPOWer: indicates the bandwidth power.  
 BDENSity: indicates the bandwidth density.  
 OFF: disables the measurements.

### Return Format

The query returns NOIS, BPOW, BDEN, or OFF.

### Example

The following command sets the measurement type of Marker 1 to noise measurement.

```
:CALCulate:MARKer1:FUNCTion NOISe
```

The following query returns NOIS.

```
:CALCulate:MARKer1:FUNCTion?
```

## :CALCulate:MARKer< n> :FUNCTion:BAND:LEFT

### Syntax

:CALCulate:MARKer< n> :FUNCTion:BAND:LEFT < freq>  
 :CALCulate:MARKer< n> :FUNCTion:BAND:LEFT?

### Description

Sets the left edge frequency or time of the signal involved in the calculation for the band function.  
 Queries the left edge frequency or time of the signal involved in the calculation for the band function.

### Parameter

| Name    | Type                    | Range           | Default                                                                                                  |
|---------|-------------------------|-----------------|----------------------------------------------------------------------------------------------------------|
| < n>    | Discrete                | 1 2 3 4 5 6 7 8 | --                                                                                                       |
| < freq> | Consecutive Real Number | 0 to band right | center frequency-bandwidth/2<br>capture time/2-bandwidth/2 (PvT)<br>sweep time/2-bandwidth/2 (zero span) |

### Remarks

This command is only valid when the corresponding bandwidth function is enabled.  
 This command is used to set the left edge time when the zero span is enabled in GPSA mode or when in the PvT view in RTSA mode.

### Return Format

The query returns the left edge frequency or time of the signal in scientific notation.

**Example**

The following command sets the left edge frequency of the signal involved in the calculation for the Marker 1 band function to 2 MHz.

```
:CALCulate:MARKer1:FUNCTION:BAND:LEFT 2000000
```

The following query returns 2.000000000e+06.

```
:CALCulate:MARKer1:FUNCTION:BAND:LEFT?
```

**:CALCulate:MARKer< n> :FUNCTION:BAND:RI GHt****Syntax**

```
:CALCulate:MARKer< n> :FUNCTION:BAND:RI GHt < freq>
```

```
:CALCulate:MARKer< n> :FUNCTION:BAND:RI GHt?
```

**Description**

Sets the right edge frequency or time of the signal involved in the calculation for the band function.

Queries the right edge frequency or time of the signal involved in the calculation for the band function.

**Parameter**

| Name    | Type                    | Range                  | Default                                                                                                     |
|---------|-------------------------|------------------------|-------------------------------------------------------------------------------------------------------------|
| < n>    | Discrete                | 1 2 3 4 5 6 7 8        | --                                                                                                          |
| < freq> | Consecutive Real Number | band left to $+\infty$ | center frequency+ bandwidth/2<br>capture time/2+ bandwidth/2 (PvT)<br>sweep time/2+ bandwidth/2 (zero span) |

**Remarks**

This command is only valid when the corresponding bandwidth function is enabled.

This command is used to set the right edge time when the zero span is enabled in GPSA mode or when in the PvT view in RTSA mode.

**Return Format**

The query returns the right edge frequency or time of the signal in scientific notation.

**Example**

The following command sets the right edge frequency of the signal involved in the calculation for the Marker 1 band function to 4 GHz.

```
:CALCulate:MARKer1:FUNCTION:BAND:RI GHt 4000000000
```

The following query returns 4.000000000e+09.

```
:CALCulate:MARKer1:FUNCTION:BAND:RI GHt?
```

## :CALCulate:MARKer< n> :FUNCTION:BAND:SPAN

### Syntax

:CALCulate:MARKer< n> :FUNCTION:BAND:SPAN <freq>

:CALCulate:MARKer< n> :FUNCTION:BAND:SPAN?

### Description

Sets the bandwidth or the time span of the signal involved in the calculation for the band function.

Queries the bandwidth or the time span of the signal involved in the calculation for the band function.

### Parameter

| Name   | Type                    | Range           | Default                                                      |
|--------|-------------------------|-----------------|--------------------------------------------------------------|
| < n>   | Discrete                | 1 2 3 4 5 6 7 8 | --                                                           |
| <freq> | Consecutive Real Number | 0 to + $\infty$ | span/2<br>capture time/20 (PvT)<br>sweep time/20 (zero span) |

### Remarks

This command is only valid when the corresponding bandwidth function is enabled.

This command is used to set the signal time when the zero span is enabled in GPSA mode or when in the PvT view in RTSA mode.

### Return Format

The query returns the bandwidth of the signal involved in the calculation for the band function in scientific notation.

### Example

The following command sets the bandwidth of the signal involved in the calculation for the Marker 1 band function to 500 MHz.

```
:CALCulate:MARKer1:FUNCTION:BAND:SPAN 500000000
```

The following query returns 5.000000000e+08.

```
:CALCulate:MARKer1:FUNCTION:BAND:SPAN?
```

## :CALCulate:MARKer< n> :FUNCTION:BAND:SPAN:AUTO

### Syntax

:CALCulate:MARKer< n> :FUNCTION:BAND:SPAN:AUTO OFF|ON|0|1

:CALCulate:MARKer< n> :FUNCTION:BAND:SPAN:AUTO?

### Description

Enables or disables the band span auto function.

Queries the status of band span auto function.

### Parameter

| Name | Type     | Range           | Default |
|------|----------|-----------------|---------|
| < n> | Discrete | 1 2 3 4 5 6 7 8 | --      |
| --   | Bool     | OFF ON 0 1      | ON 1    |

### Remarks

This command is only valid when the corresponding bandwidth function is enabled.

**Return Format**

The query returns 1 or 0.

**Example**

The following command enables the band span auto function of Marker 1.

```
:CALCulate:MARKer1:FUNCTION:BAND:SPAN:AUTO ON
```

```
or :CALCulate:MARKer1:FUNCTION:BAND:SPAN:AUTO 1
```

The following query returns 1.

```
:CALCulate:MARKer1:FUNCTION:BAND:SPAN:AUTO?
```

**:CALCulate:MARKer< n> :LI Nes[:STATe]****Syntax**

```
:CALCulate:MARKer< n> :LI Nes[:STATe] OFF|ON|0|1
```

```
:CALCulate:MARKer< n> :LI Nes[:STATe] ?
```

**Description**

Enables or disables the marker line of the specified marker.

Queries the status for the marker line of the specified marker.

**Parameter**

| Name | Type     | Range           | Default |
|------|----------|-----------------|---------|
| < n> | Discrete | 1 2 3 4 5 6 7 8 | ——      |
| ——   | Bool     | OFF ON 0 1      | OFF     |

**Remarks**

If the marker is not visible in the selected area, enable the marker line function to extend the marker line to the display area for better observation.

The PvT only supports setting the marker on Trace 1.

**Return Format**

The query returns 0 or 1.

**Example**

The following command enables the marker line of Marker 1.

```
:CALCulate:MARKer1:LI Nes:STATe ON or :CALCulate:MARKer1:LI Nes:STATe 1
```

The following query returns 1.

```
:CALCulate:MARKer1:LI Nes:STATe?
```

**:CALCulate:MARKer< n> :MAXimum:LEFT****Syntax**

```
:CALCulate:MARKer< n> :MAXimum:LEFT
```

**Description**

Searches for and marks the nearest peak which is located at the left side of the current peak and meets the peak search condition.

**Parameter**

| Name | Type     | Range           | Default |
|------|----------|-----------------|---------|
| < n> | Discrete | 1 2 3 4 5 6 7 8 | —       |

**Remarks**

When no peak is found, a prompt message "No peak found" is displayed on the screen.

**Example**

The following command performs one left peak search, and marks with Marker 2.

```
:CALCulate:MARKer2:MAXimum:LEFT
```

**:CALCulate:MARKer< n> :MAXimum[:MAX]****Syntax**

```
:CALCulate:MARKer< n> :MAXimum[:MAX]
```

**Description**

Performs one peak search based on the search mode set by the [:CALCulate:MARKer:PEAK:SEARch:MODE](#) command and marks it with the specified marker.

**Parameter**

| Name | Type     | Range           | Default |
|------|----------|-----------------|---------|
| < n> | Discrete | 1 2 3 4 5 6 7 8 | —       |

**Remarks**

When no peak is found, a prompt message "No peak found" is displayed on the screen.

**Example**

The following command performs one peak search, and marks with Marker 2.

```
:CALCulate:MARKer2:MAXimum:MAX
```

**:CALCulate:MARKer< n> :MAXimum:NEXT****Syntax**

```
:CALCulate:MARKer< n> :MAXimum:NEXT
```

**Description**

Searches for and marks the peak whose amplitude on the trace is next to that of the current peak and which meets the peak search condition.

**Parameter**

| Name | Type     | Range           | Default |
|------|----------|-----------------|---------|
| < n> | Discrete | 1 2 3 4 5 6 7 8 | —       |

**Remarks**

When no peak is found, a prompt message "No peak found" is displayed on the screen.

**Example**

The following command performs one next peak search, and marks with Marker 2.

```
:CALCulate:MARKer2:MAXimum:NEXT
```

## :CALCulate:MARKer< n> :MAXimum:RI GHt

### Syntax

:CALCulate:MARKer< n> :MAXimum:RI GHt

### Description

Searches for and marks the nearest peak which is located at the right side of the current peak and meets the peak search condition.

### Parameter

| Name | Type     | Range           | Default |
|------|----------|-----------------|---------|
| < n> | Discrete | 1 2 3 4 5 6 7 8 | —       |

### Remarks

When no peak is found, a prompt message "No peak found" is displayed on the screen.

### Example

The following command performs one right peak search, and marks with Marker 2.

:CALCulate:MARKer2:MAXimum:RI GHt

## :CALCulate:MARKer< n> :MI Nimum

### Syntax

:CALCulate:MARKer< n> :MI Nimum

### Description

Searches for and marks the peak with the minimum amplitude on the trace.

### Parameter

| Name | Type     | Range           | Default |
|------|----------|-----------------|---------|
| < n> | Discrete | 1 2 3 4 5 6 7 8 | —       |

### Remarks

When no peak is found, a prompt message "No peak found" is displayed on the screen.

### Example

The following command performs one minimum search, and marks it with Marker 2.

:CALCulate:MARKer2:MI Nimum

## :CALCulate:MARKer< n> :MODE

### Syntax

:CALCulate:MARKer< n> :MODE POSition| DELTa| FI Xed| OFF

:CALCulate:MARKer< n> :MODE?

### Description

Sets the type of the specified marker.

Queries the type of the specified marker.

### Parameter

| Name | Type     | Range                        | Default |
|------|----------|------------------------------|---------|
| < n> | Discrete | 1 2 3 4 5 6 7 8              | —       |
| —    | Keyword  | POSition  DELTa  FI Xed  OFF | OFF     |

### Remarks

POSition: indicates the normal marker.

DELTA: indicates difference between two data points.

FIXed: indicates that the marker is fixed.

OFF: turns off the selected marker.

The PvT only supports setting the marker on Trace 1.

### Return Format

The query returns POS, DELT, FIX, or OFF.

### Example

The following command sets the type of Marker 1 to Position.

```
:CALCulate:MARKer1:MODE POSition
```

The following query returns POS.

```
:CALCulate:MARKer1:MODE?
```

## :CALCulate:MARKer< n> :PTPeak

### Syntax

:CALCulate:MARKer< n> :PTPeak

### Description

Performs the peak-peak search and marks it with a specified marker.

### Parameter

| Name | Type     | Range           | Default |
|------|----------|-----------------|---------|
| < n> | Discrete | 1 2 3 4 5 6 7 8 | —       |

### Remarks

After the command is executed, the marker mode of the specified marker selects "Delta" automatically. The peak search results will be marked by the reference marker (by default, the next marker), and the minimum search will be marked by the Delta marker.

### Example

The following command performs the peak-peak search, and marks the peak-peak position with the reference marker (Marker 2) and the Delta marker (Marker 1), respectively.

```
:CALCulate:MARKer1:PTPeak
```

## :CALCulate:MARKer< n> :REFErrence

### Syntax

```
:CALCulate:MARKer< n> :REFErrence < integer>
:CALCulate:MARKer< n> :REFErrence?
```

### Description

Sets the reference marker for the specified marker.  
Queries the reference marker for the specified marker.

### Parameter

| Name       | Type     | Range           | Default                                                                                         |
|------------|----------|-----------------|-------------------------------------------------------------------------------------------------|
| < n>       | Discrete | 1 2 3 4 5 6 7 8 | —                                                                                               |
| < integer> | Integer  | 1 to 8          | By default, the marker following the specified marker is considered to be the reference marker. |

### Remarks

Each marker can have another marker to be its reference marker.  
If the current marker is a Delta marker, the measurement result of the marker will be determined by the reference marker.  
Any marker cannot have itself to be the reference marker.  
The PvT only supports setting the marker on Trace 1.

### Example

The following command sets the reference marker for Marker 1 to 2.  
:CALCulate:MARKer1:REFErrence 2

The following query returns 2.  
:CALCulate:MARKer1:REFErrence?

## :CALCulate:MARKer< n> [:SET]:CENTer

### Syntax

```
:CALCulate:MARKer< n> [:SET]:CENTer
```

### Description

Sets the frequency of the specified marker to the center frequency of the analyzer.

### Parameter

| Name | Type     | Range           | Default |
|------|----------|-----------------|---------|
| < n> | Discrete | 1 2 3 4 5 6 7 8 | —       |

### Remarks

If the marker mode of the specified marker is Position or Fixed, the center frequency will be set to the frequency of the marker.  
If the specified marker mode is Delta, the center frequency will be set to the frequency of the Delta marker.  
This function is invalid in zero span.  
This command is invalid for the PvT measurement mode in RTSA.

### Example

The following command sets the frequency at Marker 1 (Position) to the center frequency.  
:CALCulate:MARKer1:SET:CENTer

## :CALCulate:MARKer< n> [:SET]:DELTA:CENTer

### Syntax

:CALCulate:MARKer< n> [:SET]:DELTA:CENTer

### Description

Sets the frequency difference of the specified Delta marker to the center frequency of the analyzer.

### Parameter

| Name | Type     | Range           | Default |
|------|----------|-----------------|---------|
| < n> | Discrete | 1 2 3 4 5 6 7 8 | —       |

### Remarks

It is only valid when the current marker mode is "Delta".

This function is invalid in zero span.

This command is invalid for the PVT measurement mode in RTSA.

### Example

The following command sets the frequency difference of the Delta Marker 1 to the center frequency of the analyzer.

```
:CALCulate:MARKer1:SET:DELTA:CENTer
```

## :CALCulate:MARKer< n> [:SET]:DELTA:SPAN

### Syntax

:CALCulate:MARKer< n> [:SET]:DELTA:SPAN

### Description

Sets the frequency difference of the specified Delta marker to the span of the analyzer.

### Parameter

| Name | Type     | Range           | Default |
|------|----------|-----------------|---------|
| < n> | Discrete | 1 2 3 4 5 6 7 8 | —       |

### Remarks

It is only valid when the current marker mode is "Delta".

This function is invalid in zero span.

This command is invalid for the PVT measurement mode in RTSA.

### Example

The following command sets the frequency difference of the Delta Marker 1 to the span of the analyzer.

```
:CALCulate:MARKer1:SET:DELTA:SPAN
```

## :CALCulate:MARKer< n> [:SET]:RLEVel

### Syntax

:CALCulate:MARKer< n> [:SET]:RLEVel

### Description

Sets the amplitude of the specified marker to the reference level of the analyzer.

### Parameter

| Name | Type     | Range           | Default |
|------|----------|-----------------|---------|
| < n> | Discrete | 1 2 3 4 5 6 7 8 | —       |

### Remarks

If the marker mode of the specified marker is Position or Fixed, the reference level will be set to the amplitude of the marker.

If the specified marker mode is Delta and the current marker is the reference marker, then the reference level is set to the amplitude of the reference marker; if the current marker is the Delta marker, then the reference level is set to the amplitude of the Delta marker.

### Example

The following command sets the amplitude of Marker 2 (Position) to the reference level of the analyzer.

```
:CALCulate:MARKer2:SET:RLEVel
```

## :CALCulate:MARKer< n> [:SET]:START

### Syntax

:CALCulate:MARKer< n> [:SET]:START

### Description

Sets the frequency of the specified marker to the start frequency of the analyzer.

### Parameter

| Name | Type     | Range           | Default |
|------|----------|-----------------|---------|
| < n> | Discrete | 1 2 3 4 5 6 7 8 | —       |

### Remarks

If the marker mode of the specified marker is Position or Fixed, the start frequency will be set to the frequency of the marker.

If the specified marker mode is Delta, the start frequency will be set to the frequency of the Delta marker.

This function is invalid in zero span.

This command is invalid for the PvT measurement mode in RTSA.

### Example

The following command sets the frequency of Marker 3 (Position) to the start frequency of the analyzer.

```
:CALCulate:MARKer3:SET:START
```

## :CALCulate:MARKer< n> [:SET]:STEP

### Syntax

:CALCulate:MARKer< n> [:SET]:STEP

### Description

Sets the frequency of the specified marker to the center frequency step of the analyzer.

### Parameter

| Name | Type     | Range           | Default |
|------|----------|-----------------|---------|
| < n> | Discrete | 1 2 3 4 5 6 7 8 | —       |

### Remarks

If the marker mode of the specified marker is Position or Fixed, the center frequency step will be set to the frequency of the marker.

If the specified marker mode is Delta, the center frequency step will be set to the frequency difference between the Delta marker and the reference marker.

This function is invalid in zero span.

This command is invalid for the PVT measurement mode in RTSA.

### Example

The following command sets the frequency of Marker 4 (Position) to the center frequency step of the analyzer.

```
:CALCulate:MARKer4:SET:STEP
```

## :CALCulate:MARKer< n> [:SET]:STOP

### Syntax

:CALCulate:MARKer< n> [:SET]:STOP

### Description

Sets the frequency of the specified marker to the stop frequency of the analyzer.

### Parameter

| Name | Type     | Range           | Default |
|------|----------|-----------------|---------|
| < n> | Discrete | 1 2 3 4 5 6 7 8 | —       |

### Remarks

If the marker mode of the specified marker is Position or Fixed, the stop frequency will be set to the frequency of the marker.

If the specified marker mode is Delta, the stop frequency will be set to the frequency of the Delta marker.

This function is invalid in zero span.

This command is invalid for the PVT measurement mode in RTSA.

### Example

The following command sets the frequency of Marker 2 (Position) to the stop frequency of the analyzer.

```
:CALCulate:MARKer2:SET:STOP
```

## :CALCulate:MARKer< n> :STATe

### Syntax

```
:CALCulate:MARKer< n> :STATe OFF|ON|0|1
:CALCulate:MARKer< n> :STATe?
```

### Description

Enables or disables the specified marker.  
Queries the status for the specified marker.

### Parameter

| Name | Type     | Range           | Default |
|------|----------|-----------------|---------|
| < n> | Discrete | 1 2 3 4 5 6 7 8 | —       |
| —    | Bool     | OFF ON 0 1      | OFF 0   |

### Remarks

If the specified marker is disabled currently, by default, the marker mode is "Position" when you send the command to enable the marker.  
The PvT only supports setting the marker on Trace 1.

### Return Format

The query returns 0 or 1.

### Example

The following command enables Marker 1.  
:CALCulate:MARKer1:STATe ON or :CALCulate:MARKer1:STATe 1

The following query returns 1.  
:CALCulate:MARKer1:STATe?

## :CALCulate:MARKer< n> :TRACe

### Syntax

```
:CALCulate:MARKer< n> :TRACe <integer>
:CALCulate:MARKer< n> :TRACe?
```

### Description

Sets the marker trace for the specified marker.  
Queries the marker trace for the specified marker.

### Parameter

| Name       | Type     | Range           | Default |
|------------|----------|-----------------|---------|
| < n>       | Discrete | 1 2 3 4 5 6 7 8 | —       |
| < integer> | Discrete | 1 2 3 4 5 6     | 1       |

### Remarks

< integer> indicates the marker trace, and its range is from Trace 1 to Trace 6. The selected trace must be enabled.

You can also run the [:CALCulate:MARKer< n> :TRACe:AUTO](#) command to set the marker trace of the specified marker to "Auto".

The PvT does not support this command.

**Return Format**

The query returns any integer from 1 to 6.

If the marker trace is set to "Auto", the query command returns a trace number on which the marker is placed.

**Example**

The following command sets the marker trace of Marker 1 to Trace 2.

:CALCulate:MARKer1:TRACe 2

The following query returns 2.

:CALCulate:MARKer1:TRACe?

**:CALCulate:MARKer< n> :TRACe:AUTO****Syntax**

:CALCulate:MARKer< n> :TRACe:AUTO OFF|ON|0|1

:CALCulate:MARKer< n> :TRACe:AUTO?

**Description**

Enables or disables the auto trace marking of the specified marker.

Queries the status of the auto trace marking of the specified marker.

**Parameter**

| Name | Type     | Range           | Default |
|------|----------|-----------------|---------|
| < n> | Discrete | 1 2 3 4 5 6 7 8 | —       |
| —    | Bool     | OFF ON 0 1      | ON 1    |

**Remarks**

When you disable the auto marking of the trace, the currently enabled marker will stay on the corresponding trace.

The PvT does not support this command.

**Return Format**

The query returns 0 or 1.

**Example**

The following command sets the marker trace of Marker 1 to Auto.

:CALCulate:MARKer1:TRACe:AUTO ON or :CALCulate:MARKer1:TRACe:AUTO 1

The following query returns 1.

:CALCulate:MARKer1:TRACe:AUTO?

## :CALCulate:MARKer< n> :X

### Syntax

```
:CALCulate:MARKer< n> :X < param>
:CALCulate:MARKer< n> :X?
```

### Description

Sets the X-axis value of the specified marker. Its default unit is Hz.  
Queries the X-axis value of the specified marker.

### Parameter

| Name     | Type                    | Range                       | Default |
|----------|-------------------------|-----------------------------|---------|
| < n>     | Discrete                | 1 2 3 4 5 6 7 8             | —       |
| < param> | Consecutive Real Number | Refer to " <b>Remarks</b> " | —       |

### Remarks

If the readout mode is frequency, it cannot exceed the frequency range of the sweep and the units available are Hz (default), kHz, MHz, and GHz.

If the readout mode is time, it cannot exceed the time range of the sweep and the units available are s (default), μs, ms, and ks.

< param> can be any value within the available range of the current X axis.

If the marker mode of the specified marker is Position or Fixed, this command sets the X value of the marker.

If the specified marker mode is Delta, this command sets the X value of the delta marker relative to the reference marker.

The PvT only supports setting the marker on Trace 1.

### Return Format

The query returns the X-axis value of the marker in scientific notation.

### Example

The following command sets the X-axis value of Marker 1 to 150 MHz (the readout mode is frequency).

```
:CALCulate:MARKer1:X 150000000
```

The following query returns 1.500000000e+08.

```
:CALCulate:MARKer1:X?
```

## :CALCulate:MARKer< n> :X:READout

### Syntax

```
:CALCulate:MARKer< n> :X:READout FREQuency| TIME| ITIME| PERiod
:CALCulate:MARKer< n> :X:READout?
```

### Description

Sets the readout mode of the X axis of the specified marker.

Queries the readout mode of the X axis of the specified marker.

### Parameter

| Name | Type     | Range                          | Default                     |
|------|----------|--------------------------------|-----------------------------|
| < n> | Discrete | 1 2 3 4 5 6 7 8                | —                           |
| —    | Keyword  | FREQuency  TIME  ITIME  PERiod | Refer to " <b>Remarks</b> " |

**Remarks**

FREQuency: indicates frequency. It is the default readout mode in non-zero span mode.

TIME: indicates time. It is the default readout mode in zero span mode.

ITIMe: indicates the reciprocal of time. It is available only in zero span mode and a Delta marker is selected.

PERiod: indicates period. It is not available in zero span mode.

The PvT does not support this command.

**Return Format**

The query returns FREQ, TIME, ITIM, or PER.

**Example**

The following command sets the readout mode of the X axis of Marker 1 to "Time".

```
:CALCulate:MARKer1:X:READout TIME
```

The following query returns TIME.

```
:CALCulate:MARKer1:X:READout?
```

**:CALCulate:MARKer< n> :X:READout:AUTO****Syntax**

```
:CALCulate:MARKer< n> :X:READout:AUTO OFF|ON|0|1
```

```
:CALCulate:MARKer< n> :X:READout:AUTO?
```

**Description**

Enables or disables the auto readout mode of the specified marker.

Queries the auto readout mode of the specified marker.

**Parameter**

| Name | Type     | Range           | Default |
|------|----------|-----------------|---------|
| < n> | Discrete | 1 2 3 4 5 6 7 8 | —       |
| —    | Bool     | OFF ON 0 1      | ON 1    |

**Remarks**

When the auto readout mode is enabled, if the marker trace changes, the readout mode will be re-determined based on the destination trace.

The PvT does not support this command.

**Return Format**

The query returns 0 or 1.

**Example**

The following command enables the auto readout mode of Marker 1 in the X-axis.

```
:CALCulate:MARKer1:X:READout:AUTO ON or :CALCulate:MARKer1:X:READout:AUTO 1
```

The following query returns 1.

```
:CALCulate:MARKer1:X:READout:AUTO?
```

## :CALCulate:MARKer< n> :Y

### Syntax

```
:CALCulate:MARKer< n> :Y < amp>
:CALCulate:MARKer< n> :Y?
```

### Description

Sets the Y-axis value of the specified fixed marker.  
Queries the Y-axis value of the specified marker, and its default unit is dBm.

### Parameter

| Name   | Type                    | Range              | Default |
|--------|-------------------------|--------------------|---------|
| < n>   | Discrete                | 1 2 3 4 5 6 7 8    | —       |
| < amp> | Consecutive Real Number | -170 dBm to 30 dBm | —       |

### Remarks

This setting command is only available when there is only one marker and it is a fixed marker.  
If the marker mode of the specified marker is Position or Fixed, the query command queries the Y value of the marker.  
If the marker mode of the specified marker is Delta, the query command queries the Y-axis difference between the reference marker and the Delta marker.  
The PvT only supports setting the marker on Trace 1.

### Return Format

The query returns the Y-axis value in scientific notation.

### Example

The following command sets the Y-axis value of Marker 1 is -59.6 dBm.  
:CALCulate:MARKer1:Y -59.6

The following query returns -5.960000000e+ 01.  
:CALCulate:MARKer1:Y?

## :CALCulate:MARKer< n> :Z:POSition

### Syntax

```
:CALCulate:MARKer< n> :Z:POSition < integer>
:CALCulate:MARKer< n> :Z:POSition?
```

### Description

Sets the trace number of the trace where the marker stays in the Spectrogram view.  
Queries the trace number of the trace where the marker stays in the Spectrogram view.

### Parameter

| Name       | Type     | Range           | Default |
|------------|----------|-----------------|---------|
| < n>       | Discrete | 1 2 3 4 5 6 7 8 | —       |
| < integer> | Integer  | 1 to 8192       | 1       |

### Remarks

Only when you select the Spectrogram view of the Spectrogram, Density Spectrogram or PvT Spectrogram in RTSA mode, can this command be effective.

### Return Format

The query returns the trace number in integer.

**Example**

The following command sets the trace number of the trace where the marker1 stays in to 100.

```
:CALCulate:MARKer1:Z:POSition 100
```

The following query returns 100.

```
:CALCulate:MARKer1:Z:POSition?
```

**:CALCulate:MATH****Syntax**

```
:CALCulate:MATH
```

```
<trace_destination>,<function_math>,<trace1_operation>,<trace2_operation>,<rel_offset>,<rel_reference>
```

```
:CALCulate:MATH? TRACE1|TRACE2|TRACE3|TRACE4|TRACE5|TRACE6
```

**Description**

Sets mathematical operations between traces and, in some cases, user-defined offsets.

Queries the math operation function of the specified trace.

**Parameter**

| Name                | Type                    | Range                                             | Default |
|---------------------|-------------------------|---------------------------------------------------|---------|
| <trace_destination> | Keyword                 | TRACE1 TRACE2 TRACE3 TRACE4 TRACE5 TRACE6         | —       |
| <function_math>     | Keyword                 | PDIFference PSUM LOFFset LMOFFset LDIFference OFF | OFF     |
| <trace1_operation>  | Keyword                 | TRACE1 TRACE2 TRACE3 TRACE4 TRACE5 TRACE6         | TRACE5  |
| <trace2_operation>  | Keyword                 | TRACE1 TRACE2 TRACE3 TRACE4 TRACE5 TRACE6         | TRACE6  |
| <rel_offset>        | Consecutive Real Number | -100 dB to 100 dB                                 | 0 dB    |
| <rel_reference>     | Consecutive Real Number | -170 dBm to 30 dBm                                | 0 dBm   |

**Remarks**

PDIFference: indicates the power difference between Operand 1 and Operand 2.

PSUM: indicates the power sum between Operand 1 and Operand 2.

LOFFset: indicates the sum between Operand 1 and the offset parameter <rel\_offset>.

LMOFFset: indicates the difference between Operand 1 and the offset parameter <rel\_offset>.

LDIFference: indicates that Operand 1 (Op1) minus Operand 2 (Op2), and then plus the reference parameter value <rel\_reference>.

OFF: disables the operation function.

<rel\_offset> indicates the log offset, and its unit is dB; <rel\_reference> indicates the log reference, and its unit is dBm.

For certain operation, if some parameters are irrelevant in the operation, you can replace these parameters with "," in the command. The returned parameters are separated by commas. The returned results of the irrelevant parameters are undefined. The empty fields are replaced by ",".

**Return Format**

The query returns the operation function of the specified trace, namely, PDIF, PSUM, LOFF, LMOFF, LDIF, or OFF.

**Example**

The following command sets Trace 1 to the power difference math operation, and sets Operand Trace 1 to Trace 4 and Operand Trace 2 to Trace 5.

:CALCulate:MATH TRACE1,PDIFference,TRACE4,TRACE5,,

The following query returns PDIFference.

:CALCulate:MATH? TRACE1

## :CALCulate:NTData[:STATe]

### Syntax

:CALCulate:NTData[:STATe] OFF|ON|0|1

:CALCulate:NTData[:STATe]?

### Description

Enables or disables the normalize function.

Queries the status of normalization.

### Parameter

| Name | Type | Range      | Default |
|------|------|------------|---------|
| —    | Bool | OFF ON 0 1 | OFF 0   |

### Remarks

This command is only available for RSA3045-TG/RSA3030-TG working in GPSA mode.

This command is only valid when the tracking generator function is enabled.

### Return Format

The query returns 0 or 1.

### Example

The following command enables normalization.

:CALCulate:NTData:STATe ON or :CALCulate:NTData:STATe 1

The following query returns 1.

:CALCulate:NTData:STATe?

## :CALibration Commands

### Command List:

- ◆ [:CALibration:\[ALL\]](#)
- ◆ [:CALibration:AUTO](#)

### :CALibration:[ALL]

#### Syntax

:CALibration:[ALL]

#### Description

Executes self-calibration immediately.

#### Remarks

If the self-calibration succeeded, the command :CALibration:[ALL]? returns 0; if failed, the command :CALibration:[ALL]? returns 1.

#### Example

The following command executes the self-calibration immediately.  
:CALibration:ALL

### :CALibration:AUTO

#### Syntax

:CALibration:AUTO OFF|ON|0|1  
:CALibration:AUTO?

#### Description

Enables or disables auto calibration.  
Query the setting status of auto calibration.

#### Parameter

| Name | Type | Range      | Default |
|------|------|------------|---------|
| ——   | Bool | OFF ON 0 1 | OFF 0   |

#### Return Format

The query returns 0 or 1.

#### Example

The following command enables auto calibration.  
:CALibration:AUTO ON or :CALibration:AUTO 1

The following query returns 1.  
:CALibration:AUTO?

## :CONFigure Commands

### Command List:

- ◆ [:CONFigure?](#)
- ◆ [:CONFigure:ACPower\\*](#)
- ◆ [:CONFigure:CNRatio\\*](#)
- ◆ [:CONFigure:DENSity](#)
- ◆ [:CONFigure:DSPEctrogram](#)
- ◆ [:CONFigure:EBWidth\\*](#)
- ◆ [:CONFigure:HDISt\\*](#)
- ◆ [:CONFigure:LPSTep](#)
- ◆ [:CONFigure:MCHPower\\*](#)
- ◆ [:CONFigure:NORMal](#)
- ◆ [:CONFigure:OBWidth\\*](#)
- ◆ [:CONFigure:PSGRam](#)
- ◆ [:CONFigure:PSpectrum](#)
- ◆ [:CONFigure:PVT](#)
- ◆ [:CONFigure:SANalyzer](#)
- ◆ [:CONFigure:SPEctrogram](#)
- ◆ [:CONFigure:TOI\\*](#)
- ◆ [:CONFigure:TPOWer\\*](#)

### Remarks:

The commands with \* are only available for the RSA3000 series that has been installed with advanced measurement kit option.

## :CONFigure?

### Syntax

:CONFigure?

### Description

Queries the current measurement function.

### Return Format

The query returns SAN, TPOW, ACP, MCHP, OBW, EBW, CNR, HARM, TOI, NORM, DENS, SPEC, DSPE, PVT, PSP, or PSGR. The explanations for the return results are as follows: SAN (swept SA), TPOW (time-domain power), ACP (adjacent power), MCHP (multi-channel power), OBW (occupied bandwidth), EBW (emission bandwidth), CNR (carrier/noise ratio), HARM (harmonics distortion), TOI (third-order intercept), NORM (Normal), DENS (Density), SPEC (Spectrogram), DSPE (Density Spectrogram), PVT (power versus time), PSP (PVT Spectrum), PSGR (PVT Spectrogram).

## **:CONFigure:ACPower**

### **Syntax**

:CONFigure:ACPower

### **Description**

Sets the analyzer to be in the ACP measurement state.

### **Remarks**

This command is only available for GPSA mode.

Running this command will discontinue the current measurement and restore the specified measurement of the instrument to the default factory state.

After changing any measurement setup, running the [:READ Commands](#) will initialize the measurement, and you do not have to set the measurement to the default.

## **:CONFigure:CNRatio**

### **Syntax**

:CONFigure:CNRatio

### **Description**

Sets the analyzer to be in the CNR measurement state.

### **Remarks**

This command is only available for GPSA mode.

Running this command will discontinue the current measurement and restore the specified measurement of the instrument to the default factory state.

After changing any measurement setup, running the [:READ Commands](#) will initialize the measurement, and you do not have to set the measurement to the default.

## **:CONFigure:DENSity**

### **Syntax**

:CONFigure:DENSity

### **Description**

Sets the analyzer to be in the Density measurement state.

### **Remarks**

This command is only available for RTSA mode.

Running this command will discontinue the current measurement and restore the specified measurement of the instrument to the default factory state.

## :CONFigure:DSPEctrogram

### Syntax

:CONFigure:DSPEctrogram

### Description

Sets the analyzer to be in the Density Spectrogram measurement state.

### Remarks

This command is only available for RTSA mode.

Running this command will discontinue the current measurement and restore the specified measurement of the instrument to the default factory state.

## :CONFigure:EBWidth

### Syntax

:CONFigure:EBWidth

### Description

Sets the analyzer to be in the EBW measurement state.

### Remarks

This command is only available for GPSA mode.

Running this command will discontinue the current measurement and restore the specified measurement of the instrument to the default factory state.

After changing any measurement setup, running the [:READ Commands](#) will initialize the measurement, and you do not have to set the measurement to the default.

## :CONFigure:HDI St

### Syntax

:CONFigure:HDI St

### Description

Sets the analyzer to be in the harmonic distortion measurement state.

### Remarks

This command is only available for GPSA mode.

Running this command will discontinue the current measurement and restore the specified measurement of the instrument to the default factory state.

After changing any measurement setup, running the [:READ Commands](#) will initialize the measurement, and you do not have to set the measurement to the default.

## :CONFigure:LPSTep

### Syntax

:CONFigure:LPSTep

### Description

Resets all the parameters in the current measurement mode.

## **:CONFigure:MCHPower**

### **Syntax**

:CONFigure:MCHPower

### **Description**

Sets the analyzer to be in the MCHP measurement state.

### **Remarks**

This command is only available for GPSA mode.

Running this command will discontinue the current measurement and restore the specified measurement of the instrument to the default factory state.

After changing any measurement setup, running the [:READ Commands](#) will initialize the measurement, and you do not have to set the measurement to the default.

## **:CONFigure:NORMal**

### **Syntax**

:CONFigure:NORMal

### **Description**

Sets the analyzer to be in the Normal measurement state.

### **Remarks**

This command is only available for RTSA mode.

Running this command will discontinue the current measurement and restore the specified measurement of the instrument to the default factory state.

## **:CONFigure:OBWidth**

### **Syntax**

:CONFigure:OBWidth

### **Description**

Sets the analyzer to be in the OBW measurement state.

### **Remarks**

This command is only available for GPSA mode.

Running this command will discontinue the current measurement and restore the specified measurement of the instrument to the default factory state.

After changing any measurement setup, running the [:READ Commands](#) will initialize the measurement, and you do not have to set the measurement to the default.

## **:CONFigure:PSGRam**

### **Syntax**

:CONFigure:PSGRam

### **Description**

Sets the analyzer to be in the PvT Spectrogram measurement state.

### **Remarks**

This command is only available for RTSA mode.

Running this command will discontinue the current measurement and restore the specified measurement of the instrument to the default factory state.

## **:CONFigure:PSpectrum**

### **Syntax**

:CONFigure:PSpectrum

### **Description**

Sets the analyzer to be in the PvT Spectrum measurement state.

### **Remarks**

This command is only available for RTSA mode.

Running this command will discontinue the current measurement and restore the specified measurement of the instrument to the default factory state.

## **:CONFigure:PVT**

### **Syntax**

:CONFigure:PVT

### **Description**

Sets the analyzer to be in the PvT measurement state.

### **Remarks**

This command is only available for RTSA mode.

Running this command will discontinue the current measurement and restore the specified measurement of the instrument to the default factory state.

## **:CONFigure:SANalyzer**

### **Syntax**

:CONFigure:SANalyzer

### **Description**

This command is only available for GPSA mode.

This command will disable the currently activated measurement function, and set the analyzer to the swept SA mode.

## :CONFigure:SPECTrogram

### Syntax

:CONFigure:SPECTrogram

### Description

Sets the analyzer to be in the Spectrogram measurement state.

### Remarks

This command is only available for RTSA mode.

Running this command will discontinue the current measurement and restore the specified measurement of the instrument to the default factory state.

## :CONFigure:TOI

### Syntax

:CONFigure:TOI

### Description

Sets the analyzer to be in the TOI measurement state.

### Remarks

This command is only available for GPSA mode.

Running this command will discontinue the current measurement and restore the specified measurement of the instrument to the default factory state.

After changing any measurement setup, running the [:READ Commands](#) will initialize the measurement, and you do not have to set the measurement to the default.

## :CONFigure:TPOWer

### Syntax

:CONFigure:TPOWer

### Description

Sets the analyzer to be in the time-domain power measurement state.

### Remarks

This command is only available for GPSA mode.

Running this command will discontinue the current measurement and restore the specified measurement of the instrument to the default factory state.

After changing any measurement setup, running the [:READ Commands](#) will initialize the measurement, and you do not have to set the measurement to the default.

## :COUPle Commands

### Command List:

◆ [:COUPle](#)

## :COUPle

### Syntax

:COUPle ALL

### Description

Sets all the manual/auto settings in the current measurement mode to "Auto".

## :DI SPlay Commands

### Command List:

- ◆ [:DI SPlay:BACKlight](#)
- ◆ [:DI SPlay:ENABle](#)
- ◆ [:DI SPlay:GRATicule\[:STATe\]](#)
- ◆ [:DI SPlay:HDMI\[:STATe\]](#)
- ◆ [:DI SPlay:PVTime:WINDow:TRACe:X\[:SCALe\]:COUPle](#)
- ◆ [:DI SPlay:PVTime:WINDow:TRACe:X\[:SCALe\]:PDI Vision](#)
- ◆ [:DI SPlay:PVTime:WINDow:TRACe:X\[:SCALe\]:RLEVel](#)
- ◆ [:DI SPlay:PVTime:WINDow:TRACe:X\[:SCALe\]:RPOSition](#)
- ◆ [:DI SPlay:PVTime:WINDow:TRACe:Y\[:SCALe\]:PDI Vision](#)
- ◆ [:DI SPlay:PVTime:WINDow:TRACe:Y\[:SCALe\]:RLEVel <real>](#)
- ◆ [:DI SPlay:VIEW:DENSity:AADJust](#)
- ◆ [:DI SPlay:VIEW:DENSity:CNONlinear](#)
- ◆ [:DI SPlay:VIEW:DENSity:CPALettes](#)
- ◆ [:DI SPlay:VIEW:DENSity:HDHue](#)
- ◆ [:DI SPlay:VIEW:DENSity:LDHue](#)
- ◆ [:DI SPlay:VIEW:DENSity:PERStence](#)
- ◆ [:DI SPlay:VIEW:DENSity:PERStence:INFinite](#)
- ◆ [:DI SPlay:VIEW\[:SElect\]](#)
- ◆ [:DI SPlay:VIEW:SPECTrogram:AADJust](#)
- ◆ [:DI SPlay:VIEW:SPECTrogram:BOTTom](#)
- ◆ [:DI SPlay:VIEW:SPECTrogram:HUE](#)
- ◆ [:DI SPlay:VIEW:SPECTrogram:POSition](#)
- ◆ [:DI SPlay:VIEW:SPECTrogram:REFerence](#)
- ◆ [:DI SPlay:VIEW:SPECTrogram:TRACe:COUPle](#)
- ◆ [:DI SPlay:VIEW:SPECTrogram:TRACe:SElection](#)
- ◆ [:DI SPlay:WINDow:SElect](#)
- ◆ [:DI SPlay:WINDow:TRACe:Y:DLINe](#)
- ◆ [:DI SPlay:WINDow:TRACe:Y:DLINe:STATe](#)
- ◆ [:DI SPlay:WINDow:TRACe:Y\[:SCALe\]:NRLevel\\*](#)
- ◆ [:DI SPlay:WINDow:TRACe:Y\[:SCALe\]:NRPosition\\*](#)
- ◆ [:DI SPlay:WINDow:TRACe:Y\[:SCALe\]:PDI Vision](#)
- ◆ [:DI SPlay:WINDow:TRACe:Y\[:SCALe\]:RLEVel](#)
- ◆ [:DI SPlay:WINDow:TRACe:Y\[:SCALe\]:RLEVel:OFFSet](#)
- ◆ [:DI SPlay:WINDow:TRACe:Y\[:SCALe\]:SPACing](#)

### Remarks:

The commands with \* are only available for RSA3045-TG/RSA3030-TG.

## :DISPlay:BACKlight

### Syntax

:DISPlay:BACKlight < integer>  
:DISPlay:BACKlight?

### Description

Sets the brightness of the backlight of LCD.  
Queries the brightness of the backlight of LCD.

### Parameter

| Name       | Type    | Range    | Default |
|------------|---------|----------|---------|
| < integer> | Integer | 1 to 100 | 100     |

### Return Format

The query returns the brightness of the LCD backlight in integer.

### Example

The following command sets the brightness of the LCD backlight to 50.  
:DISPlay:BACKlight 50

The following query returns 50.  
:DISPlay:BACKlight?

## :DISPlay:ENABLE

### Syntax

:DISPlay:ENABLE OFF|ON|0|1  
:DISPlay:ENABLE?

### Description

Turns on or off the LCD.  
Queries the status of the LCD.

### Parameter

| Name | Type | Range      | Default |
|------|------|------------|---------|
| —    | Bool | OFF ON 0 1 | ON 1    |

### Return Format

The query returns 0 or 1.

### Example

The following command enables the LCD.  
:DISPlay:ENABLE ON or :DISPlay:ENABLE 1

The following query returns 1.  
:DISPlay:ENABLE?

## :DISPlay:GRATicule[:STATe]

### Syntax

:DISPlay:GRATicule[:STATe] OFF|ON|0|1

:DISPlay:GRATicule[:STATe]?

### Description

Enables or disables the display of the graticule.

Queries the display of the graticule.

### Parameter

| Name | Type | Range      | Default |
|------|------|------------|---------|
| ——   | Bool | OFF ON 0 1 | ON 1    |

### Return Format

The query returns 0 or 1.

### Example

The following command enables display of the graticule.

:DISPlay:GRATicule:STATe ON or :DISPlay:GRATicule:STATe 1

The following query returns 1.

:DISPlay:GRATicule:STATe?

## :DISPlay:HDMI[:STATe]

### Syntax

:DISPlay:HDMI[:STATe] OFF|ON|0|1

:DISPlay:HDMI[:STATe]?

### Description

Enables or disables HDMI.

Queries the status of HDMI.

### Parameter

| Name | Type | Range      | Default |
|------|------|------------|---------|
| ——   | Bool | OFF ON 0 1 | OFF 0   |

### Return Format

The query returns 0 or 1.

### Example

The following command enables HDMI.

:DISPlay:HDMI:STATe ON or :DISPlay:HDMI:STATe 1

The following query returns 1.

:DISPlay:HDMI:STATe?

## :DISPlay:PVTime:WINDow:TRACe:X[:SCALe]:COUPle

### Syntax

:DISPlay:PVTime:WINDow:TRACe:X[:SCALe]:COUPle OFF|ON|0|1  
 :DISPlay:PVTime:WINDow:TRACe:X[:SCALe]:COUPle?

### Description

Enables or disables the auto scale function for the horizontal axis in the PvT view.  
 Queries the status of the auto scale function for the horizontal axis in the PvT view.

### Parameter

| Name | Type | Range      | Default |
|------|------|------------|---------|
| --   | Bool | OFF ON 0 1 | ON 1    |

### Remarks

This command is only valid when the PvT view in RTSA mode is selected.

### Return Format

The query returns 1 or 0.

### Example

The following command enables the auto scale function for the horizontal axis in the PvT view.

:DISPlay:PVTime:WINDow:TRACe:X:SCALe:COUPle ON  
 or :DISPlay:PVTime:WINDow:TRACe:X:SCALe:COUPle 1

The following query returns 1.

:DISPlay:PVTime:WINDow:TRACe:X:SCALe:COUPle?

## :DISPlay:PVTime:WINDow:TRACe:X[:SCALe]:PDI Vision

### Syntax

:DISPlay:PVTime:WINDow:TRACe:X[:SCALe]:PDI Vision <time>  
 :DISPlay:PVTime:WINDow:TRACe:X[:SCALe]:PDI Vision?

### Description

Sets the unit per division in the horizontal axis of the PvT view.  
 Queries the unit per division in the horizontal axis of the PvT view.

### Parameter

| Name   | Type                    | Range             | Default             |
|--------|-------------------------|-------------------|---------------------|
| <time> | Consecutive Real Number | 20 $\mu$ s to 4 s | acquisition time/10 |

### Remarks

This command is only valid when the PvT view in RTSA mode is selected.

### Return Format

The query returns the X-axis scale value in scientific notation.

### Example

The following command sets the horizontal scale in the PvT view to 40  $\mu$ s.

:DISPlay:PVTime:WINDow:TRACe:X:SCALe:PDI Vision 0.00004

The following query returns 4.000000000e-07.  
 :DISPlay:PVTime:WINDow:TRACe:X:SCALe:PDI Vision?

## :DISPlay:PVTime:WINDow:TRACe:X[:SCALe]:RLEVel

### Syntax

:DISPlay:PVTime:WINDow:TRACe:X[:SCALe]:RLEVel <real>  
 :DISPlay:PVTime:WINDow:TRACe:X[:SCALe]:RLEVel?

### Description

Sets the reference time for the horizontal axis in the PvT view.  
 Queries the reference time for the horizontal axis in the PvT view.

### Parameter

| Name   | Type                    | Range        | Default   |
|--------|-------------------------|--------------|-----------|
| <real> | Consecutive Real Number | -1 s to 40 s | 0 $\mu$ s |

### Remarks

This command is only valid when the PvT view in RTSA mode is selected.

### Return Format

The query returns the reference time for the horizontal axis in scientific notation.

### Example

The following command sets the reference time for the horizontal axis in the PvT view to 2 s.  
 :DISPlay:PVTime:WINDow:TRACe:X:SCALe:RLEVel 2

The following query returns 2.000000000e+00.  
 :DISPlay:PVTime:WINDow:TRACe:X:SCALe:RLEVel?

## :DISPlay:PVTime:WINDow:TRACe:X[:SCALe]:RPOSition

### Syntax

:DISPlay:PVTime:WINDow:TRACe:X[:SCALe]:RPOSition LEFT|CENTer|RIGHT  
 :DISPlay:PVTime:WINDow:TRACe:X[:SCALe]:RPOSition?

### Description

Sets the position of the reference time in the horizontal axis of the PvT view.  
 Queries the position of the reference time in the horizontal axis of the PvT view.

### Parameter

| Name | Type    | Range             | Default |
|------|---------|-------------------|---------|
| --   | Keyword | LEFT CENTer RIGHT | LEFT    |

### Remarks

This command is only valid when the PvT view in RTSA mode is selected.

### Return Format

The query returns LEFT, CENT, or RIGH.

### Example

The following command sets the position of the reference time in the horizontal axis of the PvT view to

"Left".

```
:DISPlay:PVTime:WINDow:TRACe:X:SCALe:RPOSition LEFT
```

The following query returns LEFT.

```
:DISPlay:PVTime:WINDow:TRACe:X:SCALe:RPOSition?
```

## :DISPlay:PVTime:WINDow:TRACe:Y[:SCALe]:PDI Vision

### Syntax

```
:DISPlay:PVTime:WINDow:TRACe:Y[:SCALe]:PDI Vision <rel_ampl>
```

```
:DISPlay:PVTime:WINDow:TRACe:Y[:SCALe]:PDI Vision?
```

### Description

Sets the unit per division in the vertical axis of the PvT view.

Queries the unit per division in the vertical axis of the PvT view.

### Parameter

| Name   | Type                    | Range           | Default |
|--------|-------------------------|-----------------|---------|
| <time> | Consecutive Real Number | 0.1 dB to 20 dB | 10 dB   |

### Remarks

This command is only valid when the PvT view in RTSA mode is selected.

### Return Format

The query returns the Y-axis scale value in scientific notation.

### Example

The following command sets the vertical scale in the PvT view to 15 dB.

```
:DISPlay:PVTime:WINDow:TRACe:Y:SCALe:PDI Vision 15
```

The following query returns 1.500000000e+01.

```
:DISPlay:PVTime:WINDow:TRACe:Y:SCALe:PDI Vision?
```

## :DISPlay:PVTime:WINDow:TRACe:Y[:SCALe]:RLEVel <real>

### Syntax

```
:DISPlay:PVTime:WINDow:TRACe:Y[:SCALe]:RLEVel <real>
```

```
:DISPlay:PVTime:WINDow:TRACe:Y[:SCALe]:RLEVel?
```

### Description

Sets the reference level for the vertical axis in the PvT view.

Queries the reference level for the vertical axis in the PvT view.

### Parameter

| Name   | Type                    | Range               | Default |
|--------|-------------------------|---------------------|---------|
| <n>    | Discrete                | 1 2 3               | 1       |
| <real> | Consecutive Real Number | -250 dBm to 250 dBm | 0 dBm   |

### Remarks

This command is only valid when the PvT view in RTSA mode is selected.

**Return Format**

The query returns the reference level for the vertical axis in scientific notation.

**Example**

The following command sets the reference level for the vertical axis in the PvT view to 100 dBm.

```
:DISPlay:PVTime:WINDow:TRACe:Y:SCALE:RLEVel 100
```

The following query returns 1.000000000e+02.

```
:DISPlay:PVTime:WINDow2:TRACe:Y:SCALE:RLEVel?
```

**:DISPlay:VIEW:DENSITY:AADJust****Syntax**

```
:DISPlay:VIEW:DENSITY:AADJust
```

**Description**

Sets the highest density hue to the highest density value found in the current bitmap; sets the lowest density hue to the lowest non-zero density value found in the current bitmap.

**Remarks**

This command is only available for Density and Density Spectrogram in RTSA mode.

**:DISPlay:VIEW:DENSITY:CNONlinear****Syntax**

```
:DISPlay:VIEW:DENSITY:CNONlinear <real>
```

```
:DISPlay:VIEW:DENSITY:CNONlinear?
```

**Description**

Sets the curve nonlinearity.

Queries the curve nonlinearity.

**Parameter**

| Name   | Type                    | Range       | Default |
|--------|-------------------------|-------------|---------|
| <real> | Consecutive Real Number | -100 to 100 | 75      |

**Remarks**

This command is only available for Density and Density Spectrogram in RTSA mode.

Within the range between the highest density hue and lowest density hue, setting the curve nonlinearity can change the gradient among different density hues, making the displayed results move towards either the higher or lower end of the gradient. Increasing the curve nonlinearity value will compress the colors towards the higher end of the color bar, and decreasing the nonlinearity value will compress the colors towards the lower end of the color bar.

**Return Format**

The query returns the curve nonlinearity value in scientific notation.

**Example**

The following command sets the curve nonlinearity value to 50.

```
:DISPlay:VIEW:DENSITY:CNONlinear 50
```

The following query returns 5.000000e+01.

```
:DISPlay:VIEW:DENSITY:CNONlinear?
```

## :DISPlay:VIEW:DENSity:CPALettes

### Syntax

```
:DISPlay:VIEW:DENSity:CPALettes COOL|WARM|RADar|FIRE|FROSt
:DISPlay:VIEW:DENSity:CPALettes?
```

### Description

Sets the color palette of Density.  
Queries the color palette of Density.

### Parameter

| Name | Type    | Range                      | Default |
|------|---------|----------------------------|---------|
| —    | Keyword | COOL WARM RADar FIRE FROSt | WARM    |

### Remarks

This command is only available for Density and Density Spectrogram in RTSA mode.

### Return Format

The query returns COOL, WARM, RAD, FIRE, or FROS.

### Example

The following command sets the color palette of Density to Cool.

```
:DISPlay:VIEW:DENSity:CPALettes COOL
```

The following query returns COOL.

```
:DISPlay:VIEW:DENSity:CPALettes?
```

## :DISPlay:VIEW:DENSity:HDHue

### Syntax

```
:DISPlay:VIEW:DENSity:HDHue <real>
:DISPlay:VIEW:DENSity:HDHue?
```

### Description

Sets the highest density hue.  
Queries the highest density hue.

### Parameter

| Name   | Type                    | Range      | Default |
|--------|-------------------------|------------|---------|
| <real> | Consecutive Real Number | 0.1 to 100 | 100     |

### Remarks

This command is only available for Density and Density Spectrogram in RTSA mode.

### Return Format

The query returns the highest density hue in scientific notation.

### Example

The following command sets the highest density hue to 60.

```
:DISPlay:VIEW:DENSity:HDHue 60
```

The following query returns 6.000000e+01.

```
:DISPlay:VIEW:DENSity:HDHue?
```

## :DISPlay:VIEW:DENSity:LDHue

### Syntax

:DISPlay:VIEW:DENSity:LDHue <real>

:DISPlay:VIEW:DENSity:LDHue?

### Description

Sets the lowest density hue.

Queries the lowest density hue.

### Parameter

| Name   | Type                    | Range     | Default |
|--------|-------------------------|-----------|---------|
| <real> | Consecutive Real Number | 0 to 99.9 | 0       |

### Remarks

This command is only available for Density and Density Spectrogram in RTSA mode.

### Return Format

The query returns the lowest density hue in scientific notation.

### Example

The following command sets the lowest density hue to 30.

:DISPlay:VIEW:DENSity:LDHue 30

The following query returns 3.000000e+ 01.

:DISPlay:VIEW:DENSity:LDHue?

## :DISPlay:VIEW:DENSity:PERSistence

### Syntax

:DISPlay:VIEW:DENSity:PERSistence <time>

:DISPlay:VIEW:DENSity:PERSistence?

### Description

Sets the persistence time.

Queries the persistence time.

### Parameter

| Name   | Type                    | Range       | Default |
|--------|-------------------------|-------------|---------|
| <time> | Consecutive Real Number | 0 s to 10 s | 300 ms  |

### Remarks

This command is only available for Density and Density Spectrogram in RTSA mode.

### Return Format

The query returns the persistence time in scientific notation.

### Example

The following command sets the persistence time to 5 s.

:DISPlay:VIEW:DENSity:PERSistence 5

The following query returns 5.000000e+ 00.

:DISPlay:VIEW:DENSity:PERSistence?

## :DISPlay:VIEW:DENSITY:PERSISTence:INFinite

### Syntax

:DISPlay:VIEW:DENSITY:PERSISTence:INFinite OFF|ON|0|1  
 :DISPlay:VIEW:DENSITY:PERSISTence:INFinite?

### Description

Enables or disables the infinite mode of the persistence time.  
 Queries the status of the infinite mode of the persistence time.

### Parameter

| Name | Type | Range      | Default |
|------|------|------------|---------|
| —    | Bool | OFF ON 0 1 | OFF 0   |

### Remarks

This command is only available for Density and Density Spectrogram in RTSA mode.

### Return Format

The query returns 0 or 1.

### Example

The following command enables the infinite mode of the persistence time.

:DISPlay:VIEW:DENSITY:PERSISTence:INFinite ON or :DISPlay:VIEW:DENSITY:PERSISTence:INFinite 1

The following query returns 1.

:DISPlay:VIEW:DENSITY:PERSISTence:INFinite?

## :DISPlay:VIEW[:SElect]

### Syntax

:DISPlay:VIEW[:SElect] NORMal|SPECtrogram|DENSITY|DSPectrogram|PVT|PVTspectrum|PSPectrogram  
 :DISPlay:VIEW[:SElect]?

### Description

Sets the current display view.  
 Queries the current display view.

### Parameter

| Name | Type    | Range                                                                | Default |
|------|---------|----------------------------------------------------------------------|---------|
| —    | Keyword | NORMal SPECtrogram DENSITY DSPectrogram PVT PVTspectrum PSPectrogram | NORMal  |

### Remarks

NORMal: indicates the normal view.  
 SPECtrogram: indicates Spectrogram.  
 DENSITY: indicates Density.  
 DSPectrogram: indicates Density and Density Spectrogram.  
 PVT: indicates Power versus Time spectrum.  
 PVTspectrum: indicates PVT Spectrum.  
 PSPectrogram: indicates PVT Spectrogram.  
 This command is only available for RTSA mode.

### Return Format

The query returns NORM, SPEC, DENS, DSP, PVT, PVTs, or PSP.

**Example**

The following commands sets the current view to Spectrogram.

```
:DISPlay:VIEW:SElect SPECTrogram
```

The following query returns SPEC.

```
:DISPlay:VIEW:SElect?
```

**:DISPlay:VIEW:SPECTrogram:AADJust****Syntax**

```
:DISPlay:VIEW:SPECTrogram:AADJust
```

**Description**

Auto adjusts the reference hue position and bottom hue position based on the highest amplitude value and the lowest amplitude value found in the spectrogram, respectively.

**Remarks**

The reference hue position is set to the highest amplitude value, and the bottom hue position to the lowest amplitude value.

This command is only available for Spectrogram in RTSA mode or the view that contains the spectrogram.

**:DISPlay:VIEW:SPECTrogram:BOTTom****Syntax**

```
:DISPlay:VIEW:SPECTrogram:BOTTom <integer>
```

```
:DISPlay:VIEW:SPECTrogram:BOTTom?
```

**Description**

Sets the bottom hue position displayed in the graticule.

Queries the bottom hue position displayed in the graticule.

**Parameter**

| Name      | Type    | Range                                                  | Default |
|-----------|---------|--------------------------------------------------------|---------|
| <integer> | Integer | 0 to min (90%, with the reference position value -10%) | 0       |

**Remarks**

Any amplitudes lower than the bottom hue position are displayed as black.

This command is only available for Spectrogram in RTSA mode or the view that contains the spectrogram.

**Return Format**

Queries the bottom hue position displayed in the graticule in integer.

**Example**

The following command sets the bottom hue position displayed in the graticule to 40.

```
:DISPlay:VIEW:SPECTrogram:BOTTom 40
```

The following query returns 40.

```
:DISPlay:VIEW:SPECTrogram:BOTTom?
```

## :DISPlay:VIEW:SPECTrogram:HUE

### Syntax

```
:DISPlay:VIEW:SPECTrogram:HUE <real>
:DISPlay:VIEW:SPECTrogram:HUE?
```

### Description

Sets the reference hue.

Queries the reference hue.

### Parameter

| Name   | Type                    | Range      | Default |
|--------|-------------------------|------------|---------|
| <real> | Consecutive Real Number | 0 to 359.9 | 0       |

### Remarks

The reference hue indicates the hue value at the top of the color bar in the spectrogram.

This command is only available for Spectrogram in RTSA mode or the view that contains the spectrogram.

### Return Format

The query returns the reference hue value in scientific notation.

### Example

The following command sets the reference hue value to 120.

```
:DISPlay:VIEW:SPECTrogram:HUE 120
```

The following query returns 1.200000e+02.

```
:DISPlay:VIEW:SPECTrogram:HUE?
```

## :DISPlay:VIEW:SPECTrogram:POSition

### Syntax

```
:DISPlay:VIEW:SPECTrogram:POSition <integer>
:DISPlay:VIEW:SPECTrogram:POSition?
```

### Description

Sets the trace to be displayed in the spectrogram.

Queries the trace to be displayed in the spectrogram.

### Parameter

| Name      | Type    | Range             | Default |
|-----------|---------|-------------------|---------|
| <integer> | Integer | 1 to 8,192 (RTSA) | 1       |

### Remarks

This command is only available for Spectrogram in RTSA mode or the view that contains the spectrogram.

In the trace window displayed in the spectrogram mode, you can determine the trace either by the trace number or the trace time. Trace 1 indicates the latest trace. If you select the trace by trace time, the trace that is closest to the set time will be selected.

### Return Format

The query returns the displayed trace number in integer.

### Example

The following command sets the displayed trace number to 146.

```
:DISPlay:VIEW:SPECTrogram:POSition 146
```

The following query returns 146.

:DISPlay:VIEW:SPECTrogram:POSition?

## :DISPlay:VIEW:SPECTrogram:REference

### Syntax

:DISPlay:VIEW:SPECTrogram:REference <integer>

:DISPlay:VIEW:SPECTrogram:REference?

### Description

Sets the position of the reference hue displayed in the graticule.

Queries the position of the reference hue displayed in the graticule.

### Parameter

| Name      | Type    | Range                                    | Default |
|-----------|---------|------------------------------------------|---------|
| <integer> | Integer | Max (10%, bottom hue value + 10%) to 100 | 100     |

### Remarks

This command is only available for Spectrogram in RTSA mode or the view that contains the spectrogram.

### Return Format

Queries the reference hue position displayed in the graticule in integer.

### Example

The following command sets the reference hue position displayed in the graticule to 60.

:DISPlay:VIEW:SPECTrogram:REference 60

The following query returns 60.

:DISPlay:VIEW:SPECTrogram:REference?

## :DISPlay:VIEW:SPECTrogram:TRACe:COUPle

### Syntax

:DISPlay:VIEW:SPECTrogram:TRACe:COUPle ON|OFF|1|0

:DISPlay:VIEW:SPECTrogram:TRACe:COUPle?

### Description

Enables or disables coupling the marker to the trace.

Queries whether the marker is coupled to the trace.

### Parameter

| Name | Type | Range      | Default |
|------|------|------------|---------|
| —    | Bool | OFF ON 0 1 | OFF 0   |

### Remarks

When you select "Off", the command allows you to fix the current marker to the displayed trace that is active when the marker is turned on.

When you select "On", the marker will stay on the selected trace and change with the trace.

This command is only available for Spectrogram in RTSA mode or the view that contains the spectrogram.

### Return Format

The query returns 0 or 1.

**Example**

The following command enables coupling the marker to the trace.

```
:DISPlay:VIEW:SPECTrogram:TRACe:COUPle ON or :DISPlay:VIEW:SPECTrogram:TRACe:COUPle 1
```

The following query returns 1.

```
:DISPlay:VIEW:SPECTrogram:TRACe:COUPle?
```

**:DISPlay:VIEW:SPECTrogram:TRACe:SElection****Syntax**

```
:DISPlay:VIEW:SPECTrogram:TRACe:SElection TIME|TNUMber
```

```
:DISPlay:VIEW:SPECTrogram:TRACe:SElection?
```

**Description**

Sets the selection method for the displayed trace.

Queries the selection method for the displayed trace.

**Parameter**

| Name | Type    | Range        | Default |
|------|---------|--------------|---------|
| —    | Keyword | TIME TNUMber | TNUMber |

**Remarks**

TIME: indicates the trace time.

TNUMber: indicates the trace number.

Each trace is associated with a time value that represents the acquisition time. The formula is as follows:

Trace Time = Trace Number x Acquisition Time.

This command is only available for Spectrogram in RTSA mode or the view that contains the spectrogram.

**Return Format**

The query returns TIME or TNUM.

**Example**

The following command sets the selection method for the displayed trace to Time.

```
:DISPlay:VIEW:SPECTrogram:TRACe:SElection TIME
```

The following query returns TIME.

```
:DISPlay:VIEW:SPECTrogram:TRACe:SElection?
```

## :DISPlay:WINDow:SElect

### Syntax

```
:DISPlay:WINDow:SElect SPECTrum|PVT
:DISPlay:WINDow:SElect?
```

### Description

Selects a window in the current view.  
Queries the currently selected window type.

### Parameter

| Name | Type    | Range        | Default |
|------|---------|--------------|---------|
| —    | Keyword | SPECTrum PVT | —       |

### Remarks

This command is only available for Density Spectrogram, PVT Spectrum, or PVT Spectrogram in RTSA mode.

### Return Format

The query returns SPEC or PVT.

### Example

The following command selects the PVT window in the PVT Spectrum view.

```
:DISPlay:WINDow:SElect PVT
```

The following query returns PVT.

```
:DISPlay:WINDow:SElect?
```

## :DISPlay:WINDow:TRACe:Y:DLINe

### Syntax

```
:DISPlay:WINDow:TRACe:Y:DLINe < ampl>
:DISPlay:WINDow:TRACe:Y:DLINe?
```

### Description

Sets the position of the display line. Its default unit is dBm.  
Queries the position of the display line.

### Parameter

| Name    | Type                    | Range                   | Default |
|---------|-------------------------|-------------------------|---------|
| < ampl> | Consecutive Real Number | Current amplitude range | -25 dBm |

### Remarks

By default, the display line is disabled. When the display line is enabled for the first time, its position is -25 dBm.

### Return Format

The query returns the position of the display line in scientific notation.

### Example

The following command sets the position of the display line to -10 dBm.

```
:DISPlay:WINDow:TRACe:Y:DLINe -10
```

The following query returns -1.000000e+01.

```
:DISPlay:WINDow:TRACe:Y:DLINe?
```

## :DISPlay:WINDow:TRACe:Y:DLINe:STATe

### Syntax

:DISPlay:WINDow:TRACe:Y:DLINe:STATe OFF|ON|0|1  
:DISPlay:WINDow:TRACe:Y:DLINe:STATe?

### Description

Turns on or off the display line.

Queries the status of the display line.

### Parameter

| Name | Type | Range      | Default |
|------|------|------------|---------|
| —    | Bool | OFF ON 0 1 | OFF 0   |

### Return Format

The query returns 0 or 1.

### Example

The following command enables the display line.

:DISPlay:WINDow:TRACe:Y:DLINe:STATe ON or :DISPlay:WINDow:TRACe:Y:DLINe:STATe 1

The following query returns 1.

:DISPlay:WINDow:TRACe:Y:DLINe:STATe?

## :DISPlay:WINDow:TRACe:Y[:SCALE]:NRLevel

### Syntax

:DISPlay:WINDow:TRACe:Y[:SCALE]:NRLevel <rel\_ampl>  
:DISPlay:WINDow:TRACe:Y[:SCALE]:NRLevel?

### Description

Sets the reference level of normalization.

Queries the reference level of normalization.

### Parameter

| Name       | Type                    | Range             | Default |
|------------|-------------------------|-------------------|---------|
| <rel_ampl> | Consecutive Real Number | -200 dB to 200 dB | 10 dB   |

### Remarks

This command is only available for RSA3045-TG/ RSA3030-TG working in GPSA mode.

This command is only valid when the tracking generator function is enabled.

### Return Format

The query returns the reference level of normalization in scientific notation.

### Example

The following command sets the reference level of normalization to -20 dB.

:DISPlay:WINDow:TRACe:Y:SCALE:NRLevel -20

The following query returns -2.000000e+01.

:DISPlay:WINDow:TRACe:Y:SCALE:NRLevel?

## :DISPlay:WINDow:TRACe:Y[:SCALe]:NRPosition

### Syntax

:DISPlay:WINDow:TRACe:Y[:SCALe]:NRPosition <integer>  
 :DISPlay:WINDow:TRACe:Y[:SCALe]:NRPosition?

### Description

Sets the reference position of normalization.  
 Queries the reference position of normalization.

### Parameter

| Name      | Type    | Range      | Default |
|-----------|---------|------------|---------|
| <integer> | Integer | 0% to 100% | 100%    |

### Remarks

This command is only available for RSA3045-TG/RSA3030-TG working in GPSA mode.  
 This command is only valid when the tracking generator function is enabled.

### Return Format

The query returns the reference position of normalization in integer.

### Example

The following command sets the reference position of normalization to 50%.  
 :DISPlay:WINDow:TRACe:Y:SCALe:NRPosition 50

The following query returns 50.  
 :DISPlay:WINDow:TRACe:Y:SCALe:NRPosition?

## :DISPlay:WINDow:TRACe:Y[:SCALe]:PDI Vision

### Syntax

:DISPlay:WINDow:TRACe:Y[:SCALe]:PDI Vision <rel\_ampl>  
 :DISPlay:WINDow:TRACe:Y[:SCALe]:PDI Vision?

### Description

Sets the Y-axis scale type.  
 Queries the Y-axis scale type.

### Parameter

| Name       | Type                    | Range           | Default |
|------------|-------------------------|-----------------|---------|
| <rel_ampl> | Consecutive Real Number | 0.1 dB to 20 dB | 10 dB   |

### Return Format

The query returns the Y-axis scale value in scientific notation.

### Example

The following command sets the Y-axis scale value to 15 dB.  
 :DISPlay:WINDow:TRACe:Y:SCALe:PDI Vision 15

The following query returns 1.500000e+ 01.  
 :DISPlay:WINDow:TRACe:Y:SCALe:PDI Vision?

## :DISPlay:WINDow:TRACe:Y[:SCALe]:RLEVel

### Syntax

:DISPlay:WINDow:TRACe:Y[:SCALe]:RLEVel < ampl>  
 :DISPlay:WINDow:TRACe:Y[:SCALe]:RLEVel?

### Description

Sets the reference level.  
 Queries the reference level.

### Parameter

| Name    | Type                    | Range              | Default |
|---------|-------------------------|--------------------|---------|
| < ampl> | Consecutive Real Number | -170 dBm to 30 dBm | 0 dBm   |

### Return Format

The query returns the reference level in scientific notation.

### Example

The following command sets the reference level to -10 dB.

:DISPlay:WINDow:TRACe:Y:SCALe:RLEVel -10

The following query returns -1.000000e+01.

:DISPlay:WINDow:TRACe:Y:SCALe:RLEVel?

## :DISPlay:WINDow:TRACe:Y[:SCALe]:RLEVel:OFFSet

### Syntax

:DISPlay:WINDow:TRACe:Y[:SCALe]:RLEVel:OFFSet < rel\_ampl>  
 :DISPlay:WINDow:TRACe:Y[:SCALe]:RLEVel:OFFSet?

### Description

Sets the reference level offset.  
 Queries the reference level offset.

### Parameter

| Name        | Type                    | Range             | Default |
|-------------|-------------------------|-------------------|---------|
| < rel_ampl> | Consecutive Real Number | -300 dB to 300 dB | 0 dB    |

### Remarks

The offset of the reference level only modifies the reference level and amplitude readout of the marker, but does not change the position of the curve.

### Return Format

The query returns the offset of the reference level in scientific notation.

### Example

The following command sets the reference level offset to 10 dB.

:DISPlay:WINDow:TRACe:Y:SCALe:RLEVel:OFFSet 10

The following query returns 1.000000e+01.

:DISPlay:WINDow:TRACe:Y:SCALe:RLEVel:OFFSet?

## :DISPlay:WINDow:TRACe:Y[:SCALe]:SPACing

### Syntax

:DISPlay:WINDow:TRACe:Y[:SCALe]:SPACing LINear|LOGarithmic

:DISPlay:WINDow:TRACe:Y[:SCALe]:SPACing?

### Description

Sets the Y-axis scale type.

Queries the Y-axis scale type.

### Parameter

| Name | Type    | Range              | Default     |
|------|---------|--------------------|-------------|
| —    | Keyword | LINear LOGarithmic | LOGarithmic |

### Remarks

LINear: indicates linear.

LOGarithmic: indicates log.

### Return Format

The query returns LIN or LOG.

### Example

The following command sets the Y-axis scale type to LOG.

:DISPlay:WINDow:TRACe:Y:SCALe:SPACing LOGarithmic

The following query returns LOG.

:DISPlay:WINDow:TRACe:Y:SCALe:SPACing?

## :FETCh Commands

### Command List:

- ◆ [:FETCh:ACPower?](#)
- ◆ [:FETCh:ACPower:LOWer?](#)
- ◆ [:FETCh:ACPower:MAIN?](#)
- ◆ [:FETCh:ACPower:UPPer?](#)
- ◆ [:FETCh:CNRatio?](#)
- ◆ [:FETCh:CNRatio:CARRier?](#)
- ◆ [:FETCh:CNRatio:CNRatio?](#)
- ◆ [:FETCh:CNRatio:NOISe?](#)
- ◆ [:FETCh:EBWidth?](#)
- ◆ [:FETCh:HARMonics:AMPLitude:ALL?](#)
- ◆ [:FETCh:HARMonics:AMPLitude? <n>](#)
- ◆ [:FETCh:HARMonics\[:DISTortion\]?](#)
- ◆ [:FETCh:HARMonics:FREQuency:ALL?](#)
- ◆ [:FETCh:HARMonics:FREQuency? <n>](#)
- ◆ [:FETCh:HARMonics:FUNDamental?](#)
- ◆ [:FETCh:OBWidth?](#)
- ◆ [:FETCh:OBWidth:OBWidth?](#)
- ◆ [:FETCh:OBWidth:OBWidth:FERRor?](#)
- ◆ [:FETCh:SANalyzer<n>?](#)
- ◆ [:FETCh:TOIntercept?](#)
- ◆ [:FETCh:TOIntercept:IP3?](#)
- ◆ [:FETCh:TPOWer?](#)

### Remarks:

The :FETCh commands are only available for the RSA3000 series that has been installed with advanced measurement kit (AMK) option.

## :FETCh:ACPower?

### Syntax

:FETCh:ACPower?

### Description

Queries the results of adjacent channel power measurement.

### Remarks

This command is only valid when the adjacent channel power measurement in GPSA mode is enabled.

The power unit of the return value is consistent with the current Y-axis unit.

This command will select data from the latest measurement results, and then transmit the data to the output buffer.

### Return Format

The query returns 5 values (main channel power, upper channel power, the power difference between the upper channel and the main channel (in dBc), the lower channel power, the power difference between the lower channel and the main channel (in dBc)) in scientific notation, and the values are separated by commas.

### Example

The following query returns

-5.150423000e+ 01,-5.173441000e+ 01,-2.301865000e-01,-5.142665000e+ 01,7.757568000e-02.

:FETCh:ACPower?

## :FETCh:ACPower:LOWer?

### Syntax

:FETCh:ACPower:LOWer?

### Description

Queries the lower channel power of the adjacent channel power measurement.

### Remarks

This command is only valid when the adjacent channel power measurement in GPSA mode is enabled.

The power unit of the return value is consistent with the current Y-axis unit.

This command will select data from the latest measurement results, and then transmit the data to the output buffer.

### Return Format

The query returns the lower channel power in scientific notation.

### Example

The following query returns -5.142665000e+ 01.

:FETCh:ACPower:LOWer?

## **:FETCh:ACPower:MAIN?**

### **Syntax**

:FETCh:ACPower:MAIN?

### **Description**

Queries the main channel power of the adjacent channel power measurement.

### **Remarks**

This command is only valid when the adjacent channel power measurement in GPSA mode is enabled.

The power unit of the return value is consistent with the current Y-axis unit.

This command will select data from the latest measurement results, and then transmit the data to the output buffer.

### **Return Format**

The query returns the main channel power in scientific notation.

### **Example**

The following query returns -5.150423000e+01.

:FETCh:ACPower:MAIN?

## **:FETCh:ACPower:UPPer?**

### **Syntax**

:FETCh:ACPower:UPPer?

### **Description**

Queries the upper channel power of the adjacent channel power measurement.

### **Remarks**

This command is only valid when the adjacent channel power measurement in GPSA mode is enabled.

The power unit of the return value is consistent with the current Y-axis unit.

This command will select data from the latest measurement results, and then transmit the data to the output buffer.

### **Return Format**

The query returns the upper channel power in scientific notation.

### **Example**

The following query returns -5.173441000e+01.

:FETCh:ACPower:UPPer?

## :FETCh:CNRatio?

### Syntax

:FETCh:CNRatio?

### Description

Queries the results of C/N ratio measurement.

### Remarks

This command is only valid when the C/N ratio measurement in GPSA mode is enabled.

The power unit of the return value is consistent with the current Y-axis unit.

This command will select data from the latest measurement results, and then transmit the data to the output buffer.

### Return Format

The query returns the carrier power, noise power, and C/N ratio (in dB) in scientific notation. They are separated by commas.

### Example

The following query returns -6.048788000e+ 01,-6.186192000e+ 01,1.374039000e+ 00.

:FETCh:CNRatio?

## :FETCh:CNRatio:CARRier?

### Syntax

:FETCh:CNRatio:CARRier?

### Description

Queries the carrier power.

### Remarks

This command is only valid when the C/N ratio measurement in GPSA mode is enabled.

The power unit of the return value is consistent with the current Y-axis unit.

This command will select data from the latest measurement results, and then transmit the data to the output buffer.

### Return Format

The query returns the carrier power in scientific notation.

### Example

The following query returns -1.484203000e+ 01.

:FETCh:CNRatio:CARRier?

## **:FETCh:CNRatio:CNRatio?**

### **Syntax**

:FETCh:CNRatio:CNRatio?

### **Description**

Queries the results of C/N ratio measurement.

### **Remarks**

This command is only valid when the C/N ratio measurement in GPSA mode is enabled.

This command will select data from the latest measurement results, and then transmit the data to the output buffer.

### **Return Format**

The query returns the C/N ratio in scientific notation.

### **Example**

The following query returns 8.956909000e-02.

:FETCh:CNRatio:CNRatio?

## **:FETCh:CNRatio:NOI Se?**

### **Syntax**

:FETCh:CNRatio:NOI Se?

### **Description**

Queries the noise power.

### **Remarks**

This command is only valid when the C/N ratio measurement in GPSA mode is enabled.

The power unit of the return value is consistent with the current Y-axis unit.

This command will select data from the latest measurement results, and then transmit the data to the output buffer.

### **Return Format**

The query returns the noise power in scientific notation.

### **Example**

The following query returns -1.442294000e+ 01.

:FETCh:CNRatio:NOI Se?

## :FETCh:EBWidth?

### Syntax

:FETCh:EBWidth?

### Description

Queries the result of the emission bandwidth measurement.

### Remarks

This command is only valid when the emission bandwidth measurement in GPSA mode is enabled.  
This command will select data from the latest measurement results, and then transmit the data to the output buffer.

### Return Format

The query returns the emission bandwidth in integer, and its unit is Hz.

### Example

The following query returns 5.000000000e+ 04.  
:FETCh:EBWidth?

## :FETCh:HARMonics:AMPLitude:ALL?

### Syntax

:FETCh:HARMonics:AMPLitude:ALL?

### Description

Queries the amplitudes of the first 10 harmonics. The first harmonic is the fundamental waveform.

### Remarks

This command is only valid when the harmonic distortion measurement in GPSA mode is enabled.  
The amplitude unit of the return value is consistent with the current Y-axis unit.  
This command will select data from the latest measurement results, and then transmit the data to the output buffer.

### Return Format

The query returns the amplitudes of the first 10 harmonics in scientific notation (separated by commas).  
If the number of harmonics under measurement is less than 10, the harmonics that are not measured do not have return values.

### Example

The following query returns  
-1.692102000e+ 01,-6.458423000e+ 01,-7.509421000e+ 01,-7.924328000e+ 01,-7.847027000e+ 01,-7.885  
457000e+ 01,-7.882358000e+ 01,-7.921457000e+ 01,-7.923057000e+ 01,-7.915358000e+ 01.  
:FETCh:HARMonics:AMPLitude:ALL?

## :FETCh:HARMonics:AMPLitude? < n>

### Syntax

:FETCh:HARMonics:AMPLitude? < n>

### Description

Queries the amplitude of the specified harmonic.

### Parameter

| Name | Type    | Range   | Default |
|------|---------|---------|---------|
| < n> | Integer | 1 to 10 | ---     |

### Remarks

This command is only valid when the harmonic distortion measurement in GPSA mode is enabled.

The amplitude unit of the return value is consistent with the current Y-axis unit.

The query returns --- when the harmonic read does not have data.

This command will select data from the latest measurement results, and then transmit the data to the output buffer.

### Return Format

The query returns the amplitude of the specified harmonic in scientific notation.

### Example

The following query returns -1.692102000e+ 01.

:FETCh:HARMonics:AMPLitude? 1

## :FETCh:HARMonics[:DI STortion]?

### Syntax

:FETCh:HARMonics[:DI STortion]?

### Description

Queries the percentage of the total harmonic distortion.

### Remarks

This command is only valid when the harmonic distortion measurement in GPSA mode is enabled.

This command will select data from the latest measurement results, and then transmit the data to the output buffer.

### Return Format

The query returns the percentage of the total harmonic distortion in scientific notation.

### Example

The following query returns 1.510000000e-02.

:FETCh:HARMonics:DI STortion?

## :FETCh:HARMonics:FREQuency:ALL?

### Syntax

:FETCh:HARMonics:FREQuency:ALL?

### Description

Queries the frequencies of the first 10 harmonics. The first harmonic is the fundamental waveform.

### Remarks

This command is only valid when the harmonic distortion measurement in GPSA mode is enabled.  
This command will select data from the latest measurement results, and then transmit the data to the output buffer.

### Return Format

The query returns the frequencies of the first 10 harmonics in scientific notation (separated by commas). The unit is Hz.  
If the number of harmonics under measurement is less than 10, the harmonics that are not measured do not have return values.

### Example

The following query returns  
4.550000000e+07,9.100000000e+07,1.365000000e+08,1.820000000e+08,2.275000000e+08,2.73000000e+08,3.185000000e+08,3.640000000e+08,4.095000000e+08,4.550000000e+08.  
:FETCh:HARMonics:FREQuency:ALL?

## :FETCh:HARMonics:FREQuency? <n>

### Syntax

:FETCh:HARMonics:FREQuency? <n>

### Description

Queries the frequency of the specified harmonic.

### Parameter

| Name | Type    | Range   | Default |
|------|---------|---------|---------|
| <n>  | Integer | 1 to 10 | —       |

### Remarks

This command is only valid when the harmonic distortion measurement in GPSA mode is enabled.  
The query returns --- when the harmonic read does not have data.  
This command will select data from the latest measurement results, and then transmit the data to the output buffer.

### Return Format

The query returns the frequency of the specified harmonic in scientific notation. The unit is Hz.

### Example

The following query returns 4.550000000e+07.  
:FETCh:HARMonics:FREQuency? 1

## :FETCh:HARMonics:FUNDamental?

### Syntax

:FETCh:HARMonics:FUNDamental?

### Description

Queries the frequency of the fundamental waveform.

### Remarks

This command is only valid when the harmonic distortion measurement in GPSA mode is enabled.

This command has the same function as [:FETCh:HARMonics:FREQuency? 1](#).

This command will select data from the latest measurement results, and then transmit the data to the output buffer.

### Return Format

The query returns the frequency of the fundamental waveform in scientific notation. The unit is Hz.

### Example

The following query returns 4.550000000e+07.

:FETCh:HARMonics:FUNDamental?

## :FETCh:OBWidth?

### Syntax

:FETCh:OBWidth?

### Description

Queries the results of the occupied bandwidth measurement.

### Remarks

This command is only valid when the occupied bandwidth measurement in GPSA mode is enabled.

This command will select data from the latest measurement results, and then transmit the data to the output buffer.

### Return Format

The query returns the occupied bandwidth (Hz) and the transmit frequency error (Hz) in scientific notation. They are separated by commas.

### Example

The following query returns 1.860000000e+06,2.000000000e+04.

:FETCh:OBWidth?

## :FETCh:OBWidth:OBWidth?

### Syntax

:FETCh:OBWidth:OBWidth?

### Description

Queries the occupied bandwidth.

### Remarks

This command is only valid when the occupied bandwidth measurement in GPSA mode is enabled. This command will select data from the latest measurement results, and then transmit the data to the output buffer.

### Return Format

The query returns the occupied bandwidth in scientific notation. Its unit is Hz.

### Example

The following query returns 1.860000000e+ 06.

:FETCh:OBWidth:OBWidth?

## :FETCh:OBWidth:OBWidth:FERRor?

### Syntax

:FETCh:OBWidth:OBWidth:FERRor?

### Description

Queries the transmit frequency error.

### Remarks

This command is only valid when the occupied bandwidth measurement in GPSA mode is enabled. This command will select data from the latest measurement results, and then transmit the data to the output buffer.

### Return Format

The query returns the transmit frequency error in scientific notation. The unit is Hz.

### Example

The following query returns 2.000000000e+ 04.

:FETCh:OBWidth:OBWidth:FERRor?

## :FETCh:SANalyzer< n> ?

### Syntax

:FETCh:SANalyzer< n> ?

### Description

Queries the measurement results or trace data in the cache area.

### Parameter

| Name | Type     | Range       | Default |
|------|----------|-------------|---------|
| < n> | Discrete | 1 2 3 4 5 6 | —       |

### Remarks

This command is only valid in GPSA mode.

This command will select data from the latest measurement results, and then transmit the data to the output buffer.

### Return Format

The query returns the measurement data of Trace n in scientific notation. The measurement data are returned in x,y pairs, separated by commas. The unit is Hz.

### Example

The following query returns

0,0,0,0,-1.000000000e+02,0,801,0,0,0,-3.19e+01,0.00e+00,0.00e+00,0.00e+00,0.00e+00,0.00e+00,0.00e+00,0.00e+00.

:FETCh:SANalyzer1?

## :FETCh:TOI ntercept?

### Syntax

:FETCh:TOI ntercept?

### Description

Queries the results of TOI measurement.

### Remarks

This command is only valid when the TOI measurement in GPSA mode is enabled.

The amplitude unit of the return value is consistent with the current Y-axis unit.

This command will select data from the latest measurement results, and then transmit the data to the output buffer.

### Return Format

The query returns the measurement results of TOI in the following format:

Base Lower frequency (Hz), amplitude, Base Upper frequency (Hz), amplitude, 3rd Order Lower frequency (Hz), amplitude, intercept, 3rd Order Upper frequency (Hz), amplitude, intercept.

### Example

The following query returns

1.500450000e+09,-8.131735000e+01,1.500450000e+09,-8.131735000e+01,1.500450000e+09,-8.131735000e+01,-8.131735000e+01,1.500450000e+09,-8.131735000e+01,-8.131735000e+01.

:FETCh:TOI ntercept?

## :FETCh:TOI ntercept:IP3?

### Syntax

:FETCh:TOI ntercept:IP3?

### Description

The query returns the minor value of the intercepts of the Third Order Lower and Third Order Upper.

### Remarks

This command is only valid when the TOI measurement in GPSA mode is enabled.

This command will select data from the latest measurement results, and then transmit the data to the output buffer.

### Return Format

The query returns the minor one in scientific notation.

### Example

The following query returns -8.131735000e+01.

:FETCh:TOI ntercept:IP3?

## :FETCh:TPOWer?

### Syntax

:FETCh:TPOWer?

### Description

Queries the result of T-power measurement.

### Remarks

This command is only valid when the T-power measurement in GPSA mode is enabled.

The power unit of the return value is consistent with the current Y-axis unit.

This command will select data from the latest measurement results, and then transmit the data to the output buffer.

### Return Format

The query returns the T-power measurement result in scientific notation.

### Example

The following query returns -1.658941000e+01.

:FETCh:TPOWer?

## :FORMat Commands

### Command List:

- ◆ [:FORMat:BORDER](#)
- ◆ [:FORMat\[:TRACe\]\[:DATA\]](#)

## :FORMat:BORDER

### Syntax

:FORMat:BORDER NORMal| SWAPped  
:FORMat:BORDER?

### Description

Selects the binary data byte order for data transmission.  
Queries the binary data byte order for data transmission.

### Parameter

| Name | Type    | Range           | Default |
|------|---------|-----------------|---------|
| —    | Keyword | NORMal  SWAPped | NORMal  |

### Remarks

NORMal: indicates that the byte sequence begins with the most significant byte (MSB) and ends with the least significant byte (LSB).

SWAPped: indicates that the byte sequence begins with the least significant byte (LSB) and ends with the most significant byte (MSB).

### Return Format

The query returns NORM or SWAP.

### Example

The following command queries the binary data byte order for data transmission to NORMal.

:FORMat:BORDER NORMal

The following query returns NORM.

:FORMat:BORDER?

## :FORMat[:TRACe][:DATA]

### Syntax

:FORMat[:TRACe][:DATA] ASCii|INTeger,32|REAL,32|REAL,64  
:FORMat[:TRACe][:DATA] ?

### Description

Sets the input/output format of the trace data.

Queries the input/output format of the trace data.

### Parameter

| Name | Type    | Range                            | Default |
|------|---------|----------------------------------|---------|
| ——   | Keyword | ASCii INTeger,32 REAL,32 REAL,64 | ASCii   |

### Remarks

ASCii: ASCII characters, separated by commas.

INTeger,32: binary 32-bit integer values.

REAL,32: binary 32-bit real values.

REAL,64: binary 64-bit real values.

### Return Format

The query returns ASC,8, INT,32, REAL,32, or REAL,64.

### Example

The following command sets the input/output format of the trace data to REAL,32.

:FORMat:TRACe:DATA REAL,32

The following query returns REAL,32.

:FORMat:TRACe:DATA?

## IEEE 488.2 Common Commands

IEEE 488.2 common commands are used to operate or query the status registers. For the structure of the status register, refer to "[:STATus Commands](#)".

### Command List:

- ◆ [\\*CLS](#)
- ◆ [\\*ESE](#)
- ◆ [\\*ESR?](#)
- ◆ [\\*IDN?](#)
- ◆ [\\*OPC](#)
- ◆ [\\*RCL](#)
- ◆ [\\*RST](#)
- ◆ [\\*SAV](#)
- ◆ [\\*SRE](#)
- ◆ [\\*STB?](#)
- ◆ [\\*TRG](#)
- ◆ [\\*TST?](#)
- ◆ [\\*WAI](#)

## \* CLS

### Syntax

\* CLS

### Description

Clears all the event registers and status byte registers.

## \* ESE

### Syntax

\* ESE <value>

\* ESE?

### Description

Sets the enable register for the standard event status register.

Queries the enable register for the standard event status register.

### Parameter

| Name    | Type    | Range                       | Default |
|---------|---------|-----------------------------|---------|
| <value> | Integer | Refer to " <b>Remarks</b> " | 0       |

### Remarks

Bit 2, bit 3, bit 4, and bit 7 are reserved; you can set their values but they will not affect the system. Bit 1 and bit 6 are not used and are always treated as 0; therefore, the range of <value> are the decimal numbers corresponding to the binary numbers ranging from 00000000 (0 in decimal) to 11111111 (255 in decimal) and of which bit 1 and bit 6 are 0.

### Return Format

The query returns an integer. The integer equals to the binary-weighted sum of all the bits set in the register. For example, the query returns 144 if bit 4 (16 in decimal) and bit 7 (128 in decimal) are enabled.

### Example

The following command sets the enable register for the standard event status register to 16.

\* ESE 16

The following query returns 16.

\* ESE?

## \* ESR?

### Syntax

\* ESR?

### Description

Queries and clears the event register for the standard event status register.

### Remarks

Bit 1 and bit 6 in the standard event status register are not in use, and are regarded as 0. The query returns a decimal value that corresponds to the binary values ranging from 00000000 (0 in decimal) to 11111111 (255 in decimal) and of which bit 1 and bit 6 are 0.

### Return Format

The query returns an integer. The integer equals to the binary-weighted sum of all the bits set in the register. For example, the query returns 144 if bit 4 (16 in decimal) and bit 7 (128 in decimal) are enabled.

### Example

The following query returns 24 (bit 3 and bit 4 have been set).

\* ESR?

## \* IDN?

### Syntax

\* IDN?

### Description

Queries the ID string of the instrument.

### Return Format

The query returns the ID string in the following format:

Rigol Technologies,< model> ,< serial number> ,XX.XX.XX

< model> : instrument model

< serial number> : serial number of the instrument

XX.XX.XX: software version of the instrument

### Example

The following query returns Rigol Technologies,RSA3045,RSA3A200100001,00.01.00.

\* IDN?

## \* OPC

### Syntax

\* OPC

\* OPC?

### Description

Sets bit 0 (Operation Complete, OPC) in the standard event status register to 1 after the current operation is finished.

Queries whether the current operation is finished.

### Return Format

The query returns 1 after the current operation is finished; otherwise, the query returns 0.

## \* RCL

### Syntax

\* RCL <integer>

### Description

Recalls the selected register.

### Parameter

| Name      | Type    | Range   | Default |
|-----------|---------|---------|---------|
| <integer> | Integer | 1 to 16 | —       |

### Example

The following command recalls Register 1.

\* RCL 1

## \* RST

### Syntax

\* RST

### Description

Restores the instrument to its factory default settings.

**\* SAV****Syntax**

\*SAV <integer>

**Description**

Saves the current instrument state to the selected register.

**Parameter**

| Name      | Type    | Range   | Default |
|-----------|---------|---------|---------|
| <integer> | Integer | 1 to 16 | —       |

**Example**

The following command saves the current instrument state to Register 1.

\*SAV 1

**\* SRE****Syntax**

\*SRE <value>

\*SRE?

**Description**

Sets the enable register for the status byte register.

Queries the enable register for the status byte register.

**Parameter**

| Name    | Type    | Range              | Default |
|---------|---------|--------------------|---------|
| <value> | Integer | Refer to "Remarks" | 0       |

**Remarks**

Bit 0 and Bit 1 are not used and are always treated as 0; therefore, the range of <value> are the decimal numbers corresponding to the binary numbers ranging from 00000000 (0 in decimal) to 11111111 (255 in decimal) and of which Bit 0 and Bit 1 are 0.

**Return Format**

The query returns an integer. The integer equals to the binary-weighted sum of all the bits set in the register. For example, the query returns 144 if Bit 4 (16 in decimal) and Bit 7 (128 in decimal) are enabled.

**Example**

The following command sets the enable register for the status byte register to 16.

\*SRE 16

The following query returns 16.

\*SRE?

## \* STB?

### Syntax

\* STB?

### Description

Queries the event register for the status byte register.

### Remarks

Bit 0 and Bit 1 in the status byte register are not in use, and are regarded as 0. The query returns a decimal value that corresponds to the binary values ranging from 00000000 (0 in decimal) to 11111111 (255 in decimal) and of which Bit 0 and Bit 1 are 0.

### Return Format

The query returns an integer. The integer equals to the binary-weighted sum of all the bits set in the register. For example, the query returns 144 if bit 4 (16 in decimal) and bit 7 (128 in decimal) are enabled.

### Example

The following query returns 24 (bit 3 and bit 4 have been set).

\* STB?

## \* TRG

### Syntax

\* TRG

### Description

Triggers a sweep or measurement immediately.

## \* TST?

### Syntax

\* TST?

### Description

Queries whether the self-check operation is finished.

### Remarks

The query returns 0 or 1. A zero is returned if the test is successful, 1 if it fails.

## \* WAI

### Syntax

\* WAI

### Description

Wait for the operation to finish.

## :I NI Tiate Commands

### Command List:

- ◆ [:I NI Tiate:CONTInuous](#)
- ◆ [:I NI Tiate\[:I MMediate\]](#)

### Remarks:

The commands with \* are only available for the RSA3000 series that has been installed with advanced measurement kit option.

## :I NI Tiate:CONTInuous

### Syntax

```
:I NI Tiate:CONTInuous OFF|ON|0|1
:I NI Tiate:CONTInuous?
```

### Description

Selects continuous (ON|1) or single (OFF|0) measurement mode.  
Queries the current measurement mode.

### Parameter

| Name | Type | Range      | Default |
|------|------|------------|---------|
| —    | Bool | OFF ON 0 1 | ON 1    |

### Return Format

The query returns 0 or 1.

### Example

The following command sets the instrument to sweep continuously.  
:I NI Tiate:CONTInuous ON or :I NI Tiate:CONTInuous 1

The following query returns 1.  
:I NI Tiate:CONTInuous?

## :I NI Tiate[:I MMediate]

### Syntax

```
:I NI Tiate[:I MMediate]
```

### Description

In non-measurement state, initialize a sweep.  
In measurement state, trigger a measurement.

### Explanation

Use the :FETCh? command to transmit a measurement result from the internal memory to the output buffer.

## :INSTRument Commands

### Command List:

- ◆ [:INSTRument:COUPle:FREQuency:CENTer](#)
- ◆ [:INSTRument:DEFault](#)
- ◆ [:INSTRument:NSElect](#)
- ◆ [:INSTRument\[:SElect\]](#)

## :INSTRument:COUPle:FREQuency:CENTer

### Syntax

:INSTRument:COUPle:FREQuency:CENTer ALL|NONE  
:INSTRument:COUPle:FREQuency:CENTer?

### Description

Turns on or off the global center frequency of the instrument.  
Queries the setting status of the global center frequency of the instrument.

### Parameter

| Name | Type    | Range    | Default |
|------|---------|----------|---------|
| —    | Keyword | ALL NONE | NONE    |

### Remarks

NONE: turns off the global center frequency.

ALL: turns on the global center frequency.

If you execute this command in any mode, the center frequency of the current mode is set to the global center frequency. Adjusting the center frequency in a mode, while the global center frequency is on, will modify the global center frequency.

### Return Format

The query returns ALL or NONE.

### Example

The following command enables the global center frequency of the instrument.

:INSTRument:COUPle:FREQuency:CENTer ALL

The following query returns ALL.

:INSTRument:COUPle:FREQuency:CENTer?

## :INSTRument:DEFault

### Syntax

:INSTRument:DEFault

### Description

Resets the parameters of the current mode to the factory default settings.

## **:INSTRument:NSElect** **:INSTRument[:SElect]**

### **Syntax**

```
:INSTRument:NSElect 1|2  
:INSTRument:NSElect?  
:INSTRument[:SElect] SA|RTSA  
:INSTRument[:SElect]?
```

### **Description**

Selects the working mode of the instrument.  
Queries the working mode of the instrument.

### **Parameter**

| Name | Type     | Range   | Default |
|------|----------|---------|---------|
| ——   | Discrete | 1 2     | 1       |
| ——   | Keyword  | SA RTSA | SA      |

### **Remarks**

The above parameters 1|2 correspond to SA|RTSA, respectively.  
After running the command of switching the working mode, we recommend you set the timeout value to 8 s first and then perform the follow-up operation. Or, you can perform the next step after a delay of 8 s.

### **Example**

The following command sets the working mode of the instrument to GPSA.  
:INSTRument:NSElect 1  
:INSTRument:SElect SA

The following query returns 1 or SA.  
:INSTRument:NSElect?  
:INSTRument:SElect?

## :MMEMory Commands

### Command List:

- ◆ [:MMEMory:DELeTe](#)
- ◆ [:MMEMory:LOAD:FMT](#)
- ◆ [:MMEMory:LOAD:LI Mit](#)
- ◆ [:MMEMory:LOAD:STATe](#)
- ◆ [:MMEMory:LOAD:TRACe](#)
- ◆ [:MMEMory:LOAD:TRACe:DATA](#)
- ◆ [:MMEMory:MOVE](#)
- ◆ [:MMEMory:STORe:LI Mit](#)
- ◆ [:MMEMory:STORe:MTABLe](#)
- ◆ [:MMEMory:STORe:PTABLe](#)
- ◆ [:MMEMory:STORe:RESults](#)
- ◆ [:MMEMory:STORe:SCReen](#)
- ◆ [:MMEMory:STORe:STATe](#)
- ◆ [:MMEMory:STORe:TRACe](#)
- ◆ [:MMEMory:STORe:TRACe:DATA](#)

### Remarks:

The mode name in the following path contains "gpsa" and rtSa" modes. The return path takes "gpsa" as an example.

## :MMEMory:DELeTe

### Syntax

:MMEMory:DELeTe <file\_name>

### Description

Deletes a specified file.

### Parameter

| Name        | Type         | Range | Default |
|-------------|--------------|-------|---------|
| <file_name> | ASCII String | —     | —       |

### Remarks

<file\_name> should contain the path and the filename.  
This operation fails if the specified file does not exist.

### Example

The following command deletes the "state1.sta" file from the "/gpsa/state" folder.  
:MMEMory:DELeTe /gpsa/state/state1.sta

## :MMEMory:LOAD:FMT

### Syntax

:MMEMory:LOAD:FMT <label>,<file\_name>

### Description

Loads the edited FMT file (.csv).

### Parameter

| Name        | Type         | Range       | Default |
|-------------|--------------|-------------|---------|
| <label>     | Keyword      | UPPer LOWer | ----    |
| <file_name> | ASCII String | ----        | ----    |

### Remarks

This operation fails if the specified file does not exist.

This command is only available for RTSA mode.

### Example

The following command loads the FMT file (mask1.csv) to the upper mask.

```
:MMEMory:LOAD:FMT upper,mask1.csv
```

## :MMEMory:LOAD:LI Mit

### Syntax

:MMEMory:LOAD:LI Mit <label>,<file\_name>

### Description

Imports the edited limit line file (.csv).

### Parameter

| Name        | Type         | Range                                     | Default |
|-------------|--------------|-------------------------------------------|---------|
| <label>     | Keyword      | LLINE1 LLINE2 LLINE3 LLINE4 LLINE5 LLINE6 | ----    |
| <file_name> | ASCII String | ----                                      | ----    |

### Remarks

This operation fails if the specified file does not exist.

### Example

The following command imports the limit line file (upp1.csv) to Limit1.

```
:MMEMory:LOAD:LI Mit LLINE1,upp1.csv
```

## :MMEMory:LOAD:STATe

### Syntax

:MMEMory:LOAD:STATe <file\_name>

### Description

Imports the specified state file (.sta).

### Parameter

| Name        | Type         | Range | Default |
|-------------|--------------|-------|---------|
| <file_name> | ASCII String | ——    | ——      |

### Remarks

This operation fails if the specified file does not exist.

### Example

The following command imports the state file (state1.sta) to the instrument.

```
:MMEMory:LOAD:STATe state1.sta
```

## :MMEMory:LOAD:TRACe

### Syntax

:MMEMory:LOAD:TRACe <label>,<file\_name>

### Description

Imports the specified trace file (.trs) and the current instrument state.

### Parameter

| Name        | Type         | Range                                     | Default |
|-------------|--------------|-------------------------------------------|---------|
| <label>     | Keyword      | TRACE1 TRACE2 TRACE3 TRACE4 TRACE5 TRACE6 | ——      |
| <file_name> | ASCII String | ——                                        | ——      |

### Remarks

This operation fails if the specified file does not exist.

### Example

The following command imports the current instrument state and the trace file (t1.trs) to Trace2.

```
:MMEMory:LOAD:TRACe TRACE2,t1.trs
```

## :MMEMory:LOAD:TRACe:DATA

### Syntax

:MMEMory:LOAD:TRACe:DATA <label>,<file\_name>

### Description

Imports the specified measurement data file (.csv).

### Parameter

| Name        | Type         | Range                                     | Default |
|-------------|--------------|-------------------------------------------|---------|
| <label>     | Keyword      | TRACE1 TRACE2 TRACE3 TRACE4 TRACE5 TRACE6 | ——      |
| <file_name> | ASCII String | ——                                        | ——      |

### Remarks

This operation fails if the specified file does not exist.

### Example

The following command imports the measurement data file (trace1.csv) to Trace1.

```
:MMEMory:LOAD:TRACe:DATA TRACE1,trace1.csv
```

## :MMEMory:MOVE

### Syntax

:MMEMory:MOVE <file\_name1>,<file\_name2>

### Description

Renames the specified file <file\_name1> as <file\_name2>.

### Parameter

| Name         | Type         | Range | Default |
|--------------|--------------|-------|---------|
| <file_name1> | ASCII String | ——    | ——      |
| <file_name2> | ASCII String | ——    | ——      |

### Remarks

<file\_name1> and <file\_name2> should contain the path and the filename.

This operation fails if the file with the specified filename does not exist.

### Example

The following command renames the state file (state1.sta) in the folder (/gpsa/state) as "state2.sta".

```
:MMEMory:MOVE /gpsa/state/state1.sta,/gpsa/state/state2.sta
```

## :MMEMory:STORe:LI Mit

### Syntax

:MMEMory:STORe:LI Mit <label>,<file\_name>

### Description

Saves the currently edited limit line with a specified filename suffixed with ".csv" by default (you do not have to add the suffix manually) to a default path ("/mode name"/limit).

### Parameter

| Name        | Type         | Range                                     | Default |
|-------------|--------------|-------------------------------------------|---------|
| <label>     | Keyword      | LLINE1 LLINE2 LLINE3 LLINE4 LLINE5 LLINE6 | —       |
| <file_name> | ASCII String | —                                         | —       |

### Remarks

If the specified file already exists, overwrite it.

### Example

The following command saves Limit1 with the filename "low" to the folder (/gpsa/limit).

```
:MMEMory:STORe:LI Mit LLINE1,low
```

## :MMEMory:STORe:MTABle

### Syntax

:MMEMory:STORe:MTABle <file\_name>

### Description

Saves the marker table with a specified filename suffixed with ".csv" by default (you do not have to add the suffix manually) to the default path ("/mode name"/measdata).

### Parameter

| Name        | Type         | Range | Default |
|-------------|--------------|-------|---------|
| <file_name> | ASCII String | —     | —       |

### Remarks

If the specified file already exists, overwrite it.

### Example

The following command saves the marker table with the specified filename "MAK1" to the folder (/gpsa/measdata).

```
:MMEMory:STORe:MTABle MAK1
```

## :MMEMory:STORe:PTABle

### Syntax

:MMEMory:STORe:PTABle <file\_name>

### Description

Saves the peak table with a specified filename suffixed with ".csv" by default (you do not have to add the suffix manually) to the default path ("/mode name"/measdata).

### Parameter

| Name        | Type         | Range | Default |
|-------------|--------------|-------|---------|
| <file_name> | ASCII String | —     | —       |

### Remarks

If the specified file already exists, overwrite it.

### Example

The following command saves the peak table with the specified filename "PT1" to the folder (/gpsa/measdata).

```
:MMEMory:STORe:PTABle PT1
```

## :MMEMory:STORe:RESults

### Syntax

:MMEMory:STORe:RESults <file\_name>

### Description

Saves the current measurement results with a specified filename suffixed with ".csv" by default (you do not have to add the suffix manually) to the default path ("/mode name"/measdata).

### Parameter

| Name        | Type         | Range | Default |
|-------------|--------------|-------|---------|
| <file_name> | ASCII String | —     | —       |

### Remarks

If the specified file already exists, overwrite it.

This command is only valid when the advanced measurement function in GPSA mode is enabled.

### Example

The following command saves the current measurement results with the specified filename "data" to the folder (/gpsa/measdata).

```
:MMEMory:STORe:RESults data
```

## :MMEMory:STORe:SCReen

### Syntax

:MMEMory:STORe:SCReen <file\_name>

### Description

Saves the current screen image with the specified filename suffixed with ".jpg", ".png", or ".bmp" to the default path ("/mode name"/screen).

**Parameter**

| Name        | Type         | Range | Default |
|-------------|--------------|-------|---------|
| <file_name> | ASCII String | —     | —       |

**Remarks**

If the specified file already exists, overwrite it.

If a suffix (.jpg/.png/.bmp) following the filename exists, you can save the current screen image with a different format based on its different suffix.

If no suffix is followed with the filename, then by default, the current screen image is saved in the currently selected format.

**Example**

The following command saves the current screen image with the filename "screen.jpg" to the folder (/gpsa/screen).

```
:MMEMory:STORe:SCReen screen.jpg
```

**:MMEMory:STORe:STATe****Syntax**

```
:MMEMory:STORe:STATe <file_name>
```

**Description**

Saves the current instrument state with the specified filename suffixed with ".sta" to the default path ("/mode name"/state).

**Parameter**

| Name        | Type         | Range | Default |
|-------------|--------------|-------|---------|
| <file_name> | ASCII String | —     | —       |

**Remarks**

If the specified file already exists, overwrite it.

**Example**

The following command saves the current instrument state with the filename "state.sta" to the folder (/gpsa/state).

```
:MMEMory:STORe:STATe state
```

**:MMEMory:STORe:TRACe****Syntax**

```
:MMEMory:STORe:TRACe <label>,<file_name>
```

**Description**

Saves the specified trace+ state file with filename suffixed with ".trs" to the default path ("/mode name"/tracestate).

**Parameter**

| Name        | Type         | Range                                     | Default |
|-------------|--------------|-------------------------------------------|---------|
| <label>     | Keyword      | TRACE1 TRACE2 TRACE3 TRACE4 TRACE5 TRACE6 | —       |
| <file_name> | ASCII String | —                                         | —       |

**Remarks**

If the specified file already exists, overwrite it.

**Example**

The following command saves the current instrument state and Trace 1 with the filename "mystate.trs" to the folder (/gpsa/tracestate).

```
:MMEMory:STORe:TRACe TRACE1,mystate
```

**:MMEMory:STORe:TRACe:DATA****Syntax**

```
:MMEMory:STORe:TRACe:DATA <label>,<file_name>
```

**Description**

Saves the trace measurement results with a specified filename suffixed with ".csv" to the default path ("/mode name"/measdata).

**Parameter**

| Name        | Type         | Range                                     | Default |
|-------------|--------------|-------------------------------------------|---------|
| <label>     | Keyword      | TRACE1 TRACE2 TRACE3 TRACE4 TRACE5 TRACE6 | ——      |
| <file_name> | ASCII String | ——                                        | ——      |

**Remarks**

If the specified file already exists, overwrite it.

**Example**

The following command saves Trace1 measurement data with the specified filename "mydata.csv" to the folder (/gpsa/measdata).

```
:MMEMory:STORe:TRACe:DATA TRACE1,mydata
```

## :OUTPut Commands

### Command List:

◆ [:OUTPut\[:EXternal\]\[:STATe\]](#)

## :OUTPut[:EXternal][:STATe]

### Syntax

:OUTPut[:EXternal][:STATe] OFF|ON|0|1

:OUTPut[:EXternal][:STATe]?

### Description

Enables or disables the output of the tracking generator.

Queries the on/off status of the tracking generator.

### Parameter

| Name | Type | Range      | Default |
|------|------|------------|---------|
| —    | Bool | OFF ON 0 1 | OFF 0   |

### Remarks

This command is only available for RSA3045-TG/RSA3030-TG working in GPSA mode.

### Return Format

The query returns 0 or 1.

### Example

The following command enables the output of the tracking generator.

:OUTPut:EXternal:STATe ON or :OUTPut:EXternal:STATe 1

The following query returns 1.

:OUTPut:EXternal:STATe?

## :READ Commands

Both the :READ and [:FETCh Commands](#) are used to acquire the measurement results. The difference is that the [:FETCh Commands](#) acquire the measurement results immediately, whereas the :READ commands start a measurement and return the measurement results after the measurement operation is finished.

### Command List:

- ◆ [:READ:ACPower?](#)
- ◆ [:READ:ACPower:LOWer?](#)
- ◆ [:READ:ACPower:MAIN?](#)
- ◆ [:READ:ACPower:UPPer?](#)
- ◆ [:READ:CNRatio?](#)
- ◆ [:READ:CNRatio:CARRier?](#)
- ◆ [:READ:CNRatio:CNRatio?](#)
- ◆ [:READ:CNRatio:NOISe?](#)
- ◆ [:READ:EBWidth?](#)
- ◆ [:READ:HARMonics:AMPLitude:ALL?](#)
- ◆ [:READ:HARMonics:AMPLitude? <n>](#)
- ◆ [:READ:HARMonics\[:DISTortion\]?](#)
- ◆ [:READ:HARMonics:FREQuency:ALL?](#)
- ◆ [:READ:HARMonics:FREQuency? <n>](#)
- ◆ [:READ:HARMonics:FUNDamental?](#)
- ◆ [:READ:OBWidth?](#)
- ◆ [:READ:OBWidth:OBWidth?](#)
- ◆ [:READ:OBWidth:OBWidth:FERRor?](#)
- ◆ [:READ:SANalyzer<n>?](#)
- ◆ [:READ:TOIntercept?](#)
- ◆ [:READ:TOIntercept:IP3?](#)
- ◆ [:READ:TPOWER?](#)

### Remarks:

The :READ commands are only available for the RSA3000 series that has been installed with advanced measurement kit (AMK) option.

## **:READ:ACPower?**

### **Syntax**

:READ:ACPower?

### **Description**

Executes one adjacent channel power measurement and returns the measurement results.

### **Remarks**

The power unit of the returned value is consistent with the current Y-axis unit.

This command will select data from the latest measurement results, and then transmit the data to the output buffer.

This command is only available for GPSA mode.

### **Return Format**

The query returns 5 values (main channel power, upper channel power, the power difference between the upper channel and the main channel (in dBc), the lower channel power, the power difference between the lower channel and the main channel (in dBc)) in scientific notation, and the values are separated by commas. For example,

-5.150423000e+01,-5.173441000e+01,-2.301865000e-01,-5.142665000e+01,7.757568000e-02

## **:READ:ACPower:LOWer?**

### **Syntax**

:READ:ACPower:LOWer?

### **Description**

Executes one adjacent channel power measurement and returns the lower channel power.

### **Remarks**

The power unit of the return value is consistent with the current Y-axis unit.

This command will select data from the latest measurement results, and then transmit the data to the output buffer.

This command is only available for GPSA mode.

### **Return Format**

The query returns the lower channel power in scientific notation. For example, -5.142665000e+01

## **:READ:ACPower:MAIN?**

### **Syntax**

:READ:ACPower:MAIN?

### **Description**

Executes one adjacent channel power measurement and returns the main channel power.

### **Remarks**

The power unit of the return value is consistent with the current Y-axis unit.

This command will select data from the latest measurement results, and then transmit the data to the output buffer.

This command is only available for GPSA mode.

### **Return Format**

The query returns the main channel power in scientific notation. For example, -5.150423000e+01

## **:READ:ACPower:UPPer?**

### **Syntax**

:READ:ACPower:UPPer?

### **Description**

Executes one adjacent channel power measurement and returns the upper channel power.

### **Remarks**

The power unit of the return value is consistent with the current Y-axis unit.

This command will select data from the latest measurement results, and then transmit the data to the output buffer.

This command is only available for GPSA mode.

### **Return Format**

The query returns the upper channel power in scientific notation. For example, -5.173441000e+ 01

## **:READ:CNRatio?**

### **Syntax**

:READ:CNRatio?

### **Description**

Executes one C/N ratio measurement and returns the measurement results.

### **Remarks**

The power unit of the return value is consistent with the current Y-axis unit.

This command will select data from the latest measurement results, and then transmit the data to the output buffer.

This command is only available for GPSA mode.

### **Return Format**

The query returns the carrier power, noise power, and C/N ratio (in dB) in scientific notation (separated by commas).

For example, -6.048788000e+ 01,-6.186192000e+ 01,1.374039000e+ 00

## **:READ:CNRatio:CARRier?**

### **Syntax**

:READ:CNRatio:CARRier?

### **Description**

Executes one C/N ratio measurement and returns the carrier power.

### **Remarks**

The power unit of the return value is consistent with the current Y-axis unit.

This command will select data from the latest measurement results, and then transmit the data to the output buffer.

This command is only available for GPSA mode.

### **Return Format**

The query returns the carrier power in scientific notation. For example, -1.484203000e+ 01

## **:READ:CNRatio:CNRatio?**

### **Syntax**

:READ:CNRatio:CNRatio?

### **Description**

Executes one C/N ratio measurement and returns the C/N ratio.

### **Remarks**

This command will select data from the latest measurement results, and then transmit the data to the output buffer.

This command is only available for GPSA mode.

### **Return Format**

The query returns the C/N ratio in scientific notation. For example, 8.956909000e-02

## **:READ:CNRatio:NOI Se?**

### **Syntax**

:READ:CNRatio:NOI Se?

### **Description**

Executes one C/N ratio measurement and returns the noise power.

### **Remarks**

The power unit of the return value is consistent with the current Y-axis unit.

This command will select data from the latest measurement results, and then transmit the data to the output buffer.

This command is only available for GPSA mode.

### **Return Format**

The query returns the noise power in scientific notation. For example, -1.442294000e+01

## **:READ:EBWidth?**

### **Syntax**

:READ:EBWidth?

### **Description**

Executes one emission bandwidth measurement and returns the measurement results.

### **Remarks**

This command will select data from the latest measurement results, and then transmit the data to the output buffer.

This command is only available for GPSA mode.

### **Return Format**

The query returns the emission bandwidth measurement results in scientific notation, and its unit is Hz. For example, 5.000000000+04

## :READ:HARMonics:AMPLitude:ALL?

### Syntax

:READ:HARMonics:AMPLitude:ALL?

### Description

Executes one harmonic distortion measurement and returns the amplitudes of the first 10 harmonics.

### Remarks

If the number of harmonics under measurement is less than 10, the harmonics that are not measured do not have return values.

The amplitude unit of the return value is consistent with the current Y-axis unit.

This command will select data from the latest measurement results, and then transmit the data to the output buffer.

This command is only available for GPSA mode.

### Return Format

The query returns the amplitudes of the first 10 harmonics in scientific notation. They are separated by commas.

For example,

-1.692102000e+01,-6.458423000e+01,-7.509421000e+01,-7.924328000e+01,-7.847027000e+01,-7.885457000e+01,-7.882358000e+01,-7.921457000e+01,-7.923057000e+01,-7.915358000e+01

## :READ:HARMonics:AMPLitude? < n>

### Syntax

:READ:HARMonics:AMPLitude? < n>

### Description

Executes one harmonic distortion measurement and returns the amplitude of the specified harmonic.

### Parameter

| Name | Type    | Range   | Default |
|------|---------|---------|---------|
| < n> | Integer | 1 to 10 | —       |

### Remarks

The amplitude unit of the return value is consistent with the current Y-axis unit.

The query returns --- when the harmonic read does not have data.

This command will select data from the latest measurement results, and then transmit the data to the output buffer.

This command is only available for GPSA mode.

### Return Format

The query returns the amplitude of the specified harmonic in scientific notation.

For example, -1.692102000e+01

## :READ:HARMonics[:DIStortion]?

### Syntax

:READ:HARMonics[:DIStortion]?

### Description

Executes one harmonic distortion measurement and returns the percentage of the total harmonic distortion.

### Remarks

This command will select data from the latest measurement results, and then transmit the data to the output buffer.

This command is only available for GPSA mode.

### Return Format

Queries the percentage of the total harmonic distortion in scientific notation. For example, 2.490393000e+02

## :READ:HARMonics:FREQuency:ALL?

### Syntax

:READ:HARMonics:FREQuency:ALL?

### Description

Executes one harmonic distortion measurement and returns the frequencies of the first 10 harmonics.

### Remarks

If the number of harmonics under measurement is less than 10, the harmonics that are not measured do not have return values.

This command will select data from the latest measurement results, and then transmit the data to the output buffer.

This command is only available for GPSA mode.

### Return Format

The query returns the frequencies in scientific notation (separated by commas). The unit is Hz. For example, 4.550000000e+07,9.100000000e+07,1.365000000e+07

## :READ:HARMonics:FREQuency? <n>

### Syntax

:READ:HARMonics:FREQuency? <n>

### Description

Executes one harmonic distortion measurement and returns the frequency of the specified harmonic.

### Parameter

| Name | Type    | Range   | Default |
|------|---------|---------|---------|
| <n>  | Integer | 1 to 10 | —       |

### Remarks

The query returns --- when the harmonic read does not have data.

This command will select data from the latest measurement results, and then transmit the data to the output buffer.

This command is only available for GPSA mode.

**Return Format**

The query returns the frequency of the specified harmonic in scientific notation. The unit is Hz. For example, 4.550000000e+07

**:READ:HARMonics:FUNDamental?****Syntax**

:READ:HARMonics:FUNDamental?

**Description**

Executes one harmonic distortion measurement and returns the frequency of the fundamental waveform.

**Remarks**

This command has the same function as :READ:HARMonics:FREQuency? 1.

This command will select data from the latest measurement results, and then transmit the data to the output buffer.

This command is only available for GPSA mode.

**Return Format**

The query returns the frequency of the fundamental waveform in scientific notation. The unit is Hz. For example, 4.550000000e+07

**:READ:OBWidth?****Syntax**

:READ:OBWidth?

**Description**

Executes one occupied bandwidth measurement and returns the measurement results.

**Remarks**

This command will select data from the latest measurement results, and then transmit the data to the output buffer.

This command is only available for GPSA mode.

**Return Format**

The query returns the occupied bandwidth (Hz) and the transmit frequency error (Hz) in scientific notation (separated by commas).

For example, 1.860000000e+06,2.000000000e+04

## **:READ:OBWidth:OBWidth?**

### **Syntax**

:READ:OBWidth:OBWidth?

### **Description**

Executes one occupied bandwidth measurement and returns the occupied bandwidth.

### **Remarks**

This command will select data from the latest measurement results, and then transmit the data to the output buffer.

This command is only available for GPSA mode.

### **Return Format**

The query returns the occupied bandwidth in scientific notation. Its unit is Hz. For example, 1.860000000e+06

## **:READ:OBWidth:OBWidth:FERRor?**

### **Syntax**

:READ:OBWidth:OBWidth:FERRor?

### **Description**

Executes one occupied bandwidth measurement and returns the transmit frequency error.

### **Remarks**

This command will select data from the latest measurement results, and then transmit the data to the output buffer.

This command is only available for GPSA mode.

### **Return Format**

The query returns the transmit frequency error in scientific notation. The unit is Hz. For example, 2.000000000+04

## :READ:SANalyzer< n> ?

### Syntax

:READ:SANalyzer< n> ?

### Description

Queries the measurement results or trace data in the buffer.

### Parameter

| Name | Type     | Range       | Default |
|------|----------|-------------|---------|
| < n> | Discrete | 1 2 3 4 5 6 | —       |

### Remarks

This command will select data from the latest measurement results, and then transmit the data to the output buffer.

This command is only available for GPSA mode.

### Return Format

The query returns the measurement data of Trace n in scientific notation. The measurement data are returned with a pair of data (x,y), separated by commas. The unit is Hz.

### Example

The following query returns

0,0,0,0,-1.000000000e+02,0,801,0,0,0,-3.19e+01,0.00e+00,0.00e+00,0.00e+00,0.00e+00,0.00e+00,0.00e+00,0.00e+00

:FETCh:SANalyzer1?

## :READ:TOI ntercept?

### Syntax

:READ:TOI ntercept?

### Description

Executes one TOI measurement and returns the measurement results.

### Remarks

The amplitude unit of the return value is consistent with the current Y-axis unit.

This command will select data from the latest measurement results, and then transmit the data to the output buffer.

This command is only available for GPSA mode.

### Return Format

The query returns the measurement results of TOI in the following format:

Base Lower frequency (Hz), amplitude, Base Upper frequency (Hz), amplitude, 3rd Order Lower frequency (Hz), amplitude, intercept, 3rd Order Upper frequency (Hz), amplitude, intercept.

For example,

1.500450000e+09,-8.131735000e+01,1.500450000e+09,-8.131735000e+01,1.500450000e+09,-8.131735000e+01,-8.131735000e+01,1.500450000e+09,-8.131735000e+01,-8.131735000e+01

## **:READ:TOIntercept:IP3?**

### **Syntax**

:READ:TOIntercept:IP3?

### **Description**

Executes one TOI measurement and returns the value of the intercept of the Third Order Lower or the Third Order Upper, whichever is smaller.

### **Remarks**

This command will select data from the latest measurement results, and then transmit the data to the output buffer.

This command is only available for GPSA mode.

### **Return Format**

The query returns the minor one in scientific notation. For example, -8.131735000e+01

## **:READ:TPOWer?**

### **Syntax**

:READ:TPOWer?

### **Description**

Executes one T-power measurement and returns the measurement results.

### **Remarks**

The power unit of the return value is consistent with the current Y-axis unit.

This command will select data from the latest measurement results, and then transmit the data to the output buffer.

This command is only available for GPSA mode.

### **Return Format**

The query returns the T-power measurement result in scientific notation. For example, -1.658941000e+01

## [ :SENSe] Commands

### Command List:

- ◆ [\[:SENSe\]:ACPower:AVERage:COUNT](#)
- ◆ [\[:SENSe\]:ACPower:AVERage\[:STATe\]](#)
- ◆ [\[:SENSe\]:ACPower:AVERage:TCONtrol](#)
- ◆ [\[:SENSe\]:ACPower:BANDwidth:ACHannel](#)
- ◆ [\[:SENSe\]:ACPower:BANDwidth:INTEgration](#)
- ◆ [\[:SENSe\]:ACPower:CSPacing](#)
- ◆ [\[:SENSe\]:ACQuisition:TIME](#)
- ◆ [\[:SENSe\]:ACQuisition:TIME:AUTO](#)
- ◆ [\[:SENSe\]:ACQuisition:TIME:PVTime](#)
- ◆ [\[:SENSe\]:ACQuisition:TIME:PVTime:AUTO](#)
- ◆ [\[:SENSe\]:AVERage:COUNT](#)
- ◆ [\[:SENSe\]:AVERage:COUNT:CURRent?](#)
- ◆ [:TRACe:AVERage:COUNT:CURRent?](#)
- ◆ [\[:SENSe\]:AVERage:TYPE](#)
- ◆ [:TRACe<n>:AVERage:TYPE](#)
- ◆ [\[:SENSe\]:AVERage:TYPE:AUTO](#)
- ◆ [\[:SENSe\]:BANDwidth|BWIDth:EMI Filter:STATe](#)
- ◆ [\[:SENSe\]:BANDwidth|BWIDth\[:RESolution\]](#)
- ◆ [\[:SENSe\]:BANDwidth|BWIDth\[:RESolution\]:AUTO](#)
- ◆ [\[:SENSe\]:BANDwidth|BWIDth\[:RESolution\]:SElect](#)
- ◆ [\[:SENSe\]:BANDwidth|BWIDth\[:RESolution\]:SElect:AUTO\[:STATe\]](#)
- ◆ [\[:SENSe\]:BANDwidth|BWIDth:SHAPE](#)
- ◆ [\[:SENSe\]:BANDwidth|BWIDth:VIDeo](#)
- ◆ [\[:SENSe\]:BANDwidth|BWIDth:VIDeo:AUTO](#)
- ◆ [\[:SENSe\]:BANDwidth|BWIDth:VIDeo:RATio](#)
- ◆ [\[:SENSe\]:BANDwidth|BWIDth:VIDeo:RATio:AUTO](#)
- ◆ [\[:SENSe\]:AVERage:COUNT:CURRent](#)
- ◆ [\[:SENSe\]:CNRatio:AVERage:COUNT\\*](#)
- ◆ [\[:SENSe\]:CNRatio:AVERage\[:STATe\] \\*](#)
- ◆ [\[:SENSe\]:CNRatio:AVERage:TCONtrol\\*](#)
- ◆ [\[:SENSe\]:CNRatio:BANDwidth:INTEgration\\*](#)
- ◆ [\[:SENSe\]:CNRatio:BANDwidth:NOISe\\*](#)
- ◆ [\[:SENSe\]:CNRatio:OFFSet\\*](#)
- ◆ [:INPut:IMPedance](#)
- ◆ [\[:SENSe\]:CORRection:IMPedance\[:INPut\]\[:MAGNitude\]](#)
- ◆ [\[:SENSe\]:CORRection:SA\[:RF\]:GAIN](#)

- ◆ [\[:SENSe\]:DEMod](#)
- ◆ [\[:SENSe\]:DEMod:GAIN:AUTO](#)
- ◆ [\[:SENSe\]:DEMod:GAIN:INCRement](#)
- ◆ [\[:SENSe\]:DEMod:STATe](#)
- ◆ [\[:SENSe\]:DETEctor:TRACe:PVTIme](#)
- ◆ [\[:SENSe\]:DETEctor\[:FUNCTion\]](#)
- ◆ [\[:SENSe\]:DETEctor:TRACe<n>](#)
- ◆ [\[:SENSe\]:DETEctor:TRACe<n>:AUTO](#)
- ◆ [\[:SENSe\]:EBWidth:AVERage:COUNt\\*](#)
- ◆ [\[:SENSe\]:EBWidth:AVERage\[:STATe\]\\*](#)
- ◆ [\[:SENSe\]:EBWidth:AVERage:TCONtrol\\*](#)
- ◆ [\[:SENSe\]:EBWidth:FREQuency:SPAN\\*](#)
- ◆ [\[:SENSe\]:EBWidth:MAXHold:STATe\\*](#)
- ◆ [\[:SENSe\]:EBWidth:XDB\\*](#)
- ◆ [\[:SENSe\]:FREQuency:CENTer](#)
- ◆ [\[:SENSe\]:FREQuency:CENTer:STEP:AUTO](#)
- ◆ [\[:SENSe\]:FREQuency:CENTer:STEP\[:INCRement\]](#)
- ◆ [\[:SENSe\]:FREQuency:OFFSet](#)
- ◆ [\[:SENSe\]:FREQuency:SPAN](#)
- ◆ [\[:SENSe\]:FREQuency:SPAN:BANDwidth\[:RESolution\]:RATio](#)
- ◆ [\[:SENSe\]:FREQuency:SPAN:BANDwidth\[:RESolution\]:RATio:AUTO](#)
- ◆ [\[:SENSe\]:FREQuency:SPAN:FULL](#)
- ◆ [\[:SENSe\]:FREQuency:SPAN:PREVious](#)
- ◆ [\[:SENSe\]:FREQuency:SPAN:ZERO](#)
- ◆ [\[:SENSe\]:FREQuency:STARt](#)
- ◆ [\[:SENSe\]:FREQuency:STOP](#)
- ◆ [\[:SENSe\]:FREQuency:TUNE:IMMediate](#)
- ◆ [\[:SENSe\]:HDISt:AVERage:COUNt\\*](#)
- ◆ [\[:SENSe\]:HDISt:AVERage\[:STATe\]\\*](#)
- ◆ [\[:SENSe\]:HDISt:AVERage:TCONtrol\\*](#)
- ◆ [\[:SENSe\]:HDISt:NUMBers\\*](#)
- ◆ [\[:SENSe\]:HDISt:TIME\\*](#)
- ◆ [\[:SENSe\]:MCHPower:AVERage:COUNt\\*](#)
- ◆ [\[:SENSe\]:MCHPower:AVERage\[:STATe\]\\*](#)
- ◆ [\[:SENSe\]:MCHPower:AVERage:TCONtrol\\*](#)
- ◆ [\[:SENSe\]:OBWidth:AVERage:COUNt\\*](#)
- ◆ [\[:SENSe\]:OBWidth:AVERage\[:STATe\]\\*](#)
- ◆ [\[:SENSe\]:OBWidth:AVERage:TCONtrol\\*](#)
- ◆ [\[:SENSe\]:OBWidth:FREQuency:SPAN\\*](#)

- ◆ [\[:SENSe\]:OBWidth:MAXHold:STATe\\*](#)
- ◆ [\[:SENSe\]:OBWidth:PERCent\\*](#)
- ◆ [\[:SENSe\]:POWer\[:RF\]:ATTenuation](#)
- ◆ [\[:SENSe\]:POWer\[:RF\]:ATTenuation:AUTO](#)
- ◆ [\[:SENSe\]:POWer\[:RF\]:GAIN\[:STATe\]](#)
- ◆ [\[:SENSe\]:POWer\[:RF\]:MI Xer:RANGe\[:UPPer\]](#)
- ◆ [\[:SENSe\]:SIGCapture:2FSK:AMPDown\\*](#)
- ◆ [\[:SENSe\]:SIGCapture:2FSK:AMPUp\\*](#)
- ◆ [\[:SENSe\]:SIGCapture:2FSK:MARK1:FREQ\\*](#)
- ◆ [\[:SENSe\]:SIGCapture:2FSK:MARK1:SWitch\[:STATe\]\\*](#)
- ◆ [\[:SENSe\]:SIGCapture:2FSK:MARK2:FREQ\\*](#)
- ◆ [\[:SENSe\]:SIGCapture:2FSK:MARK2:SWitch\[:STATe\]\\*](#)
- ◆ [\[:SENSe\]:SIGCapture:2FSK:MAXHold\[:STATe\]\\*](#)
- ◆ [\[:SENSe\]:SIGCapture:2FSK:PEAKAmp?\\*](#)
- ◆ [\[:SENSe\]:SIGCapture:2FSK:PEAKFreq?\\*](#)
- ◆ [\[:SENSe\]:SIGCapture:2FSK:PF?](#)
- ◆ [\[:SENSe\]:SIGCapture:2FSK:PFSWitch\[:STATe\]\\*](#)
- ◆ [\[:SENSe\]:SIGCapture:2FSK:RESet\\*](#)
- ◆ [\[:SENSe\]:SIGCapture:2FSK:SIGNal\\*](#)
- ◆ [\[:SENSe\]:SWEep:POINts](#)
- ◆ [\[:SENSe\]:SWEep:TIME](#)
- ◆ [\[:SENSe\]:SWEep:TIME:AUTO](#)
- ◆ [\[:SENSe\]:SWEep:TIME:AUTO:RULEs](#)
- ◆ [\[:SENSe\]:TOI:AVERage:COUNt\\*](#)
- ◆ [\[:SENSe\]:TOI:AVERage\[:STATe\]\\*](#)
- ◆ [\[:SENSe\]:TOI:AVERage:TCONtrol\\*](#)
- ◆ [\[:SENSe\]:TOI:FREQuency:SPAN\\*](#)
- ◆ [\[:SENSe\]:TPOWer:AVERage:COUNt\\*](#)
- ◆ [\[:SENSe\]:TPOWer:AVERage\[:STATe\]\\*](#)
- ◆ [\[:SENSe\]:TPOWer:AVERage:TCONtrol\\*](#)
- ◆ [\[:SENSe\]:TPOWer:LLIMit\\*](#)
- ◆ [\[:SENSe\]:TPOWer:MODE\\*](#)
- ◆ [\[:SENSe\]:TPOWer:RLIMit\\*](#)

**Remarks:** The commands with \* are only available for the RSA3000 series that has been installed with advanced measurement kit (AMK) option.

## [ :SENSe]:ACPower:AVERage:COUNT

### Syntax

[ :SENSe]:ACPower:AVERage:COUNT <integer>

[ :SENSe]:ACPower:AVERage:COUNT?

### Description

Sets the average count of the ACP measurement.

Queries the average count of the ACP measurement.

### Parameter

| Name      | Type    | Range      | Default |
|-----------|---------|------------|---------|
| <integer> | Integer | 1 to 1,000 | 10      |

### Remarks

This command is only valid when the adjacent channel power measurement in GPSA mode is enabled.

### Return Format

The query returns the average count in integer.

### Example

The following command sets the average count to 100.

:SENSe:ACPower:AVERage:COUNT 100

The following query returns 100.

:SENSe:ACPower:AVERage:COUNT?

## [ :SENSe]:ACPower:AVERage[:STATe]

### Syntax

[ :SENSe]:ACPower:AVERage[:STATe] OFF|ON|0|1

[ :SENSe]:ACPower:AVERage[:STATe]?

### Description

Enables or disables the average measurement function of the ACP measurement.

Queries the status of the average measurement function of the ACP measurement.

### Parameter

| Name | Type | Range      | Default |
|------|------|------------|---------|
| —    | Bool | OFF ON 0 1 | OFF 0   |

### Remarks

This command is only valid when the adjacent channel power measurement in GPSA mode is enabled.

### Return Format

The query returns 0 or 1.

### Example

The following command enables the average measurement function.

:SENSe:ACPower:AVERage:STATe ON or :SENSe:ACPower:AVERage:STATe 1

The following query returns 1.

:SENSe:ACPower:AVERage:STATe?

## [ :SENSe]:ACPower:AVERage:TCONtrol

### Syntax

[ :SENSe]:ACPower:AVERage:TCONtrol EXPonential| REPeat  
[ :SENSe]:ACPower:AVERage:TCONtrol?

### Description

Selects the average mode of the adjacent channel power measurement.  
Queries the average mode of the adjacent channel power measurement.

### Parameter

| Name | Type    | Range               | Default     |
|------|---------|---------------------|-------------|
| —    | Keyword | EXPonential  REPeat | EXPonential |

### Remarks

EXPonential: indicates the exponential average.

REPeat: indicates the repeat average.

When "EXPonential" is selected, the result is the exponential average of the measurement results obtained in the past N times (N is specified in [\[:SENSe\]:ACPower:AVERage:COUNt](#)).

When "REPeat" is selected, the result is the arithmetic average of the measurement results obtained in the past N times (N is specified in [\[:SENSe\]:ACPower:AVERage:COUNt](#)).

This command is only valid when the adjacent channel power measurement in GPSA mode is enabled.

### Return Format

The query returns EXP or REP.

### Example

The following command selects repeat average to the average mode.

```
:SENSe:ACPower:AVERage:TCONtrol REPeat
```

The following query returns REP.

```
:SENSe:ACPower:AVERage:TCONtrol?
```

## [ :SENSe]:ACPower:BANDwidth:ACHannel

### Syntax

[ :SENSe]:ACPower:BANDwidth:ACHannel <freq>  
[ :SENSe]:ACPower:BANDwidth:ACHannel?

### Description

Sets the bandwidth of the adjacent channel.

Queries the bandwidth of the adjacent channel.

### Parameter

| Name   | Type                    | Range <sup>[1]</sup> | Default |
|--------|-------------------------|----------------------|---------|
| <freq> | Consecutive Real Number | 33 Hz to 1.5 GHz     | 2 MHz   |

**Note<sup>[1]</sup>:** If option RSA3000-BW1 is installed, the available range is from 3 Hz to 1.5 GHz.

### Remarks

This command is only valid when the adjacent channel power measurement in GPSA mode is enabled.

### Return Format

The query returns the bandwidth of the adjacent channel in scientific notation. Its unit is Hz.

**Example**

The following command sets the bandwidth of the adjacent channel to 1 MHz.

```
:SENSe:ACPower:BANDwidth:ACHannel 1000000
```

The following query returns 1.000000000e+06.

```
:SENSe:ACPower:BANDwidth:ACHannel?
```

**[ :SENSe]:ACPower:BANDwidth:INTegration****Syntax**

```
[ :SENSe]:ACPower:BANDwidth:INTegration <freq>
```

```
[ :SENSe]:ACPower:BANDwidth:INTegration?
```

**Description**

Sets the bandwidth of the main channel.

Queries the bandwidth of the main channel.

**Parameter**

| Name   | Type                    | Range <sup>[1]</sup> | Default |
|--------|-------------------------|----------------------|---------|
| <freq> | Consecutive Real Number | 33 Hz to 1.5 GHz     | 2 MHz   |

**Note<sup>[1]</sup>:** If option RSA3000-BW1 is installed, the available range is from 3 Hz to 1.5 GHz.

**Remarks**

This command is only valid when the adjacent channel power measurement in GPSA mode is enabled.

**Return Format**

The query returns the bandwidth of the main channel in scientific notation. Its unit is Hz.

**Example**

The following command sets the bandwidth of the main channel to 1 MHz.

```
:SENSe:ACPower:BANDwidth:INTegration 1000000
```

The following query returns 1.000000000e+06.

```
:SENSe:ACPower:BANDwidth:INTegration?
```

**[ :SENSe]:ACPower:CSPacing****Syntax**

```
[ :SENSe]:ACPower:CSPacing <freq>
```

```
[ :SENSe]:ACPower:CSPacing?
```

**Description**

Sets the center frequency difference (channel spacing) between the main channel and the adjacent channels.

Queries the channel spacing.

**Parameter**

| Name   | Type                    | Range <sup>[1]</sup> | Default |
|--------|-------------------------|----------------------|---------|
| <freq> | Consecutive Real Number | 33 Hz to 1.5 GHz     | 2 MHz   |

**Note<sup>[1]</sup>:** If option RSA3000-BW1 is installed, the available range is from 3 Hz to 1.5 GHz.

**Remarks**

This command is only valid when the adjacent channel power measurement in GPSA mode is enabled.

**Return Format**

The query returns the channel spacing in scientific notation. Its unit is Hz.

**Example**

The following command sets the channel spacing to 1 MHz.

```
:SENSe:ACPower:CSPacing 1000000
```

The following query returns 1.000000000e+06.

```
:SENSe:ACPower:CSPacing?
```

**[ :SENSe]:ACQuisition:TI ME****Syntax**

```
[ :SENSe]:ACQuisition:TIME <time>
```

```
[ :SENSe]:ACQuisition:TIME?
```

**Description**

Sets the acquisition time for producing one single trace or one bitmap. The unit is s.

Queries the acquisition time for producing one single trace or one bitmap.

**Parameter**

| Name   | Type                    | Range                                    | Default |
|--------|-------------------------|------------------------------------------|---------|
| <time> | Consecutive Real Number | 32 ms to 40 s<br>100 μs to 40 s (others) | 1 ms    |

**Remarks**

This command is only available for Normal, Density, Spectrogram, and Density Spectrogram in RTSA mode.

**Return Format**

The query returns the acquisition time in scientific notation.

**Example**

The following command sets the acquisition time to 0.5 s.

```
:SENSe:ACQuisition:TIME 0.5
```

The following query returns 5.000000000e-01.

```
:SENSe:ACQuisition:TIME?
```

**[ :SENSe]:ACQuisition:TI ME:AUTO****Syntax**

```
[ :SENSe]:ACQuisition:TIME:AUTO OFF|ON|0|1
```

```
[ :SENSe]:ACQuisition:TIME:AUTO?
```

**Description**

Enables or disables the auto acquisition time for producing one single trace or one bitmap.

Queries the status of the auto acquisition time for producing one single trace or one bitmap.

**Parameter**

| Name | Type | Range      | Default |
|------|------|------------|---------|
| —    | Bool | OFF ON 0 1 | ON 1    |

**Remarks**

This command is only available for Normal, Density, Spectrogram, and Density Spectrogram in RTSA mode.

**Return Format**

The query returns 0 or 1.

**Example**

The following command enables the auto acquisition time for producing one single trace or one bitmap.

:SENSe:ACQuisition:TIME:AUTO ON or :SENSe:ACQuisition:TIME:AUTO 1

The following query returns 1.

:SENSe:ACQuisition:TIME:AUTO?

**[ :SENSe ] :ACQuisition:TI ME:PVTime****Syntax**

[ :SENSe ] :ACQuisition:TIME:PVTime <time>

[ :SENSe ] :ACQuisition:TIME:PVTime?

**Description**

Sets the acquisition time for producing a trace. The unit is s.

Queries the acquisition time for producing a trace.

**Parameter**

| Name   | Type                    | Range       | Default |
|--------|-------------------------|-------------|---------|
| <time> | Consecutive Real Number | 0 s to 40 s | 30 ms   |

**Remarks**

This command is only available for PvT, PvT Spectrum, and PvT Spectrogram in RTSA mode.

**Return Format**

The query returns the acquisition time in scientific notation.

**Example**

The following command sets the acquisition time to 0.5 s.

:SENSe:ACQuisition:TIME:PVTime 0.5

The following query returns 5.000000000e-01.

:SENSe:ACQuisition:TIME:PVTime?

## [ :SENSe]:ACQuisition:TI ME:PVTime:AUTO

### Syntax

```
[ :SENSe]:ACQuisition:TIME:PVTime:AUTO OFF|ON|0|1
[ :SENSe]:ACQuisition:TIME:PVTime:AUTO?
```

### Description

Enables or disables the auto acquisition time for producing a trace.  
Queries the status of the auto acquisition time for producing a trace.

### Parameter

| Name | Type | Range      | Default |
|------|------|------------|---------|
| —    | Bool | OFF ON 0 1 | ON 1    |

### Remarks

This command is only available for PvT, PvT Spectrum, and PvT Spectrogram in RTSA mode.

### Return Format

The query returns 0 or 1.

### Example

The following command enables the auto acquisition time for producing all the traces.  
:SENSe:ACQuisition:TIME:PVTime:AUTO ON or :SENSe:ACQuisition:TIME:PVTime:AUTO 1

The following query returns 1.  
:SENSe:ACQuisition:TIME:PVTime:AUTO?

## [ :SENSe]:AVERAge:COUNT :TRACe:AVERAge:COUNT

### Syntax

```
[ :SENSe]:AVERAge:COUNT <integer>
[ :SENSe]:AVERAge:COUNT?
:TRACe:AVERAge:COUNT <integer>
:TRACe:AVERAge:COUNT?
```

### Description

Sets the trace average count of the current measurement.  
Queries the trace average count of the current measurement.

### Parameter

| Name      | Type    | Range       | Default |
|-----------|---------|-------------|---------|
| <integer> | Integer | 1 to 10,000 | 100     |

### Remarks

This command is available for the swept SA measurement in GPSA mode and the RTSA mode.

### Return Format

The query returns the average count in integer.

### Example

The following command sets the average count to 100.  
:SENSe:AVERAge:COUNT 100  
:TRACe:AVERAge:COUNT 100

The following query returns 100.

```
:SENSe:AVERage:COUNT?
```

```
:TRACe:AVERage:COUNT?
```

## **[ :SENSe]:AVERage:COUNT:CURRent? :TRACe:AVERage:COUNT:CURRent?**

### **Syntax**

```
[ :SENSe]:AVERage:COUNT:CURRent?
```

```
:TRACe:AVERage:COUNT:CURRent?
```

### **Description**

Queries the current average times of the average trace.

### **Return Format**

The query returns the current trace average count in integer.

## **[ :SENSe]:AVERage:TYPE :TRACe< n> :AVERage:TYPE**

### **Syntax**

```
[ :SENSe]:AVERage:TYPE LOG| RMS| SCALar
```

```
[ :SENSe]:AVERage:TYPE?
```

```
:TRACe< n> :AVERage:TYPE LOG| RMS| SCALar
```

```
:TRACe< n> :AVERage:TYPE?
```

### **Description**

Selects the average type of the swept SA analysis measurement.

Queries the average type of the swept SA analysis measurement.

### **Parameter**

| Name | Type     | Range            | Default |
|------|----------|------------------|---------|
| < n> | Discrete | 1 2 3 4 5 6      | —       |
| —    | Keyword  | LOG  RMS  SCALar | LOG     |

### **Remarks**

LOG: indicates the log-power averaging.

RMS: indicates the power averaging.

SCALar: indicates the voltage averaging.

This command is only available for GPSA mode.

### **Return Format**

The query returns LOG, RMS, or SCAL.

### **Example**

The following command sets the average type of the swept SA analysis mode to Log.

```
:SENSe:AVERage:TYPE LOG
```

```
:TRACe1:AVERage:TYPE LOG
```

The following query returns LOG.

```
:SENSe:AVERage:TYPE?
```

```
:TRACe1:AVERage:TYPE LOG
```

## [ :SENSe]:AVERage:TYPE:AUTO

### Syntax

```
[ :SENSe]:AVERage:TYPE:AUTO OFF|ON|0|1
[ :SENSe]:AVERage:TYPE:AUTO?
```

### Description

Enables or disables the auto state for the average type of the swept SA analysis measurement.  
Queries the on/off auto state for the average type of the swept SA analysis measurement.

### Parameter

| Name | Type | Range      | Default |
|------|------|------------|---------|
| —    | Bool | OFF ON 0 1 | OFF 0   |

### Remarks

This command is only available for GPSA mode.

### Return Format

The query returns 0 or 1.

### Example

The following command enables the auto state for the average type of the swept SA analysis mode.  
:SENSe:AVERage:TYPE:AUTO ON or :SENSe:AVERage:TYPE:AUTO 1

The following query returns 1.  
:SENSe:AVERage:TYPE:AUTO?

## [ :SENSe]:BANDwidth| BWIDth:EMI Filter:STATe

### Syntax

```
[ :SENSe]:BANDwidth| BWIDth:EMI Filter:STATe OFF|ON|0|1
[ :SENSe]:BANDwidth| BWIDth:EMI Filter:STATe?
```

### Description

Enables or disables the EMI filter.  
Queries the status of the EMI filter.

### Parameter

| Name | Type | Range      | Default |
|------|------|------------|---------|
| —    | Bool | OFF ON 0 1 | OFF 0   |

### Remarks

ON: selects the EMI filter (-6 dB bandwidth).  
OFF: selects the Gaussian filter (-3 dB bandwidth).  
This command is only available for GPSA mode installed with the RSA3000-EMC option.

### Return Format

The query returns 0 or 1.

### Example

The following command selects the EMI filter.  
:SENSe:BANDwidth:EMI Filter:STATe ON or :SENSe:BANDwidth:EMI Filter:STATe 1  
:SENSe:BWIDth:EMI Filter:STATe ON or :SENSe:BWIDth:EMI Filter:STATe 1

The following query returns 1.

```
:SENSe:BANDwidth:EMI Filter:STATe?
```

```
:SENSe:BWIDth:EMI Filter:STATe?
```

## [ :SENSe]:BANDwidth| BWIDth[:RESolution]

### Syntax

```
[ :SENSe]:BANDwidth| BWIDth[:RESolution] <freq>
```

```
[ :SENSe]:BANDwidth| BWIDth[:RESolution] ?
```

### Description

Sets the resolution bandwidth (RBW).

Queries the resolution bandwidth.

### Parameter

| Name   | Type     | Range <sup>[1]</sup>            | Default |
|--------|----------|---------------------------------|---------|
| <freq> | Discrete | 10 Hz to 3 MHz (at 1-3-10 step) | 3 MHz   |

**Note<sup>[1]</sup>:** If option RSA3000-BW1 is installed, the available range is from 1 Hz to 10 MHz.

### Remarks

The setting command is only available for GPSA mode. The query command is available for both the GPSA and RTSA modes.

### Return Format

In GPSA mode, the query returns the resolution bandwidth in scientific notation. Its unit is Hz.

In RTSA mode, the query returns the specific frequency resolution value of RBW1 through RBW6.

### Example

The following command sets RBW to 1,000 Hz.

```
:SENSe:BANDwidth:RESolution 1000
```

```
:SENSe:BWIDth:RESolution 1000
```

The following query returns 1.000000000e+03.

```
:SENSe:BANDwidth:RESolution?
```

```
:SENSe:BWIDth:RESolution?
```

## [ :SENSe]:BANDwidth| BWIDth[:RESolution]:AUTO

### Syntax

```
[ :SENSe]:BANDwidth| BWIDth[:RESolution]:AUTO OFF| ON| 0| 1
```

```
[ :SENSe]:BANDwidth| BWIDth[:RESolution]:AUTO?
```

### Description

Enables or disables the auto setting mode of RBW.

Queries the status of the auto setting mode of RBW.

### Parameter

| Name | Type | Range         | Default |
|------|------|---------------|---------|
| —    | Bool | OFF  ON  0  1 | ON  1   |

### Remarks

In auto mode, the resolution bandwidth changes with the span (non-zero span).

This command is only available for GPSA mode.

### Return Format

The query returns 0 or 1.

### Example

The following command enables the auto setting mode of RBW.

```
:SENSe:BANDwidth:RESolution:AUTO ON or :SENSe:BANDwidth:RESolution:AUTO 1
:SENSe:BWIDth:RESolution:AUTO ON or :SENSe:BWIDth:RESolution:AUTO 1
```

The following query returns 1.

```
:SENSe:BANDwidth:RESolution:AUTO?
:SENSe:BWIDth:RESolution:AUTO?
```

## [ :SENSe]:BANDwidth| BWIDth[:RESolution]:SElect

### Syntax

```
[ :SENSe]:BANDwidth| BWIDth[:RESolution]:SElect RBW1| RBW2| RBW3| RBW4| RBW5| RBW6
[ :SENSe]:BANDwidth| BWIDth[:RESolution]:SElect?
```

### Description

Sets the resolution bandwidth (RBW).

Queries the resolution bandwidth.

### Parameter

| Name | Type    | Range                              | Default |
|------|---------|------------------------------------|---------|
| —    | Keyword | RBW1  RBW2  RBW3  RBW4  RBW5  RBW6 | RBW2    |

### Remarks

This command is only available for RTSA mode. For details, refer to *RSA3000 User's Guide*.

RBW1, RBW2, RBW3, RBW4, RBW5, and RBW6 corresponds to 98.823 kHz, 197.64 kHz, 395.29 kHz, 790.59 kHz, 1.5811 MHz, and 3.1623 MHz, respectively.

### Return Format

The query returns RBW1, RBW2, RBW3, RBW4, RBW5, or RBW6.

### Example

The following command selects RBW1.

```
:SENSe:BANDwidth:RESolution:SElect RBW1
:SENSe:BWIDth:RESolution:SElect RBW1
```

The following query returns RBW1.

```
:SENSe:BANDwidth:RESolution:SElect?
:SENSe:BWIDth:RESolution:SElect?
```

## **[ :SENSe]:BANDwidth| BWIDth[:RESolution]:SElect:AUTO[:STATe]**

### **Syntax**

[ :SENSe]:BANDwidth| BWIDth[:RESolution]:SElect:AUTO[:STATe] OFF|ON|0|1

[ :SENSe]:BANDwidth| BWIDth[:RESolution]:SElect:AUTO[:STATe]?

### **Description**

Enables or disables the auto setting mode of RBW.

Queries the status of the auto setting mode of RBW.

### **Parameter**

| Name | Type | Range      | Default |
|------|------|------------|---------|
| —    | Bool | OFF ON 0 1 | ON 1    |

### **Remarks**

This command is only available for RTSA mode.

### **Return Format**

The query returns 0 or 1.

### **Example**

The following command enables the auto setting mode of RBW.

:SENSe:BANDwidth:RESolution:SElect:AUTO:STATe ON

or :SENSe:BANDwidth:RESolution:SElect:AUTO:STATe 1

:SENSe:BWIDth:RESolution:SElect:AUTO:STATe ON or :SENSe:BWIDth:RESolution:SElect:AUTO:STATe 1

The following query returns 1.

:SENSe:BANDwidth:RESolution:SElect:AUTO:STATe?

:SENSe:BWIDth:RESolution:SElect:AUTO:STATe?

## **[ :SENSe]:BANDwidth| BWIDth:SHAPE**

### **Syntax**

[ :SENSe]:BANDwidth| BWIDth:SHAPE GAUSSian|FLATtop|BHARris|RECTangular|HANNing|KAISer

[ :SENSe]:BANDwidth| BWIDth:SHAPE?

### **Description**

Sets the filter type.

Queries the filter type.

### **Parameter**

| Name | Type    | Range                                               | Default  |
|------|---------|-----------------------------------------------------|----------|
| —    | Keyword | GAUSSian FLATtop BHARris RECTangular HANNing KAISer | GAUSSian |

### **Remarks**

This command is only available for RTSA mode.

When "Rectangular" is set to be the filter type, the analyzer automatically selects RBW1, and "RBW2 through RBW6" is disabled.

### **Return Format**

The query returns GAUS, FLAT, BHAR, RECT, HANN, or KAIS.

### **Example**

The following command sets the filter type to Gaussian.

```
:SENSe:BANDwidth:SHAPE GAUSSian
:SENSe:BWIDth:SHAPE GAUSSian
```

The following query returns GAUS.

```
:SENSe:BANDwidth:SHAPE?
:SENSe:BWIDth:SHAPE GAUSSian
```

## [ :SENSe]:BANDwidth| BWIDth:VDeo

### Syntax

```
[ :SENSe]:BANDwidth| BWIDth:VDeo <freq>
[ :SENSe]:BANDwidth| BWIDth:VDeo?
```

### Description

Sets the video bandwidth (VBW).  
Queries the video bandwidth.

### Parameter

| Name   | Type     | Range                           | Default |
|--------|----------|---------------------------------|---------|
| <freq> | Discrete | 1 Hz to 10 MHz (at 1-3-10 step) | 3 MHz   |

### Remarks

This command is only available for GPSA mode.

### Return Format

The query returns the video bandwidth in scientific notation. Its unit is Hz.

### Example

The following command sets the video bandwidth to 1,000 Hz.

```
:SENSe:BANDwidth:VDeo 1000
:SENSe:BWIDth:VDeo 1000
```

The following query returns 1.000000000e+03.

```
:SENSe:BANDwidth:VDeo?
:SENSe:BWIDth:VDeo?
```

## [ :SENSe]:BANDwidth| BWIDth:VDeo:AUTO

### Syntax

```
[ :SENSe]:BANDwidth| BWIDth:VDeo:AUTO OFF|ON|0|1
[ :SENSe]:BANDwidth| BWIDth:VDeo:AUTO?
```

### Description

Enables or disables the auto setting mode of VBW.  
Queries the status of the auto setting mode of VBW.

### Parameter

| Name | Type | Range      | Default |
|------|------|------------|---------|
| —    | Bool | OFF ON 0 1 | ON 1    |

### Remarks

This command is only available for GPSA mode.

**Return Format**

The query returns 0 or 1.

**Example**

The following command enables the auto setting mode of VBW.

:SENSe:BANDwidth:VIDeo:AUTO ON or :SENSe:BANDwidth:VIDeo:AUTO 1

:SENSe:BWIDth:VIDeo:AUTO ON or :SENSe:BWIDth:VIDeo:AUTO 1

The following query returns 1.

:SENSe:BANDwidth:VIDeo:AUTO?

:SENSe:BWIDth:VIDeo:AUTO?

**[ :SENSe]:BANDwidth| BWIDth:VIDeo:RATio****Syntax**

[ :SENSe]:BANDwidth| BWIDth:VIDeo:RATio < number>

[ :SENSe]:BANDwidth| BWIDth:VIDeo:RATio?

**Description**

Sets the V/R ratio, that is, the ratio of VBW to RBW.

Queries the V/R ratio.

**Parameter**

| Name      | Type     | Range                                 | Default |
|-----------|----------|---------------------------------------|---------|
| < number> | Discrete | 0.00001 to 3,000,000 (at 1-3-10 step) | 1       |

**Remarks**

This command is valid for the swept measurement in GPSA mode.

**Return Format**

The query returns the V/R ratio in scientific notation.

**Example**

The following command sets the V/R ratio to 0.01.

:SENSe:BANDwidth:VIDeo:RATio 0.01

:SENSe:BWIDth:VIDeo:RATio 0.01

The following query returns 1.000000000e-02.

:SENSe:BANDwidth:VIDeo:RATio?

:SENSe:BWIDth:VIDeo:RATio?

## [ :SENSe]:BANDwidth| BWIDth:VIDeo:RATio:AUTO

### Syntax

[ :SENSe]:BANDwidth| BWIDth:VIDeo:RATio:AUTO OFF| ON| 0| 1  
[ :SENSe]:BANDwidth| BWIDth:VIDeo:RATio:AUTO?

### Description

Enables or disables the auto setting mode of V/R ratio.

Queries the status of the auto setting mode of V/R ratio.

### Parameter

| Name | Type | Range         | Default |
|------|------|---------------|---------|
| —    | Bool | OFF  ON  0  1 | ON  1   |

### Remarks

This command is valid for the swept SA measurement in GPSA mode.

### Return Format

The query returns 0 or 1.

### Example

The following command enables the auto setting mode of V/R ratio.

:SENSe:BANDwidth:VIDeo:AUTO ON or :SENSe:BANDwidth:VIDeo:AUTO 1

:SENSe:BWIDth:VIDeo:AUTO ON or :SENSe:BWIDth:VIDeo:AUTO 1

The following query returns 1.

:SENSe:BANDwidth:VIDeo:AUTO?

:SENSe:BWIDth:VIDeo:AUTO?

## [ :SENSe]:CNRatio:AVERage:COUNT

### Syntax

[ :SENSe]:CNRatio:AVERage:COUNT < integer>  
[ :SENSe]:CNRatio:AVERage:COUNT?

### Description

Sets the average count of the C/N ratio measurement.

Queries the average count of the C/N ratio measurement.

### Parameter

| Name       | Type    | Range      | Default |
|------------|---------|------------|---------|
| < integer> | Integer | 1 to 1,000 | 10      |

### Remarks

This command is only valid when the C/N ratio measurement in GPSA mode is enabled.

### Return Format

The query returns the average count in integer.

### Example

The following command sets the average count to 100.

:SENSe:CNRatio:AVERage:COUNT 100

The following query returns 100.

```
:SENSe:CNRatio:AVERage:COUNT?
```

## [ :SENSe]:CNRatio:AVERage[:STATe]

### Syntax

```
[ :SENSe]:CNRatio:AVERage[:STATe] OFF|ON|0|1
```

```
[ :SENSe]:CNRatio:AVERage[:STATe]?
```

### Description

Enables or disables the average measurement function of the C/N ratio measurement.

Queries the status of the average measurement function of the C/N ratio measurement.

### Parameter

| Name | Type | Range      | Default |
|------|------|------------|---------|
| ---  | Bool | OFF ON 0 1 | OFF 0   |

### Remarks

This command is only valid when the C/N ratio measurement in GPSA mode is enabled.

### Return Format

The query returns 0 or 1.

### Example

The following command enables the average measurement function.

```
:SENSe:CNRatio:AVERage:STATe 1 or :SENSe:CNRatio:AVERage:STATe ON
```

The following query returns 1.

```
:SENSe:CNRatio:AVERage:STATe?
```

## [ :SENSe]:CNRatio:AVERage:TCONtrol

### Syntax

```
[ :SENSe]:CNRatio:AVERage:TCONtrol EXPonential|REPeat
```

```
[ :SENSe]:CNRatio:AVERage:TCONtrol?
```

### Description

Sets the average mode of the C/N ratio measurement.

Queries the average mode of the C/N ratio measurement.

### Parameter

| Name | Type    | Range              | Default     |
|------|---------|--------------------|-------------|
| ---  | Keyword | EXPonential REPeat | EXPonential |

### Remarks

EXPonential: indicates the exponential average.

REPeat: indicates the repeat average.

When "EXPonential" is selected, the result is the exponential average of the measurement results obtained in the past N times (N is specified in [\[:SENSe\]:CNRatio:AVERage:COUNT](#)).

When "REPeat" is selected, the result is the arithmetic average of the measurement results obtained in the past N times (N is specified in [\[:SENSe\]:CNRatio:AVERage:COUNT](#)).

This command is only valid when the C/N ratio measurement in GPSA mode is enabled.

### Return Format

The query returns EXP or REP.

### Example

The following command sets repeat average to the average mode.

```
:SENSe:CNRatio:AVERAge:TCONtrol REPeat
```

The following query returns REP.

```
:SENSe:CNRatio:AVERAge:TCONtrol?
```

## [ :SENSe]:CNRatio:BANDwidth:INTegration

### Syntax

```
[ :SENSe]:CNRatio:BANDwidth:INTegration <freq>
```

```
[ :SENSe]:CNRatio:BANDwidth:INTegration?
```

### Description

Sets the carrier bandwidth.

Queries the carrier bandwidth.

### Parameter

| Name   | Type                    | Range <sup>[1]</sup> | Default |
|--------|-------------------------|----------------------|---------|
| <freq> | Consecutive Real Number | 33 Hz to 1.5 GHz     | 2 MHz   |

**Note<sup>[1]</sup>:** If option RSA3000-BW1 is installed, the available range is from 3 Hz to 1.5 GHz.

### Remarks

This command is only valid when the C/N ratio measurement in GPSA mode is enabled.

The carrier bandwidth is correlated with the noise bandwidth. The range of the carrier bandwidth is from (noise bandwidth/20 to noise bandwidth×20).

### Return Format

The query returns the carrier bandwidth in integer. Its unit is Hz.

### Example

The following command sets the carrier bandwidth to 1 MHz.

```
:SENSe:CNRatio:BANDwidth:INTegration 1000000 or :SENSe:CNRatio:BANDwidth:INTegration 1MHz
```

The following query returns 1000000.

```
:SENSe:CNRatio:BANDwidth:INTegration?
```

## [ :SENSe]:CNRatio:BANDwidth:NOI Se

### Syntax

```
[ :SENSe]:CNRatio:BANDwidth:NOI Se <freq>
```

```
[ :SENSe]:CNRatio:BANDwidth:NOI Se?
```

### Description

Sets the noise bandwidth.

Queries the noise bandwidth.

**Parameter**

| Name   | Type                    | Range <sup>[1]</sup> | Default |
|--------|-------------------------|----------------------|---------|
| <freq> | Consecutive Real Number | 33 Hz to 1.5 GHz     | 2 MHz   |

**Note<sup>[1]</sup>:** If option RSA3000-BW1 is installed, the available range is from 3 Hz to 1.5 GHz.

**Remarks**

This command is only valid when the C/N ratio measurement in GPSA mode is enabled.

**Return Format**

The query returns the noise bandwidth in integer. Its unit is Hz.

**Example**

The following command sets the noise bandwidth to 1 MHz.

```
:SENSe:CNRatio:BANDwidth:NOISe 1000000
```

The following query returns 1000000.

```
:SENSe:CNRatio:BANDwidth:NOISe?
```

**[ :SENSe]:CNRatio:OFFSet****Syntax**

```
[ :SENSe]:CNRatio:OFFSet <freq>
```

```
[ :SENSe]:CNRatio:OFFSet?
```

**Description**

Sets the center frequency difference (offset frequency) between the carrier waveform and the noise. Queries the offset frequency.

**Parameter**

| Name   | Type                    | Range <sup>[1]</sup> | Default |
|--------|-------------------------|----------------------|---------|
| <freq> | Consecutive Real Number | 33 Hz to 1.5 GHz     | 2 MHz   |

**Note<sup>[1]</sup>:** If option RSA3000-BW1 is installed, the available range is from 3 Hz to 1.5 GHz.

**Remarks**

This command is only valid when the C/N ratio measurement in GPSA mode is enabled.

**Return Format**

The query returns the offset frequency in integer. Its unit is Hz.

**Example**

The following command sets the offset frequency to 1 MHz.

```
:SENSe:CNRatio:OFFSet 1000000
```

The following query returns 1000000.

```
:SENSe:CNRatio:OFFSet?
```

## **:INPut:IMPedance** **[ :SENSe]:CORRection:IMPedance[:INPut][:MAGNitude]**

### **Syntax**

```
:INPut:IMPedance 50|75
:INPut:IMPedance?
[:SENSe]:CORRection:IMPedance[:INPut][:MAGNitude] 50|75
[:SENSe]:CORRection:IMPedance[:INPut][:MAGNitude]?
```

### **Description**

Sets the input impedance for voltage-to-power conversion. The unit is  $\Omega$ .  
Queries the input impedance for voltage-to-power conversion.

### **Parameter**

| Name | Type     | Range | Default |
|------|----------|-------|---------|
| —    | Discrete | 50 75 | 50      |

### **Remarks**

If the output impedance of the system under measurement is 75  $\Omega$ , you should use a 75  $\Omega$  to 50  $\Omega$  adapter (option) supplied by **RIGOL** to connect the analyzer with the system under test, and then set the input impedance to 75  $\Omega$ .

### **Return Format**

The query returns 50 or 75.

### **Example**

The following command sets the input impedance to 75  $\Omega$ .  
:INPut:IMPedance 75 or :SENSe:CORRection:IMPedance:INPut:MAGNitude 75

The following query returns 75.  
:INPut:IMPedance? or :SENSe:CORRection:IMPedance:INPut:MAGNitude?

## **[ :SENSe]:CORRection:SA[:RF]:GAI N**

### **Syntax**

```
[ :SENSe]:CORRection:SA[:RF]:GAI N <rel_ampl>
[:SENSe]:CORRection:SA[:RF]:GAI N?
```

### **Description**

Sets the external gain.  
Queries the external gain.

### **Parameter**

| Name       | Type                    | Range             | Default |
|------------|-------------------------|-------------------|---------|
| <rel_ampl> | Consecutive Real Number | -120 dB to 120 dB | 0 dB    |

### **Return Format**

The query returns the external gain value in scientific notation. The unit is dB.

### **Example**

The following command set the external gain value to 20 dB.  
:SENSe:CORRection:SA:RF:GAIN 20

The following query returns 2.000000000e+01.  
:SENSe:CORRection:SA:RF:GAIN?

## [ :SENSe]:DEMod

### Syntax

[ :SENSe]:DEMod AM|FM|OFF

[ :SENSe]:DEMod?

### Description

Sets the demodulation type or disables the demodulation.

Queries the demodulation type.

### Parameter

| Name | Type    | Range     | Default |
|------|---------|-----------|---------|
| ---- | Keyword | AM FM OFF | OFF     |

### Remarks

AM: indicates the amplitude modulation.

FM: indicates the frequency modulation.

OFF: indicates off.

When AM or FM is selected, the spectrum analyzer enables the demodulation function automatically.

When OFF is selected, the spectrum analyzer disables the demodulation function.

This command is only available for GPSA mode.

### Return Format

The query returns AM, FM, or OFF.

### Example

The following command sets the demodulation type to AM.

:SENSe:DEMod AM

The following query returns AM.

:SENSe:DEMod?

## [ :SENSe]:DEMod:GAIN:AUTO

### Syntax

[ :SENSe]:DEMod:GAIN:AUTO OFF|ON|0|1

[ :SENSe]:DEMod:GAIN:AUTO?

### Description

Enables or disables the auto setting mode of the signal gain.

Queries the status of the signal gain.

### Parameter

| Name | Type | Range      | Default |
|------|------|------------|---------|
| ---- | Bool | OFF ON 0 1 | ON 1    |

### Remarks

This command is only valid when the demodulation function (i.g. AM or FM is selected to be the demodulation type) in GPSA mode is enabled.

### Return Format

The query returns 0 or 1.

**Example**

The following command enables the auto setting mode of signal gain.

```
:SENSe:DEMod:GAIN:AUTO ON or :SENSe:DEMod:GAIN:AUTO 1
```

The following query returns 1.

```
:SENSe:DEMod:GAIN:AUTO?
```

**[ :SENSe ] :DEMod:GAIN:INCRement****Syntax**

```
[ :SENSe ] :DEMod:GAIN:INCRement <integer>
```

```
[ :SENSe ] :DEMod:GAIN:INCRement?
```

**Description**

Sets the signal gain.

Queries the signal gain.

**Parameter**

| Name      | Type    | Range  | Default |
|-----------|---------|--------|---------|
| <integer> | Integer | 1 to 7 | 7       |

**Remarks**

This command is only valid when the demodulation function (i.g. AM or FM is selected to be the demodulation type) in GPSA mode is enabled.

**Return Format**

The query returns the signal gain in integer.

**Example**

The following command sets the signal gain to 4.

```
:SENSe:DEMod:GAIN:INCRement 4
```

The following query returns 4.

```
:SENSe:DEMod:GAIN:INCRement?
```

**[ :SENSe ] :DEMod:STATe****Syntax**

```
[ :SENSe ] :DEMod:STATe OFF|ON|0|1
```

```
[ :SENSe ] :DEMod:STATe?
```

**Description**

Enables or disables the demodulation function.

Queries the status of the demodulation function.

**Parameter**

| Name | Type | Range      | Default |
|------|------|------------|---------|
| —    | Bool | OFF ON 0 1 | OFF 0   |

**Remarks**

When the demodulation function is enabled, the analyzer selects AM as the demodulation type by default. This command is only available for GPSA mode.

**Return Format**

The query returns 0 or 1.

**Example**

The following command enables the demodulation function.

:SENSe:DEMod:STATe 1 or :SENSe:DEMod:STATe ON

The following query returns 1.

:SENSe:DEMod:STATe?

**[ :SENSe ] :DETEctor:TRACe:PVTTime****Syntax**

[ :SENSe ] :DETEctor:TRACe:PVTTime AVERAge|NEGAtive|POSitive|SAMPle

[ :SENSe ] :DETEctor:TRACe:PVTTime?

**Description**

Sets the detector type of the trace in the PVT view.

Queries the detector type of the trace in the PVT view.

**Parameter**

| Name | Type    | Range                            | Default  |
|------|---------|----------------------------------|----------|
| —    | Keyword | AVERAge NEGAtive POSitive SAMPle | POSitive |

**Remarks**

AVERAge: indicates the voltage average.

NEGAtive: indicates the negative peak.

POSitive: indicates the positive peak.

SAMPle: indicates the sample detector.

This command is only available for RTSA mode.

**Return Format**

The query returns AVER, NEG, POS, or SAMP.

**Example**

The following command sets the trace detector type to Positive.

:SENSe:DETEctor:TRACe:PVTTime POSitive

The following query returns POS.

:SENSe:DETEctor:TRACe:PVTTime?

## **[ :SENSe]:DETEctor[:FUNCTION] [:SENSe]:DETEctor:TRACe< n>**

### **Syntax**

[ :SENSe]:DETEctor[:FUNCTION] AVERage(VAVerage)|NEGative|NORMal|POSitive|SAMPLE|QPEak|RAVerage (RMS)

[ :SENSe]:DETEctor[:FUNCTION] ?

[ :SENSe]:DETEctor:TRACe< n> AVERage(VAVerage)|NEGative|NORMal|POSitive|SAMPLE|QPEak|RAVerage (RMS)

[ :SENSe]:DETEctor:TRACe< n> ?

### **Description**

Sets the detector type for the specified trace.

Queries the detector type for the specified trace.

### **Parameter**

| Name | Type     | Range                                                                           | Default  |
|------|----------|---------------------------------------------------------------------------------|----------|
| < n> | Discrete | 1 2 3 4 5 6                                                                     | 1        |
| —    | Keyword  | AVERage (VAVerage)<br> NEGative NORMal POSitive <br>SAMPLE QPEak RAVerage (RMS) | POSitive |

### **Remarks**

AVERage|VAVerage: indicates the voltage average.

NEGative: indicates the negative peak.

NORMal: indicates the normal detector.

POSitive: indicates the positive peak.

SAMPLE: indicates the sample detector.

QPEak: indicates the quasi-peak. It is only available for the instrument installed with the RSA3000-EMC option.

RAVerage|RMS: indicates the RMS average.

In RTSA mode, the parameters in this command only include AVERage, NEGative, POSitive, and SAMPLE.

### **Return Format**

The query returns AVER, NEG, NORM, POS, SAMP, QPE, or RAV.

### **Example**

The following command sets the detector type of Trace 1 to Positive.

```
:SENSe:DETEctor:FUNCTION POSitive
```

```
:SENSe:DETEctor:TRACe1 POSitive
```

The following query returns POS.

```
:SENSe:DETEctor:FUNCTION?
```

```
:SENSe:DETEctor:TRACe1?
```

## [ :SENSe]:DETEctor:TRACe< n> :AUTO

### Syntax

```
[ :SENSe]:DETEctor:TRACe< n> :AUTO OFF|ON|0|1
```

```
[ :SENSe]:DETEctor:TRACe< n> :AUTO?
```

### Description

Enables or disables the Detector Auto function for the specified trace.

Queries the status of the Detector Auto function for the specified trace.

### Parameter

| Name | Type     | Range       | Default |
|------|----------|-------------|---------|
| < n> | Discrete | 1 2 3 4 5 6 | 1       |
| —    | Bool     | OFF ON 0 1  | ON 1    |

### Return Format

The query returns 0 or 1.

### Example

The following command enables the Detector Auto function for Trace 1.

```
:SENSe:DETEctor:TRACe1:AUTO ON or :SENSe:DETEctor:TRACe1:AUTO 1
```

The following query returns 1.

```
:SENSe:DETEctor:TRACe1:AUTO?
```

## [ :SENSe]:EBWidth:AVERage:COUNT

### Syntax

```
[ :SENSe]:EBWidth:AVERage:COUNT < integer>
```

```
[ :SENSe]:EBWidth:AVERage:COUNT?
```

### Description

Sets the average count of the emission bandwidth measurement.

Queries the average count of the emission bandwidth measurement.

### Parameter

| Name       | Type    | Range      | Default |
|------------|---------|------------|---------|
| < integer> | Integer | 1 to 1,000 | 10      |

### Remarks

This command is only valid when the emission bandwidth measurement in GPSA mode is enabled.

### Return Format

The query returns the average count in integer.

### Example

The following command sets the average count to 100.

```
:SENSe:EBWidth:AVERage:COUNT 100
```

The following query returns 100.

```
:SENSe:EBWidth:AVERage:COUNT?
```

## [ :SENSe]:EBWidth:AVERage[:STATe]

### Syntax

[ :SENSe]:EBWidth:AVERage[:STATe] OFF|ON|0|1  
[ :SENSe]:EBWidth:AVERage[:STATe]?

### Description

Enables or disables the average measurement function of the emission bandwidth measurement.  
Queries the status of the average measurement function of the emission bandwidth measurement.

### Parameter

| Name | Type | Range      | Default |
|------|------|------------|---------|
| —    | Bool | OFF ON 0 1 | OFF 0   |

### Remarks

This command is only valid when the emission bandwidth measurement in GPSA mode is enabled.

### Return Format

The query returns 0 or 1.

### Example

The following command enables the average measurement function.

:SENSe:EBWidth:AVERage:STATe ON or :SENSe:EBWidth:AVERage:STATe 1

The following query returns 1.

:SENSe:EBWidth:AVERage:STATe?

## [ :SENSe]:EBWidth:AVERage:TCONtrol

### Syntax

[ :SENSe]:EBWidth:AVERage:TCONtrol EXPonential|REPeat  
[ :SENSe]:EBWidth:AVERage:TCONtrol?

### Description

Sets the average mode of the emission bandwidth measurement.  
Queries the average mode of the emission bandwidth measurement.

### Parameter

| Name | Type    | Range              | Default     |
|------|---------|--------------------|-------------|
| —    | Keyword | EXPonential REPeat | EXPonential |

### Remarks

EXPonential: indicates the exponential average.

REPeat: indicates the repeat average.

When "EXPonential" is selected, the result is the exponential average of the measurement results obtained in the past N times (N is specified in [\[:SENSe\]:EBWidth:AVERage:COUNt](#)).

When "REPeat" is selected, the result is the arithmetic average of the measurement results obtained in the past N times (N is specified in [\[:SENSe\]:EBWidth:AVERage:COUNt](#)).

This command is only valid when the emission bandwidth measurement in GPSA mode is enabled.

### Return Format

The query returns EXP or REP.

**Example**

The following command sets the average mode to Repeat.

```
:SENSe:EBWidth:AVERage:TCONtrol REPeat
```

The following query returns REP.

```
:SENSe:EBWidth:AVERage:TCONtrol?
```

**[ :SENSe]:EBWidth:FREQuency:SPAN****Syntax**

```
[ :SENSe]:EBWidth:FREQuency:SPAN <freq>
```

```
[ :SENSe]:EBWidth:FREQuency:SPAN?
```

**Description**

Sets the span of the emission bandwidth measurement.

Queries the span of the emission bandwidth measurement.

**Parameter**

| Name   | Type                    | Range <sup>[1]</sup> | Default |
|--------|-------------------------|----------------------|---------|
| <freq> | Consecutive Real Number | 100 Hz to 4.5 GHz    | 2 MHz   |

**Note<sup>[1]</sup>:** If option RSA3000-BW1 is installed, the available range is from 10 Hz to 4.5 GHz.

**Remarks**

This command is only valid when the emission bandwidth measurement in GPSA mode is enabled.

This setting changes the span of the analyzer.

**Return Format**

The query returns the span in scientific notation. Its unit is Hz.

**Example**

The following command sets the span to 10 MHz.

```
:SENSe:EBWidth:FREQuency:SPAN 10000000
```

The following query returns 1.0000000000e+07.

```
:SENSe:EBWidth:FREQuency:SPAN?
```

**[ :SENSe]:EBWidth:MAXHold:STATe****Syntax**

```
[ :SENSe]:EBWidth:MAXHold:STATe OFF|ON|0|1
```

```
[ :SENSe]:EBWidth:MAXHold:STATe?
```

**Description**

Enables or disables the EBW Max Hold.

Queries the status of the EBW Max Hold.

**Parameter**

| Name | Type | Range      | Default |
|------|------|------------|---------|
| —    | Bool | OFF ON 0 1 | OFF 0   |

**Remarks**

This command is only valid when the emission bandwidth measurement in GPSA mode is enabled.

When Max Hold is enabled, each measurement result is compared with the previous result, and then display whichever is the maximum.

When Max Hold is disabled, the current measurement result is displayed.

Max Hold and average measurement are mutually exclusive. When Max Hold is enabled, the average measurement will be automatically disabled.

### Return Format

The query returns 0 or 1.

### Example

The following command enables the Max Hold.

```
:SENSe:EBWidth:MAXHold:STATe ON or :SENSe:EBWidth:MAXHold:STATe 1
```

The following query returns 1.

```
:SENSe:EBWidth:MAXHold:STATe?
```

## [ :SENSe]:EBWidth:XDB

### Syntax

```
[ :SENSe]:EBWidth:XDB <real_amp>
```

```
[ :SENSe]:EBWidth:XDB?
```

### Description

Sets the value of X dB for the EBW measurement.

Queries the value of X dB for the EBW measurement.

### Parameter

| Name       | Type                    | Range              | Default |
|------------|-------------------------|--------------------|---------|
| <real_amp> | Consecutive Real Number | -100 dB to -0.1 dB | -10 dB  |

### Remarks

This command is only valid when the emission bandwidth measurement in GPSA mode is enabled.

### Return Format

The query returns the X dB value in scientific notation.

### Example

The following command sets the X dB value to -20 dB.

```
:SENSe:EBWidth:XDB -20
```

The following query returns -2.000000000e+01.

```
:SENSe:EBWidth:XDB?
```

## [ :SENSe]:FREQuency:CENTer

### Syntax

```
[ :SENSe]:FREQuency:CENTer <freq>
```

```
[ :SENSe]:FREQuency:CENTer?
```

### Description

Sets the center frequency.

Queries the center frequency.

### Parameter

| Name   | Type                    | Range     |                                                                                                   | Default |
|--------|-------------------------|-----------|---------------------------------------------------------------------------------------------------|---------|
| <freq> | Consecutive Real Number | GPSA mode | 0 Hz to Fmax <sup>[1]</sup> (zero span)<br>50 Hz to (Fmax - 50 Hz) (non-zero span) <sup>[2]</sup> | Fmax/2  |
|        |                         | RTSA mode | 2.5 kHz to (Fmax - 2.5 kHz)                                                                       |         |

**Note<sup>[1]</sup>:** Fmax (maximum measurement frequency) is determined by the model of the spectrum analyzer. The maximum frequency of RSA3030/RSA3030-TG is 3 GHz, and the maximum frequency of RSA3045/RSA3045-TG is 4.5 GHz.

**Note<sup>[2]</sup>:** In non-zero span, if the option RSA3030-BW1 is installed, its range is from 5 Hz to (Fmax – 5 Hz).

### Return Format

The query returns the center frequency in scientific notation. The unit is Hz.

### Example

The following command sets the center frequency to 1 MHz.

```
:SENSe:FREQuency:CENTer 1000000
```

The following query returns 1.000000000e+ 06.

```
:SENSe:FREQuency:CENTer?
```

## [ :SENSe]:FREQuency:CENTer:STEP:AUTO

### Syntax

```
[ :SENSe]:FREQuency:CENTer:STEP:AUTO OFF|ON|0|1
```

```
[ :SENSe]:FREQuency:CENTer:STEP:AUTO?
```

### Description

Enables or disables the auto setting mode of the CF step.

Queries the status of the auto setting mode of the CF step.

### Parameter

| Name | Type | Range      | Default |
|------|------|------------|---------|
| —    | Bool | OFF ON 0 1 | ON 1    |

### Return Format

The query returns 0 or 1.

### Example

The following command enables the auto setting mode of the CF step.

```
:SENSe:FREQuency:CENTer:STEP:AUTO ON or :SENSe:FREQuency:CENTer:STEP:AUTO 1
```

The following query returns 1.

```
:SENSe:FREQuency:CENTer:STEP:AUTO?
```

## [ :SENSe]:FREQuency:CENTer:STEP[:I NCRement]

### Syntax

[ :SENSe]:FREQuency:CENTer:STEP[:I NCRement] <freq>  
[ :SENSe]:FREQuency:CENTer:STEP[:I NCRement] ?

### Description

Sets the CF step.  
Queries the CF step.

### Parameter

| Name   | Type                    | Range         | Default |
|--------|-------------------------|---------------|---------|
| <freq> | Consecutive Real Number | -Fmax to Fmax | Fmax/10 |

### Return Format

The query returns the center frequency step in scientific notation. The unit is Hz.

### Example

The following command sets the CF step to 100 kHz.  
:SENSe:FREQuency:CENTer:STEP:INCRement 100000

The following query returns 1.000000000e+05.  
:SENSe:FREQuency:CENTer:STEP:INCRement?

## [ :SENSe]:FREQuency:OFFSet

### Syntax

[ :SENSe]:FREQuency:OFFSet <freq>  
[ :SENSe]:FREQuency:OFFSet ?

### Description

Sets the frequency offset.  
Queries the frequency offset.

### Parameter

| Name   | Type                    | Range               | Default |
|--------|-------------------------|---------------------|---------|
| <freq> | Consecutive Real Number | -500 GHz to 500 GHz | 0 Hz    |

### Remarks

The change of this parameter only changes the display values of the center frequency, start frequency, and stop frequency; but does not affect any hardware settings of the spectrum analyzer.

### Return Format

The query returns the frequency offset in scientific notation. The unit is Hz.

### Example

The following command sets the frequency offset to 1 MHz.  
:SENSe:FREQuency:OFFSet 1000000

The following query returns 1.000000000e+06.  
:SENSe:FREQuency:OFFSet ?

## [ :SENSe]:FREQuency:SPAN

### Syntax

```
[ :SENSe]:FREQuency:SPAN <freq>
```

```
[ :SENSe]:FREQuency:SPAN?
```

### Description

Sets the span.

Queries the span.

### Parameter

| Name   | Type        | Range                                           | Default                           |
|--------|-------------|-------------------------------------------------|-----------------------------------|
| <freq> | Consecutive | GPSA mode <sup>[1]</sup> : 0 Hz, 100 Hz to Fmax | GPSA mode: Fmaxz                  |
|        | Real Number | RTSA mode <sup>[2]</sup> : 5 kHz to 10 MHz      | RTSA mode <sup>[3]</sup> : 10 MHz |

**Note<sup>[1]</sup>:** When the span of the analyzer is set to 0 Hz, the spectrum analyzer enters the zero span mode. If option RSA3000-BW1 is installed, the available range is from 10 Hz to Fmax in non-zero span.

**Note<sup>[2]</sup>:** If the option RSA3000-B25 is installed, its range is from 5 kHz to 25 MHz. If the option RSA3000-B40 is installed, its range is from 5 kHz to 40 MHz.

**Note<sup>[3]</sup>:** If the option RSA3000-B25 is installed, the default value is 25 MHz. If the option RSA3000-B40 is installed, the default value is 40 MHz.

### Remarks

The span can be set to 0 only in GPSA mode. After the span is set to 0, the instrument enters the zero span mode; the X axis changes from frequency to time, and the instrument only displays signals whose frequency equals to the center frequency.

### Return Format

The query returns the span in scientific notation. Its unit is Hz.

### Example

The following command sets the span to 20 MHz.

```
:SENSe:FREQuency:SPAN 20000000
```

The following query returns 2.000000000e+ 07.

```
:SENSe:FREQuency:SPAN?
```

## [ :SENSe]:FREQuency:SPAN:BANDwidth[:RESolution]:RATio

### Syntax

```
[ :SENSe]:FREQuency:SPAN:BANDwidth[:RESolution]:RATio <integer>
```

```
[ :SENSe]:FREQuency:SPAN:BANDwidth[:RESolution]:RATio?
```

### Description

Sets the ratio of span to RBW.

Queries the ratio of span to RBW.

### Parameter

| Name      | Type     | Range       | Default |
|-----------|----------|-------------|---------|
| <integer> | Discrete | 2 to 10,000 | 106     |

### Remarks

This command is valid for the swept SA measurement in GPSA mode.

### Return Format

The query returns the span/RBW ratio in integer.

**Example**

The following command sets the span/RBW ratio to 100.

```
:SENSe:FREQuency:SPAN:BANDwidth:RESolution:RATio 100
```

The following query returns 100.

```
:SENSe:FREQuency:SPAN:BANDwidth:RESolution:RATio?
```

**[ :SENSe]:FREQuency:SPAN:BANDwidth[:RESolution]:RATio:AUTO****Syntax**

```
[ :SENSe]:FREQuency:SPAN:BANDwidth[:RESolution]:RATio:AUTO OFF|ON|0|1
```

```
[ :SENSe]:FREQuency:SPAN:BANDwidth[:RESolution]:RATio:AUTO?
```

**Description**

Enables or disables the auto setting mode of span/RBW ratio.

Queries the status of the auto setting mode of span/RBW ratio.

**Parameter**

| Name | Type | Range      | Default |
|------|------|------------|---------|
| —    | Bool | OFF ON 0 1 | ON 1    |

**Remarks**

This command is valid for the swept SA measurement in GPSA mode.

**Return Format**

The query returns 0 or 1.

**Example**

The following command enables the auto setting mode of span/bandwidth ratio.

```
:SENSe:FREQuency:SPAN:BANDwidth:RESolution:RATio:AUTO ON
```

```
or :SENSe:FREQuency:SPAN:BANDwidth:RESolution:RATio:AUTO 1
```

The following query returns 1.

```
:SENSe:FREQuency:SPAN:BANDwidth:RESolution:RATio:AUTO?
```

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

```
[ :SENSe]:FREQuency:SPAN:FULL
```

**Description**

Sets the maximum span (full span).

**[ :SENSe]:FREQuency:SPAN:PREVious****Syntax**

```
[ :SENSe]:FREQuency:SPAN:PREVious
```

**Description**

Sets the span to the last set value.

## [ :SENSe]:FREQuency:SPAN:ZERO

### Syntax

```
[ :SENSe]:FREQuency:SPAN:ZERO
```

### Description

Sets the span to 0 (zero span).

### Remarks

This command is only available for GPSA mode.

## [ :SENSe]:FREQuency:STArT

### Syntax

```
[ :SENSe]:FREQuency:STArT <freq>
```

```
[ :SENSe]:FREQuency:STArT?
```

### Description

Sets the start frequency.

Queries the start frequency.

### Parameter

| Name   | Type                    | Range <sup>[1]</sup> |                                                                                       | Default                    |
|--------|-------------------------|----------------------|---------------------------------------------------------------------------------------|----------------------------|
| <freq> | Consecutive Real Number | GPSA mode            | 0 Hz to Fmax (zero span)<br>0 Hz to (Fmax - 100 Hz)<br>(non-zero span) <sup>[1]</sup> | center<br>frequency-span/2 |
|        |                         | RTSA mode            | 0 Hz to (Fmax - 5 kHz)                                                                |                            |

**Note**<sup>[1]</sup>: In non-zero span, if the option RSA3030-BW1 is installed, the available range is from 0 Hz to (Fmax-10 Hz).

### Return Format

The query returns the start frequency in scientific notation. The unit is Hz.

### Example

The following command sets the start frequency to 100 MHz.

```
:SENSe:FREQuency:STArT 100000000
```

The following query returns 1.000000000e+08.

```
:SENSe:FREQuency:STArT?
```

## [ :SENSe]:FREQuency:STOP

### Syntax

```
[ :SENSe]:FREQuency:STOP <freq>
[ :SENSe]:FREQuency:STOP?
```

### Description

Sets the stop frequency.  
Queries the stop frequency.

### Parameter

| Name   | Type                    | Range <sup>[1]</sup> |                                                                           | Default                     |
|--------|-------------------------|----------------------|---------------------------------------------------------------------------|-----------------------------|
| <freq> | Consecutive Real Number | GPSA mode            | 0 Hz to Fmax (zero span)<br>100 Hz to Fmax (non-zero span) <sup>[1]</sup> | center<br>frequency+ span/2 |
|        |                         | RTSA mode            | 5 kHz to Fmax                                                             |                             |

**Note<sup>[1]</sup>:** In non-zero span, if the option RSA3030-BW1 is installed, the available range is 10 Hz to Fmax.

### Return Format

The query returns the stop frequency in scientific notation. The unit is Hz.

### Example

The following command sets the stop frequency to 10 MHz.

```
:SENSe:FREQuency:STOP 10000000
```

The following query returns 1.000000000e+07.

```
:SENSe:FREQuency:STOP?
```

## [ :SENSe]:FREQuency:TUNE:IMMediate

### Syntax

```
[ :SENSe]:FREQuency:TUNE:IMMediate
```

### Description

Automatically searches for the signal within the full frequency range, and adjusts the frequency and amplitude for optimal display effect of the signal.

### Remarks

This command is invalid when the advance measurement function (AMK) is enabled in GPSA mode or when in RTSA mode.

## [ :SENSe]:HDISt:AVERage:COUNT

### Syntax

```
[ :SENSe]:HDISt:AVERage:COUNT <integer>
[ :SENSe]:HDISt:AVERage:COUNT?
```

### Description

Sets the average count of the harmonic distortion measurement.  
Queries the average count of the harmonic distortion measurement.

**Parameter**

| Name      | Type    | Range      | Default |
|-----------|---------|------------|---------|
| <integer> | Integer | 1 to 1,000 | 10      |

**Remarks**

This command is only valid when the harmonic distortion measurement in GPSA mode is enabled.

**Return Format**

The query returns the average count in integer.

**Example**

The following command sets the average count to 100.

```
:SENSe:HDISt:AVERage:COUNT 100
```

The following query returns 100.

```
:SENSe:HDISt:AVERage:COUNT?
```

**[ :SENSe ] :HDISt:AVERage[ :STATe ]****Syntax**

```
[ :SENSe ] :HDISt:AVERage[ :STATe ] OFF|ON|0|1
```

```
[ :SENSe ] :HDISt:AVERage[ :STATe ] ?
```

**Description**

Enables or disables the average measurement function of the harmonic distortion measurement.

Queries the status of the average measurement function of the harmonic distortion measurement.

**Parameter**

| Name | Type | Range      | Default |
|------|------|------------|---------|
| —    | Bool | OFF ON 0 1 | OFF 0   |

**Remarks**

This command is only valid when the harmonic distortion measurement in GPSA mode is enabled.

**Return Format**

The query returns 0 or 1.

**Example**

The following command enables the average measurement function.

```
:SENSe:HDISt:AVERage:STATe ON or :SENSe:HDISt:AVERage:STATe 1
```

The following query returns 1.

```
:SENSe:HDISt:AVERage:STATe?
```

## [ :SENSe]:HDISt:AVERage:TCONtrol

### Syntax

```
[ :SENSe]:HDISt:AVERage:TCONtrol EXPonential| REPeat
[ :SENSe]:HDISt:AVERage:TCONtrol?
```

### Description

Sets the average mode of the harmonic distortion measurement.  
Queries the average mode of the harmonic distortion measurement.

### Parameter

| Name | Type    | Range               | Default     |
|------|---------|---------------------|-------------|
| —    | Keyword | EXPonential  REPeat | EXPonential |

### Remarks

EXPonential: indicates the exponential average.

REPeat: indicates the repeat average.

When "EXPonential" is selected, the result is the exponential average of the measurement results obtained in the past N times (N is specified in [\[:SENSe\]:HDISt:AVERage:COUNt](#) ).

When "REPeat" is selected, the result is the arithmetic average of the measurement results obtained in the past N times (N is specified in [\[:SENSe\]:HDISt:AVERage:COUNt](#) ).

This command is only valid when the harmonic distortion measurement in GPSA mode is enabled.

### Return Format

The query returns EXP or REP.

### Example

The following command sets the average mode to Repeat.

```
:SENSe:HDISt:AVERage:TCONtrol REPeat
```

The following query returns REP.

```
:SENSe:HDISt:AVERage:TCONtrol?
```

## [ :SENSe]:HDISt:NUMBers

### Syntax

```
[ :SENSe]:HDISt:NUMBers <integer>
[ :SENSe]:HDISt:NUMBers?
```

### Description

Sets the number of harmonics to be measured.  
Queries the number of harmonics to be measured.

### Parameter

| Name      | Type    | Range   | Default |
|-----------|---------|---------|---------|
| <integer> | Integer | 2 to 10 | 10      |

### Remarks

This command is only valid when the harmonic distortion measurement in GPSA mode is enabled.

### Return Format

The query returns the number of harmonics in integer.

**Example**

The following command sets the number of harmonics to 5.

```
:SENSe:HDISt:NUMBers 5
```

The following query returns 5.

```
:SENSe:HDISt:NUMBers?
```

**[ :SENSe]:HDISt:TIME****Syntax**

```
[ :SENSe]:HDISt:TIME <time>
```

```
[ :SENSe]:HDISt:TIME?
```

**Description**

Sets the sweep time of the harmonic distortion measurement.

Queries the sweep time of the harmonic distortion measurement.

**Parameter**

| Name   | Type                    | Range             | Default |
|--------|-------------------------|-------------------|---------|
| <time> | Consecutive Real Number | 1 $\mu$ s to 6 ks | 1 ms    |

**Remarks**

This command is only valid when the harmonic distortion measurement in GPSA mode is enabled.

This setting changes the sweep time of the analyzer.

**Return Format**

The query returns the sweep time in scientific notation. The unit is s.

**Example**

The following command sets the sweep time to 100 ms.

```
:SENSe:HDISt:TIME 0.1
```

The following query returns 1.000000000e-01.

```
:SENSe:HDISt:TIME?
```

**[ :SENSe]:MCHPower:AVERage:COUNT****Syntax**

```
[ :SENSe]:MCHPower:AVERage:COUNT <integer>
```

```
[ :SENSe]:MCHPower:AVERage:COUNT?
```

**Description**

Sets the average count of the multi-channel power measurement.

Queries the average count of the multi-channel power measurement.

**Parameter**

| Name      | Type    | Range      | Default |
|-----------|---------|------------|---------|
| <integer> | Integer | 1 to 1,000 | 10      |

**Remarks**

This command is only valid when the multi-channel power measurement in GPSA mode is enabled.

**Return Format**

The query returns the average count in integer.

**Example**

The following command sets the average count to 100.

```
:SENSe:MCHPower:AVERAge:COUNt 100
```

The following query returns 100.

```
:SENSe:MCHPower:AVERAge:COUNt?
```

**[ :SENSe]:MCHPower:AVERAge[:STATe]****Syntax**

```
[ :SENSe]:MCHPower:AVERAge[:STATe] OFF|ON|0|1
```

```
[ :SENSe]:MCHPower:AVERAge[:STATe]?
```

**Description**

Enables or disables the average measurement function of the multi-channel power measurement.

Queries the status of the average measurement function of the multi-channel power measurement.

**Parameter**

| Name | Type | Range      | Default |
|------|------|------------|---------|
| —    | Bool | OFF ON 0 1 | ON 1    |

**Remarks**

This command is only valid when the multi-channel power measurement in GPSA mode is enabled.

**Return Format**

The query returns 0 or 1.

**Example**

The following command enables the average measurement function.

```
:SENSe:MCHPower:AVERAge:STATe ON or :SENSe:MCHPower:AVERAge:STATe 1
```

The following query returns 1.

```
:SENSe:MCHPower:AVERAge:STATe?
```

**[ :SENSe]:MCHPower:AVERAge:TCONtrol****Syntax**

```
[ :SENSe]:MCHPower:AVERAge:TCONtrol EXPonential|REPeat
```

```
[ :SENSe]:MCHPower:AVERAge:TCONtrol?
```

**Description**

Selects the average mode of the multi-channel power measurement.

Queries the average mode of the multi-channel power measurement.

**Parameter**

| Name | Type    | Range              | Default     |
|------|---------|--------------------|-------------|
| —    | Keyword | EXPonential REPeat | EXPonential |

**Remarks**

EXPonential: indicates the exponential average.

REPeat: indicates the repeat average.

When "EXPonential" is selected, the result is the exponential average of the measurement results obtained in the past N times (N is specified in [\[:SENSe\]:MCHPower:AVERage:COUNT](#)).

When "REPeat" is selected, the result is the arithmetic average of the measurement results obtained in the past N times (N is specified in [\[:SENSe\]:MCHPower:AVERage:COUNT](#)).

This command is only valid when the multi-channel power measurement in GPSA mode is enabled.

**Return Format**

The query returns EXP or REP.

**Example**

The following command sets the average mode to Repeat.

```
:SENSe:MCHPower:AVERage:TCONtrol REPeat
```

The following query returns REP.

```
:SENSe:MCHPower:AVERage:TCONtrol?
```

**[\[:SENSe\]:OBWidth:AVERage:COUNT](#)****Syntax**

```
[ :SENSe]:OBWidth:AVERage:COUNT <integer>
```

```
[ :SENSe]:OBWidth:AVERage:COUNT?
```

**Description**

Sets the average count of the occupied bandwidth measurement.

Queries the average count of the occupied bandwidth measurement.

**Parameter**

| Name      | Type    | Range      | Default |
|-----------|---------|------------|---------|
| <integer> | Integer | 1 to 1,000 | 10      |

**Remarks**

This command is only valid when the occupied bandwidth measurement in GPSA mode is enabled.

**Return Format**

The query returns the average count in integer.

**Example**

The following command sets the average count to 100.

```
:SENSe:OBWidth:AVERage:COUNT 100
```

The following query returns 100.

```
:SENSe:OBWidth:AVERage:COUNT?
```

**[\[:SENSe\]:OBWidth:AVERage\[:STATe\]](#)****Syntax**

```
[ :SENSe]:OBWidth:AVERage[:STATe] OFF|ON|0|1
```

```
[ :SENSe]:OBWidth:AVERage[:STATe]?
```

**Description**

Enables or disables the average measurement function of the occupied bandwidth measurement.  
 Queries the status of the average measurement function of the occupied bandwidth measurement.

**Parameter**

| Name | Type | Range      | Default |
|------|------|------------|---------|
| —    | Bool | OFF ON 0 1 | ON 1    |

**Remarks**

This command is only valid when the occupied bandwidth measurement in GPSA mode is enabled.

**Return Format**

The query returns 0 or 1.

**Example**

The following command enables the average measurement function.

:SENSe:OBWidth:AVERage:STATe ON or :SENSe:ACPower:AVERage:STATe 1

The following query returns 1.

:SENSe:OBWidth:AVERage:STATe?

**[ :SENSe]:OBWidth:AVERage:TCONtrol****Syntax**

[ :SENSe]:OBWidth:AVERage:TCONtrol EXPonential|REPeat

[ :SENSe]:OBWidth:AVERage:TCONtrol?

**Description**

Selects the average mode of the occupied bandwidth measurement.

Queries the average mode of the occupied bandwidth measurement.

**Parameter**

| Name | Type    | Range              | Default     |
|------|---------|--------------------|-------------|
| —    | Keyword | EXPonential REPeat | EXPonential |

**Remarks**

EXPonential: indicates the exponential average.

REPeat: indicates the repeat average.

When "EXPonential" is selected, the result is the exponential average of the measurement results obtained in the past N times (N is specified in [\[:SENSe\]:OBWidth:AVERage:COUNt](#)).

When "REPeat" is selected, the result is the arithmetic average of the measurement results obtained in the past N times (N is specified in [\[:SENSe\]:OBWidth:AVERage:COUNt](#)).

This command is only valid when the occupied bandwidth measurement in GPSA mode is enabled.

**Return Format**

The query returns EXP or REP.

**Example**

The following command sets the average mode to Repeat.

:SENSe:OBWidth:AVERage:TCONtrol REPeat

The following query returns REP.

:SENSe:OBWidth:AVERage:TCONtrol?

## [ :SENSe]:OBWidth:FREQuency:SPAN

### Syntax

[ :SENSe]:OBWidth:FREQuency:SPAN <freq>

[ :SENSe]:OBWidth:FREQuency:SPAN?

### Description

Sets the span of the occupied bandwidth measurement.

Queries the span of the occupied bandwidth measurement.

### Parameter

| Name   | Type                    | Range <sup>[1]</sup> | Default |
|--------|-------------------------|----------------------|---------|
| <freq> | Consecutive Real Number | 100 Hz to 4.5 GHz    | 2 MHz   |

**Note<sup>[1]</sup>:** If option RSA3000-BW1 is installed, the available range is from 10 Hz to 4.5 GHz.

### Remarks

This command is only valid when the occupied bandwidth measurement in GPSA mode is enabled.

This setting changes the span of the analyzer.

### Return Format

The query returns the span in scientific notation. The unit is Hz.

### Example

The following command sets the span to 1 MHz.

```
:SENSe:OBWidth:FREQuency:SPAN 1000000
```

The following query returns 1.0000000000e+ 06.

```
:SENSe:OBWidth:FREQuency:SPAN?
```

## [ :SENSe]:OBWidth:MAXHold:STATe

### Syntax

[ :SENSe]:OBWidth:MAXHold:STATe OFF|ON|0|1

[ :SENSe]:OBWidth:MAXHold:STATe?

### Description

Enables or disables Max Hold of the occupied bandwidth measurement.

Queries the Max Hold state of the occupied bandwidth measurement.

### Parameter

| Name | Type | Range      | Default |
|------|------|------------|---------|
| —    | Bool | OFF ON 0 1 | OFF 0   |

### Remarks

This command is only valid when the occupied bandwidth measurement in GPSA mode is enabled.

When Max Hold is enabled, each measurement result is compared with the previous result, and then display whichever is the maximum.

When Max Hold is disabled, the current measurement result is displayed.

Max Hold and average measurement mode are mutually exclusive. When Max Hold is enabled, the average measurement mode will be automatically disabled.

### Return Format

The query returns 0 or 1.

**Example**

The following command enables the Max Hold.

```
:SENSe:OBWidth:MAXHold:STATe ON or :SENSe:OBWidth:MAXHold:STATe 1
```

The following query returns 1.

```
:SENSe:OBWidth:MAXHold:STATe?
```

**[ :SENSe]:OBWidth:PERCent****Syntax**

```
[ :SENSe]:OBWidth:PERCent <real>
```

```
[ :SENSe]:OBWidth:PERCent?
```

**Description**

Sets the percentage the signal power takes up in the whole span power (power ratio).

Queries the power ratio of the occupied bandwidth measurement.

**Parameter**

| Name   | Type                    | Range      | Default |
|--------|-------------------------|------------|---------|
| <real> | Consecutive Real Number | 1 to 99.99 | 99      |

**Remarks**

This command is only valid when the occupied bandwidth measurement in GPSA mode is enabled.

The range of <real> is from 1 to 99.99. Therefore, set the range of the corresponding parameter for the analyzer to "1%-99.99%".

**Return Format**

The query returns the percentage in scientific notation.

**Example**

The following command sets the power ratio to 90%.

```
:SENSe:OBWidth:PERCent 90
```

The following query returns 9.000000000e+01.

```
:SENSe:OBWidth:PERCent?
```

**[ :SENSe]:POWer[:RF]:ATTenuation****Syntax**

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

```
[ :SENSe]:POWer[:RF]:ATTenuation?
```

**Description**

Sets the attenuation of the RF front-end attenuator.

Queries the attenuation of the RF front-end attenuator.

**Parameter**

| Name   | Type    | Range         | Default |
|--------|---------|---------------|---------|
| <real> | Integer | 0 dB to 50 dB | 10 dB   |

**Return Format**

The query returns the attenuation in integer. The unit is dB.

**Example**

The following command sets the attenuation to 20 dB.

```
:SENSe:POWer:RF:ATTenuation 20
```

The following query returns 20.

```
:SENSe:POWer:RF:ATTenuation?
```

**[ :SENSe]:POWer[:RF]:ATTenuation:AUTO****Syntax**

```
[ :SENSe]:POWer[:RF]:ATTenuation:AUTO OFF|ON|0|1
```

```
[ :SENSe]:POWer[:RF]:ATTenuation:AUTO?
```

**Description**

Enables or disables the auto setting mode of the input attenuation.

Queries the status of the auto setting mode of the input attenuation.

**Parameter**

| Name | Type | Range      | Default |
|------|------|------------|---------|
| —    | Bool | OFF ON 0 1 | ON 1    |

**Return Format**

The query returns 0 or 1.

**Example**

The following command disables the auto setting mode of the input attenuation.

```
:SENSe:POWer:RF:ATTenuation:AUTO OFF or :SENSe:POWer:RF:ATTenuation:AUTO 0
```

The following query returns 0.

```
:SENSe:POWer:RF:ATTenuation:AUTO?
```

**[ :SENSe]:POWer[:RF]:GAI N[:STATe]****Syntax**

```
[ :SENSe]:POWer[:RF]:GAI N[:STATe] OFF|ON|0|1
```

```
[ :SENSe]:POWer[:RF]:GAI N[:STATe]?
```

**Description**

Enable or disable the preamplifier.

Queries the status of the preamplifier.

**Parameter**

| Name | Type | Range      | Default |
|------|------|------------|---------|
| —    | Bool | OFF ON 0 1 | OFF 0   |

**Return Format**

The query returns 0 or 1.

**Example**

The following command enables the preamplifier.

```
:SENSe:POWer:RF:GAIN:STATe ON or :SENSe:POWer:RF:GAIN:STATe 1
```

The following query returns 1.

```
:SENSe:POWer:RF:GAIN:STATe?
```

## [ :SENSe]:POWer[:RF]:MI Xer:RANGe[:UPPer]

### Syntax

```
[ :SENSe]:POWer[:RF]:MI Xer:RANGe[:UPPer] < ampl>
[ :SENSe]:POWer[:RF]:MI Xer:RANGe[:UPPer] ?
```

### Description

Sets the maximum power of the input mixer.  
Queries the maximum power of the input mixer.

### Parameter

| Name    | Type                    | Range              | Default |
|---------|-------------------------|--------------------|---------|
| < ampl> | Consecutive Real Number | -50 dBm to -10 dBm | -10 dBm |

### Return Format

The query returns the maximum power of the input mixer in scientific notation. The unit is dBm.

### Example

The following command sets the maximum power of the input mixer to -20 dBm.

```
:SENSe:POWer:RF:MI Xer:RANGe:UPPer -20
```

The following query returns -2.000000000e+ 01.

```
:SENSe:POWer:RF:MI Xer:RANGe:UPPer?
```

## [ :SENSe]:SI GCapture:2FSK:AMPDown

### Syntax

```
[ :SENSe]:SIGCapture:2FSK:AMPDown < ample>
[ :SENSe]:SIGCapture:2FSK:AMPDown?
```

### Description

Sets the lower limit of the amplitude of 2FSK signal.  
Queries the lower limit of the amplitude of 2FSK signal.

### Parameter

| Name     | Type                    | Range                             | Default  |
|----------|-------------------------|-----------------------------------|----------|
| < ample> | Consecutive Real Number | -400 dBm to amplitude upper limit | -100 dBm |

### Remarks

This command is only available for RTSA mode and valid when the SSC function is enabled.

### Return Format

The query returns the amplitude lower limit in scientific notation.

### Example

The following command sets the amplitude lower limit to -20 dBm.

```
:SENSe:SIGCapture:2FSK:AMPDown -20
```

The following query returns -2.000000000e+ 01.

```
:SENSe:SIGCapture:2FSK:AMPDown?
```

## [ :SENSe]:SIGCapture:2FSK:AMPU

### Syntax

[ :SENSe]:SIGCapture:2FSK:AMPU < ample>

[ :SENSe]:SIGCapture:2FSK:AMPU?

### Description

Sets the upper limit of the amplitude of 2FSK signal.

Queries the upper limit of the amplitude of 2FSK signal.

### Parameter

| Name     | Type                    | Range               | Default |
|----------|-------------------------|---------------------|---------|
| < ample> | Consecutive Real Number | -100 dBm to 320 dBm | -10 dBm |

### Remarks

This command is only available for RTSA mode and valid when the SSC function is enabled.

### Return Format

The query returns the amplitude upper limit in scientific notation.

### Example

The following command sets the amplitude upper limit to -20 dBm.

:SENSe:SIGCapture:2FSK:AMPU -20

The following query returns -2.000000000e+01.

:SENSe:SIGCapture:2FSK:AMPU?

## [ :SENSe]:SIGCapture:2FSK:MARK1:FREQ

### Syntax

[ :SENSe]:SIGCapture:2FSK:MARK1:FREQ < freq>

[ :SENSe]:SIGCapture:2FSK:MARK1:FREQ?

### Description

Sets the frequency value at Marker 1.

Queries the frequency value at Marker 1.

### Parameter

| Name    | Type                    | Range                                 | Default         |
|---------|-------------------------|---------------------------------------|-----------------|
| < freq> | Consecutive Real Number | (start frequency) to (stop frequency) | Start Frequency |

### Remarks

This command is only available for RTSA mode and valid when the SSC function is enabled.

### Return Format

The query returns the frequency value of Marker 1 in scientific notation. The unit is Hz.

### Example

The following command sets the frequency value at Marker 1 to 1 MHz.

:SENSe:SIGCapture:2FSK:MARK1:FREQ 1000000

The following query returns 1.000000000e+06.

:SENSe:SIGCapture:2FSK:MARK1:FREQ?

## **[ :SENSe]:SIGCapture:2FSK:MARK1:SWitch[:STATe]**

### **Syntax**

```
[ :SENSe]:SIGCapture:2FSK:MARK1:SWitch[:STATe] OFF|ON|0|1
[:SENSe]:SIGCapture:2FSK:MARK1:SWitch[:STATe]?
```

### **Description**

Enables or disables Mark 1.

Queries the on/off status of Marker 1.

### **Parameter**

| Name | Type | Range      | Default |
|------|------|------------|---------|
| --   | Bool | OFF ON 0 1 | OFF 0   |

### **Remarks**

This command is only available for RTSA mode and valid when the SSC function is enabled.

### **Return Format**

The query returns 0 or 1.

### **Example**

The following command enables Marker 1.

```
:SENSe:SIGCapture:2FSK:MARK1:SWitch:STATe ON or :SENSe:SIGCapture:2FSK:MARK1:SWitch:STATe 1
```

The following query returns 1.

```
:SENSe:SIGCapture:2FSK:MARK1:SWitch:STATe?
```

## **[ :SENSe]:SIGCapture:2FSK:MARK2:FREQ**

### **Syntax**

```
[ :SENSe]:SIGCapture:2FSK:MARK2:FREQ <freq>
[:SENSe]:SIGCapture:2FSK:MARK2:FREQ?
```

### **Description**

Sets the frequency value at Marker 2.

Queries the frequency value at Marker 2.

### **Parameter**

| Name   | Type                    | Range                                 | Default        |
|--------|-------------------------|---------------------------------------|----------------|
| <freq> | Consecutive Real Number | (start frequency) to (stop frequency) | Stop Frequency |

### **Remarks**

This command is only available for RTSA mode and valid when the SSC function is enabled.

### **Return Format**

The query returns the frequency value of Marker 2 in integer. The unit is Hz.

### **Example**

The following command sets the frequency value at Marker 2 to 1 MHz.

```
:SENSe:SIGCapture:2FSK:MARK2:FREQ 1000000
```

The following query returns 1000000.

```
:SENSe:SIGCapture:2FSK:MARK2:FREQ?
```

## [ :SENSe]:SIGCapture:2FSK:MARK2:SWitch[:STATe]

### Syntax

[ :SENSe]:SIGCapture:2FSK:MARK2:SWitch[:STATe] OFF|ON|0|1

[ :SENSe]:SIGCapture:2FSK:MARK2:SWitch[:STATe] ?

### Description

Enables or disables Mark 2.

Queries the on/off status of Marker 2.

### Parameter

| Name | Type | Range      | Default |
|------|------|------------|---------|
| --   | Bool | OFF ON 0 1 | OFF 0   |

### Remarks

This command is only available for RTSA mode and valid when the SSC function is enabled.

### Return Format

The query returns 0 or 1.

### Example

The following command enables Marker 2.

:SENSe:SIGCapture:2FSK:MARK2:SWitch:STATe ON or :SENSe:SIGCapture:2FSK:MARK2:SWitch:STATe 1

The following query returns 1.

:SENSe:SIGCapture:2FSK:MARK2:SWitch:STATe?

## [ :SENSe]:SIGCapture:2FSK:MAXHold[:STATe]

### Syntax

[ :SENSe]:SIGCapture:2FSK:MAXHold[:STATe] OFF|ON|0|1

[ :SENSe]:SIGCapture:2FSK:MAXHold[:STATe] ?

### Description

Enables or disables the Max Hold.

Queries the on/off status of the Max Hold.

### Parameter

| Name | Type | Range      | Default |
|------|------|------------|---------|
| --   | Bool | OFF ON 0 1 | OFF 0   |

### Remarks

When the Max Hold is enabled (i.g. the SSC function is enabled), the signal captured each time will be compared with the last captured signal, whichever is the larger value will be displayed as the max hold trace.

### Return Format

The query returns 0 or 1.

### Example

The following command enables the Max Hold.

:SENSe:SIGCapture:2FSK:MAXHold:STATe ON or :SENSe:SIGCapture:2FSK:MAXHold:STATe 1

The following query returns 1.

:SENSe:SIGCapture:2FSK:MAXHold:STATe?

## [ :SENSe]:SIGCapture:2FSK:PEAKAmp?

### Syntax

[ :SENSe]:SIGCapture:2FSK:PEAKAmp? <peaknum>

### Description

Queries the amplitude of the nth peak in the SSC measurement results.

### Parameter

| Name      | Type    | Range  | Default |
|-----------|---------|--------|---------|
| <peaknum> | Integer | 1 to 6 | 1       |

### Remarks

This command is only available for RTSA mode and valid when the SSC function is enabled.

### Return Format

The query returns the amplitude of the nth peak (i.g. the value of the parameter <peaknum>) in scientific notation.

### Example

The following command queries the amplitude of the nth peak in the SSC measurement results, and returns 1.960000000e+01.

```
:SENSe:SIGCapture:2FSK:PEAKAmp? 1
```

## [ :SENSe]:SIGCapture:2FSK:PEAKFreq?

### Syntax

[ :SENSe]:SIGCapture:2FSK:PEAKFreq? <peaknum>

### Description

Queries the frequency of the nth peak in the SSC measurement results.

### Parameter

| Name      | Type    | Range  | Default |
|-----------|---------|--------|---------|
| <peaknum> | Integer | 1 to 6 | 1       |

### Remarks

This command is only available for RTSA mode and valid when the SSC function is enabled.

### Return Format

The query returns the frequency of the nth peak (i.g. the value of the parameter <peaknum>) in scientific notation.

### Example

The following command queries the frequency of the 1st peak in the SSC measurement results, and returns 1.000000000e+09.

```
:SENSe:SIGCapture:2FSK:PEAKFreq? 1
```

## [ :SENSe]:SIGCapture:2FSK:PF?

### Syntax

[ :SENSe]:SIGCapture:2FSK:PF?

### Description

Queries the test result of the Pass/Fail function.

### Remarks

This command is only available for RTSA mode and valid when the SSC function is enabled.

### Return Format

The query returns PASS or FAIL.

### Example

The following query returns PASS.

:SENSe:SIGCapture:2FSK:PF?

## [ :SENSe]:SIGCapture:2FSK:PFSWitch[:STATe]

### Syntax

[ :SENSe]:SIGCapture:2FSK:PFSWitch[:STATe] OFF|ON|0|1

[ :SENSe]:SIGCapture:2FSK:PFSWitch[:STATe] ?

### Description

Enables or disables the Pass/Fail function.

Queries the on/off status of the Pass/Fail function.

### Parameter

| Name | Type | Range      | Default |
|------|------|------------|---------|
| --   | Bool | OFF ON 0 1 | OFF 0   |

### Remarks

This command is only available for RTSA mode and valid when the SSC function is enabled.

### Return Format

The query returns 0 or 1.

### Example

The following command enables the the Pass/Fail function.

:SENSe:SIGCapture:2FSK:PFSWitch:STATe ON or :SENSe:SIGCapture:2FSK:PFSWitch:STATe 1

The following query returns 1.

:SENSe:SIGCapture:2FSK:PFSWitch:STATe?

## **[ :SENSe]:SIGCapture:2FSK:RESet**

### **Syntax**

[ :SENSe]:SIGCapture:2FSK:RESet

### **Description**

Performs the reset operation, deletes the original signal, and restarts to capture the signal.

### **Remarks**

This command is only available for RTSA mode and valid when the SSC function is enabled.

## **[ :SENSe]:SIGCapture:2FSK:SIGNAL**

### **Syntax**

[ :SENSe]:SIGCapture:2FSK:SIGNAL 0|1|2

[ :SENSe]:SIGCapture:2FSK:SIGNAL?

### **Description**

Selects the signal whose Limit value is required to be modified in the pass/fail test.

Queries the signal whose Limit value is required to be modified in the pass/fail test.

### **Parameter**

| Name | Type     | Range | Default |
|------|----------|-------|---------|
| --   | Discrete | 0 1 2 | 0       |

### **Remarks**

0: selects to modify Signal 1.

1: selects to modify Signal 2.

2: selects to modify Signal 3.

This command is only available for RTSA mode and valid when the SSC function is enabled.

### **Return Format**

The query returns 0, 1, or 2.

### **Example**

The following command selects to modify Signal 1.

:SENSe:SIGCapture:2FSK:SIGNAL 0

The following query returns 0.

:SENSe:SIGCapture:2FSK:SIGNAL?

## [ :SENSe]:SWEep:POINts

### Syntax

[ :SENSe]:SWEep:POINts <integer>

[ :SENSe]:SWEep:POINts?

### Description

Sets the number of sweep points.

Queries the number of sweep points.

### Parameter

| Name      | Type    | Range         | Default |
|-----------|---------|---------------|---------|
| <integer> | Integer | 101 to 10,001 | 801     |

### Remarks

This command is only available for GPSA mode.

### Return Format

The query returns the number of sweep points in integer.

### Example

The following command sets the number of sweep points to 650.

```
:SENSe:SWEep:POINts 650
```

The following query returns 650.

```
:SENSe:SWEep:POINts?
```

## [ :SENSe]:SWEep:TIME

### Syntax

[ :SENSe]:SWEep:TIME <time>

[ :SENSe]:SWEep:TIME?

### Description

Sets the sweep time.

Queries the sweep time.

### Parameter

| Name   | Type                    | Range                                                               | Default |
|--------|-------------------------|---------------------------------------------------------------------|---------|
| <time> | Consecutive Real Number | 1 ms to 4,000 s (non-zero span)<br>1 $\mu$ s to 6,000 s (zero span) | 1 ms    |

### Remarks

This command is only available for GPSA mode.

### Return Format

The query returns the sweep time in scientific notation. The unit is s.

### Example

The following command sets the sweep time to 100 ms.

```
:SENSe:SWEep:TIME 0.1
```

The following query returns 1.000000000e-01.

```
:SENSe:SWEep:TIME?
```

## [ :SENSe]:SWEep:TI ME:AUTO

### Syntax

```
[ :SENSe]:SWEep:TIME:AUTO OFF|ON|0|1
[ :SENSe]:SWEep:TIME:AUTO?
```

### Description

Enables or disables auto sweep time.  
Queries the status of the auto sweep time.

### Parameter

| Name | Type | Range      | Default |
|------|------|------------|---------|
| —    | Bool | OFF ON 0 1 | ON 1    |

### Remarks

This command is only available for GPSA mode.

### Return Format

The query returns 0 or 1.

### Example

The following command enables the auto sweep time.  
:SENSe:SWEep:TIME:AUTO ON or :SENSe:SWEep:TIME:AUTO 1

The following query returns 1.  
:SENSe:SWEep:TIME:AUTO?

## [ :SENSe]:SWEep:TI ME:AUTO:RULEs

### Syntax

```
[ :SENSe]:SWEep:TIME:AUTO:RULEs NORMal|ACCuracy
[ :SENSe]:SWEep:TIME:AUTO:RULEs?
```

### Description

Selects the sweep type.  
Queries the sweep type.

### Parameter

| Name | Type    | Range           | Default |
|------|---------|-----------------|---------|
| —    | Keyword | NORMal ACCuracy | NORMal  |

### Remarks

NORMal: indicates normal.  
ACCuracy: indicates accuracy.  
This command is only available for GPSA mode

### Return Format

The query returns NORM or ACC.

### Example

The following command selects the sweep type to Accuracy.  
:SENSe:SWEep:TIME:AUTO:RULEs ACCuracy

The following query returns ACC.  
:SENSe:SWEep:TIME:AUTO:RULEs?

## [ :SENSe]:TOI :AVERAge:COUNT

### Syntax

```
[ :SENSe]:TOI:AVERAge:COUNT <integer>
[ :SENSe]:TOI:AVERAge:COUNT?
```

### Description

Sets the average count of the TOI measurement.  
Queries the average count of the TOI measurement.

### Parameter

| Name      | Type    | Range      | Default |
|-----------|---------|------------|---------|
| <integer> | Integer | 1 to 1,000 | 10      |

### Remarks

This command is only valid when the TOI measurement in GPSA mode is enabled.

### Return Format

The query returns the average count in integer.

### Example

The following command sets the average count to 100.  
:SENSe:TOI:AVERAge:COUNT 100

The following query returns 100.  
:SENSe:TOI:AVERAge:COUNT?

## [ :SENSe]:TOI :AVERAge[:STATe]

### Syntax

```
[ :SENSe]:TOI:AVERAge[:STATe] OFF|ON|0|1
[ :SENSe]:TOI:AVERAge[:STATe]?
```

### Description

Enables or disables the average measurement function of the TOI measurement.  
Queries the status of the average measurement function of the TOI measurement.

### Parameter

| Name | Type | Range      | Default |
|------|------|------------|---------|
| —    | Bool | OFF ON 0 1 | ON 1    |

### Remarks

This command is only valid when the TOI measurement in GPSA mode is enabled.

### Return Format

The query returns 0 or 1.

### Example

The following command enables the average measurement function.  
:SENSe:TOI:AVERAge:STATe ON or :SENSe:TOI:AVERAge:STATe 1

The following query returns 1.  
:SENSe:TOI:AVERAge:STATe?

## [ :SENSe]:TOI :AVERage:TCONtrol

### Syntax

```
[ :SENSe]:TOI :AVERage:TCONtrol EXPonential| REPeat
[ :SENSe]:TOI :AVERage:TCONtrol?
```

### Description

Selects the average mode of the TOI measurement.

Queries the average mode of the TOI measurement.

### Parameter

| Name | Type    | Range               | Default     |
|------|---------|---------------------|-------------|
| —    | Keyword | EXPonential  REPeat | EXPonential |

### Remarks

EXPonential: indicates the exponential average.

REPeat: indicates the repeat average.

When "EXPonential" is selected, the result is the exponential average of the measurement results obtained in the past N times (N is specified in [\[ :SENSe\]:TOI :AVERage:COUNt](#)).

When "REPeat" is selected, the result is the arithmetic average of the measurement results obtained in the past N times (N is specified in [\[ :SENSe\]:TOI :AVERage:COUNt](#)).

This command is only valid when the TOI measurement in GPSA mode is enabled.

### Return Format

The query returns EXP or REP.

### Example

The following command sets the average mode to Repeat.

```
:SENSe:TOI :AVERage:TCONtrol REPeat
```

The following query returns REP.

```
:SENSe:TOI :AVERage:TCONtrol?
```

## [ :SENSe]:TOI :FREQuency:SPAN

### Syntax

```
[ :SENSe]:TOI :FREQuency:SPAN <freq>
[ :SENSe]:TOI :FREQuency:SPAN?
```

### Description

Sets the span of the TOI measurement.

Queries the span of the TOI measurement.

### Parameter

| Name   | Type                    | Range <sup>[1]</sup> | Default |
|--------|-------------------------|----------------------|---------|
| <freq> | Consecutive Real Number | 100 Hz to 4.5 GHz    | 2 MHz   |

**Note**<sup>[1]</sup>: If option RSA3000-BW1 is installed, the available range is from 10 Hz to 4.5 GHz.

### Remarks

This command is only valid when the TOI measurement in GPSA mode is enabled.

This setting will change the span of the analyzer.

### Return Format

The query returns the span in scientific notation. The unit is Hz.

**Example**

The following command sets the span to 1 MHz.

```
:SENSe:TOI:FREQuency:SPAN 1000000
```

The following query returns 1.000000000e+06.

```
:SENSe:TOI:FREQuency:SPAN?
```

**[ :SENSe]:TPOWer:AVERAge:COUNT****Syntax**

```
[ :SENSe]:TPOWer:AVERAge:COUNT <integer>
```

```
[ :SENSe]:TPOWer:AVERAge:COUNT?
```

**Description**

Sets the average count of the T-power measurement.

Queries the average count of the T-power measurement.

**Parameter**

| Name      | Type    | Range      | Default |
|-----------|---------|------------|---------|
| <integer> | Integer | 1 to 1,000 | 10      |

**Remarks**

This command is only valid when the T-power measurement in GPSA mode is enabled.

**Return Format**

The query returns the average count in integer.

**Example**

The following command sets the average count to 100.

```
:SENSe:TPOWer:AVERAge:COUNT 100
```

The following query returns 100.

```
:SENSe:TPOWer:AVERAge:COUNT?
```

**[ :SENSe]:TPOWer:AVERAge[:STATe]****Syntax**

```
[ :SENSe]:TPOWer:AVERAge[:STATe] OFF|ON|0|1
```

```
[ :SENSe]:TPOWer:AVERAge[:STATe]?
```

**Description**

Enables or disables the average measurement function of the T-power measurement.

Queries the status of the average measurement function of the T-power measurement.

**Parameter**

| Name | Type | Range      | Default |
|------|------|------------|---------|
| —    | Bool | OFF ON 0 1 | OFF 0   |

**Remarks**

This command is only valid when the T-power measurement in GPSA mode is enabled.

**Return Format**

The query returns 0 or 1.

**Example**

The following command enables the average measurement.

```
:SENSe:TPOWer:AVERAge:STATe ON or :SENSe:TPOWer:AVERAge:STATe 1
```

The following query returns 1.

```
:SENSe:TPOWer:AVERAge:STATe?
```

**[ :SENSe]:TPOWer:AVERAge:TCONtrol****Syntax**

```
[ :SENSe]:TPOWer:AVERAge:TCONtrol EXPonential| REPeat
```

```
[ :SENSe]:TPOWer:AVERAge:TCONtrol?
```

**Description**

Sets the average mode of the T-power measurement.

Queries the average mode of the T-power measurement.

**Parameter**

| Name | Type    | Range               | Default     |
|------|---------|---------------------|-------------|
| —    | Keyword | EXPonential  REPeat | EXPonential |

**Remarks**

EXPonential: indicates the exponential average.

REPeat: indicates the repeat average.

When "EXPonential" is selected, the result is the exponential average of the measurement results obtained in the past N times (N is specified in "[\[:SENSe\]:TPOWer:AVERAge:COUNt](#)").

When "REPeat" is selected, the result is the arithmetic average of the measurement results obtained in the past N times (N is specified in "[\[:SENSe\]:TPOWer:AVERAge:COUNt](#)").

This command is only valid when the T-power measurement in GPSA mode is enabled.

**Return Format**

The query returns EXP or REP.

**Example**

The following command sets the average mode to Repeat.

```
:SENSe:TPOWer:AVERAge:TCONtrol REPeat
```

The following query returns REP.

```
:SENSe:TPOWer:AVERAge:TCONtrol?
```

**[ :SENSe]:TPOWer:LLIMit****Syntax**

```
[ :SENSe]:TPOWer:LLIMit <time>
```

```
[ :SENSe]:TPOWer:LLIMit?
```

**Description**

Sets the start line for the T-power measurement.

Queries the start line for the T-power measurement.

**Parameter**

| Name   | Type                    | Range                                     | Default   |
|--------|-------------------------|-------------------------------------------|-----------|
| <time> | Consecutive Real Number | 0 $\mu$ s to (current value of stop line) | 0 $\mu$ s |

**Remarks**

This command is only valid when the T-power measurement in GPSA mode is enabled.

**Return Format**

The query returns the start line in scientific notation. The unit is s.

**Example**

The following command sets the start line to 5 ms.

```
:SENSe:TPOWer:LLIMit 0.005
```

The following query returns 5.000000000e-03.

```
:SENSe:TPOWer:LLIMit?
```

**[ :SENSe]:TPOWer:MODE****Syntax**

```
[ :SENSe]:TPOWer:MODE AVERage| PEAK| RMS
```

```
[ :SENSe]:TPOWer:MODE?
```

**Description**

Sets the power type for the T-power measurement.

Queries the power type for the T-power measurement.

**Parameter**

| Name | Type    | Range              | Default |
|------|---------|--------------------|---------|
| —    | Keyword | AVERage  PEAK  RMS | PEAK    |

**Remarks**

AVERage: indicates average power.

PEAK: indicates peak power.

RMS: indicates the RMS power.

This command is only valid when the T-power measurement in GPSA mode is enabled.

**Return Format**

The query returns AVER, PEAK, or RMS.

**Example**

The following command sets the power type to Average.

```
:SENSe:TPOWer:MODE AVERage
```

The following query returns AVER.

```
:SENSe:TPOWer:MODE?
```

## [ :SENSe]:TPOWer:RLIMit

### Syntax

[ :SENSe]:TPOWer:RLIMit <time>

[ :SENSe]:TPOWer:RLIMit?

### Description

Sets the stop line for the T-power measurement.

Queries the stop line for the T-power measurement.

### Parameter

| Name   | Type                    | Range                                                         | Default |
|--------|-------------------------|---------------------------------------------------------------|---------|
| <time> | Consecutive Real Number | current value of start line to<br>current value of sweep time | 1 ms    |

### Remarks

This command is only valid when the T-power measurement in GPSA mode is enabled.

### Return Format

The query returns the stop line in scientific notation. The unit is s.

### Example

The following command sets the stop line to 10 ms.

:SENSe:TPOWer:RLIMit 0.01

The following query returns 1.000000000e-02.

:SENSe:TPOWer:RLIMit?

## :SOURce Commands

### Command List:

- ◆ [:SOURce:CORRection:OFFSet](#)
- ◆ [:SOURce\[:EXTeRnal\]:POWeR\[:LEVel\]\[:IMMediate\]\[:AMPLitude\]](#)
- ◆ [:SOURce:TRACe:REFerence:STATe](#)
- ◆ [:SOURce:TRACe:STORref](#)

### Remarks:

The :SOURce commands are only available for RSA3045-TG/RSA3030-TG working in GPSA mode.

## :SOURce:CORRection:OFFSet

### Syntax

```
:SOURce:CORRection:OFFSet <rel_ampl>
:SOURce:CORRection:OFFSet?
```

### Description

Sets the offset of the output amplitude of the tracking generator.  
Queries the offset of the output amplitude of the tracking generator.

### Parameter

| Name       | Type                    | Range             | Default |
|------------|-------------------------|-------------------|---------|
| <rel_ampl> | Consecutive Real Number | -200 dB to 200 dB | 0 dB    |

### Return Format

The query returns the offset in scientific notation.  
This parameter only changes the readout of the tracking generator output power, but does not affect the actual output power.

### Example

The following command sets the offset to 10 dB.  
:SOURce:CORRection:OFFSet 10

The following query returns 1.000000000e+ 01.  
:SOURce:CORRection:OFFSet?

## :SOURce[:EXTeRnal]:POWeR[:LEVel][:IMMediate][:AMPLitude]

### Syntax

```
:SOURce[:EXTeRnal]:POWeR[:LEVel][:IMMediate][:AMPLitude] <ampl>
:SOURce[:EXTeRnal]:POWeR[:LEVel][:IMMediate][:AMPLitude]?
```

### Description

Sets the output amplitude of the tracking generator.  
Queries the output amplitude of the tracking generator.

### Parameter

| Name   | Type                    | Range            | Default |
|--------|-------------------------|------------------|---------|
| <ampl> | Consecutive Real Number | -40 dBm to 0 dBm | -40 dBm |

**Return Format**

The query returns the output amplitude in scientific notation.

**Example**

The following command sets the output amplitude to -10 dB.

```
:SOURce:EXtErnal:POWeR:LEVel:IMMediate:AMPLitude -10
```

The following query returns -1.000000000e+01.

```
:SOURce:EXtErnal:POWeR:LEVel:IMMediate:AMPLitude?
```

**:SOURce:TRACe:REFerence:STATe****Syntax**

```
:SOURce:TRACe:REFerence:STATe OFF|ON|0|1
```

```
:SOURce:TRACe:REFerence:STATe?
```

**Description**

Sets whether to display the reference trace of normalization.

Queries whether to display the reference trace of normalization.

**Parameter**

| Name | Type | Range      | Default |
|------|------|------------|---------|
| —    | Bool | OFF ON 0 1 | OFF 0   |

**Remarks**

This command is only valid when the tracking generator function is enabled.

**Return Format**

The query returns 0 or 1.

**Example**

The following command enables the reference trace.

```
:SOURce:TRACe:REF:STATe ON or :SOURce:TRACe:REF:STATe 1
```

The following query returns 1.

```
:SOURce:TRACe:REF:STATe?
```

**:SOURce:TRACe:STORref****Syntax**

```
:SOURce:TRACe:STORref
```

**Description**

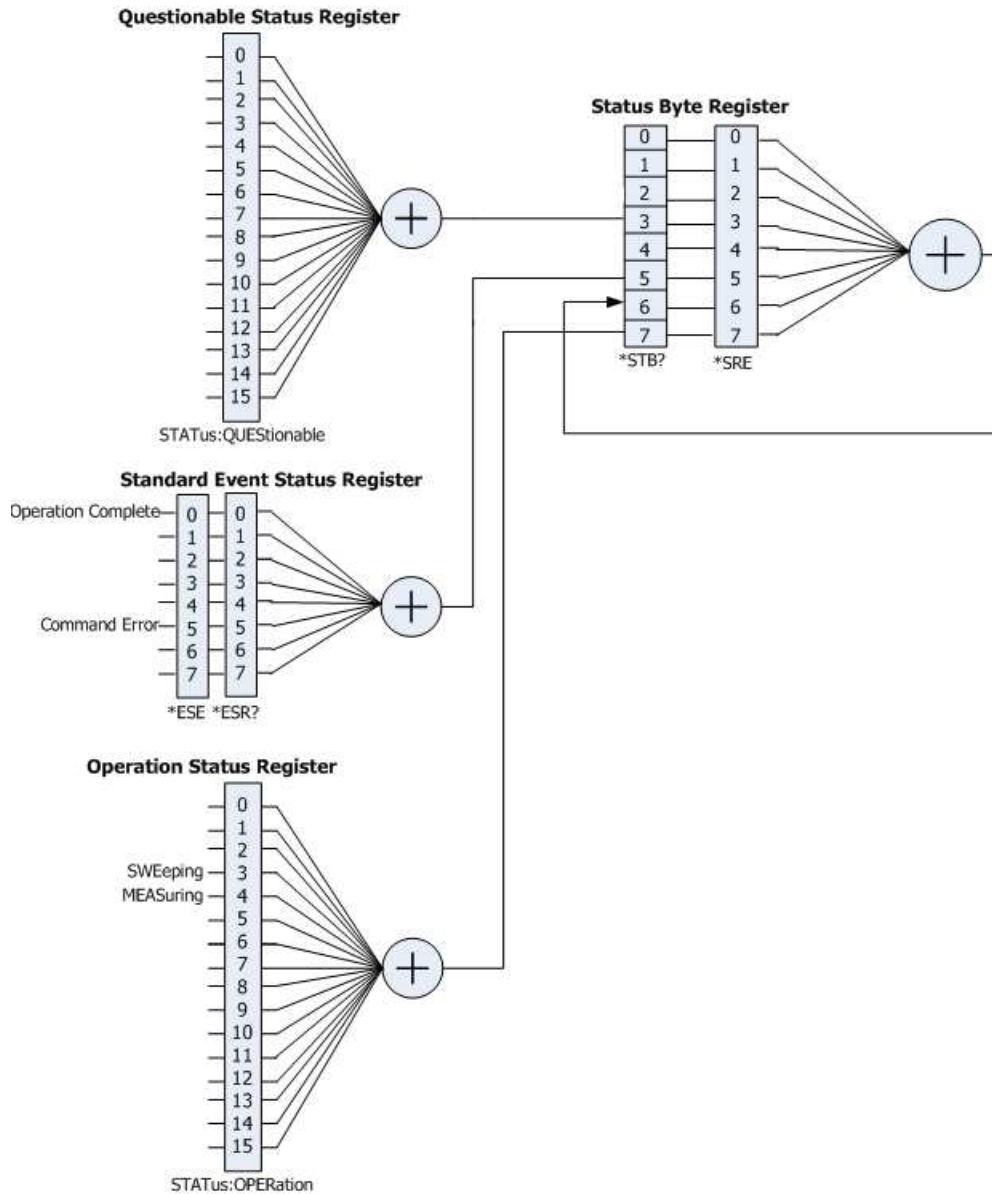
Saves the reference trace of normalization.

**Remarks**

This command is only valid when the tracking generator function is enabled.

## :STATus Commands

The :STATus commands and [IEEE 488.2 Common Commands](#) are used to operate or query the status registers. The structure of the status register is shown in the following figure. The STATus commands are used to set and query the questionable status register and operation status register. IEEE488.2 common commands can be used to operate on the standard event status register and status byte register.



### Command List:

- ◆ [:STATus:OPERation:CONDition?](#)
- ◆ [:STATus:OPERation:ENABLE](#)
- ◆ [:STATus:OPERation\[:EVENT\]?](#)
- ◆ [:STATus:PRESet](#)
- ◆ [:STATus:QUESTionable:CONDition?](#)
- ◆ [:STATus:QUESTionable:ENABLE](#)
- ◆ [:STATus:QUESTionable\[:EVENT\]?](#)

## :STATus:OPERation:CONDition?

### Syntax

:STATus:OPERation:CONDition?

### Description

Queries the condition register of the Operation Status Register.

### Return Format

The query returns the condition register in integer. For example, 24.

## :STATus:OPERation:ENABLE

### Syntax

:STATus:OPERation:ENABLE < integer>  
:STATus:OPERation:ENABLE?

### Description

Sets the enable register of the Operation Status Register.

Queries the enable register of the Operation Status Register.

### Parameter

| Name       | Type    | Range              | Default |
|------------|---------|--------------------|---------|
| < integer> | Integer | Refer to "Remarks" | 0       |

### Remarks

The following table lists the bit definitions for the operation status register. The bit 0 to bit 2, bit 5 to bit 7, bit 13 and bit 14 are reserved; they can be set but will not affect the instrument. The bit 15 and bit 12 to bit 8 are not used and are always treated as 0; therefore, the range of < integer> are the decimal numbers corresponding to the binary numbers ranging from 0000000000000000 (0 in decimal) to 1111111111111111 (32767 in decimal) and of which the bit 15 and bit 12 to bit 8 are 0.

| Bit | Value | Definition |
|-----|-------|------------|
| 0   | 1     | Reserved   |
| 1   | 2     | Reserved   |
| 2   | 4     | Reserved   |
| 3   | 8     | SWEEping   |
| 4   | 16    | MEASuring  |
| 5   | 32    | Reserved   |
| 6   | 64    | Reserved   |
| 7   | 128   | Reserved   |
| 8   | 0     | Not Used   |
| 9   | 0     | Not Used   |
| 10  | 0     | Not Used   |
| 11  | 0     | Not Used   |
| 12  | 0     | Not Used   |
| 13  | 8192  | Reserved   |
| 14  | 16384 | Reserved   |
| 15  | 0     | Not Used   |

### Return Format

The query returns the enable register of the Operation Status Register in integer.

**Example**

The following command sets the enable register of the Operation Status Register to 100.

```
:STATus:OPERation:ENABLE 100
```

The following query returns 100.

```
:STATus:OPERation:ENABLE?
```

**:STATus:OPERation[:EVENT]?****Syntax**

```
:STATus:OPERation[:EVENT]?
```

**Description**

Queries the event register of the Operation Status Register.

**Return Format**

The query returns the event register of the Operation Status Register in integer. For example, 24.

**:STATus:PRESet****Syntax**

```
:STATus:PRESet
```

**Description**

Clears the enable register of the Operation Status Register and Questionable Status Register.

**:STATus:QUEStionable:CONDition?****Syntax**

```
:STATus:QUEStionable:CONDition?
```

**Description**

Queries the condition register of the Questionable Status Register.

**Return Format**

The query returns the condition register of the Questionable Status Register in integer. For example, 0.

**:STATus:QUEStionable:ENABLE****Syntax**

```
:STATus:QUEStionable:ENABLE <integer>  
:STATus:QUEStionable:ENABLE?
```

**Description**

Sets the enable register of the Questionable Status Register.

Queries the enable register of the Questionable Status Register.

**Parameter**

| Name      | Type    | Range              | Default |
|-----------|---------|--------------------|---------|
| <integer> | Integer | Refer to "Remarks" | 0       |

**Remarks**

The following table lists the bit definitions for the questionable status register. The bit 0 to bit 8, bit 13 and bit 14 are reserved; they can be set but will not affect the instrument. The bit 15 and bit 12 to bit 9 are not used and are always treated as 0; therefore, the range of <integer> are the decimal numbers corresponding to the binary numbers ranging from 0000000000000000 (0 in decimal) to 1111111111111111 (32767 in decimal) and of which the bit 15 and bit 12 to bit 9 are 0.

| Bit | Value | Definition |
|-----|-------|------------|
| 0   | 1     | Reserved   |
| 1   | 2     | Reserved   |
| 2   | 4     | Reserved   |
| 3   | 8     | Reserved   |
| 4   | 16    | Reserved   |
| 5   | 32    | Reserved   |
| 6   | 64    | Reserved   |
| 7   | 128   | Reserved   |
| 8   | 256   | Reserved   |
| 9   | 0     | Not Used   |
| 10  | 0     | Not Used   |
| 11  | 0     | Not Used   |
| 12  | 0     | Not Used   |
| 13  | 8192  | Reserved   |
| 14  | 16384 | Reserved   |
| 15  | 0     | Not Used   |

**Return Format**

The query returns the enable register of the Questionable Status Register in integer.

**Example**

The following command sets the enable register of the Questionable Status Register to 100.  
:STATus:QUESTionable:ENABLE 100

The following query returns 100.  
:STATus:QUESTionable:ENABLE?

**:STATus:QUESTionable[:EVENT]?****Syntax**

:STATus:QUESTionable[:EVENT]?

**Description**

Queries the event register of the Questionable Status Register.

**Return Format**

Queries the event register of the Questionable Status Register in integer. For example, 0.

## :SYSTem Commands

### Command List:

- ◆ [:SYSTem:BEEPer:STATe](#)
- ◆ [:SYSTem:COMMUnicate:LAN\[:SELF\]:APPLy](#)
- ◆ [:SYSTem:COMMUnicate:LAN\[:SELF\]:AUTOp:STATe](#)
- ◆ [:SYSTem:COMMUnicate:LAN\[:SELF\]:DHCP:STATe](#)
- ◆ [:SYSTem:COMMUnicate:LAN\[:SELF\]:IP:ADDRes](#)
- ◆ [:SYSTem:COMMUnicate:LAN\[:SELF\]:IP:DNS:AUTO\[:STATe\]](#)
- ◆ [:SYSTem:COMMUnicate:LAN\[:SELF\]:IP:DNSBack](#)
- ◆ [:SYSTem:COMMUnicate:LAN\[:SELF\]:IP:DNSPreferred](#)
- ◆ [:SYSTem:COMMUnicate:LAN\[:SELF\]:IP:DNSServer](#)
- ◆ [:SYSTem:COMMUnicate:LAN\[:SELF\]:IP:GATeway](#)
- ◆ [:SYSTem:COMMUnicate:LAN\[:SELF\]:IP:SUBMask](#)
- ◆ [:SYSTem:COMMUnicate:LAN\[:SELF\]:MANuip:STATe](#)
- ◆ [:SYSTem:COMMUnicate:LAN\[:SELF\]:MDNS:STATe](#)
- ◆ [:SYSTem:COMMUnicate:LAN\[:SELF\]:RESet](#)
- ◆ [:SYSTem:CONFigure:INFormation?](#)
- ◆ [:SYSTem:DATE](#)
- ◆ [:SYSTem:FSWitch\[:STATe\]](#)
- ◆ [:SYSTem:LANGuage](#)
- ◆ [:SYSTem:LKEY](#)
- ◆ [:SYSTem:OPTion:STATe?](#)
- ◆ [:SYSTem:PON:TYPE](#)
- ◆ [:SYSTem:PRESet](#)
- ◆ [:SYSTem:PRESet:SAVE](#)
- ◆ [:SYSTem:PRESet:TYPE](#)
- ◆ [:SYSTem:PRESet:USER:SAVE](#)
- ◆ [:SYSTem:SCPI:DISPlay](#)
- ◆ [:SYSTem:SHOW](#)
- ◆ [:SYSTem:TIME](#)

## :SYSTem:BEEPer:STATe

### Syntax

:SYSTem:BEEPer:STATe OFF|ON|0|1  
:SYSTem:BEEPer:STATe?

### Description

Turns on or off the beeper.  
Queries the status of the beeper.

### Parameter

| Name | Type | Range      | Default |
|------|------|------------|---------|
| ——   | Bool | OFF ON 0 1 | OFF 0   |

### Return Format

The query returns 0 or 1.

### Example

The following command turns on the beeper.  
:SYSTem:BEEPer:STATe ON or :SYSTem:BEEPer:STATe 1

The following query returns 1.  
:SYSTem:BEEPer:STATe?

## :SYSTem:COMMunicate:LAN[:SELF]:APPLy

### Syntax

:SYSTem:COMMunicate:LAN[:SELF]:APPLy

### Description

Applies the LAN interface parameter settings.

## :SYSTem:COMMunicate:LAN[:SELF]:AUTOip:STATe

### Syntax

:SYSTem:COMMunicate:LAN[:SELF]:AUTOip:STATe OFF|ON|0|1  
:SYSTem:COMMunicate:LAN[:SELF]:AUTOip:STATe?

### Description

Enables or disables the auto IP setting mode.  
Queries the status of the auto IP setting mode.

### Parameter

| Name | Type | Range      | Default |
|------|------|------------|---------|
| ——   | Bool | OFF ON 0 1 | OFF 0   |

### Remarks

The analyzer attempts to acquire the IP address configuration according to the sequence of DHCP, Auto IP, and Manual IP. These three modes cannot be disabled at the same time.

ON|1: enables the auto IP mode.

OFF|0: disables the auto IP mode and selects DHCP configuration mode.

**Return Format**

The query returns 0 or 1.

**Example**

The following command enables the auto IP setting mode.

```
:SYSTem:COMMUnicate:LAN:SELF:AUTOip:STATe ON or :SYSTem:COMMUnicate:LAN:SELF:AUTOip:STATe 1
```

The following query returns 1.

```
:SYSTem:COMMUnicate:LAN:SELF:AUTOip:STATe?
```

**:SYSTem:COMMUnicate:LAN[:SELF]:DHCP:STATe****Syntax**

```
:SYSTem:COMMUnicate:LAN[:SELF]:DHCP:STATe OFF|ON|0|1
```

```
:SYSTem:COMMUnicate:LAN[:SELF]:DHCP:STATe?
```

**Description**

Enables or disables the DHCP configuration mode.

Queries the status of the DHCP configuration mode.

**Parameter**

| Name | Type | Range      | Default |
|------|------|------------|---------|
| —    | Bool | OFF ON 0 1 | ON 1    |

**Remarks**

The analyzer attempts to acquire the IP address configuration according to the sequence of DHCP, Auto IP, and Manual IP. These three modes cannot be disabled at the same time.

ON|1: enables the DHCP configuration mode.

OFF|0: disables the DHCP configuration mode and selects the auto IP mode.

**Return Format**

The query returns 0 or 1.

**Example**

The following command enables the DHCP configuration mode.

```
:SYSTem:COMMUnicate:LAN:SELF:DHCP:STATe ON or :SYSTem:COMMUnicate:LAN:SELF:DHCP:STATe 1
```

The following query returns 1.

```
:SYSTem:COMMUnicate:LAN:SELF:DHCP:STATe?
```

**:SYSTem:COMMUnicate:LAN[:SELF]:IP:ADDRESS****Syntax**

```
:SYSTem:COMMUnicate:LAN[:SELF]:IP:ADDRESS <ip_address>
```

```
:SYSTem:COMMUnicate:LAN[:SELF]:IP:ADDRESS?
```

**Description**

Sets the IP address.

Queries the IP address.

**Parameter**

| Name         | Type         | Range              | Default |
|--------------|--------------|--------------------|---------|
| <ip_address> | ASCII String | Refer to "Remarks" | —       |

**Remarks**

The format of <ip\_address> is "nnn.nnn.nnn.nnn". The range for the first segment (nnn) of the address is from 1 to 223 (except 127); and the range for the other three segments is from 0 to 255.  
This command is only valid in manual IP setting mode.

**Return Format**

The query returns the current IP address in the format of "nnn.nnn.nnn.nnn".

**Example**

The following command sets the IP address to "172.16.3.199".  
:SYSTem:COMMunicate:LAN:SELF:IP:ADDRes 172.16.3.199

The following query returns 172.16.3.199.  
:SYSTem:COMMunicate:LAN:SELF:IP:ADDRes?

**:SYSTem:COMMunicate:LAN[:SELF]:IP:DNS:AUTO[:STATe]****Syntax**

:SYSTem:COMMunicate:LAN[:SELF]:IP:DNS:AUTO[:STATe] OFF|ON|0|1  
:SYSTem:COMMunicate:LAN[:SELF]:IP:DNS:AUTO[:STATe]?

**Description**

Sets the mode to obtain the DNS.  
Queries the mode to obtain the DNS.

**Parameter**

| Name | Type | Range      | Default |
|------|------|------------|---------|
| —    | Bool | OFF ON 0 1 | OFF 0   |

**Remarks**

ON|1: enables the auto mode of DNS, that is, selects the auto mode.  
OFF|0: disables the auto mode of DNS, that is, selects the manual mode.

**Return Format**

The query returns 0 or 1.

**Example**

The following command enables the auto mode of DNS.  
:SYSTem:COMMunicate:LAN:SELF:IP:DNS:AUTO:STATe ON  
or :SYSTem:COMMunicate:LAN:SELF:IP:DNS:AUTO:STATe 1

The following query returns 1.  
:SYSTem:COMMunicate:LAN:SELF:IP:DNS:AUTO:STATe?

## :SYSTem:COMMunicate:LAN[:SELF]:IP:DNSBack

### Syntax

:SYSTem:COMMunicate:LAN[:SELF]:IP:DNSBack <ip\_address>  
:SYSTem:COMMunicate:LAN[:SELF]:IP:DNSBack?

### Description

Sets the backup address for the DNS.  
Queries the backup address for the DNS.

### Parameter

| Name         | Type         | Range                       | Default |
|--------------|--------------|-----------------------------|---------|
| <ip_address> | ASCII String | Refer to " <b>Remarks</b> " | ——      |

### Remarks

The format of <ip\_address> is "nnn.nnn.nnn.nnn". The range for the first segment (nnn) of the address is from 1 to 223 (except 127); and the range for the other three segments is from 0 to 255.  
This command is only valid in manual DNS setting mode.

### Return Format

The query returns the current backup DNS address in the format of "nnn.nnn.nnn.nnn".

### Example

The following command sets the backup address of DNS to "172.16.2.2".  
:SYSTem:COMMunicate:LAN:SELF:IP:DNSBack 172.16.2.2

The following query returns 172.16.2.2.  
:SYSTem:COMMunicate:LAN:SELF:IP:DNSBack?

## :SYSTem:COMMunicate:LAN[:SELF]:IP:DNSPreferred :SYSTem:COMMunicate:LAN[:SELF]:IP:DNSServer

### Syntax

:SYSTem:COMMunicate:LAN[:SELF]:IP:DNSPreferred <ip\_address>  
:SYSTem:COMMunicate:LAN[:SELF]:IP:DNSPreferred?  
:SYSTem:COMMunicate:LAN[:SELF]:IP:DNSServer <ip\_address>  
:SYSTem:COMMunicate:LAN[:SELF]:IP:DNSServer?

### Description

Sets the preferred address for the DNS.  
Queries the preferred address for the DNS.

### Parameter

| Name         | Type         | Range                       | Default |
|--------------|--------------|-----------------------------|---------|
| <ip_address> | ASCII String | Refer to " <b>Remarks</b> " | ——      |

### Remarks

The format of <ip\_address> is "nnn.nnn.nnn.nnn". The range for the first segment (nnn) of the address is from 1 to 223 (except 127); and the range for the other three segments is from 0 to 255.  
This command is only valid in manual DNS setting mode.

### Return Format

The query returns the currently preferred DNS address in the format of "nnn.nnn.nnn.nnn".

**Example**

The following command sets the preferred DNS address to "172.16.2.2".

```
:SYSTem:COMMunicate:LAN:SELF:IP:DNSPreferred 172.16.2.2
```

```
:SYSTem:COMMunicate:LAN:SELF:IP:DNSServer 172.16.2.2
```

The following query returns 172.16.2.2.

```
:SYSTem:COMMunicate:LAN:SELF:IP:DNSPreferred?
```

```
:SYSTem:COMMunicate:LAN:SELF:IP:DNSServer?
```

**:SYSTem:COMMunicate:LAN[:SELF]:IP:GATeway****Syntax**

```
:SYSTem:COMMunicate:LAN[:SELF]:IP:GATeway <ip_address>
```

```
:SYSTem:COMMunicate:LAN[:SELF]:IP:GATeway?
```

**Description**

Sets the default gateway.

Queries the default gateway.

**Parameter**

| Name         | Type         | Range                       | Default |
|--------------|--------------|-----------------------------|---------|
| <ip_address> | ASCII String | Refer to " <b>Remarks</b> " | —       |

**Remarks**

The format of <ip\_address> is "nnn.nnn.nnn.nnn". The range for the first segment (nnn) of the address is from 1 to 223 (except 127); and the range for the other three segments is from 0 to 255.

This command is only valid in manual IP setting mode.

**Return Format**

The query returns the current default gateway in the format of "nnn.nnn.nnn.nnn".

**Example**

The following command sets the default gateway to "172.16.3.1".

```
:SYSTem:COMMunicate:LAN:SELF:IP:GATeway 172.16.3.1
```

The following query returns 172.16.3.1.

```
:SYSTem:COMMunicate:LAN:SELF:IP:GATeway?
```

**:SYSTem:COMMunicate:LAN[:SELF]:IP:SUBMask****Syntax**

```
:SYSTem:COMMunicate:LAN[:SELF]:IP:SUBMask<ip_address>
```

```
:SYSTem:COMMunicate:LAN[:SELF]:IP:SUBMask?
```

**Description**

Sets the subnet mask.

Queries the subnet mask.

**Parameter**

| Name         | Type         | Range                       | Default |
|--------------|--------------|-----------------------------|---------|
| <ip_address> | ASCII String | Refer to " <b>Remarks</b> " | —       |

**Remarks**

The format of <ip\_address> is "nnn.nnn.nnn.nnn". The range for the first segment (nnn) is from 0 to 255. This command is only valid in manual IP setting mode.

**Return Format**

The query returns the current subnet mask in the format of "nnn.nnn.nnn.nnn".

**Example**

The following command sets the subnet mask to "255.255.255.0".

```
:SYSTem:COMMunicate:LAN:SELF:IP:SUBMask 255.255.255.0
```

The following query returns 255.255.255.0.

```
:SYSTem:COMMunicate:LAN:SELF:IP:SUBMask?
```

**:SYSTem:COMMunicate:LAN[:SELF]:MANuip:STATe****Syntax**

```
:SYSTem:COMMunicate:LAN[:SELF]:MANuip:STATe OFF|ON|0|1
```

```
:SYSTem:COMMunicate:LAN[:SELF]:MANuip:STATe?
```

**Description**

Enables or disables the manual IP setting mode.

Queries the status of the manual IP setting mode.

**Parameter**

| Name | Type | Range      | Default |
|------|------|------------|---------|
| —    | Bool | OFF ON 0 1 | OFF 0   |

**Remarks**

The analyzer attempts to acquire the IP address configuration according to the sequence of DHCP, Auto IP, and Manual IP. These three modes cannot be disabled at the same time.

ON|1: enables the manual IP mode.

OFF|0: disables the manual IP mode and selects the DHCP mode.

**Return Format**

The query returns 0 or 1.

**Example**

The following command enables the manual IP setting mode.

```
:SYSTem:COMMunicate:LAN:SELF:MANuip:STATe ON or :SYSTem:COMMunicate:LAN:SELF:MANuip:STATe 1
```

The following query returns 1.

```
:SYSTem:COMMunicate:LAN:SELF:MANuip:STATe?
```

## :SYSTem:COMMunicate:LAN[:SELF]:MDNS:STATe

### Syntax

:SYSTem:COMMunicate:LAN[:SELF]:MDNS:STATe OFF|ON|0|1  
:SYSTem:COMMunicate:LAN[:SELF]:MDNS:STATe?

### Description

Enables or disables the state of the network information sending.  
Queries the on/off status of the network information sending.

### Parameter

| Name | Type | Range      | Default |
|------|------|------------|---------|
| —    | Bool | OFF ON 0 1 | OFF 0   |

### Remarks

ON|1: The state of the network information sending is turned on.  
ON|0: The state of the network information sending is turned off.

### Return Format

The query returns 0 or 1.

### Example

The following command sets the state of the network information sending to "on".

:SYSTem:COMMunicate:LAN:SELF:MDNS:STATe ON or :SYSTem:COMMunicate:LAN:SELF:MDNS:STATe 1

The following query returns 1.

:SYSTem:COMMunicate:LAN:SELF:MDNS:STATe?

## :SYSTem:COMMunicate:LAN[:SELF]:RESet

### Syntax

:SYSTem:COMMunicate:LAN[:SELF]:RESet

### Description

Resets the LAN setting: enable the DHCP, disable Auto IP and the Manual IP.

## :SYSTem:CONFigure:INformation?

### Syntax

:SYSTem:CONFigure:INformation?

### Description

Queries the system information of the spectrum analyzer.

### Remarks

The system information includes the model, serial number, software and hardware version number, and firmware version number.

### Return Format

The query returns the system information. For example,

Model:RSA3045

SN:RSA3A200100001

Main Board:00.01.00  
 Keyboard:00.01.00  
 CPU:00.01.00  
 SPU:00.01.02  
 WPU:00.01.00  
 BOOT:00.01.00  
 OS:00.01.00  
 Firmware:00.01.00

Note: When you use RSA3045-TG/RSA3030-TG, the query also returns information about "TG Board".

## :SYSTem:DATE

### Syntax

:SYSTem:DATE <year>,<month>,<day>  
 :SYSTem:DATE?

### Description

Sets the date of the instrument.  
 Queries the date of the instrument.

### Parameter

| Name    | Type         | Range        | Default |
|---------|--------------|--------------|---------|
| <year>  | ASCII String | 2000 to 2099 | ——      |
| <month> | ASCII String | 01 to 12     | ——      |
| <day>   | ASCII String | 01 to 31     | ——      |

### Return Format

The query returns the current date in the format of "YYYY,MM,DD".

### Example

The following command sets the date of the instrument to 2017/11/16.

:SYSTem:DATE 2017,11,16

The following query returns 2017,11,16

:SYSTem:DATE?

## :SYSTem:FSWitch[:STATe]

### Syntax

:SYSTem:FSWitch[:STATe] OFF|ON|0|1  
 :SYSTem:FSWitch[:STATe]?

### Description

Sets the power switch on the front panel to be turned on or off.  
 Queries whether the power switch on the front panel is turned on or off.

### Parameter

| Name | Type | Range      | Default |
|------|------|------------|---------|
| ——   | Bool | OFF ON 0 1 | ON 1    |

**Return Format**

The query returns 0 or 1.

**Example**

The following command sets the power switch on the front panel is turned off.

:SYSTem:FSWitch:STATe OFF or :SYSTem:FSWitch:STATe 0

The following query returns 0.

:SYSTem:FSWitch:STATe?

**:SYSTem:LANGuage****Syntax**

:SYSTem:LANGuage ENGLISH|CHINESE

:SYSTem:LANGuage?

**Description**

Sets the language of the instrument.

Queries the language of the instrument.

**Parameter**

| Name | Type    | Range           | Default |
|------|---------|-----------------|---------|
| —    | Keyword | ENGLISH CHINESE | ENGLISH |

**Return Format**

The query returns ENGL or CHIN.

**Example**

The following command sets the language to English.

:SYSTem:LANGuage ENGLISH

The following query returns ENGL.

:SYSTem:LANGuage?

**:SYSTem:LKEY****Syntax**

:SYSTem:LKEY <option info> @<license info>

**Description**

Installs and activates the specified option.

**Parameter**

| Name           | Type         | Range | Default |
|----------------|--------------|-------|---------|
| <option info>  | ASCII String | —     | —       |
| <license info> | ASCII String | —     | —       |

**Remarks**

The parameter <option info> indicates the order number of the option. <license info> indicates the serial number of the option.

**Example**

The following command installs the option RSA3000-PA.

:SYSTem:LKEY

RSA3000-PA@8AD12B8EBC5DF492D1D4289B7CBA5B6150BF6F5D752D645C36D74530B05F39B49C461B2  
3A50D6C94A34E06782AC4380070B0D1A86BA84E02768391FFD70C2103

**:SYSTem:OPTion:STATe?****Syntax**

:SYSTem:OPTion:STATe? <option name>

**Description**

Queries whether an option is activated or not.

**Parameter**

| Name          | Type         | Range | Default |
|---------------|--------------|-------|---------|
| <option name> | ASCII String | ——    | ——      |

**Return Format**

The query returns 0 (not activated) or 1 (activated).

**Example**

The following command queries whether the option RSA3000-PA is activated.

:SYSTem:OPTion:STATe? RSA3000-PA

**:SYSTem:PON:TYPE****Syntax**

:SYSTem:PON:TYPE PRESet|LAST

:SYSTem:PON:TYPE?

**Description**

Selects the setting type the instrument recalls at power-on.

Queries what setting type the instrument recalls at power-on.

**Parameter**

| Name | Type    | Range       | Default |
|------|---------|-------------|---------|
| ——   | Keyword | PRESet LAST | PRESet  |

**Remarks**

PRESet: indicates preset settings, which includes default factory settings and 6 user settings.

LAST: indicates the last setting.

**Return Format**

The query returns PRES or LAST.

**Example**

The following command sets the instrument to recall the last setting.

:SYSTem:PON:TYPE LAST

The following query returns LAST.

:SYSTem:PON:TYPE?

## :SYSTem:PRESet

### Syntax

:SYSTem:PRESet

### Description

Recalls the system's preset settings, and recovers the system to the state specified by the [:SYSTem:PRESet:TYPE](#) command.

## :SYSTem:PRESet:SAVE

### Syntax

:SYSTem:PRESet:SAVE USER1|USER2|USER3|USER4|USER5|USER6

### Description

Saves the specified user setting.

### Parameter

| Name | Type    | Range                               | Default |
|------|---------|-------------------------------------|---------|
| —    | Keyword | USER1 USER2 USER3 USER4 USER5 USER6 | —       |

## :SYSTem:PRESet:TYPE

### Syntax

:SYSTem:PRESet:TYPE FACTory|USER1|USER2|USER3|USER4|USER5|USER6  
:SYSTem:PRESet:TYPE?

### Description

Selects the preset type of the system to FACTory or any one of the items from User1 to User6. Queries the preset type of the system.

### Parameter

| Name | Type    | Range                                       | Default |
|------|---------|---------------------------------------------|---------|
| —    | Keyword | FACTory USER1 USER2 USER3 USER4 USER5 USER6 | FACTory |

### Return Format

The query returns FACT, USER1, USER2, USER3, USER4, USER5, or USER6.

### Example

The following command sets the preset type of the system to User5.

```
:SYSTem:PRESet:TYPE USER5
```

The following query returns USER5.

```
:SYSTem:PRESet:TYPE?
```

## :SYSTem:PRESet:USER:SAVE

### Syntax

:SYSTem:PRESet:USER:SAVE

### Description

Saves the current user setting.

### Remarks

If the current preset type is "FACTory", save the current user setting to User1. If the current preset type is "User1 through User6", save the current user setting to the specified user.

## :SYSTem:SCPI:DISPlay

### Syntax

:SYSTem:SCPI:DISPlay OFF|ON|0|1

:SYSTem:SCPI:DISPlay?

### Description

Enables or disables the SCPI display.

Queries the status for the SCPI display.

### Parameter

| Name | Type | Range      | Default |
|------|------|------------|---------|
| —    | Bool | OFF ON 0 1 | OFF 0   |

### Remarks

When you use the SCPI commands to control the instrument remotely, turning on the SCPI display will go to the specified menu of the corresponding command; turning off the SCPI Display will not go to the specified menu.

### Return Format

The query returns 0 or 1.

### Example

The following command enables the SCPI display.

:SYSTem:SCPI:DISPlay 1 or :SYSTem:SCPI:DISPlay ON

The following query returns 1.

:SYSTem:SCPI:DISPlay?

## :SYSTem:SHOW

### Syntax

:SYSTem:SHOW OFF|SYSTem|OPTion|LICense

:SYSTem:SHOW?

### Description

Displays the system-related information.

Queries the system-related information.

**Parameter**

| Name | Type    | Range                     | Default |
|------|---------|---------------------------|---------|
| —    | Keyword | OFF SYSTem OPTion LICense | OFF     |

**Remarks**

OFF: turns off the system information display.

SYSTem: displays the system information.

OPTion: displays the option information.

LICense: displays the registration information.

**Return Format**

The query returns OFF, SYST, OPT, or LIC.

**Example**

The following command displays the system information.

```
:SYSTem:SHOW SYSTem
```

The following query returns SYST.

```
:SYSTem:SHOW?
```

**:SYSTem:TIME****Syntax**

```
:SYSTem:TIME <hour>,<minute>,<second>
```

```
:SYSTem:TIME?
```

**Description**

Sets the system time of the instrument.

Queries the system time of the instrument.

**Parameter**

| Name     | Type         | Range    | Default |
|----------|--------------|----------|---------|
| <hour>   | ASCII String | 00 to 23 | —       |
| <minute> | ASCII String | 00 to 59 | —       |
| <second> | ASCII String | 00 to 59 | —       |

**Return Format**

The query returns the current system time in the format of "HH,MM,SS".

**Example**

The following command sets the system time to "15:10:30".

```
:SYSTem:TIME 15,10,30
```

The following query returns 15,10,30.

```
:SYSTem:TIME?
```

## :TRACe Commands

### Command List:

- ◆ [:TRACe:CLEAr:ALL](#)
- ◆ [:TRACe\[:DATA\]](#)
- ◆ [:TRACe:MATH:A](#)
- ◆ [:TRACe:MATH:B](#)
- ◆ [:TRACe:MATH:CONSt](#)
- ◆ [:TRACe:MATH:PEAK\[:DATA\]?](#)
- ◆ [:TRACe:MATH:PEAK:POINts?](#)
- ◆ [:TRACe:MATH:REFerence](#)
- ◆ [:TRACe:MATH:STATe](#)
- ◆ [:TRACe:MATH:TYPE](#)
- ◆ [:TRACe:PRESet:ALL](#)
- ◆ [:TRACe<n>:DISPlay:STATe](#)
- ◆ [:TRACe<n>:MODE](#)
- ◆ [:TRACe<n>:TYPE](#)
- ◆ [:TRACe<n>:UPDate:STATe](#)

## :TRACe:CLEAr:ALL

### Syntax

:TRACe:CLEAr:ALL

### Description

Clears all the traces, that is, all the points on the trace will be reset to the minimum value of the trace.

## :TRACe[:DATA]

### Syntax

:TRACe[:DATA]  
 TRACE1|TRACE2|TRACE3|TRACE4|TRACE5|TRACE6,<definite\_length\_block>|<comma\_separated\_ASCII\_data>  
 :TRACe[:DATA]?TRACE1|TRACE2|TRACE3|TRACE4|TRACE5|TRACE6

### Description

Load the user data into the specified trace.  
 Query the logarithmic data of the specified trace.

### Parameter

| Name                         | Type         | Range                                     | Default |
|------------------------------|--------------|-------------------------------------------|---------|
| --                           | Keyword      | TRACE1 TRACE2 TRACE3 TRACE4 TRACE5 TRACE6 | --      |
| <definite_length_block>      | ASCII String | --                                        | --      |
| <comma_separated_ASCII_data> | ASCII String | --                                        | --      |



[2] In the two examples here, 801 points are transmitted and the queries return 801 points. However, due to the limited space, the succeeding data is omitted here. The return values in the above examples are acquired when the trace stops sweeping.

## :TRACe:MATH:A

### Syntax

```
:TRACe:MATH:A T1|T2|T3|T4|T5|T6
```

```
:TRACe:MATH:A?
```

### Description

Sets Op1 in the trace math operation formula to Trace 1, Trace 2, Trace 3, Trace 4, Trace 5, or Trace 6. Queries which trace is Op1 represented in the trace math operation formula.

### Parameter

| Name | Type    | Range             | Default |
|------|---------|-------------------|---------|
| —    | Keyword | T1 T2 T3 T4 T5 T6 | T1      |

### Return Format

The query returns T1, T2, T3, T4, T5, or T6.

### Example

The following command sets Op1 to Trace 2.

```
:TRACe:MATH:A T2
```

The following query returns T2.

```
:TRACe:MATH:A?
```

## :TRACe:MATH:B

### Syntax

```
:TRACe:MATH:B T1|T2|T3|T4|T5|T6
```

```
:TRACe:MATH:B?
```

### Description

Sets Op2 in the trace math operation formula to Trace 1, Trace 2, Trace 3, Trace 4, Trace 5, or Trace 6. Queries which trace is Op2 represented in the trace math operation formula.

### Parameter

| Name | Type    | Range             | Default |
|------|---------|-------------------|---------|
| —    | Keyword | T1 T2 T3 T4 T5 T6 | T2      |

### Return Format

The query returns T1, T2, T3, T4, T5, or T6.

### Example

The following command sets Op2 to Trace 1.

```
:TRACe:MATH:B T1
```

The following query returns T1.

```
:TRACe:MATH:B?
```

## :TRACe:MATH:CONST

### Syntax

:TRACe:MATH:CONST <integer>  
:TRACe:MATH:CONST?

### Description

Sets the log offset in the trace math operation formula.  
Queries the log offset in the trace math operation formula.

### Parameter

| Name      | Type                    | Range             | Default |
|-----------|-------------------------|-------------------|---------|
| <integer> | Consecutive Real Number | -100 dB to 100 dB | 0 dB    |

### Return Format

The query returns the log offset in scientific notation.

### Example

The following command sets the log offset to 50 dB.  
:TRACe:MATH:CONST 50

The following query returns 5.000000000e+01.  
:TRACe:MATH:CONST?

## :TRACe:MATH:PEAK[:DATA]?

### Syntax

:TRACe:MATH:PEAK[:DATA]?

### Description

Queries the frequencies (Hz) and amplitudes (the unit is the same as the current Y-axis unit) of the peaks in the peak table.

### Remarks

This command only queries Trace 1.

### Return Format

The query returns the frequencies and amplitudes of the peaks in the peak table in the following format. At most, 20 pairs of frequencies and amplitudes can be returned.

For example, 1.000000000e+09,-1.463000000e+01,9.999890000e+08,-4.172000000e+01.

## :TRACe:MATH:PEAK:POINts?

### Syntax

:TRACe:MATH:PEAK:POINts?

### Description

Query the number of peaks in the peak table.

### Return Format

The query returns an integer ranging from 0 to 20.

## :TRACe:MATH:REFeRence

### Syntax

:TRACe:MATH:REFeRence < integer >

:TRACe:MATH:REFeRence?

### Description

Sets the log reference in the trace math operation formula.

Queries the log reference in the trace math operation formula.

### Parameter

| Name        | Type                    | Range              | Default |
|-------------|-------------------------|--------------------|---------|
| < integer > | Consecutive Real Number | -170 dBm to 30 dBm | 0 dBm   |

### Return Format

The query returns the log reference in scientific notation.

### Example

The following command sets the log reference to 50 dBm.

:TRACe:MATH:REFeRence 50

The following query returns 5.000000000e+ 01.

:TRACe:MATH:REFeRence?

## :TRACe:MATH:STATe

### Syntax

:TRACe:MATH:STATe OFF|ON|0|1

:TRACe:MATH:STATe?

### Description

Enables or disables the math operation.

Queries the status of the math operation of the trace.

### Parameter

| Name | Type | Range      | Default |
|------|------|------------|---------|
| —    | Bool | OFF ON 0 1 | OFF 0   |

### Return Format

The query returns 0 or 1.

### Example

The following command enables the math operation of the trace.

:TRACe:MATH:STATe 1 or :TRACe:MATH:STATe ON

The following query returns 1.

:TRACe:MATH:STATe?

## :TRACe:MATH:TYPE

### Syntax

:TRACe:MATH:TYPE A+ B| A-B| A+ CONST| A-CONST| A-B+ REF  
:TRACe:MATH:TYPE?

### Description

Sets the operation type of the trace.

Queries the operation type of the trace.

### Parameter

| Name | Type    | Range                                  | Default |
|------|---------|----------------------------------------|---------|
| —    | Keyword | A+ B  A-B  A+ CONST  A-CONST  A-B+ REF | A+ B    |

### Remarks

The parameter A+ B, A-B, A+ CONST, A-CONST, and A-B+ REF corresponds to Op1-Op2, Op1+ Op2, Op1+ Offset, Op1-Offset, and Op1-Op2+ Ref, respectively

### Return Format

The query returns A+ B, A-B, A+ CONST, A-CONST, or A-B+ REF.

### Example

The following command sets the operation type of the trace to A+ B.

```
:TRACe:MATH:TYPE A+ B
```

The following query returns A+ B.

```
:TRACe:MATH:TYPE?
```

## :TRACe:PRESet:ALL

### Syntax

```
:TRACe:PRESet:ALL
```

### Description

Resets all the traces.

### Remarks

Enables Trace 1, and sets other traces to be in blank state. This does not affect the trace type and the detector type.

## :TRACe< n> :DISPlay:STATe

### Syntax

```
:TRACe< n> :DISPlay[:STATe] OFF|ON|0|1
:TRACe< n> :DISPlay[:STATe]?
```

### Description

Enables or disables the display of the specified trace.  
Queries the status for the specified trace.

### Parameter

| Name | Type     | Range       | Default            |
|------|----------|-------------|--------------------|
| < n> | Discrete | 1 2 3 4 5 6 | —                  |
| —    | Bool     | OFF ON 0 1  | Refer to "Remarks" |

### Remarks

By default, Trace 1 is enabled (ON), and Trace 2 through Trace 6 are disabled (OFF).

### Return Format

The query returns 1 or 0.

### Example

The following command enables the display of Trace 1.

```
:TRACe1:DISPlay:STATe ON or :TRACe1:DISPlay:STATe 1
```

The following query returns 1.

```
:TRACe1:DISPlay:STATe?
```

## :TRACe< n> :MODE

## :TRACe< n> :TYPE

### Syntax

```
:TRACe< n> :MODE WRITe|AVERAge|MAXHold|MINHold
:TRACe< n> :MODE?
:TRACe< n> :TYPE WRITe|AVERAge|MAXHold|MINHold
:TRACe< n> :TYPE?
```

### Description

Sets the type of the specified trace.  
Queries the type of the specified trace.

### Parameter

| Name | Type     | Range                         | Default |
|------|----------|-------------------------------|---------|
| < n> | Discrete | 1 2 3 4 5 6                   | —       |
| —    | Keyword  | WRITe AVERAge MAXHold MINHold | WRITe   |

### Remarks

WRITe: indicates clear/write.

AVERAge: indicates average.

MAXHold: indicates the maximum hold.

MINHold: indicates the minimum hold.

In the window that contains the Density view, only Trace 1 is available.

In the window that contains the PVT view, only Trace 1 is available and the trace type can only be selected to "Clear Write".

**Return Format**

The query returns WRIT, AVER, MAXH, or MINH.

**Example**

The following command sets the type of Trace 1 to Max Hold.

```
:TRACe1:MODE MAXHold
```

```
:TRACe1:TYPE MAXHold
```

The following query returns MAXH.

```
:TRACe1:MODE?
```

```
:TRACe1:TYPE?
```

**:TRACe< n> :UPDate:STATe****Syntax**

```
:TRACe< n> :UPDate:STATe OFF|ON|0|1
```

```
:TRACe< n> :UPDate:STATe?
```

**Description**

Enables or disables the update of the specified trace.

Queries the status for the update of the specified trace.

**Parameter**

| Name | Type     | Range       | Default                     |
|------|----------|-------------|-----------------------------|
| < n> | Discrete | 1 2 3 4 5 6 | —                           |
| —    | Bool     | OFF ON 0 1  | Refer to " <b>Remarks</b> " |

**Remarks**

By default, Trace 1 is enabled (ON), and Trace 2 thru Trace 6 are disabled (OFF).

**Return Format**

The query returns 1 or 0.

**Example**

The following command enables the update of Trace 1.

```
:TRACe1:UPDate:STATe ON or :TRACe1:UPDate:STATe 1
```

The following query returns 1.

```
:TRACe1:UPDate:STATe?
```

## :TRIGger Commands

### Command List:

- ◆ [:TRIGger\[:SEQuence\]:ATRigger](#)
- ◆ [:TRIGger\[:SEQuence\]:ATRigger:STATe](#)
- ◆ [:TRIGger\[:SEQuence\]:EXTErnal<n>:DELAy](#)
- ◆ [:TRIGger\[:SEQuence\]:EXTErnal<n>:DELAy:STATe](#)
- ◆ [:TRIGger\[:SEQuence\]:EXTErnal<n>:SLOPe](#)
- ◆ [:TRIGger\[:SEQuence\]:FMT:APTRigger](#)
- ◆ [:TRIGger\[:SEQuence\]:FMT:CRITeria](#)
- ◆ [:TRIGger\[:SEQuence\]:FMT:DELAy](#)
- ◆ [:TRIGger\[:SEQuence\]:FMT:DELAy:STATe](#)
- ◆ [:TRIGger\[:SEQuence\]:FMT:MASK](#)
- ◆ [:TRIGger\[:SEQuence\]:FMT:MASK:EDIT](#)
- ◆ [:TRIGger\[:SEQuence\]:FMT:MASK:RELative:AMPLitude](#)
- ◆ [:TRIGger\[:SEQuence\]:FMT:MASK:RELative:FREQuency](#)
- ◆ [:TRIGger\[:SEQuence\]:FMT:MASK<n>:BUILd](#)
- ◆ [:TRIGger\[:SEQuence\]:FMT:MASK<n>:DATA](#)
- ◆ [:TRIGger\[:SEQuence\]:FMT:MASK<n>:DELeTe](#)
- ◆ [:TRIGger\[:SEQuence\]:FMT:MASK<n>:NEW](#)
- ◆ [:TRIGger\[:SEQuence\]:HOLDoff](#)
- ◆ [:TRIGger\[:SEQuence\]:HOLDoff:STATe](#)
- ◆ [:TRIGger\[:SEQuence\]:SOURce](#)
- ◆ [:TRIGger\[:SEQuence\]:VIDeo:DELAy](#)
- ◆ [:TRIGger\[:SEQuence\]:VIDeo:DELAy:STATe](#)
- ◆ [:TRIGger\[:SEQuence\]:VIDeo:LEVel](#)
- ◆ [:TRIGger\[:SEQuence\]:VIDeo:SLOPe](#)
- ◆ [:TRIGger2:MODE](#)

## :TRI Gger[:SEQuence]:ATRigger

### Syntax

```
:TRI Gger[:SEQuence]:ATRigger <time>
:TRI Gger[:SEQuence]:ATRigger?
```

### Description

Sets the time that the analyzer will wait for the trigger to be initiated automatically.  
Queries the time that the analyzer will wait for the trigger to be initiated automatically.

### Parameter

| Name   | Type                    | Range         | Default |
|--------|-------------------------|---------------|---------|
| <time> | Consecutive Real Number | 1 ms to 100 s | 100 ms  |

### Remarks

This command is only valid when the auto triggering function is enabled.

### Return Format

The query returns the time value in scientific notation. The unit is s.

### Example

The following command sets the time to 10 ms.

```
:TRI Gger:SEQuence:ATRigger 0.01
```

The following query returns 1.000000000e-02.

```
:TRI Gger:SEQuence:ATRigger?
```

## :TRI Gger[:SEQuence]:ATRigger:STATe

### Syntax

```
:TRI Gger[:SEQuence]:ATRigger:STATe OFF|ON|0|1
:TRI Gger[:SEQuence]:ATRigger:STATe?
```

### Description

Enables or disables the auto trigger function.  
Queries the setting status of auto trigger function.

### Parameter

| Name | Type | Range      | Default |
|------|------|------------|---------|
| —    | Bool | OFF ON 0 1 | OFF 0   |

### Return Format

The query returns 1 or 0.

### Example

The following command enables the auto trigger function.

```
:TRI Gger:SEQuence:ATRigger:STATe ON or :TRI Gger:SEQuence:ATRigger:STATe 1
```

The following query returns 1.

```
:TRI Gger:SEQuence:ATRigger:STATe?
```

## :TRI Gger[:SEQuence]:EXTErnal< n> :DELay

### Syntax

```
:TRI Gger[:SEQuence]:EXTErnal< n> :DELay <time>
:TRI Gger[:SEQuence]:EXTErnal< n> :DELay?
```

### Description

Sets the delay time for the external trigger.  
Queries the delay time for the external trigger.

### Parameter

| Name   | Type                    | Range               | Default   |
|--------|-------------------------|---------------------|-----------|
| < n>   | Discrete                | 1 2                 | —         |
| <time> | Consecutive Real Number | 0 $\mu$ s to 500 ms | 1 $\mu$ s |

### Remarks

When the parameter n is set to 1, it indicates External Trigger 1; when set to 2, it indicates External Trigger 2.

This command is only valid when the external trigger delay function is enabled.

### Return Format

The query returns the delay time for the external trigger in scientific notation. The unit is s.

### Example

The following command sets the delay time for External Trigger 1 to 100 ms.

```
:TRI Gger:SEQuence:EXTErnal1:DELay 0.1
```

The following query returns 1.000000000e-01.

```
:TRI Gger:SEQuence:EXTErnal1:DELay?
```

## :TRI Gger[:SEQuence]:EXTErnal< n> :DELay:STATe

### Syntax

```
:TRI Gger[:SEQuence]:EXTErnal< n> :DELay:STATe OFF|ON|0|1
:TRI Gger[:SEQuence]:EXTErnal< n> :DELay:STATe?
```

### Description

Enables or disables the external trigger delay function.  
Queries the status of the external trigger delay function.

### Parameter

| Name | Type     | Range      | Default |
|------|----------|------------|---------|
| < n> | Discrete | 1 2        | —       |
| —    | Bool     | OFF ON 0 1 | OFF 0   |

### Remarks

When the parameter n is set to 1, it indicates External Trigger 1; when set to 2, it indicates External Trigger 2.

### Return Format

The query returns 1 or 0.

### Example

The following command enables the delay function of External Trigger 1.

:TRIGger:SEquence:EXternal1:DElay:STATe ON or :TRIGger:SEquence:EXternal1:DElay:STATe 1

The following query returns 1.

:TRIGger:SEquence:EXternal1:DElay:STATe?

## :TRI Gger[:SEquence]:EXternal< n> :SLOPe

### Syntax

:TRI Gger:SEquence:EXternal< n> :SLOPe POSitive|NEGative

:TRI Gger:SEquence:EXternal< n> :SLOPe?

### Description

Sets the trigger edge for the external trigger.

Queries the trigger edge for the external trigger.

### Parameter

| Name | Type     | Range             | Default  |
|------|----------|-------------------|----------|
| < n> | Discrete | 1 2               | —        |
| —    | Keyword  | POSitive NEGative | POSitive |

### Remarks

When the parameter n is set to 1, it indicates External Trigger 1; when set to 2, it indicates External Trigger 2.

POSitive: indicates the rising edge.

NEGative: indicates the falling edge.

### Return Format

The query returns POS or NEG.

### Example

The following command sets the trigger edge of External Trigger 1 to Positive.

:TRI Gger:SEquence:EXternal1:SLOPe POSitive

The following query returns POS.

:TRI Gger:SEquence:EXternal1:SLOPe?

## :TRI Gger[:SEquence]:FMT:APTRigger

### Syntax

:TRI Gger[:SEquence]:FMT:APTRigger < number>

:TRI Gger[:SEquence]:FMT:APTRigger?

### Description

Sets the number of times for acquisition after each effective trigger is completed.

Queries the number of times for acquisition after each effective trigger is completed.

### Parameter

| Name      | Type    | Range              | Default |
|-----------|---------|--------------------|---------|
| < number> | Integer | Refer to "Remarks" | 1       |

### Remarks

The minimum value of the parameter < number> is 1. When the trigger criteria of FMT is "Inside" or

"Outside", the maximum value of the parameter <number> is 1. In the Spectrogram measurement mode, the maximum value of the parameter <number> is 10,000. In the PvT measurement mode, the maximum value of the parameter <number> is 5,000.

This command is only valid when you select FMT to be the trigger source in RTSA mode.

### Return Format

The query returns the number of acquisition times in integer.

### Example

The following command sets the number of acquisition times to 5.

```
:TRIGger:SEquence:FMT:APTRigger 5
```

The following query returns 5.

```
:TRIGger:SEquence:FMT:APTRigger?
```

## :TRIGger[:SEquence]:FMT:CRITeria

### Syntax

```
:TRIGger[:SEquence]:FMT:CRITeria ENTER|LEAVE|INSide|OUTSide|ELEave|LENTer
```

```
:TRIGger[:SEquence]:FMT:CRITeria?
```

### Description

Sets the trigger criteria for FMT.

Queries the trigger criteria for FMT.

### Parameter

| Name | Type    | Range                                    | Default |
|------|---------|------------------------------------------|---------|
| —    | Keyword | ENTER LEAVE INSide OUTSide ELEave LENTer | ENTER   |

### Remarks

This command is only valid when you select FMT to be the trigger source in RTSA mode.

### Return Format

The query returns ENT, LEAV, INS, OUTS, ELE, or LENT.

### Example

The following command sets the trigger criteria for FMT to Enter.

```
:TRIGger:SEquence:FMT:CRITeria ENTER
```

The following query returns ENT.

```
:TRIGger:SEquence:FMT:CRITeria?
```

## :TRI Gger[:SEQuence]:FMT:DELAy

### Syntax

```
:TRI Gger[:SEQuence]:FMT:DELAy <time>
:TRI Gger[:SEQuence]:FMT:DELAy?
```

### Description

Sets the delay time for FMT.

Queries the delay time for FMT.

### Parameter

| Name   | Type                    | Range          | Default   |
|--------|-------------------------|----------------|-----------|
| <time> | Consecutive Real Number | 0 us to 500 ms | 1 $\mu$ s |

### Remarks

When "Inside" or "Outside" is selected to be the trigger criteria, the FMT trigger delay command is invalid. This command is only valid when the following conditions are met: FMT is selected to be the trigger source in RTSA mode; FMT trigger delay function is enabled.

### Return Format

The query returns the delay time for FMT in scientific notation. The unit is s.

### Example

The following command sets the delay time for FMT to 100 ms.

```
:TRI Gger:SEQuence:FMT:DELAy 0.1
```

The following query returns 1.000000000e-01.

```
:TRI Gger:SEQuence:FMT:DELAy?
```

## :TRI Gger[:SEQuence]:FMT:DELAy:STATe

### Syntax

```
:TRI Gger[:SEQuence]:FMT:DELAy:STATe OFF|ON|0|1
:TRI Gger[:SEQuence]:FMT:DELAy:STATe?
```

### Description

Enables or disables the FMT trigger delay function.

Queries the status of the FMT trigger delay function.

### Parameter

| Name | Type | Range      | Default |
|------|------|------------|---------|
| —    | Bool | OFF ON 0 1 | OFF 0   |

### Remarks

This command is only valid when you select FMT to be the trigger source in RTSA mode.

### Return Format

The query returns 1 or 0.

### Example

The following command enables the FMT trigger delay function.

```
:TRI Gger:SEQuence:FMT:DELAy:STATe ON or :TRI Gger:SEQuence:FMT:DELAy:STATe 1
```

The following query returns 1.

```
:TRI Gger:SEQuence:FMT:DELAy:STATe?
```

## :TRI Gger[:SEQuence]:FMT:MASK

### Syntax

```
:TRI Gger[:SEQuence]:FMT:MASK UPPer|LOWer|BOTH
:TRI Gger[:SEQuence]:FMT:MASK?
```

### Description

Sets the mask to be used for the current trigger.  
Queries the mask to be used for the current trigger.

### Parameter

| Name | Type    | Range            | Default |
|------|---------|------------------|---------|
| —    | Keyword | UPPer LOWer BOTH | UPPer   |

### Remarks

This command is only valid when you select FMT to be the trigger source in RTSA mode.

### Return Format

The query returns UPP, LOW, or BOTH.

### Example

The following command sets the mask to be used for the current trigger to Upper.  
:TRI Gger:SEQuence:FMT:MASK UPPer

The following query returns UPP.  
:TRI Gger:SEQuence:FMT:MASK?

## :TRI Gger[:SEQuence]:FMT:MASK:EDI T

### Syntax

```
:TRI Gger[:SEQuence]:FMT:MASK:EDI T UPPer|LOWer
:TRI Gger[:SEQuence]:FMT:MASK:EDI T?
```

### Description

Sets the mask type that is currently viewed/edited.  
Queries the mask type that is currently viewed/edited.

### Parameter

| Name | Type    | Range       | Default |
|------|---------|-------------|---------|
| —    | Keyword | UPPer LOWer | UPPer   |

### Remarks

This command is only valid when you select FMT to be the trigger source in RTSA mode.

### Return Format

The query returns UPP or LOW.

### Example

The following command sets the mask that is currently viewed/edited to Upper.  
:TRI Gger:SEQuence:FMT:MASK:EDIT UPPer

The following query returns UPP.  
:TRI Gger:SEQuence:FMT:MASK:EDIT?

## :TRI Gger[:SEquence]:FMT:MASK:RELative:AMPLitude

### Syntax

```
:TRI Gger[:SEquence]:FMT:MASK:RELative:AMPLitude ON|OFF|1|0
:TRI Gger[:SEquence]:FMT:MASK:RELative:AMPLitude?
```

### Description

Sets whether the amplitudes of the mask points are coupled to the reference level of the instrument.  
Queries whether the amplitudes of the mask points are coupled to the reference level of the instrument.

### Parameter

| Name | Type | Range      | Default |
|------|------|------------|---------|
| —    | Bool | OFF ON 0 1 | OFF 0   |

### Remarks

OFF|0: disables the coupling function. That is, when "Fixed" is selected under **Y Axis Type**, the amplitude of the current mask point will not be affected by the reference level.

ON|1: enables the coupling function. That is, when "Relative" is selected under **Y Axis Type**, the amplitude of the current mask point is the difference between the point and the current reference level.

If you modify the state of this parameter after the mask has been set, the amplitude of the mask should make corresponding changes based on the state of the parameter, so as to keep the relations between the mask and the reference level of the instrument unchanged.

When on, the amplitude of the mask point is expressed as an offset from the reference level.

This command is only valid when you select FMT to be the trigger source in RTSA mode.

### Return Format

The query returns 1 or 0.

### Example

The following command enables the coupling relationship between the amplitudes of the mask points and the instrument's reference level.

```
:TRI Gger:SEquence:FMT:MASK:RELative:AMPLitude ON
or :TRI Gger:SEquence:FMT:MASK:RELative:AMPLitude 1
```

The following query returns 1.

```
:TRI Gger:SEquence:FMT:MASK:RELative:AMPLitude?
```

## :TRI Gger[:SEquence]:FMT:MASK:RELative:FREQuency

### Syntax

```
:TRI Gger[:SEquence]:FMT:MASK:RELative:FREQuency ON|OFF|1|0
:TRI Gger[:SEquence]:FMT:MASK:RELative:FREQuency?
```

### Description

Sets whether the frequencies of the mask points are coupled to the center frequency of the instrument.  
Queries whether the frequencies of the mask points are coupled to the center frequency of the instrument.

### Parameter

| Name | Type | Range      | Default |
|------|------|------------|---------|
| —    | Bool | OFF ON 0 1 | OFF 0   |

### Remarks

OFF|0: disables the coupling function. That is, when "Fixed" is selected under **X Axis Type**, the frequency of the current mask point will not be affected by the center frequency.

ON| 1: enables the coupling function. That is, when "Relative" is selected under **X Axis Type**, the frequency of the current mask point is the difference between the point and the current center frequency.

If you modify the state of this parameter after the mask has been set, the frequency of the mask should make corresponding changes based on the state of the parameter, so as to keep the relations between the mask and the center frequency of the instrument unchanged.

When on, the frequencies of the mask points are expressed as an offset from the center frequency. At this time, the frequencies of the mask points can be negative values.

This command is only valid when you select FMT to be the trigger source in RTSA mode.

### Return Format

The query returns 1 or 0.

### Example

The following command enables the coupling relationship between the frequencies of the mask points and the instrument's center frequency.

```
:TRIGger:SEquence:FMT:MASK:RELative:FREQuency ON
```

```
or :TRIGger:SEquence:FMT:MASK:RELative:FREQuency 1
```

The following query returns 1.

```
:TRIGger:SEquence:FMT:MASK:RELative:FREQuency?
```

## :TRIGger[:SEQuence]:FMT:MASK< n> :BUI Ld

### Syntax

```
:TRIGger[:SEQuence]:FMT:MASK< n> :BUI Ld TRACE1|TRACE2|TRACE3|TRACE4|TRACE5|TRACE6
```

### Description

Creates a mask from a trace.

### Parameter

| Name | Type     | Range                                     | Default |
|------|----------|-------------------------------------------|---------|
| < n> | Discrete | 1 2                                       | ——      |
| ——   | Keyword  | TRACE1 TRACE2 TRACE3 TRACE4 TRACE5 TRACE6 | ——      |

### Remarks

When < n> is set to 1, it indicates that the upper mask is being edited; when it is set to 2, it indicates that the lower mask is being edited.

This command is only valid when you select FMT to be the trigger source in RTSA mode.

### Example

The following command creates a upper mask from trace1.

```
:TRIGger:SEquence:FMT:MASK1:BUI Ld TRACE1
```

## :TRIGger[:SEQuence]:FMT:MASK< n> :DATA

### Syntax

```
:TRIGger[:SEQuence]:FMT:MASK< n> :DATA < freq> ,< ampl> { ,< freq> , < ampl> }
```

```
:TRIGger[:SEQuence]:FMT:MASK< n> :DATA?
```

### Description

Edits the mask parameters.

Queries the mask parameters.

**Parameter**

| Name   | Type                    | Range                   | Default |
|--------|-------------------------|-------------------------|---------|
| <n>    | Discrete                | 1 2                     | —       |
| <freq> | Consecutive Real Number | 0 Hz to 4.5 GHz         |         |
| <ampl> | Consecutive Real Number | -1,000 dBm to 1,000 dBm |         |

**Remarks**

When <n> is set to 1, it indicates that the upper mask is being edited; when it is set to 2, it indicates that the lower mask is being edited.

This command is only valid when you select FMT to be the trigger source in RTSA working mode.

**Return Format**

The query returns the mask parameters that are being edited currently in scientific notation.

**Example**

The following command edits the two points of the upper mask.

```
:TRIGger:SEquence:FMT:MASK1:DATA -80000000, -50, 80000000, -50
```

The following query returns -8.000000e+07, -5.000000e+01, 8.000000e+07, -5.000000e+01

```
:TRIGger:SEquence:FMT:MASK1:DATA?
```

**:TRIGger[:SEquence]:FMT:MASK<n>:DELeTe****Syntax**

```
:TRIGger[:SEquence]:FMT:MASK<n>:DELeTe
```

**Description**

Deletes all the masks.

**Parameter**

| Name | Type     | Range | Default |
|------|----------|-------|---------|
| <n>  | Discrete | 1 2   | —       |

**Remarks**

When <n> is set to 1, edit the upper mask; when set to 2, edit the lower mask.

This command is only valid when you select FMT to be the trigger source in RTSA mode.

**:TRIGger[:SEquence]:FMT:MASK<n>:NEw****Syntax**

```
:TRIGger[:SEquence]:FMT:MASK<n>:NEw
```

**Description**

Clears the currently activated mask and creates a default new mask.

**Parameter**

| Name | Type     | Range | Default |
|------|----------|-------|---------|
| <n>  | Discrete | 1 2   | —       |

**Remarks**

When <n> is set to 1, it indicates that the upper mask is being edited; when it is set to 2, it indicates that

the lower mask is being edited.

This command is only valid when you select FMT to be the trigger source in RTSA working mode.

## :TRIGger[:SEQuence]:HOLDoff

### Syntax

:TRIGger[:SEQuence]:HOLDoff <time>

:TRIGger[:SEQuence]:HOLDoff?

### Description

Sets the trigger holdoff time.

Queries the trigger holdoff time.

### Parameter

| Name   | Type                    | Range                                          | Default |
|--------|-------------------------|------------------------------------------------|---------|
| <time> | Consecutive Real Number | 100 us to 500 ms (GPSA)<br>0 us to 10 s (RTSA) | 100 ms  |

### Remarks

This command is only valid when the trigger holdoff function is enabled.

### Return Format

The query returns the trigger holdoff time in scientific notation. The unit is s.

### Example

The following command sets the trigger holdoff time to 100 ms.

```
:TRIGger:SEQuence:HOLDoff 0.1
```

The following query returns 1.000000000e-01.

```
:TRIGger:SEQuence:HOLDoff?
```

## :TRIGger[:SEQuence]:HOLDoff:STATe

### Syntax

:TRIGger[:SEQuence]:HOLDoff:STATe OFF|ON|0|1

:TRIGger[:SEQuence]:HOLDoff:STATe?

### Description

Enables or disables the trigger holdoff function.

Queries the status of the trigger holdoff function.

### Parameter

| Name | Type | Range      | Default |
|------|------|------------|---------|
| —    | Bool | OFF ON 0 1 | OFF 0   |

### Return Format

The query returns 1 or 0.

### Example

The following command enables the trigger holdoff function.

```
:TRIGger:SEQuence:HOLDoff:STATe ON or :TRIGger:SEQuence:HOLDoff:STATe 1
```

The following query returns 1.  
 :TRIGger:SEquence:HOLDoff:STATe?

## :TRIGger[:SEquence]:SOURce

### Syntax

:TRIGger[:SEquence]:SOURce EXTernal1|EXTernal2|IMMEDIATE|VIDeo|FMT|POWer  
 :TRIGger[:SEquence]:SOURce?

### Description

Sets the trigger source.  
 Queries the trigger source.

### Parameter

| Name | Type    | Range                                         | Default   |
|------|---------|-----------------------------------------------|-----------|
| —    | Keyword | EXTernal1 EXTernal2 IMMEDIATE VIDeo FMT POWer | IMMEDIATE |

### Remarks

EXTernal1: indicates External Trigger 1.  
 EXTernal2: indicates External Trigger 2.  
 IMMEDIATE: indicates the free-run trigger.  
 VIDeo: indicates the video trigger. This command is only available for GPSA mode.  
 FMT: indicates the frequency mask trigger. This command is only available for RTSA mode.  
 Power: indicates the IF power trigger. This command is only available for RTSA mode.

### Return Format

The query returns EXT1, EXT2, IMM, VID, FMT, or POW.

### Example

The following command sets the trigger source to free-run trigger.  
 :TRIGger:SEquence:SOURce IMMEDIATE

The following query returns IMM.  
 :TRIGger:SEquence:SOURce?

## :TRIGger[:SEquence]:VIDeo:DElay

### Syntax

:TRIGger[:SEquence]:VIDeo:DElay <time>  
 :TRIGger[:SEquence]:VIDeo:DElay?

### Description

Sets the delay time for the video trigger.  
 Queries the delay time for the video trigger.

### Parameter

| Name   | Type                    | Range               | Default   |
|--------|-------------------------|---------------------|-----------|
| <time> | Consecutive Real Number | 0 $\mu$ s to 500 ms | 1 $\mu$ s |

### Remarks

This command is only valid when the video trigger delay function is enabled in GPSA mode.

**Return Format**

The query returns the delay time for video trigger in scientific notation. The unit is s.

**Example**

The following command sets the delay time for the video trigger to 100 ms.

```
:TRIGger:SEquence:VIDeo:DElay 0.1
```

The following query returns 1.000000e-01.

```
:TRIGger:SEquence:VIDeo:DElay?
```

**:TRIGger[:SEquence]:VIDeo:DElay:STATe****Syntax**

```
:TRIGger[:SEquence]:VIDeo:DElay:STATe OFF|ON|0|1
```

```
:TRIGger[:SEquence]:VIDeo:DElay:STATe?
```

**Description**

Enables or disables the video trigger delay function.

Queries the status of the video trigger delay function.

**Parameter**

| Name | Type | Range      | Default |
|------|------|------------|---------|
| —    | Bool | OFF ON 0 1 | OFF 0   |

**Remarks**

This command is only available for GPSA mode.

**Return Format**

The query returns 1 or 0.

**Example**

The following command enables the delay function for the video trigger.

```
:TRIGger:SEquence:VIDeo:DElay:STATe ON or :TRIGger:SEquence:VIDeo:DElay:STATe 1
```

The following query returns 1.

```
:TRIGger:SEquence:VIDeo:DElay:STATe?
```

**:TRIGger[:SEquence]:VIDeo:LEVel****Syntax**

```
:TRIGger:SEquence:VIDeo:LEVel <ampl>
```

```
:TRIGger:SEquence:VIDeo:LEVel?
```

**Description**

Sets the trigger level of the video trigger.

Queries the trigger level of the video trigger.

**Parameter**

| Name   | Type                    | Range              | Default |
|--------|-------------------------|--------------------|---------|
| <ampl> | Consecutive Real Number | -140 dBm to 30 dBm | -25 dBm |

**Remarks**

This command is only valid when the following conditions are met: in GPSA mode; selecting the video trigger.

**Return Format**

The query returns the trigger level in scientific notation.

**Example**

The following command sets the trigger level of the video trigger to 10 dBm.

:TRIGger:SEquence:VIDeo:LEVel 10

The following query returns 1.000000e+ 01.

:TRIGger:SEquence:VIDeo:LEVel?

**:TRI Gger[:SEquence]:VI Deo:SLOPe****Syntax**

:TRI Gger[:SEquence]:VI Deo:SLOPe POSitive|NEGative

:TRI Gger[:SEquence]:VI Deo:SLOPe?

**Description**

Sets the polarity of the video trigger.

Queries the polarity of the video trigger.

**Parameter**

| Name | Type    | Range             | Default  |
|------|---------|-------------------|----------|
| —    | Keyword | POSitive NEGative | POSitive |

**Remarks**

POSitive: indicates the positive slew rate.

NEGative: indicates the negative slew rate.

This command is only available for GPSA mode.

**Return Format**

The query returns POS or NEG.

**Example**

The following command sets the polarity of the video trigger to Positive.

:TRIGger:SEquence:VIDeo:SLOPe POSitive

The following query returns POS.

:TRIGger:SEquence:VIDeo:SLOPe?

## :TRI Gger2:MODE

### Syntax

:TRI Gger2:MODE IN|OUT|0|1

:TRI Gger2:MODE?

### Description

Sets the interface type of External Trigger 2.

Queries the interface type of External Trigger 2.

### Parameter

| Name | Type    | Range      | Default |
|------|---------|------------|---------|
| ——   | Keyword | IN OUT 0 1 | IN 0    |

### Remarks

IN|0: indicates the input interface.

OUT|1: indicates the output interface.

### Return Format

The query returns 0 or 1.

### Example

The following command sets External Trigger 2 to the input interface.

:TRI Gger2:MODE IN

The following query returns 0.

:TRI Gger2:MODE?

## :UNIT Command

### Command List:

◆ [:UNIT:POWer](#)

## :UNIT:POWer

### Syntax

:UNIT:POWer DBM|DBMV|DBUV|V|W

:UNIT:POWer?

### Description

Sets the unit of the Y axis.

Queries the unit of the Y axis.

### Parameter

| Name | Type    | Range             | Default                     |
|------|---------|-------------------|-----------------------------|
| —    | Keyword | DBM DBMV DBUV V W | Refer to " <b>Remarks</b> " |

### Remarks

The default log unit is dBm.

The default linear unit is V.

### Return Format

The query returns DBM, DBMV, DBUV, V, or W.

### Example

The following command sets the amplitude unit to Watts.

:UNIT:POWer W

The following query returns W.

:UNIT:POWer?



## Chapter 3 Programming Examples

This chapter lists some programming examples to illustrate how to use commands to realize the common functions of the spectrum analyzer in the development environments such as Visual C++ 6.0, Visual Basic 6.0, and LabVIEW 2010. Also, the chapter lists some examples to illustrate how to control the spectrum analyzer to realize the common functions in Linux operating system. These examples are programmed based on NI-VISA library.

NI-VISA (National Instrument-Virtual Instrument Software Architecture), developed by NI (National Instrument), provides an advanced programming interface to communicate with various instruments through their bus lines. NI-VISA enables you to realize the communication between the analyzer and PC through instrument buses (such as USB). VISA defines a set of software commands with which users can control the instrument without the need to understand how the interface bus works. For details, please refer to the NI-VISA Help.

### Contents in this chapter:

- ◆ [Programming Instructions](#)
- ◆ [Programming Preparations](#)
- ◆ [Visual C++ 6.0 Programming Example](#)
- ◆ [Visual Basic 6.0 Programming Example](#)
- ◆ [LabVIEW 2010 Programming Example](#)
- ◆ [Linux Programming Example](#)

## Programming Instructions

This section introduces the problems that might occur during the programming process as well as their solutions. If these problems occur, please resolve them according to the corresponding instructions.

1. When you build a working environment via the network, it is recommended that you build a pure local area network.
2. If the local area network environment is complicated (e.g. many devices and broadcast messages exist), it is recommended that you add some fault tolerance during the programming process. For the details, refer to the "InstrWriteEx()" and "InstrReadEx()" for instrument write/read operations with exception handling functions in "[Visual C++ 6.0 Programming Example](#)".
3. The socket programming port No. of this device is 5555.

## Programming Preparations

The programming preparations introduced here are only applicable to programming by using Visual C++ 6.0, Visual Basic 6.0, and LabVIEW 2010 development tools in Windows operating system. For the preparations of programming in Linux operating system, refer to "[Programming Preparations](#)" in "[Linux Programming Example](#)".

First, check whether your PC has installed NI's VISA library. If not, download it from <http://www.ni.com/visa/>. In this manual, the default installation path is C:\Program Files\IVI Foundation\VISA.

Connect spectrum analyzer to the PC via the USB interface of the analyzer. Use the USB cable to connect the analyzer to the PC via the USB DEVICE interface on the rear panel of the analyzer.

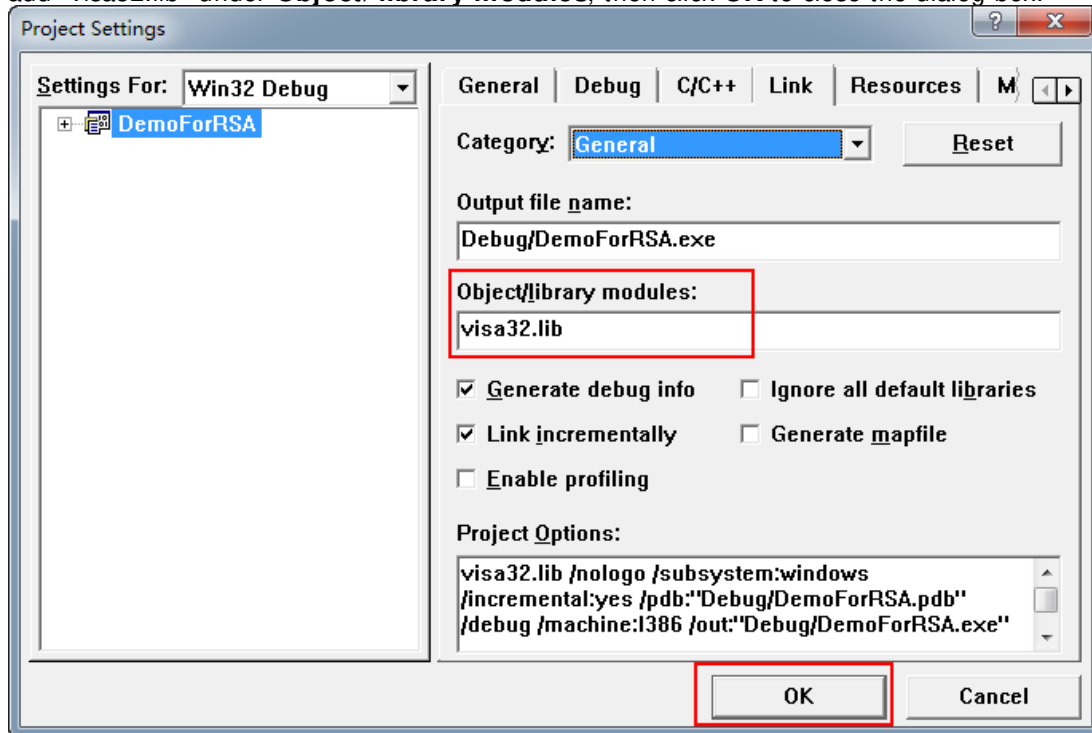
After the analyzer is connected to the PC properly, start the analyzer. In this case, "Found New Hardware Wizard" dialog box appears on the PC. Please install "USB Test and Measurement Device (IVI)" according to the wizard.

By now, the programming preparations are complete. The following parts will make a detailed introduction about the programming instances in the Visual C++ 6.0, Visual Basic 6.0, and LabVIEW 2010 development environment.

## Visual C++ 6.0 Programming Example

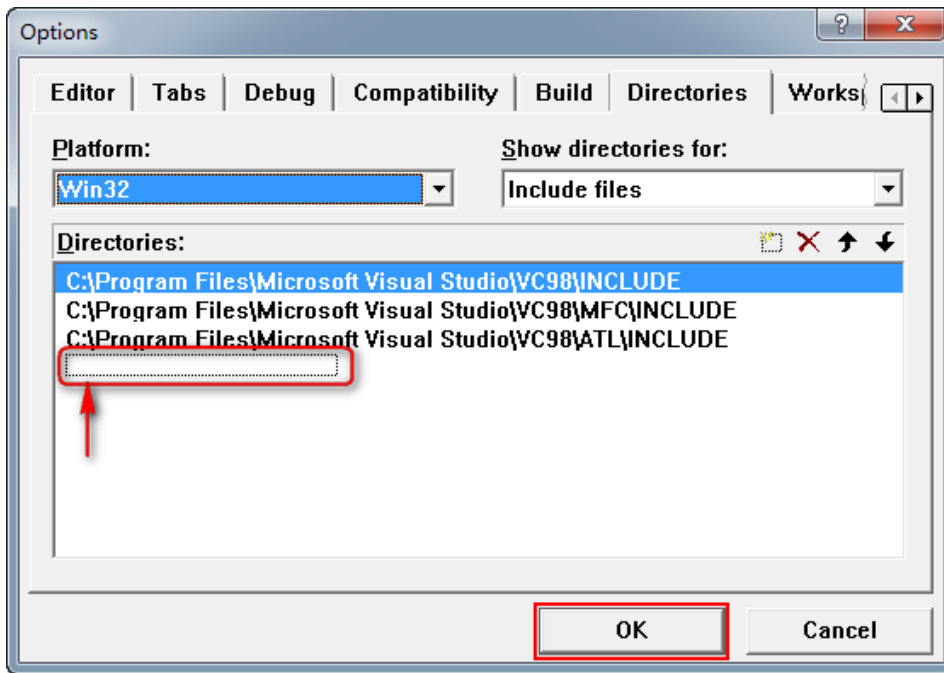
Enter the Visual C++ 6.0 programming environment, and perform the following procedures.

1. Create a MFC project based on a dialog box and name it "DemoForRSA" in this example.
2. Click **Project** → **Settings** to open the "Project Setting" dialog box. In the dialog box, click the **Link** tab, add "visa32.lib" under **Object/ library modules**, then click **OK** to close the dialog box.



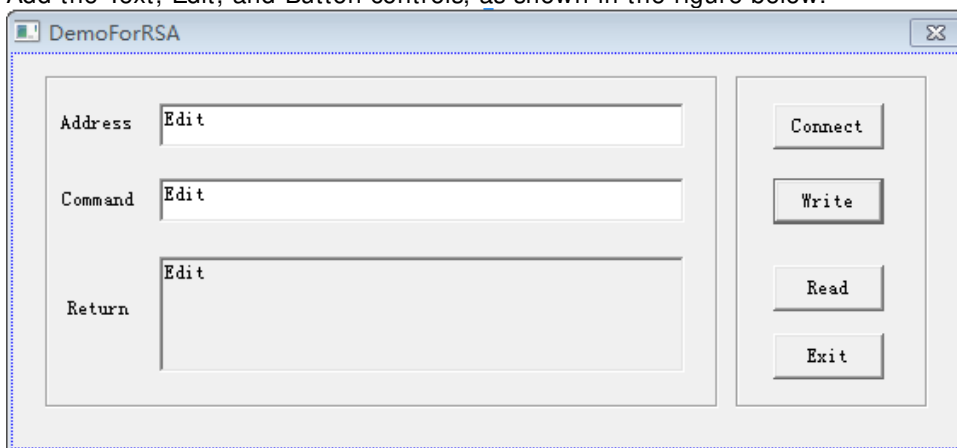
3. Click **Tools** → **Options** to open the "Options" dialog box. Then, click the **Directories** tab. Select **Include files** from the drop-down list under **Show directories for**. Double click the empty space under **Directories** to enter the specified path of Include files: C:\Program Files\IVI Foundation\ISA\WinNT\include. Click **OK** to close the dialog box. Select **Library files** from the drop-down list under **Show directories for**. Double click the empty space under **Directories** to enter the specified path of Library files: C:\Program Files\IVI Foundation\ISA\WinNT\lib\msc. Click **OK** to close the dialog box.

Note: The two paths added here are related to the installation path of NI-VISA on your PC. By default, NI-VISA is installed under C:\Program Files\IVI Foundation\ISA.



By now, VISA library has been added.

4. Add the Text, Edit, and Button controls, as shown in the figure below.



5. Add the control variables.  
Click **View** → **ClassWizard**, and then click the **Member Variables** tab to add the following three variables:  
Address CString m\_strInstrAddr  
Command CString m\_strCommand  
Return CString m\_strResult

6. Encapsulate the read and write operations of VISA.
  - 1) Encapsulate the write operation of VISA for easier operation.  

```
bool CDemoForRSADlg::InstrWrite(CString strAddr, CString strContent) //Write operation
{
    ViSession defaultRM,instr;
    ViStatus status;
    ViUInt32 retCount;
    char * SendBuf = NULL;
    char * SendAddr = NULL;
    bool bWriteOK = false;
    CString str;
```

```

// Change the address's data style from CString to char*
SendAddr = strAddr.GetBuffer(strAddr.GetLength());
strcpy(SendAddr,strAddr);
strAddr.ReleaseBuffer();

// Change the command's data style from CString to char*
SendBuf = strContent.GetBuffer(strContent.GetLength());
strcpy(SendBuf,strContent);
strContent.ReleaseBuffer();

//Open a VISA resource
status = viOpenDefaultRM(&defaultRM);
if (status < VI_SUCCESS)
{
    AfxMessageBox("No VISA resource was opened!");
    return false;
}

status = viOpen(defaultRM, SendAddr, VI_NULL, VI_NULL, &instr);

//Write command to the instrument
status = viWrite(instr, (unsigned char *)SendBuf, strlen(SendBuf), &retCount);

//Close the system
status = viClose(instr);
status = viClose(defaultRM);

return bWriteOK;
}

```

- 2) Encapsulate the read operation of VISA for easier operation.

```

bool CDemoForRSADlg::InstrRead(CString strAddr, CString *pstrResult) //Read operation
{
    ViSession defaultRM,instr;
    ViStatus status;
    ViUInt32 retCount;
    char * SendAddr = NULL;
    unsigned char RecBuf[MAX_REC_SIZE];
    bool bReadOK = false;
    CString str;

    // Change the address's data style from CString to char*
    SendAddr = strAddr.GetBuffer(strAddr.GetLength());
    strcpy(SendAddr,strAddr);
    strAddr.ReleaseBuffer();

    memset(RecBuf,0,MAX_REC_SIZE);

    //Open a VISA resource
    status = viOpenDefaultRM(&defaultRM);
    if (status < VI_SUCCESS)
    {
        // Error Initializing VISA...exiting
        AfxMessageBox("No VISA resource was opened!");
        return false;
    }
}

```

```

//Open the instrument
status = viOpen(defaultRM, SendAddr, VI_NULL, VI_NULL, &instr);

//Read from the instrument
status = viRead(instr, RecBuf, MAX_REC_SIZE, &retCount);

//Close the system
status = viClose(instr);
status = viClose(defaultRM);

(* pstrResult).Format("%s", RecBuf);

return bReadOK;
}

```

- 3) Encapsulate the read operation with exception handling function of VISA.  
 ViStatus CDemoForRSADlg::OpenVisaDevice(CString strAddr) //Open a VISA device

```

{
  ViStatus status;
  char * SendAddr = NULL;

  // Change the address's data style from CString to char*
  SendAddr = strAddr.GetBuffer(strAddr.GetLength());
  strcpy(SendAddr, strAddr);
  strAddr.ReleaseBuffer();

  //Open a VISA resource
  status = viOpenDefaultRM(&m_SessRM);

  if (status == 0)
  {
    //Open the device
    status = viOpen(m_SessRM, SendAddr, VI_NULL, VI_NULL, &m_SessInstr);

    //If you fail to open the connection, close the resource
    if (status != 0)
    {
      viClose(m_SessRM);
    }
  }

  return status;
}

```

```

ViStatus CDemoForRSADlg::CloseVisaDevice() //Close a VISA device
{
  ViStatus status;

  //Close the device
  status = viClose(m_SessInstr);

  if (status == 0)
  {
    //close the resource
    status = viClose(m_SessRM);
  }
}

```

```

    return status;
}

bool CDemoForRSADlg::InstrWriteEx(CString strAddr, CString strContent) //Write operation with
exception handling
{
    ViStatus status;
    ViUInt32 retCount;
    char * SendBuf = NULL;
    bool bWriteOK = true;

    // Change the address's data style from CString to char*
    SendBuf = strContent.GetBuffer(strContent.GetLength());
    strcpy(SendBuf, strContent);
    strContent.ReleaseBuffer();

    do
    {
        //Write command to the instrument
        status = viWrite(m_SessInstr, (unsigned char *)SendBuf, strlen(SendBuf), &retCount);

        //If an error occurs, perform error handling
        if (status < 0)
        {
            //If the time exceeds the limit value, resend the command after a delay of 1s
            if (VI_ERROR_TMO == status)
            {
                Sleep(1000);
                status = viWrite(m_SessInstr, (unsigned char *)SendBuf, strlen(SendBuf),
                &retCount);
            }
            else
            {
                //If another error occurs, reopen the connection after the connection is closed and
                resend the command
                status = CloseVisaDevice();
                Sleep(1000);
                status = OpenVisaDevice(m_strInstrAddr);
                if (status == 0)
                {
                    status = viWrite(m_SessInstr, (unsigned char *)SendBuf, strlen(SendBuf),
                    &retCount);
                }
            }
        }
    } while (status < 0);

    return bWriteOK;
}

bool CDemoForRSADlg::InstrReadEx(CString strAddr, CString * pstrResult) //Read operation with
exception handling
{
    ViStatus status;
    ViUInt32 retCount;
    char * SendAddr = NULL;

```

```

unsigned char RecBuf[MAX_REC_SIZE];
bool bReadOK = true;

// Change the address's data style from CString to char*
SendAddr = strAddr.GetBuffer(strAddr.GetLength());
strcpy(SendAddr, strAddr);
strAddr.ReleaseBuffer();

memset(RecBuf, 0, MAX_REC_SIZE);

do
{
    //Read from the instrument
    status = viRead(m_SessInstr, RecBuf, MAX_REC_SIZE, &retCount);
    if (status < 0)
    {
        //If the time exceeds the limit value, read from the instrument after a delay of 1s
        if (VI_ERROR_TMO == status)
        {
            Sleep(1000);
            status = viRead(m_SessInstr, RecBuf, MAX_REC_SIZE, &retCount);
        }
        else
        {
            //If another error occurs, reopen the connection after the connection is closed and
            // reread from instrument
            status = CloseVisaDevice();
            Sleep(1000);
            status = OpenVisaDevice(m_strInstrAddr);
            if (status == 0)
            {
                status = viRead(m_SessInstr, RecBuf, MAX_REC_SIZE, &retCount);
            }
        }
    }
} while (status < 0);

(*pstrResult).Format("%s", RecBuf);

return bReadOK;
}

```

7. Add the control message response codes.

```

1) Connect to the instrument
void CDemoForRSADlg::OnBtConnectInstr() // Connect to the instrument
{
    //TODO: Add your control notification handler code here
    ViStatus status;
    ViSession defaultRM;
    ViString expr = "?* ";
    ViPFindList findList = new unsigned long;
    ViPUInt32 retcnt = new unsigned long;
    ViChar instrDesc[1000];
    CString strSrc = "";
    CString strInstr = "";
    unsigned long i = 0;

```

```

bool bFindRSA = false;

status = viOpenDefaultRM(&defaultRM);
if (status < VI_SUCCESS)
{
    // Error Initializing VISA...exiting
    MessageBox("No VISA instrument was opened ! ");
    return ;
}

memset(instrDesc,0,1000);

// Find resource
status = viFindRsrc(defaultRM,expr,findList, retcnt, instrDesc);

for (i = 0;i < (*retcnt);i++)
{
    // Get instrument name
    strSrc.Format("%s",instrDesc);
    InstrWrite(strSrc,"*IDN?");
    ::Sleep(200);
    InstrRead(strSrc,&strInstr);

    // If the instrument(resource) belongs to the RSA series then jump out //from the loop
    strInstr.MakeUpper();
    if (strInstr.Find("RSA") >= 0)
    {
        bFindRSA = true;
        m_strInstrAddr = strSrc;
        break;
    }

    //Find next instrument
    status = viFindNext(*findList,instrDesc);
}

if (bFindRSA == false)
{
    MessageBox("Didn't find any RSA!");
}
UpdateData(false);
}

```

## 2) Write Operation

```

void CDemoForRSADlg::OnBtWrite() //Write operation
{
    //TODO: Add your control notification handler code here
    UpdateData(true);
    if (m_strInstrAddr.IsEmpty())
    {
        MessageBox("Please connect to the instrument first!");
    }
    InstrWrite(m_strInstrAddr,m_strCommand);
    m_strResult.Empty();
    UpdateData(false);
}

```

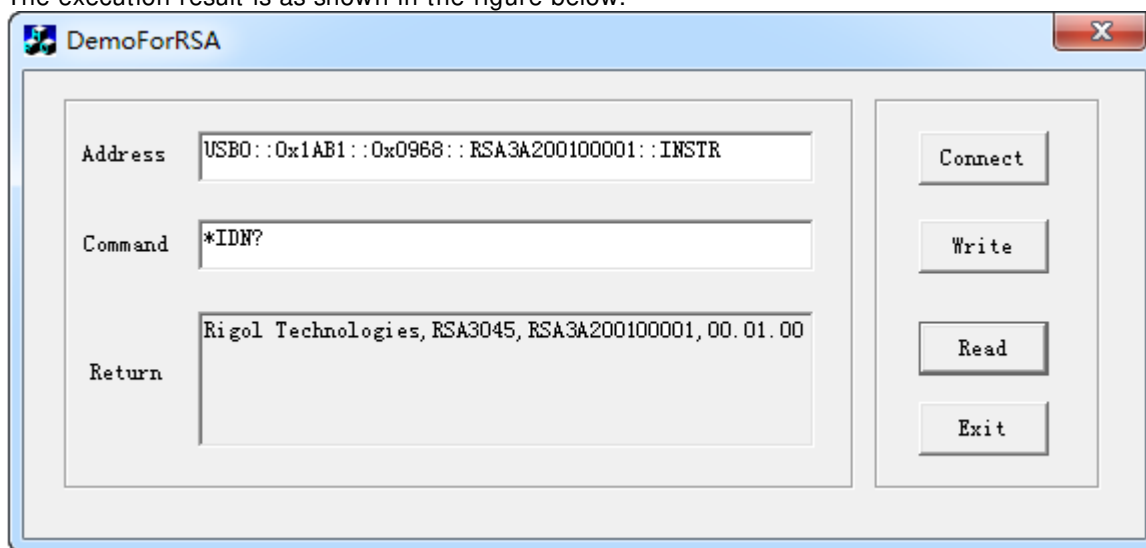
## 3) Read Operation

```
void CDemoForRSADlg::OnBtRead()           //Read operation
{
    //TODO: Add your control notification handler code here
    UpdateData(true);
    InstrRead(m_strInstrAddr,&m_strResult);
    UpdateData(false);
}
```

8. Run the results.

- 1) Click **Connect** to search for the spectrum analyzer;
- 2) Input "**\*IDN?**" in the "Command" edit box;
- 3) Click **Write** to write the command into the spectrum analyzer;
- 4) Click **Read** to read the return value.

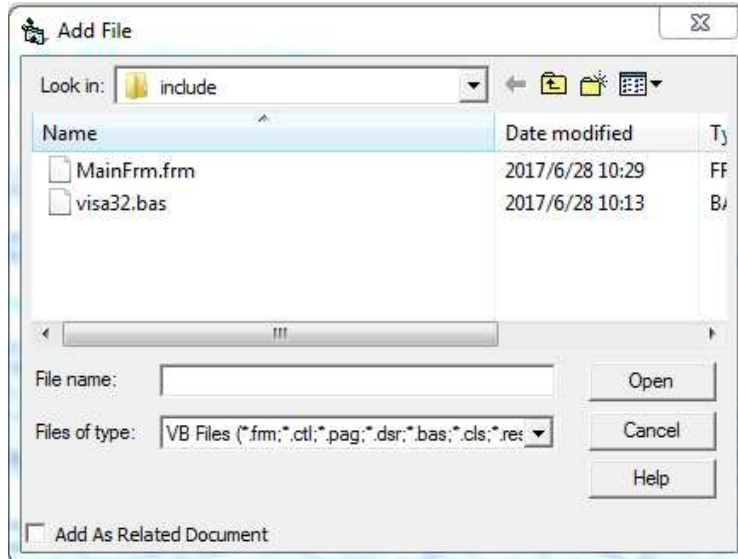
The execution result is as shown in the figure below.



## Visual Basic 6.0 Programming Example

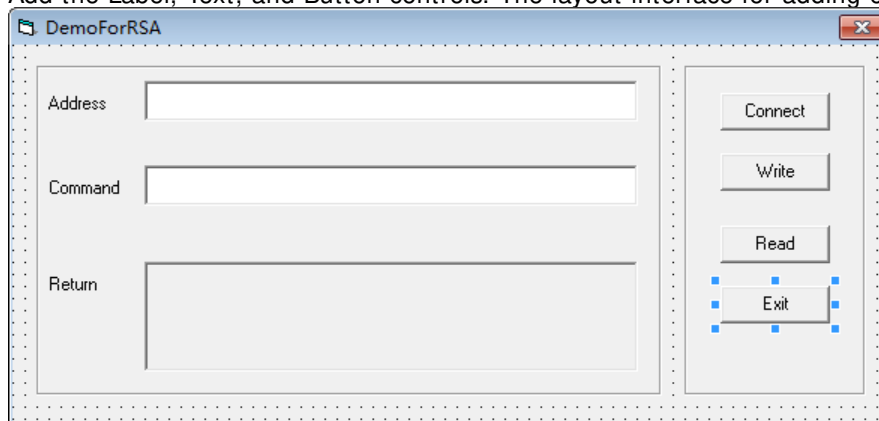
Enter the Visual Basic 6.0 programming environment, and perform the following procedures.

1. Build a standard application program project (Standard EXE), and name it "DemoForRSA".
2. Open **Project → Add File....** Search for the **visa32.bas** file from the **include** folder in the installation path of NI-VISA, and then add the file to the project. The visa32.bas module contains all VISA functions and constant statements.



Then, add the **Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)** statement into the **visa32.bas** module; or you can also create a new module to declare the Sleep function.

3. Add the Label, Text, and Button controls. The layout interface for adding controls is as follows:



4. Encapsulate the read and write operations of VISA.
  - 1) Encapsulate the write operation of VISA for easier operation.

```

'-----
'Function Name: InstrWrite
'Function: Send command to the instrument
'Input: rsrcName,instrument(resource) name
       strCmd,Command
'-----
Public Sub InstrWrite(rsrcName As String, strCmd As String)
    Dim status As Long
    Dim dfltRM As Long
    Dim sesn As Long
  
```

```

Dim rSize As Long

'Initialize the system
status = viOpenDefaultRM(dfItRM)
'Failed to initialize the system
If (status < VI_SUCCESS) Then
    MsgBox " No VISA resource was opened ! "
    Exit Sub
End If
'Open the VISA instrument
status = viOpen(dfItRM, rsrcName, VI_NULL, VI_NULL, sesn)
'Failed to open the instrument
If (status < VI_SUCCESS) Then
    MsgBox "Failed to open the instrument ! "
    Exit Sub
End If

'Write command to the instrument
status = viWrite(sesn, strCmd, Len(strCmd), rSize)
'Failed to write to the instrument
If (status < VI_SUCCESS) Then
    MsgBox " Failed to write to the instrument ! "
    Exit Sub
End If

'Close the system
status = viClose(sesn)
status = viClose(dfItRM)

End Sub

```

- 2) Encapsulate the read operation of VISA for easier operation.

```

'-----
'Function Name: InstrRead
'Function: Read the return value from the instrument
'Input: rsrcName,Resource name
'Return: The string gotten from the instrument
'-----
Public Function InstrRead(rsrcName As String) As String
    Dim status As Long
    Dim dfItRM As Long
    Dim sesn As Long
    Dim strTemp0 As String * 256
    Dim strTemp1 As String
    Dim rSize As Long

    'Begin by initializing the system
    status = viOpenDefaultRM(dfItRM)
    'Initialization failed
    If (status < VI_SUCCESS) Then
        MsgBox " Failed to open the instrument! "
        Exit Function
    End If
    'Open the instrument
    status = viOpen(dfItRM, rsrcName, VI_NULL, VI_NULL, sesn)
    'Failed to open the instrument
    If (status < VI_SUCCESS) Then

```

```

        MsgBox " Failed to open the instrument! "
    Exit Function
End If

'Read from the instrument
status = viRead(sesn, strTemp0, 256, rSize)
'Reading failed
If (status < VI_SUCCESS) Then
    MsgBox " Failed to read from the instrument! "
    Exit Function
End If

'Close the system
status = viClose(sesn)
status = viClose(dfiltRM)

'Remove the space at the end of the string
strTemp1 = Left(strTemp0, rSize)
InstrRead = strTemp1
End Function

```

5. Add the control event codes.

1) Connect to the instrument

```

'Connect to the instrument
Private Sub CmdConnect_Click()
    Const MAX_CNT = 200
    Dim status As Long
    Dim dfiltRM As Long
    Dim sesn As Long
    Dim fList As Long
    Dim buffer As String * MAX_CNT, Desc As String * 256
    Dim nList As Long, retCount As Long
    Dim rsrcName(19) As String * VI_FIND_BUFLen, instrDesc As String * VI_FIND_BUFLen
    Dim i, j As Long
    Dim strRet As String
    Dim bFindRSA As Boolean

    'Initialize the system
    status = viOpenDefaultRM(dfiltRM)
    'Initialization failed
    If (status < VI_SUCCESS) Then
        MsgBox " No VISA resource was opened ! "
        Exit Sub
    End If

    'Find instrument resource
    Call viFindRsrc(dfiltRM, "USB?* INSTR", fList, nList, rsrcName(0))
    'Get the list of the instruments (resources)
    strRet = ""
    bFindRSA = False
    For i = 0 To nList - 1
        'Get the instrument name
        InstrWrite rsrcName(i), "*IDN?"
        Sleep 200
        strRet = InstrRead(rsrcName(i))
        'Continuing searching for the resource until an RSA instrument is found
        strRet = UCase(strRet)
    Next i

```

```

j = InStr(strRet, "RSA")
If (j >= 0) Then
    bFindRSA = True
    Exit For
End If

Call viFindNext(fList + i - 1, rsrcName(i))
Next i
'Display
If (bFindRSA = True) Then
    TxtInsAddr.Text = rsrcName(i)
Else
    TxtInsAddr.Text = ""
End If
End Sub

```

2) Write Operation

'Write the command to the instrument

```

Private Sub CmdWrite_Click()
    If (TxtInsAddr.Text = "") Then
        MsgBox ("Please write the instrument address! ")
    End If

    InstrWrite TxtInsAddr.Text, TxtCommand.Text
End Sub

```

3) Read Operation

'Read the return value from the instrument

```

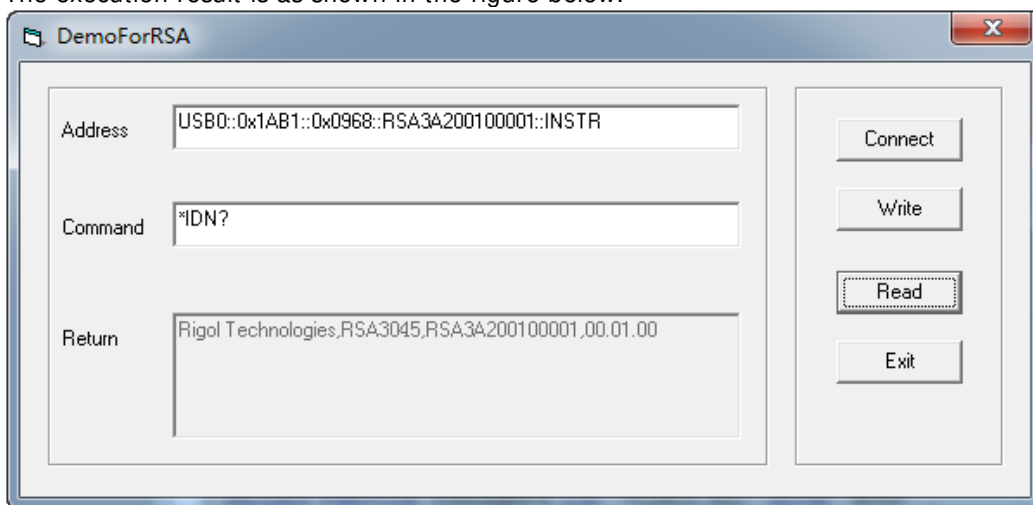
Private Sub CmdRead_Click()
    Dim strTemp As String
    strTemp = InstrRead(TxtInsAddr.Text)
    TxtReturn.Text = strTemp
End Sub

```

6. Run the results.

- 1) Click **Connect** to search for the spectrum analyzer;
- 2) Input "\*\*IDN?" in the "Command" edit box;
- 3) Click **Write** to write the command into the spectrum analyzer;
- 4) Click **Read** to read the return value.

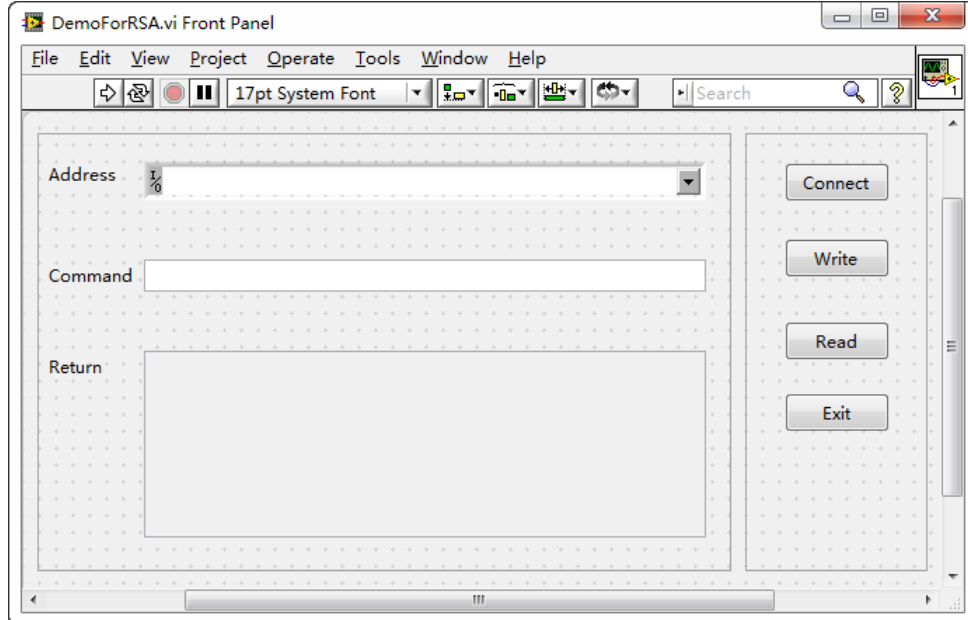
The execution result is as shown in the figure below.



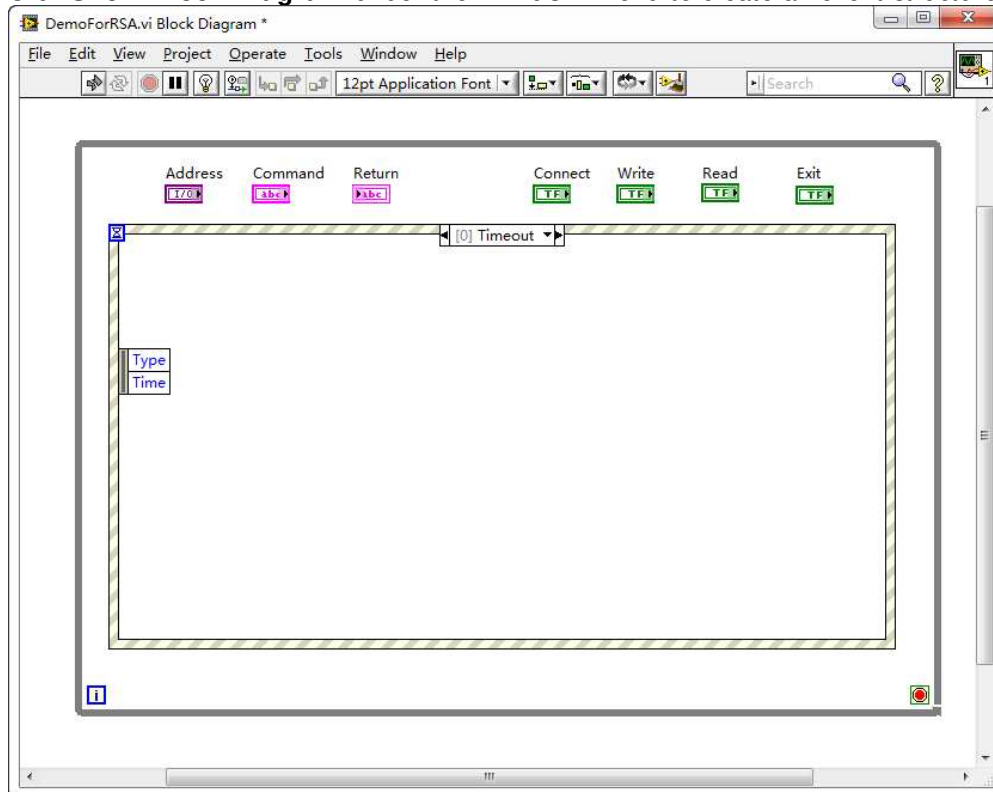
## LabVIEW 2010 Programming Example

Enter the Labview 2010 programming environment, and perform the following procedures.

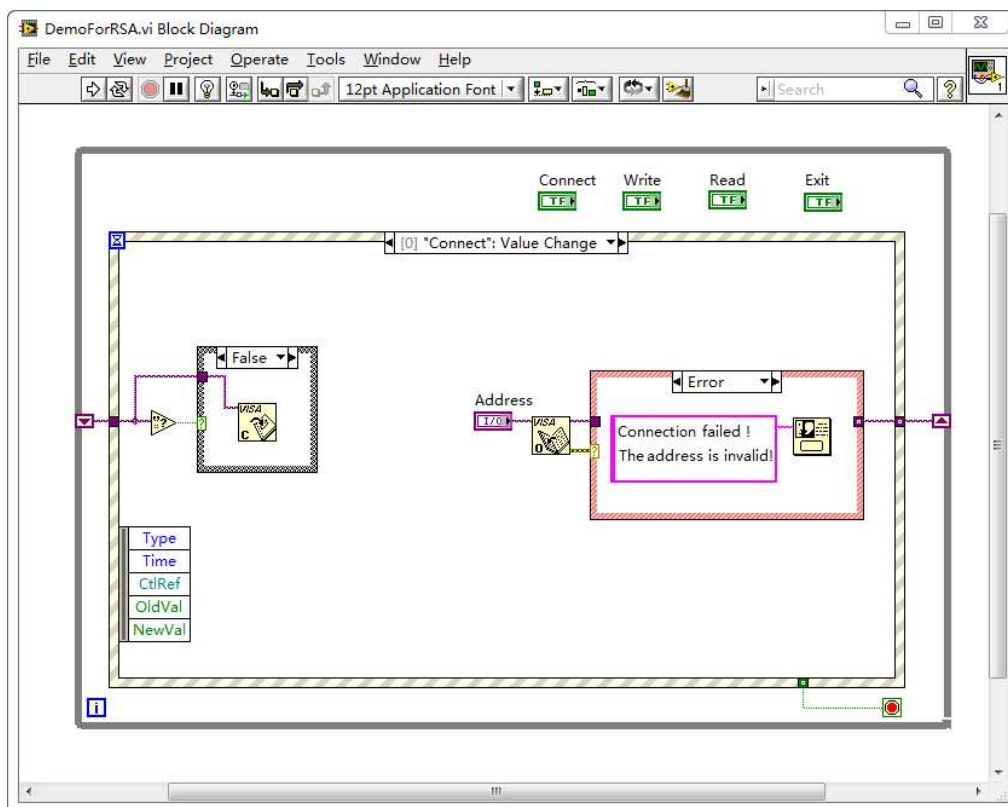
1. Create a VI file, and name it "DemoForRSA".
2. Add controls to the front panel interface, including the **Address** field, **Command** field, and **Return** field, the **Connect** button, the **Write** button, the **Read** button, and the **Exit** button.



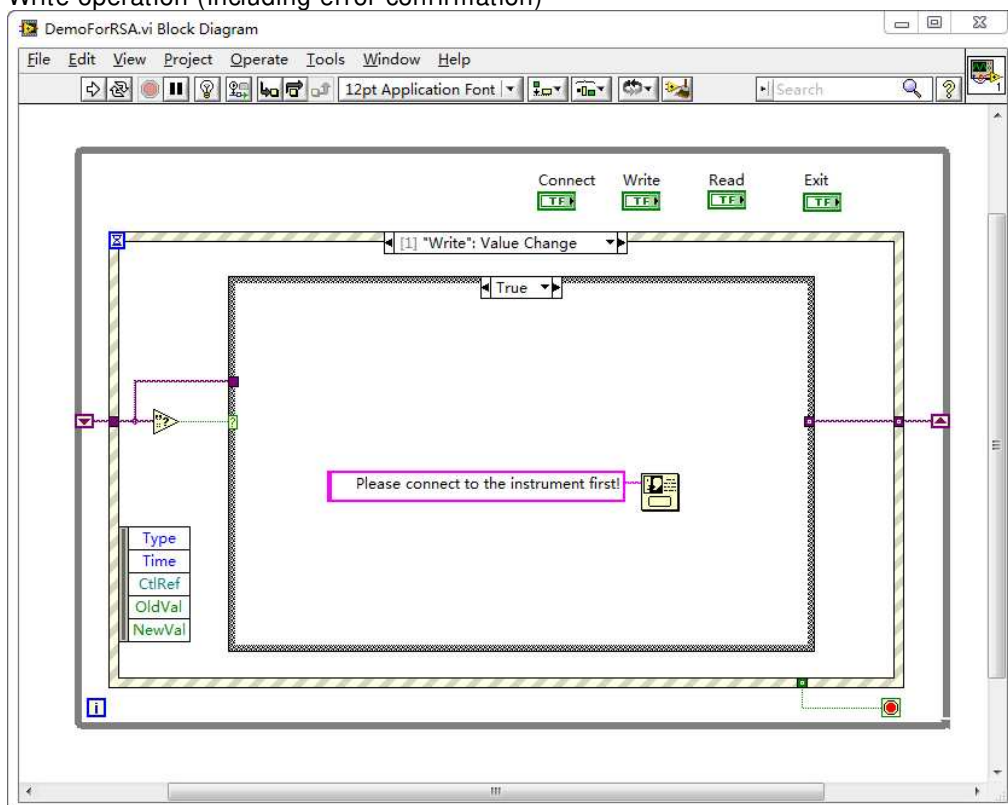
3. Click **Show Block Diagram** under the **Window** menu to create an event structure.

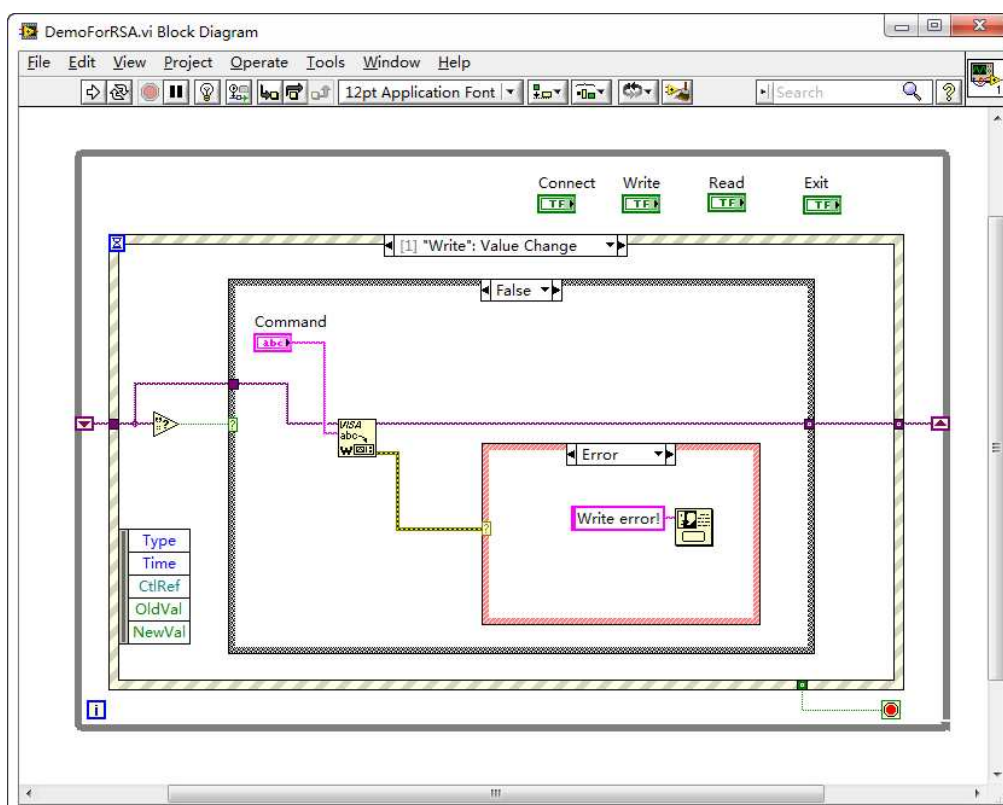


4. Add the events (including connecting to the instrument, write operation, read operation, and exit)
  - 1) Connect to the instrument

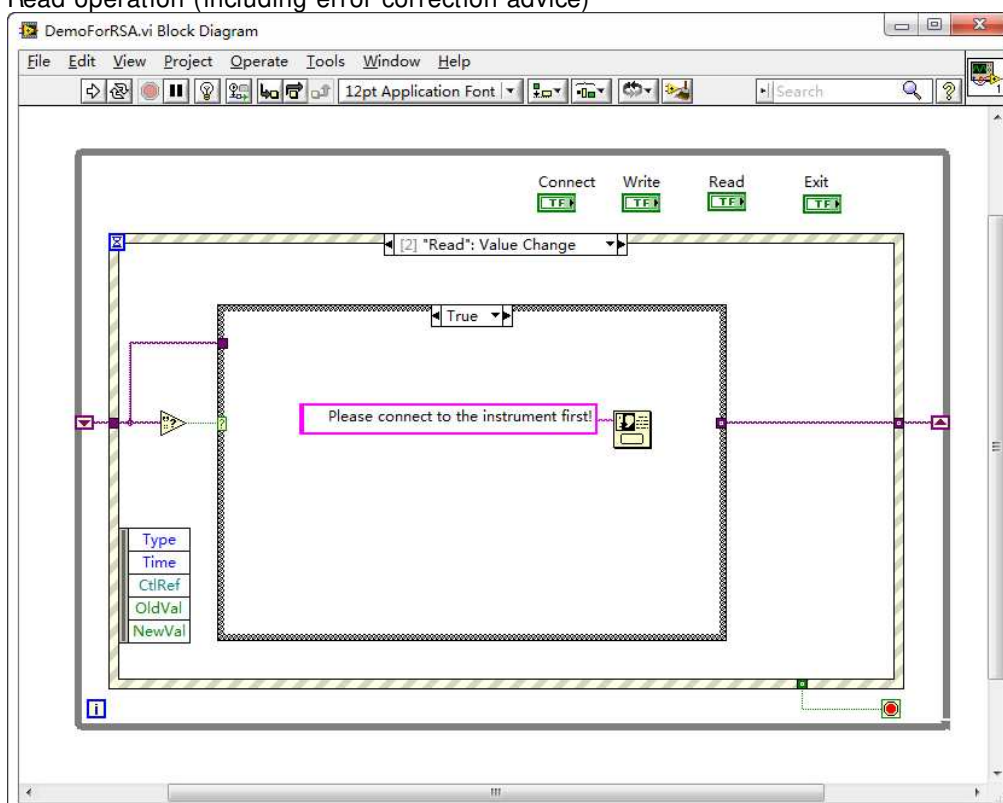


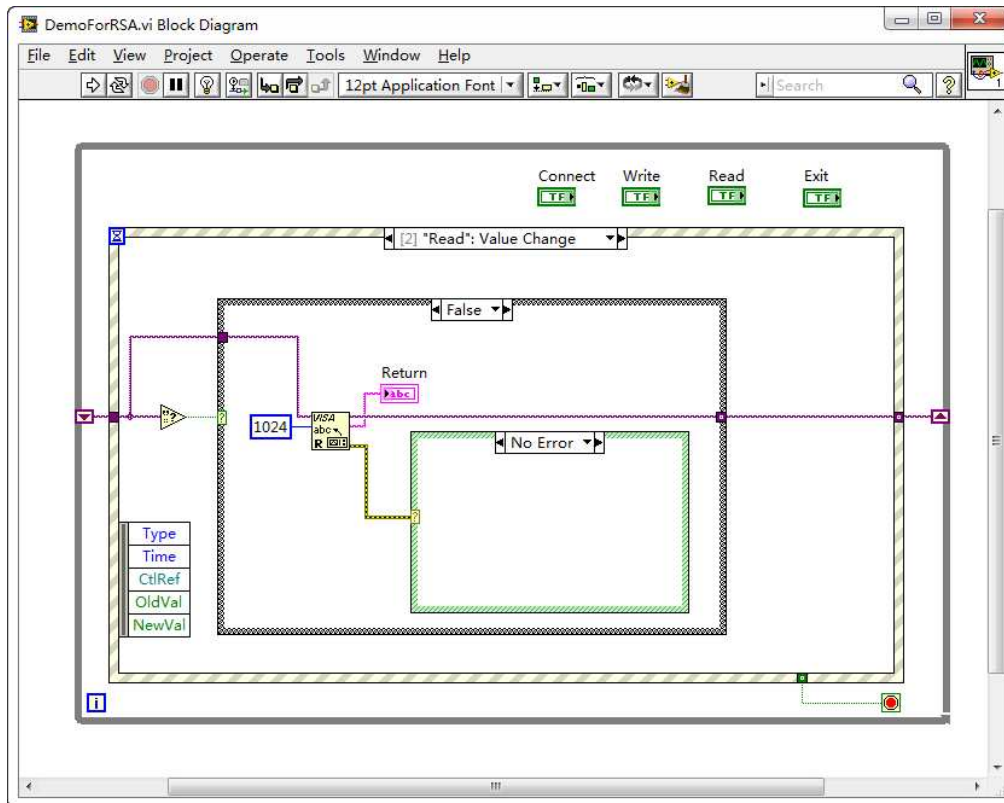
## 2) Write operation (including error confirmation)



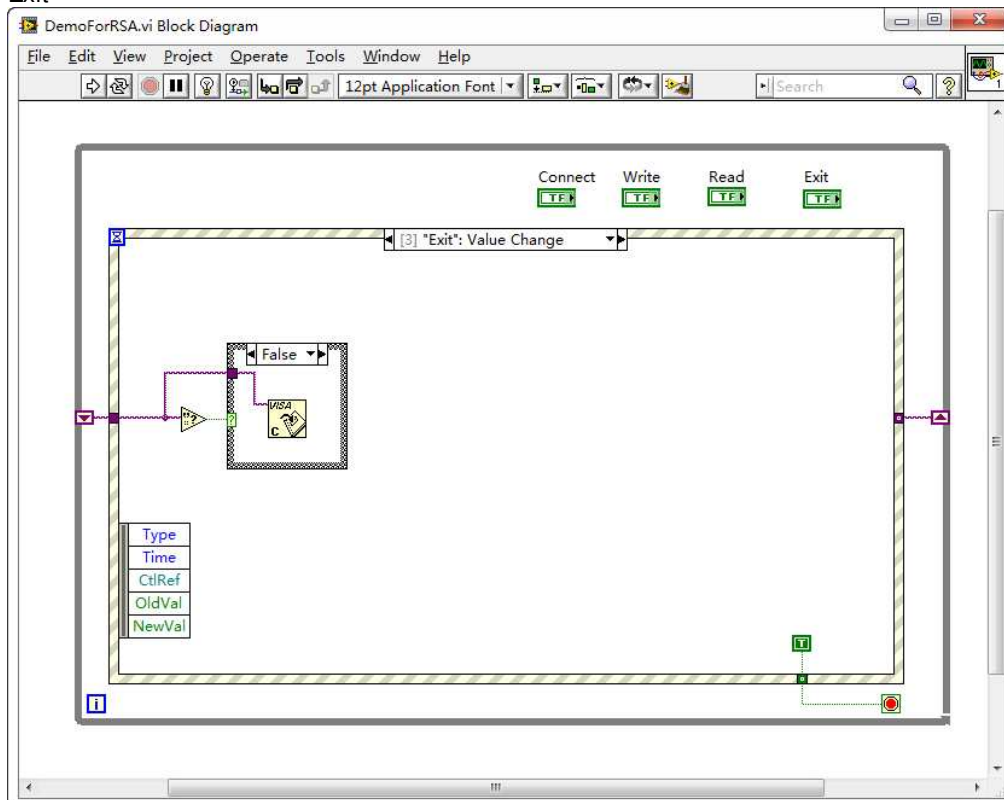


## 3) Read operation (including error correction advice)



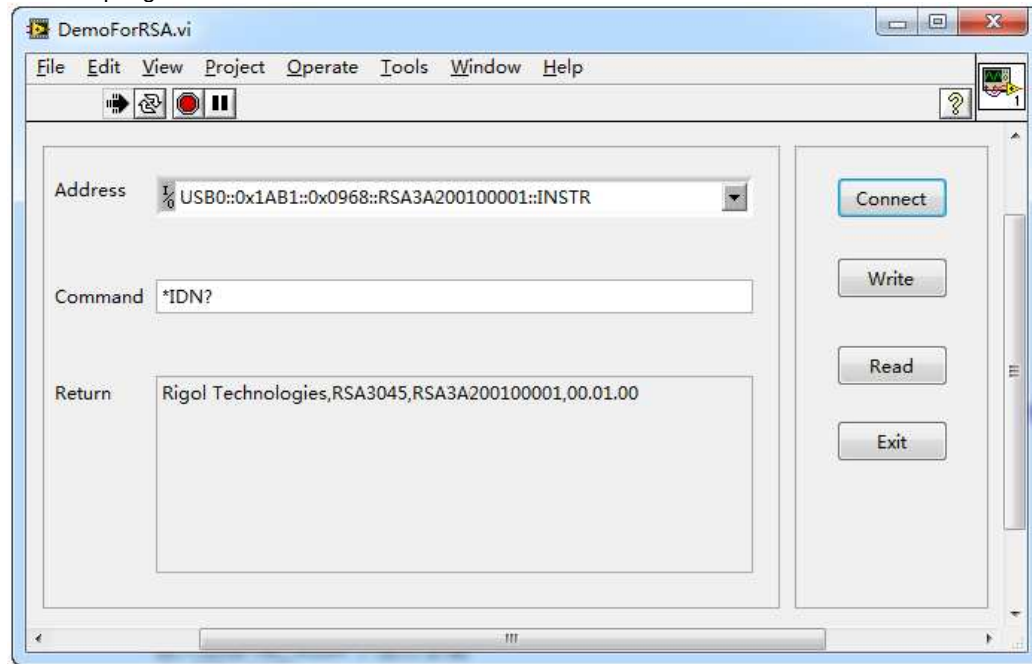


## 4) Exit



- 5) Run the program, and then the following interface is displayed below. Click the VISA resource name from the drop-down list under **Address**, and click **Connect** to connect the instrument. Then, input a command in the **Command** field. Click **Write** to write the command to the instrument. If the command is a query (e.g. \*IDN?), click **Write** to write the command into the instrument, and then click **Read**. The return value is displayed in the **Return** field. Click **Exit** to

exit the program.



## Linux Programming Example

This section illustrates how to program and control the spectrum analyzer to realize the common functions in Linux operating system.

### Programming Preparations

1. Programming environment:  
Operating system: Fedroa 8 (Linux-2.6.23)  
GCC version: gcc-4.1.2
2. Install the VISA library. First, check whether your PC has installed NI's VISA library. If not, download it from NI website (<http://www.ni.com/visa/>). The installation procedures are as follows:  
Download the VISA library NI-VISA-4.4.0.ISO from the NI website.

Create a new directory.

```
# mkdir NI_VISA
```

Mount the isofile

```
# mount -o loop -t iso9660 NI-VISA-4.4.0.iso NI_VISA
```

Enter the NI\_VISA directory to install

```
# cd NI_VISA
```

```
# ./INSTALL
```

Unmount the isofile

```
# umount NI_VISA
```

After the installation is finished, the default installation path is /usr/local.

3. Build communication between the spectrum analyzer and the PC. Use the network cable to connect the analyzer to the PC via the LAN interface on the rear panel of the analyzer. You can also use a network cable to connect the spectrum analyzer to the local area network where the PC resides.

After the spectrum analyzer is connected to the PC properly, configure the network address for the spectrum analyzer to make its address to be within the same network segment where the PC resides.

For example, if the network address and DNS setting configured for the PC are as shown in the figures below, then, the network address of the spectrum analyzer should be configured as follows.

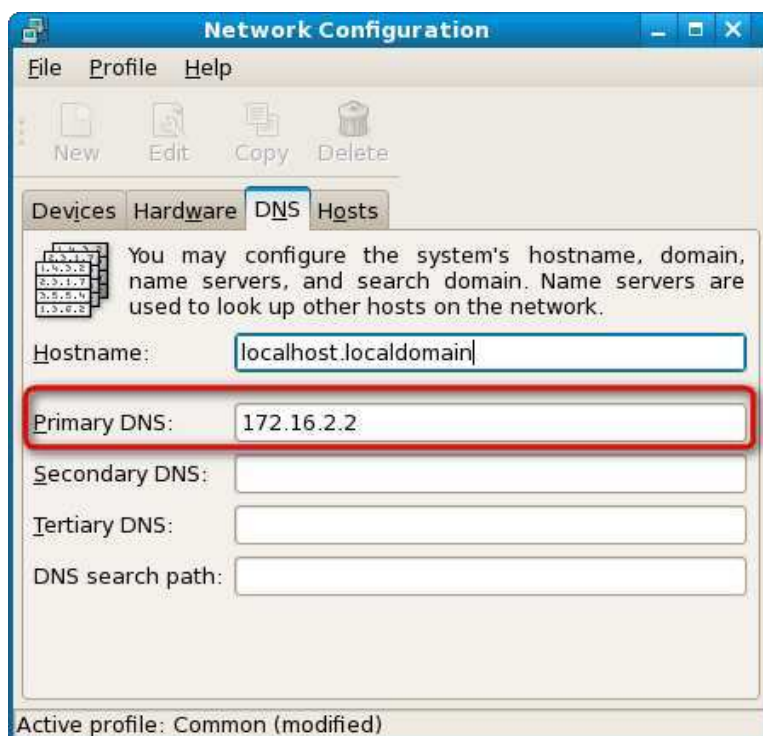
IP Address: 172.16.3.X\*

Default Gateway: 172.16.3.1

Subnet Mask: 255.255.255.0

DNS: 172.16.2.2

Note\*: X can be any value not in use from 2 to 254.



4. Use either of the following two methods to add the library location to the search path of the library, so that the program can load the installed library file automatically.

**Method 1:** Specify the search path of the library in the environment variable LD\_LIBRARY\_PATH. Operation Method: Add the library file path /usr/local/lib to the LD\_LIBRARY\_PATH variable in the /etc/profile file, as shown in the figure below.

```
123456@localhost:/home/123456
File Edit View Terminal Tabs Help
USER="`id -un`"
LOGNAME=$USER
MAIL="/var/spool/mail/$USER"
fi

HOSTNAME=`/bin/hostname`
HISTSZ=1000

if [ -z "$INPUTRC" -a ! -f "$HOME/.inputrc" ]; then
    INPUTRC=/etc/inputrc
fi
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/lib
#: /usr/local/vxipnp/linux/lib
export PATH USER LOGNAME MAIL HOSTNAME HISTSZ INPUTRC LD_LIBRARY_PATH

for i in /etc/profile.d/*.sh ; do
    if [ -r "$i" ]; then
        . $i
    fi
done

unset i
```

**Method 2:** Add the search path of the library in the `/etc/ld.so.conf` file.

Operation Method: # echo "/usr/local/lib" >> /etc/ld.so.conf, as shown in the figure below.

After setting the search path of the library in `/etc/ld.so.conf`, run the `/sbin/ldconfig` command to update `/etc/ld.so.cache` (this command should have the root permission) to ensure the location of the library when executing the program.

[illegible]

## Linux Programming Procedures

1. Edit the DemoForRSA.h header file and declare a class to encapsulate the operation and property of the instrument.

```
#ifndef DEMO_FOR_RSA_H
#define DEMO_FOR_RSA_H

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <iostream>
// #include <syswait.h>
using namespace std;

#define MAX_SEND_BUF_SIZE 50
#define MAX_REC_SIZE 300

class DemoForRSA
{
// Construction
public:
DemoForRSA();
bool InstrRead(string strAddr, string & pstrResult);
bool InstrWrite(string strAddr, string strContent);
bool ConnectInstr();

string m_strInstrAddr;
string m_strResult;
string m_strCommand;

};

void makeupper(string & instr);

#endif
```

2. Edit the DemoForRSA.cpp file to realize various operations of the instrument.

```
#include "visa.h"
#include "DemoForRSA.h"

DemoForRSA::DemoForRSA()
{
m_strInstrAddr = "";
m_strResult = "";
m_strCommand = "";
}

bool DemoForRSA::ConnectInstr()
{
{
ViUInt32 retCount;
ViStatus status;
ViSession defaultRM;
ViString expr = "?* ";
ViPFindList findList = new unsigned long;
ViPUInt32 retcnt = new unsigned long;
string strSrc = "";
```

```

string strInstr = "";
ViChar instrDesc[1000];

unsigned long i = 0;
bool bFindRSA = false;
memset(instrDesc,0,1000);

//Turn on the VISA device
status = viOpenDefaultRM(&defaultRM);

if (status < VI_SUCCESS)
{
    cout<< "No VISA equipment!"<< endl;
    return false;
}

//Search for resources
status = viFindRsrc(defaultRM,expr,findList, retcnt, instrDesc);

for (i = 0;i < (*retcnt);i++)
{
    //Acquire the instrument name
    strSrc = instrDesc;

    InstrWrite(strSrc,"*IDN?");
    usleep(200);
    InstrRead(strSrc,strInstr);

    //If the RSA series is found, then exit
    makeupper(strInstr);
    if (strInstr.find("RSA",0) > 0)
    {
        bFindRSA = true;
        m_strInstrAddr = strSrc;
        break;
    }

    //Acquire the next device
    status = viFindNext(*findList,instrDesc);
}

if (bFindRSA == false)
{
    printf("RSA device not found!\n");
    return false;
}

return true;
}

bool DemoForRSA::InstrWrite(string strAddr, string strContent) //Write operation
{
    ViSession defaultRM,instr;
    ViStatus status;
    ViUInt32 retCount;
    char * SendBuf = NULL;
    char * SendAddr = NULL;
    bool bWriteOK = false;

```

```

string str;

//Address conversion, convert the string type to char*
SendAddr = const_cast<char*>(strAddr.c_str());

//Address conversion, convert the string type to char*
SendBuf = const_cast<char*>(strContent.c_str());

//Turn on the specified device
status = viOpenDefaultRM(&defaultRM);
if (status < VI_SUCCESS)
{
    cout<< "No VISA equipment!" << endl;
    return false;
}

status = viOpen(defaultRM, SendAddr, VI_NULL, VI_NULL, &instr);

//Write command to the device
status = viWrite(instr, (unsigned char *)SendBuf, strlen(SendBuf), &retCount);

//Turn off the device
status = viClose(instr);
status = viClose(defaultRM);
return bWriteOK;
}

bool DemoForRSA::InstrRead(string strAddr, string & pstrResult) //Read operation
{
    ViSession defaultRM, instr;
    ViStatus status;
    ViUInt32 retCount;
    char* SendAddr = NULL;
    char * result = NULL;
    bool bReadOK = false;
    unsigned char RecBuf[MAX_REC_SIZE];
    string str;
    memset(RecBuf,0,MAX_REC_SIZE);

    result=(char*) malloc(MAX_REC_SIZE* sizeof(char));
    memset(result,0,MAX_REC_SIZE);

    //Address conversion, convert the string type to char*
    SendAddr=const_cast<char*>(strAddr.c_str());

    //Turn on the VISA device
    status = viOpenDefaultRM(&defaultRM);
    if (status < VI_SUCCESS)
    {
        // Error Initializing VISA...exiting
        cout<< "No VISA equipment!" << endl;
        return false;
    }

    //Turn on the specified device
    status = viOpen(defaultRM, SendAddr, VI_NULL, VI_NULL, &instr);

    //Read from the device

```

```

status = viRead(instr, RecBuf, MAX_REC_SIZE, &retCount);

//Turn off the device
status = viClose(instr);
status = viClose(defaultRM);
sprintf(result,"%s",RecBuf);
pstrResult = result;
free(result);
return bReadOK;
}

void makeupper( string &instr)
{

    string outstr = "";
    if(instr == "")
    {
        exit(0);
    }

    for(int i = 0;i < instr.length();i++)
    {
        instr[i] = toupper(instr[i]);
    }

}

```

3. Edit the function file mainloop.cpp to complete the flow control.

```
#include "DemoForRSA.h"
```

```

void menudisplay()
{
    cout<< "\t\t Please operate the instrument:\n read write quit" << endl;
}

int main()
{
    DemoForRSA demo;
    char temp[50];
    if(!demo.ConnectInstr())
    {
        cout<< "can not connect the equipment!" << endl;
        return 0;
    }
    else
    {
        cout<< "\n connect equipment success!" << endl;
        cout<< " the equipment address is : " << demo.m_strInstrAddr << endl;
    }

    while(1)
    {
        menudisplay();
        //cin>>demo.m_strCommand;
        cin.getline(temp,50);
        demo.m_strCommand=temp;
        if(demo.m_strCommand[0]='r' && demo.m_strCommand[1]='e'

```

```

    && demo.m_strCommand[2]='a' && demo.m_strCommand[3]='d')
{
    //demo.InstrWrite(demo.m_strInstrAddr,"* IDN?");
    //demo.InstrRead(demo.m_strInstrAddr,demo.m_strResult);
    cout<<"read result:"<< demo.m_strResult<< endl;
    demo.m_strResult="";
}

else if (demo.m_strCommand[0]='w' && demo.m_strCommand[1]='r'
        && demo.m_strCommand[2]='i' && demo.m_strCommand[3]='t' &&
        demo.m_strCommand[4]='e')
{
    if (demo.m_strInstrAddr=="")
    {
        cout<<"Please connect the instrument ! \n";
    }
    demo.InstrWrite(demo.m_strInstrAddr,demo.m_strCommand.substr(5,40));
    usleep(200);

    //Read operation
    demo.InstrRead(demo.m_strInstrAddr,demo.m_strResult);
}

else if (demo.m_strCommand[0] == 'q' && demo.m_strCommand[1] == 'u'
        && demo.m_strCommand[2] == 'i' && demo.m_strCommand[3] == 't')
{
    break;
}
else if(demo.m_strCommand != "")
{
    cout<<"Bad command!"<< endl;
}
}
return 1;
}

```

4. makefile file  
src = DemoForRSA.cpp mainloop.cpp DemoForRSA.h

```

obj = DemoForRSA.o mainloop.o
INCLUDE= -I/usr/local/vxipnp/linux/include
LIB= -lvisa -lc -lpthread
CC=
demo : $(obj)
$(CC) $(INCLUDE) $(LIB) -o demo $(obj)

mainloop.o : mainloop.cpp DemoForRSA.h
$(CC) -c $< -o $@
DemoForRSA.o: DemoForRSA.cpp DemoForRSA.h
$(CC) -c $< -o $@

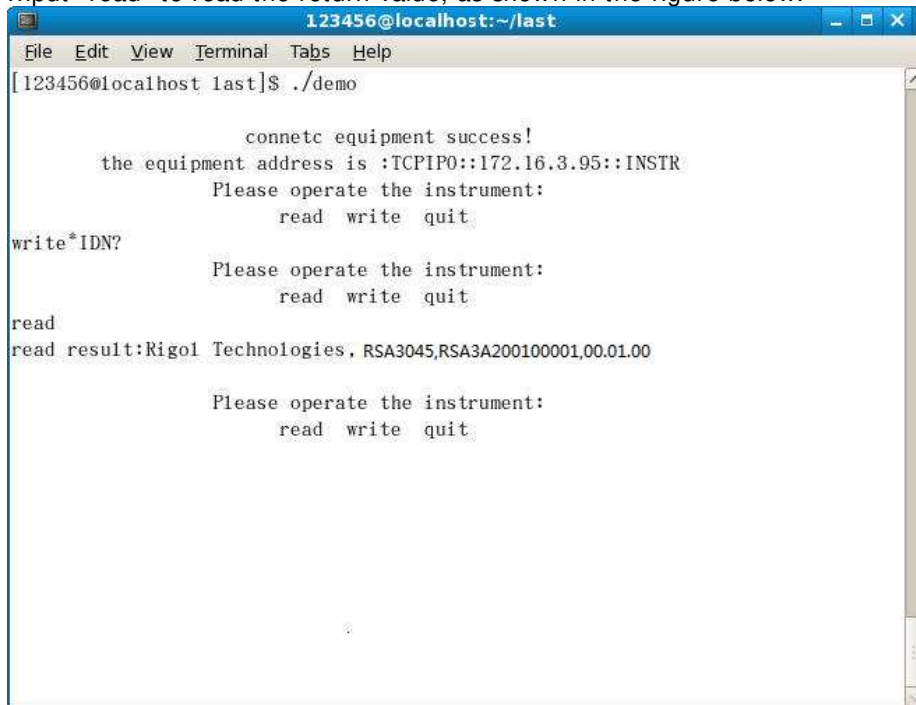
.PHONY : clean
clean:

```

```
rm demo $(obj)
```

5. Run the results.

- 1) # make
- 2) ./demo
- 3) When the program runs, the instrument is connected automatically. If no instrument is found, a prompt message "No VISA equipment!" is displayed, and the system exits the program. If the instrument is found and successfully connected, the following interface is displayed, as shown in the figure below.
- 4) Input write<command> (for example, write<\*IDN?>) to write the command into the spectrum analyzer.
- 5) Input "read" to read the return value, as shown in the figure below.



```
123456@localhost: ~/last
File Edit View Terminal Tabs Help
[123456@localhost last]$ ./demo

      connect equipment success!
the equipment address is :TCPIP0::172.16.3.95::INSTR
Please operate the instrument:
      read write quit
write*IDN?
      Please operate the instrument:
      read write quit
read
read result:Rigol Technologies, RSA3045,RSA3A200100001,00.01.00
      Please operate the instrument:
      read write quit
```



## Chapter 4 Appendix

### Appendix A: Default Settings

Sending the [\\*RST](#) command can restore the instrument to default settings, as shown in the table below.

| Parameter Name  | GPSA Parameter Value | RTSA Parameter Value    |                  |
|-----------------|----------------------|-------------------------|------------------|
|                 |                      | Non-PvT                 | PvT              |
| FREQ            |                      |                         |                  |
| Center Freq     | 2.25 GHz             | 2.25 GHz                |                  |
| Start Freq      | 0 Hz                 | 2.245 GHz               |                  |
| Stop Freq       | 4.5 GHz              | 2.255 GHz               |                  |
| CF Step         | Auto, 450 MHz        | Auto, 1 MHz             |                  |
| Freq Offset     | 0 Hz                 | 0 Hz                    |                  |
| Signal Track    | Off                  | --                      |                  |
| SPAN            |                      |                         |                  |
| Span            | 4.5 GHz              | 10 MHz                  |                  |
| Ref Value       | --                   | --                      | 0 us             |
| Scale/Div       | --                   | --                      | 3.1946 ms        |
| Ref Position    | --                   | --                      | Left             |
| Auto Scale      | --                   | --                      | Auto             |
| AMPT            |                      |                         |                  |
| Ref Level       | 0 dBm                | 0 dBm                   |                  |
| Attenuation     | Auto, 10 dB          | Auto, 10 dB             |                  |
| RF Preamp       | Off                  | Off                     |                  |
| Y Axis Unit     | dBm                  | dBm                     |                  |
| Scale Type      | Log                  | Log                     |                  |
| Scale/Div       | 10 dB                | 10 dB                   |                  |
| Max Mixer Lvl   | -10 dBm              | -10 dBm                 |                  |
| Ref Offset      | 0 dB                 | 0 dB                    |                  |
| BW              |                      |                         |                  |
| RBW             | Auto, 10 MHz         | Auto, 50.228 kHz (RBW2) | --               |
| SPAN/RBW Ratio  | Auto, 106            | --                      | --               |
| VBW             | Auto, 10 MHz         | --                      | --               |
| VBW/RBW Ratio   | Auto, 1              | --                      | --               |
| Filter Type     | Gaussian             | Kaiser                  | --               |
| Sweep           |                      |                         |                  |
| Sweep Points    | 801                  | --                      |                  |
| Sweep Time      | Auto, 1 ms           | --                      |                  |
| Acq Time        | --                   | Auto, 31.9460 ms        | Auto, 31.9460 ms |
| Sweep Time Mode | Continuous           | Continuous              |                  |
| Sweep Time Rule | Normal               | --                      |                  |
| Trigger         |                      |                         |                  |
| Trigger Source  | Free Run             | Free Run                |                  |
| Trigger Holdoff | Off, 100 ms          | Off, 100 ms             |                  |
| Auto Trig       | Off, 100 ms          | Off, 100 ms             |                  |
| Slope           | Positive             | Positive                |                  |
| Trig Delay      | Off, 1 us            | Off, 1 us               |                  |
| Trigger Level   | -25 dBm              | --                      |                  |
| Acq/Trigger     | --                   | 1                       |                  |
| Power           | --                   | 0 dB                    |                  |
| Mask Type       | --                   | Upper                   |                  |
| Trigger Mask    | --                   | Upper                   |                  |

|                             |                      |             |          |
|-----------------------------|----------------------|-------------|----------|
| Trigger Criteria            | --                   | Enter       |          |
| Active Mask                 | --                   | Upper       |          |
| Frequency                   | --                   | 0 Hz        |          |
| Amplitude                   | --                   | -100 dBm    |          |
| Build From Trace            | --                   | Trace1      |          |
| X Offset                    | --                   | 0 Hz        |          |
| Y Offset                    | --                   | 0 dB        |          |
| X Axis Type                 | --                   | Fixed       |          |
| Y Axis Type                 | --                   | Fixed       |          |
| <b>Trace</b>                |                      |             |          |
| Selected Trace              | Trace1               | Trace1      | --       |
| Trace Type                  | Clear Write          | Clear Write | --       |
| Trace Det                   | Normal               | Pos Peak    | Pos Peak |
| Detector Auto               | On                   | On          | On       |
| Trace Update                | On                   | On          | On       |
| Trace Display               | On                   | On          | On       |
| Math Function               | Off                  | Off         | --       |
| Op1                         | Trace5               | Trace5      | --       |
| Op2                         | Trace6               | Trace6      | --       |
| Offset                      | 0 dB                 | 0 dB        | --       |
| Reference                   | 0 dBm                | 0 dBm       | --       |
| <b>TG<sup>[1]</sup></b>     |                      |             |          |
| Output                      | Off                  | --          |          |
| Amplitude                   | -40 dBm              | --          |          |
| Amplitude Offset            | 0 dB                 | --          |          |
| Normalize                   | Off                  | --          |          |
| Reference Level             | 0 dB                 | --          |          |
| Reference Position          | 100%                 | --          |          |
| Reference Trace             | Off                  | --          |          |
| <b>Mode</b>                 |                      |             |          |
| Measurement Mode            | GPSA                 |             |          |
| <b>Mode Setup</b>           |                      |             |          |
| Global CF Mode              | Off, 0 Hz            | Off, 0 Hz   |          |
| <b>Meas</b>                 |                      |             |          |
| Measurement Function        | Swept SA Measurement | Normal      |          |
| <b>Measure Setup (GPSA)</b> |                      |             |          |
| <b>Swept SA Measurement</b> |                      |             |          |
| Avg Number                  | 100                  |             |          |
| Avg Mode                    | Log                  |             |          |
| Avg State                   | On                   |             |          |
| Test Limits                 | Off                  |             |          |
| Select Limit                | Limit1               |             |          |
| Limit State                 | Off                  |             |          |
| Test Trace                  | Trace1               |             |          |
| Limit Type                  | Upper                |             |          |
| X to CF                     | Relative             |             |          |
| Y to Ref                    | Relative             |             |          |
| Margin                      | Off, 0 dB            |             |          |
| Frequency                   | 0 Hz                 |             |          |
| Amplitude                   | 0 dBm                |             |          |
| Build From Trace            | Trace1               |             |          |
| X Offset                    | 0 Hz                 |             |          |
| Y Offset                    | 0 dB                 |             |          |
| Select Limit                | Limit1               |             |          |

|                                    |             |
|------------------------------------|-------------|
| Limit Type                         | Upper       |
| X To CF                            | Fixed       |
| Y To Ref                           | Fixed       |
| Margin                             | Off, 0 dB   |
| Test Trace                         | Trace1      |
| Test Limits                        | On          |
| Limit State                        | Off         |
| <b>T-Power<sup>[2]</sup></b>       |             |
| Avg Number                         | 10          |
| Avg Mode                           | Exponential |
| Avg State                          | On          |
| TP Type                            | Peak        |
| Start Line                         | 0 us        |
| Stop Line                          | 1 ms        |
| <b>ACP<sup>[2]</sup></b>           |             |
| Avg Number                         | 10          |
| Avg Mode                           | Exponential |
| Avg State                          | On          |
| Main Chan BW                       | 2 MHz       |
| Adj Chan BW                        | 2 MHz       |
| Chan Spacing                       | 2 MHz       |
| <b>Multichan Pwr<sup>[2]</sup></b> |             |
| Avg Number                         | 10          |
| Avg Mode                           | Exponential |
| Avg State                          | On          |
| Channel Span                       | 4.5 GHz     |
| Channel Sheet                      | Off         |
| Channel Freq                       | 2.25 GHz    |
| <b>Occupied BW<sup>[2]</sup></b>   |             |
| Avg Number                         | 10          |
| Avg Mode                           | Exponential |
| Avg State                          | On          |
| Max Hold                           | Off         |
| OBW Span                           | 2 MHz       |
| Power Ratio                        | 99%         |
| <b>Emission BW<sup>[2]</sup></b>   |             |
| Avg Number                         | 10          |
| Avg Mode                           | Exponential |
| Avg State                          | On          |
| Max Hold                           | Off         |
| EBW Span                           | 2 MHz       |
| EBW X dB                           | -10 dB      |
| <b>C/ N Ratio<sup>[2]</sup></b>    |             |
| Avg Number                         | 10          |
| Avg Mode                           | Exponential |
| Avg State                          | On          |
| Offset Freq                        | 2 MHz       |
| Noise BW                           | 2 MHz       |
| Carrier BW                         | 2 MHz       |
| <b>Harmo Dist<sup>[2]</sup></b>    |             |
| Avg Number                         | 10          |
| Avg Mode                           | Exponential |
| Avg State                          | On          |
| No. of Harmo                       | 10          |
| Harmonic ST                        | 1 ms        |

|                            |                |              |            |
|----------------------------|----------------|--------------|------------|
| <b>TOI<sup>[2]</sup></b>   |                |              |            |
| Avg Number                 | 10             |              |            |
| Avg Mode                   | Exponential    |              |            |
| Avg State                  | On             |              |            |
| TOI Span                   | 2 MHz          |              |            |
| <b>Measure Setup(RTSA)</b> |                |              |            |
| <b>Normal</b>              |                |              |            |
| Avg Number                 | 100            |              |            |
| Test Limits                | Off            |              |            |
| Select Limit               | Limit1         |              |            |
| Limit State                | Off            |              |            |
| Test Trace                 | Trace1         |              |            |
| Limit Type                 | Upper          |              |            |
| X to CF                    | Relative       |              |            |
| Y to Ref                   | Relative       |              |            |
| Margin                     | Off, 0 dB      |              |            |
| Frequency                  | 0 Hz           |              |            |
| Amplitude                  | 0 dBm          |              |            |
| Build From Trace           | Trace1         |              |            |
| X Offset                   | 0 Hz           |              |            |
| Y Offset                   | 0 dB           |              |            |
| <b>Density</b>             |                |              |            |
| Avg Number                 | 100            |              |            |
| Persistence                | 300 ms         |              |            |
| Pers Inf                   | Off            |              |            |
| Color Palettes             | Warm           |              |            |
| Highest Density Hue        | 100            |              |            |
| Lowest Density Hue         | 0              |              |            |
| Curve Nonlinearity         | 75             |              |            |
| Hue Truncate               | Off            |              |            |
| <b>Spectrogram</b>         |                |              |            |
| Avg Number                 | 100            |              |            |
| Display Trace              | 1              |              |            |
| Trace Selection            | Trace Number   |              |            |
| Couple Marker Trace        | Off            |              |            |
| Ref Hue                    | 0              |              |            |
| Ref Hue Pos                | 100            |              |            |
| Bottom Hue Pos             | 0              |              |            |
| <b>SSC</b>                 |                |              |            |
| Max Hold                   | Off            |              |            |
| Mark Line1                 | Off, 2.245 GHz |              |            |
| Mark Line2                 | Off, 2.255 GHz |              |            |
| Pass/Fail                  | Off            |              |            |
| Signal                     | 1              |              |            |
| Ampt Up                    | -100 dBm       |              |            |
| Ampt Down                  | -100 dBm       |              |            |
| <b>Marker</b>              |                |              |            |
| Selected Marker            | Marker1        | Marker1      | Marker1    |
| Marker Mode                | Position       | Position     | Position   |
| Reference Marker           | Marker2        | Marker2      | Marker2    |
| Marker Trace               | Auto, Trace1   | Auto, Trace1 | --         |
| Marker Freq                | 2.25 GHz       | 2.25 GHz     | 15.9730 ms |
| Marker Readout             | Frequency      | Frequency    | --         |
| Readout Auto               | On             | On           | --         |
| Line State                 | Off            | Off          | Off        |

|                             |               |               |     |
|-----------------------------|---------------|---------------|-----|
| Couple Markers              | Off           | Off           | Off |
| Marker Table                | Off           | Off           | Off |
| <b>Peak</b>                 |               |               |     |
| Cont Peak                   | Off           | Off           |     |
| Pk-Pk Search                | Maximum Value | Maximum Value |     |
| Peak Threshold              | On, -90 dBm   | On, -90 dBm   |     |
| Peak Excursion              | On, 6 dB      | On, 6 dB      |     |
| Threshold Line              | Off           | Off           |     |
| Peak Table                  | Off           | Off           |     |
| Peak Table Sort             | Amplitude     | Frequency     |     |
| Table Readout               | All           | All           |     |
| <b>Marker Func</b>          |               |               |     |
| N dB Bandwidth              | Off, -3.01 dB | Off, -3.01 dB | --  |
| Band Function               | Off           | Off           | Off |
| Marker Counter              | Off           | --            | --  |
| Gate Time                   | On, 100 ms    | --            | --  |
| <b>System<sup>[3]</sup></b> |               |               |     |
| Power On                    | Preset        | Preset        |     |
| Preset Type                 | Default       | Default       |     |
| Align Auto                  | Off           | Off           |     |
| LAN Setting Mode            | Manual        | Manual        |     |
| Display Line                | Off, -25 dBm  | Off, -25 dBm  |     |
| Graticule                   | On            | On            |     |
| HDMI                        | Off           | Off           |     |
| HDMI Resolution             | 1280*720 60Hz | 1280*720 60Hz |     |
| LCD                         | On            | On            |     |
| LCD Backlight               | 100%          | 100%          |     |
| Power Switch                | Default       | Default       |     |
| Beep Switch                 | Off           | Off           |     |
| User key                    | Off           | Off           |     |
| Language                    | English       | English       |     |

**Note<sup>[1]</sup>:** This function is only available for RSA3045-TG/RSA3030-TG working in GPSA mode.

**Note<sup>[2]</sup>:** This function is only available for RSA3000 installed with the corresponding option.

**Note<sup>[3]</sup>:** Not affected by Preset settings.

## Appendix B: Warranty

**RIGOL** TECHNOLOGIES, INC. (hereinafter referred to as **RIGOL**) warrants that the product will be free from defects in materials and workmanship within the warranty period. If a product proves defective within the warranty period, **RIGOL** guarantees free replacement or repair for the defective product.

To get repair service, please contact with your nearest **RIGOL** sales or service office.

There is no other warranty, expressed or implied, except such as is expressly set forth herein or other applicable warranty card. There is no implied warranty of merchantability or fitness for a particular purpose. Under no circumstances shall **RIGOL** be liable for any consequential, indirect, ensuing, or special damages for any breach of warranty in any case.