
Agilent Technologies E5500A SCPI Command

Reference Guide

Part number: E5500-90005
Printed in USA
June 2000

Version A.01.05

Notice

The information contained in this document is subject to change without notice.

Agilent Technologies makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Agilent Technologies shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Agilent Technologies assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Agilent Technologies.

This document contains proprietary information which is protected by copyright. All rights are reserved. No part of this document may be photocopied, reproduced, or translated to another language without prior written consent of Agilent Technologies.

U.S. Government Restricted Rights

The Software and documentation are provided with "Restricted Rights". Use, duplication or disclosure by the U.S. Government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clauses in DFARS 252.227-7013 or as set forth in subparagraph (c)(1) and (2) of the Commercial Computer Software - Restricted Rights clauses at 48 CFR 52.227-19, as applicable. The Contractor for the Software is Agilent Technologies, 3000 Hanover Street, Palo Alto, California 94304.

Trademarks

Windows NT 4.0™ is a U.S. trademarks of Microsoft Corp.

Pentium™ is a U.S. trademark of Intel Corporation

© Copyright Agilent Technologies Company 1997, 1998, 2000

SOFTWARE LICENSE TERMS

The following terms govern your use of the enclosed software programs ("Software") unless you have a separate written agreement with Agilent Technologies.

License Grant

Agilent Technologies grants you a license to Use one copy of the version of the Software identified in your documentation on any one product. "Use" means storing, loading, installing, executing or displaying the Software. You may not modify the Software or disable any licensing or control features of the Software. Additional copies of the software may be used for the sole purpose of viewing previously measured data.

Ownership

The Software is owned and copyrighted by Agilent Technologies or its third party licensors. Your license confers no title or ownership in the Software and should not be construed as a sale of any rights in the Software. Agilent Technologies third party licensors may protect their rights in the event of any violation of these terms.

Copies and Adaptations

You may only make copies or adaptations of the Software for archival purposes or when copying or adaptation is an essential step in the authorized

Use of the Software

You must reproduce all copyright notices in the original Software on all authorized copies or adaptations. You may not copy the Software onto any bulletin board or similar system.

No Disassembly or Decryption

You may not disassemble, decompile or decrypt the Software unless Agilent Technologies prior written consent is obtained. In some jurisdictions, Agilent Technologies consent may not be required for disassembly or decompilation. Upon request, you will provide Agilent Technologies with reasonably detailed information regarding any disassembly or decompilation.

Transfer

Your license will automatically terminate upon any transfer of the Software. Upon transfer, you must deliver all copies of the Software and related documentation to the transferee. The transferee must accept these License Terms as a condition to the transfer.

Third Party Software

Software may include third party software. Those third parties may protect their rights in the event of any violation of these License Terms.

Termination

Agilent Technologies may terminate your license upon notice for failure to comply with any of these License Terms. Upon termination, you must immediately destroy the Software, together with all copies, adaptations and merged portions in any form.

Export Requirements

You may not export or re-export the Software or any copy or adaptation in violation of any applicable laws or regulations.

What You'll Find in This Manual...

- **Welcome to the Agilent Technologies E5500 Phase Noise Measurement System Series of Solutions**, page 1-1
- **Configuring the SCPI Remote Interface**, page 2-1
- **Starting the SCPI Remote Interface**, page 3-1
- **Starting the SCPI Assistant**, page 4-1
- **SCPI Program Examples**, page 5-1
- **SCPI Command Reference**, page 6-1
- **Required Commands**, page 7-1
- **Abort Commands**, page 8-1
- **CALCulate Commands**, page 9-1
- **CALibrate Commands**, page 10-1
- **DISPlay Commands**, page 11-1
- **FORMat Commands**, page 12-1
- **INITiate Commands**, page 13-1
- **MMEMory Commands**, page 14-1
- **PAUSE Commands**, page 15-1
- **SENSE Commands**, page 16-1
- **SOURce Commands**, page 17-1
- **STATus Commands**, page 18-1
- **SYSTEM Commands**, page 19-1

Limited Warranty

Software

Agilent Technologies warrants that the software will perform substantially in accordance with the written materials for a period of one (1) year from the date of receipt.

Agilent Technologies does not warrant that the operation of the software will be uninterrupted or error free. In the event that this software product fails to execute its programming instructions during the warranty period, the customer's remedy shall be to return the media to Agilent Technologies for replacement. Should Agilent Technologies be unable to replace the media within a reasonable amount of time, Customer's alternate remedy shall be a refund of the purchase price upon return of all copies of the software.

Media

Agilent Technologies warrants the media upon which this product is recorded to be free from defects in materials and workmanship under normal use for a period of one (1) year from the date of purchase. In the event any media prove to be defective during the warranty period, Customer's remedy shall be to return the media to Agilent Technologies for replacement. Should Agilent Technologies be unable to replace the media within a reasonable amount of time, Customer's alternate remedy shall be a refund of the purchase price upon return of the product and all copies.

Notice of Warranty Claims

Customer shall notify Agilent Technologies in writing of any warranty claim not later than thirty (30) days after the expiration of the warranty period.

Limitation of Warranty

Agilent Technologies makes no other express warranty, whether written or oral, with respect to this product.

Any implied warranty of merchantability or fitness is limited to one (1) year duration of this written warranty.

This warranty gives specific legal rights, and Customer may also have rights which vary which vary from state to state, or province to province.

Exclusive Remedies

The remedies provided above are Customer's sole and exclusive remedies. In no event shall Agilent Technologies be liable for any direct, indirect, special, incidental, or consequential damages (including lost profit) whether based on warranty, contract, tort, or any other legal theory.

Assistance

For assistance, call your local Agilent Technologies Sales and Service Office (refer to ["Service and Support" on page -vii](#)).

Service and Support

Any adjustment, maintenance, or repair of this product must be performed by qualified personnel. Contact your customer engineer through your local Agilent Technologies Service Center. You can find a list of service centers on the web at <http://www.agilent.com/find/tmdir>.

If you do not have access to the Internet, one of these Agilent Technologies centers can direct you to your nearest Agilent Technologies representative:

United States:	Agilent Technologies Test and Measurement Call Center PO Box 4026 Englewood, CO 80155-4026 (800) 452 4844 (toll-free in US)
Canada:	Agilent Technologies Canada Ltd. 5150 Spectrum Way Mississauga, Ontario L4W 5G1 (905) 206 4725
Europe:	Agilent Technologies European Marketing Centre Postbox 999 1180 AZ Amstelveen The Netherlands (31 20) 547 9900
Japan:	Yokogawa-Agilent Technologies Ltd. Measurement Assistance Center 9-1, Takakura-Cho, Hachioji-Shi Tokyo 192, Japan (81) 426 56 7832 (81) 426 56 7840 (FAX)
Latin America:	Agilent Technologies Latin American Region Headquarters 5200 Blue Lagoon Drive, 9th Floor Miami, Florida 33126, U.S.A. (305) 267 4245 (305) 267 4288 (FAX)
Australia/New Zealand:	Agilent Technologies Australia Ltd. 31-41 Joseph Street Blackburn, Victoria 3130 Australia 1 800 629 485 (toll-free)
Asia-Pacific:	Agilent Technologies Asia Pacific Ltd. 17-21/F Shell Tower, Times Square 1 Matheson Street, Causeway Bay Hong Kong (852) 2599 7777 (852) 2506 9285 (FAX)

Contents

Notice	ii
What You'll Find in This Manual.....	v
Limited Warranty	vi
Service and Support	vii
1. Welcome to the Agilent Technologies E5500 Phase Noise Measurement System Series of Solutions	
Introduction	1-2
2. Configuring the SCPI Remote Interface	
Configuring Telnet and Sockets Interfaces	2-2
Configuring the RS-232 Interface	2-3
Configuring the VXI-11 Interface	2-4
3. Starting the SCPI Remote Interface	
Starting the SCPI Remote Interface	3-2
Starting the SCPI Remote Interface Programmatically	3-6
4. Starting the SCPI Assistant	
Starting the SCPI Assistant	4-2
5. SCPI Program Examples	
Application Environment Setup	5-2
E5500 SCPI Programming	5-4
HP Basic Program Example	5-6
6. SCPI Command Reference	
Syntax Conventions	6-2
Status Registers	6-3
Full Command List	6-7
7. Required Commands	
*CLS	7-2
*ESE	7-3
*ESR?	7-4
*IDN?	7-5
*OPC	7-6
*OPC?	7-7
*RST	7-8
*SRE	7-11
*STB?	7-12
*TST?	7-13
*WAI	7-14

8. Abort Commands	
ABORt	8-2
9. CALCulate Commands	
CALCulate:ACCumulated[PHASe]?	9-3
CALCulate:AVARiance?	9-4
CALCulate:CFSCale	9-5
CALCulate:DATA?	9-6
CALCulate:DATA:HEADer:POINTs?	9-8
CALCulate:DIPower	9-9
CALCulate:F2OScillator[:COMPare]	9-10
CALCulate:F3OScillator[:COMPare]	9-12
CALCulate:INTegral?	9-14
CALCulate:INTegral:TYPE	9-15
CALCulate:LIMit:NOISe:MAXimum	9-16
CALCulate:LIMit:NOISe:MAXimum:DELeTe	9-17
CALCulate:LIMit:NOISe:MAXimum:DELeTe:ALL	9-18
CALCulate:LIMit:NOISe:MAXimum:DISPlay	9-19
CALCulate:LIMit:NOISe:MAXimum:FAIL?	9-20
CALCulate:LIMit:NOISe:MAXimum:FAIL:ALL?	9-21
CALCulate:LIMit:NOISe:MINimum	9-22
CALCulate:LIMit:NOISe:MINimum:DELeTe	9-23
CALCulate:LIMit:NOISe:MINimum:DELeTe:ALL	9-24
CALCulate:LIMit:NOISe:MINimum:DISPlay	9-25
CALCulate:LIMit:NOISe:MINimum:FAIL?	9-26
CALCulate:LIMit:NOISe:MINimum:FAIL:ALL?	9-27
CALCulate:LIMit:SPUR:MAXimum	9-28
CALCulate:LIMit:SPUR:MAXimum:DELeTe	9-29
CALCulate:LIMit:SPUR:MAXimum:DELeTe:ALL	9-30
CALCulate:LIMit:SPUR:MAXimum:DISPlay	9-31
CALCulate:LIMit:SPUR:MAXimum:FAIL?	9-32
CALCulate:LIMit:SPUR:MAXimum:FAIL:ALL?	9-33
CALCulate:LIMit:SPUR:MINimum	9-34
CALCulate:LIMit:SPUR:MINimum:DELeTe	9-35
CALCulate:LIMit:SPUR:MINimum:DELeTe:ALL	9-36
CALCulate:LIMit:NOISe:MINimum:DISPlay	9-37
CALCulate:LIMit:SPUR:MINimum:FAIL?	9-38
CALCulate:LIMit:SPUR:MINimum:FAIL:ALL?	9-39
CALCulate:PBWidth	9-40
CALCulate:SELeCt	9-41
CALCulate:SMOothing	9-42
CALCulate:TRANsform	9-43
CALCulate:TVARiance?	9-44
CALCulate:VIEW:SPURtable	9-45
CALCulate:VIEW:TRACe	9-46
CALCulate:VIEW:XYData	9-47
CALCulate:YSHift	9-48

10.	CALibrate Commands	
	CALibrate:DETEctor:CONStant	10-2
	CALibrate:DETEctor:CONStant:METhod	10-3
	CALibrate:DETEctor:CONStant:SPUR:AMPLitude	10-5
	CALibrate:DETEctor:CONStant:SPUR:OFFSet	10-6
	CALibrate:VCO:IREStance	10-7
	CALibrate:VCO:PLLSuppress	10-8
	CALibrate:VCO:PLLSuppress:APOLe?	10-10
	CALibrate:VCO:PLLSuppress:CBWidth?	10-11
	CALibrate:VCO:PLLSuppress:DISPlay[:ALWays]	10-12
	CALibrate:VCO:PLLSuppress:ERRor	10-13
	CALibrate:VCO:PLLSuppress:ERRor:ACTion	10-14
	CALibrate:VCO:PLLSuppress:PTRange?	10-15
	CALibrate:VCO:TCONstant?	10-16
	CALibrate:VCO:TCONstant:METhod	10-17
	CALibrate:VCO:TCONstant:NOMinal	10-19
	CALibrate:VCO:TMODE	10-20
	CALibrate:VCO:VCADjust	10-21
	CALibrate:VCO:VCENter	10-22
	CALibrate:VCO:VRANge	10-23
11.	DISPlay Commands	
	DISPlay:GRAPh:BOUNds:AMPLitude	11-2
	DISPlay:GRAPh:BOUNds:AMPLitude?	11-3
	DISPlay:GRAPh:BOUNds:FREQuency	11-4
	DISPlay:GRAPh:CFSCale	11-5
	DISPlay:GRAPh:DIPower	11-6
	DISPlay:GRAPh:PBWidth	11-7
	DISPlay:GRAPh:SMOothing	11-8
	DISPlay:GRAPh:TRANsform	11-9
	DISPlay:GRAPh:YShift	11-10
	DISPlay:MEASurement:TIME?	11-11
	DISPlay:TEXT:TITLe	11-12
12.	FORMat Commands	
	FORMat[:DATA]	12-2
13.	INITiate Commands	
	INITiate[:IMMediate][:ALL]	13-2
	INITiate[:IMMediate]:CALibrate	13-3
	INITiate[:IMMediate]:MEASure	13-4
14.	MMEMory Commands	
	MMEMory:LOAD[:ALL]	14-2
	MMEMory:STORE	14-3
15.	PAUSE Commands	
	PAUSE:ABORT:AUTO	15-2

PAUSE:ADJust:LNAGain	15-3
PAUSE:ADJust:VCOCenter	15-4
PAUSE:CONNect	15-5
PAUSE:CONTInue	15-6
PAUSE:RETRy	15-7
PAUSE:SPECial	15-8

16. SENSE Commands

[SENSe:]DATA?	16-3
[SENSe:]DATA:HEADer:POINts?	16-4
[SENSe:]DCONverter:ATTenuator	16-5
[SENSe:]DCONverter:ATTenuator:AUTO	16-6
[SENSe:]DCONverter:ATTenuator:AUTO:IMMEDIATE	16-7
[SENSe:]DCONverter:BAND	16-8
[SENSe:]DCONverter:FREQuency	16-9
[SENSe:]DCONverter:IFFREquency?	16-10
[SENSe:]DCONverter:IFGain	16-11
[SENSe:]DCONverter:IFGain:AUTO	16-12
[SENSe:]DCONverter:IFGain:AUTO:IMMEDIATE	16-13
[SENSe:]DCONverter:L1BWidth	16-14
[SENSe:]DCONverter:L6BWidth	16-15
[SENSe:]DCONverter:LOPower	16-16
[SENSe:]DCONverter:LOSelect	16-17
[SENSe:]DCONverter:LOSelect:AUTO	16-18
[SENSe:]DCONverter:MEXT:BIAS	16-19
[SENSe:]DCONverter:MEXT:BIAS:STATe	16-20
[SENSe:]DCONverter:MMLO?	16-21
[SENSe:]DCONverter:PLO	16-22
[SENSe:]DCONverter:REFerence	16-23
[SENSe:]DCONverter:TSENSitivity?	16-24
[SENSe:]DCONverter:TUNE:PORT	16-26
[SENSe:]DETector:FREQuency	16-27
[SENSe:]DETector:SELEct	16-28
[SENSe:]FFT:INPut:COUPling	16-29
[SENSe:]NOISe:BBGain	16-30
[SENSe:]NOISe:FFT	16-31
[SENSe:]NOISe:MEAStpe	16-32
[SENSe:]NOISe:PULSed	16-33
[SENSe:]NOISe:QUADrature[:METHod]	16-34
[SENSe:]RANGe:FFT:AVERage:MINimum	16-35
[SENSe:]RANGe:FFT:SEGTable[:MEASurement][:QUALity]	16-36
[SENSe:]RANGe:OFFSet	16-37
[SENSe:]RANGe:SWEPt:SEGTable[:MEASurement][:QUALity]	16-38
[SENSe:]TSET:ATTenuator	16-39
[SENSe:]TSET:ATTenuator:AUTO	16-40
[SENSe:]TSET:DCBLock	16-41
[SENSe:]TSET:LNAGain	16-42

	[SENSe:]TSET:LNAGain:METhod	16-43
	[SENSe:]TSET:LNAGain:MINimum	16-44
	[SENSe:]TSET:LPF	16-45
	[SENSe:]TSET:LPF:AUTO	16-46
	[SENSe:]TSET:PLL:ATTenuator	16-47
	[SENSe:]TSET:PLL:UNLock:IGNore	16-49
	[SENSe:]TVCO	16-50
17.	SOURce Commands	
	SOURce:CALibration:FM[:DEViation]	17-2
	SOURce:CALibration:FM:INTernal:FREQuency	17-3
	SOURce:CALibration:FREQuency[:CW FIXed]	17-4
	SOURce:CALibration:POWer[:LEVel AMPLitude]	17-5
	SOURce:CARRier:FREQuency[:CW FIXed]	17-6
	SOURce:CARRier:POWer[:LEVel AMPLitude]	17-7
	SOURce:REFerence:FREQuency:DIVisor	17-8
	SOURce:REFerence:FREQuency:MULTiplier	17-9
	SOURce:REFerence:POWer[:LEVel AMPLitude]	17-10
	SOURce:RESidual:FREQuency[:CW FIXed]	17-11
	SOURce:RESidual:FREQuency:CALCulate	17-12
	SOURce:RESidual:FREQuency:DETector:COUPled	17-13
	SOURce:REFerence:FREQuency:DIVisor	17-14
	SOURce:RESidual:FREQuency:MULTiplier	17-15
	SOURce:RESidual:POWer[:LEVel AMPLitude]	17-16
18.	STATus Commands	
	STATus:ADVise:ENABle	18-4
	STATus:OPERation:CONDition?	18-5
	STATus:OPERation:ENABle	18-6
	STATus:OPERation:EVENT?	18-7
	STATus:OPERation:NTRansition	18-8
	STATus:OPERation:PTRansition	18-9
	STATus:PRESet	18-10
	STATus:QUEStionable:CONDition?	18-11
	STATus:QUEStionable:ENABle	18-12
	STATus:QUEStionable:[EVENT]?	18-13
	STATus:QUEStionable:NTRansition	18-14
	STATus:QUEStionable:PTRansition	18-15
19.	SYSTem Commands	
	SYSTem:ASSet:CALSource	19-3
	SYSTem:ASSet:CARRier	19-4
	SYSTem:ASSet:COUNter	19-5
	SYSTem:ASSet:DCONverter	19-6
	SYSTem:ASSet:DCONverter:USAGe	19-7
	SYSTem:ASSet:FFT	19-8
	SYSTem:ASSet:FFT:CUSTom[:SEGTable]	19-9
	SYSTem:ASSet:PSHifter	19-10

SYSTem:ASSet:REFerence	19-11
SYSTem:ASSet:RESidual	19-12
SYSTem:ASSet:SLAVe:SOURce	19-13
SYSTem:ASSet:SLAVe:SOURce:PORT	19-14
SYSTem:ASSet:SWANalyzer	19-15
SYSTem:ASSet:SWANalyzer:CUSTom[:SEGTaBle]	19-16
SYSTem:ASSet:TBASe	19-17
SYSTem:ASSet:TSET	19-18
SYSTem:ERRor?	19-19
SYSTem:GUI:LLOCkout[:STATe]	19-22
SYSTem:GUI:REMote[:STATe]	19-23
SYSTem:HELP:HEADers?	19-24
SYSTem:PATH:CARRier	19-25
SYSTem:PATH:DCBBanalyzer	19-26
SYSTem:PATH:FFTAnalyzer	19-27
SYSTem:PATH:SWANalyzer	19-28
SYSTem:PATH:TVCO	19-29
SYSTem:TSTart?	19-30
SYSTem:VERSIon?	19-31

1

Welcome to the Agilent Technologies E5500 Phase Noise Measurement System Series of Solutions

What You'll Find in This Chapter...

- [Introduction, page 1-2](#)

Introduction

Standard Commands for Programmable Instruments (SCPI) defines how you communicate with an instrument from a computer. It a programming language designed specifically for electronic test and measurement instruments.

The E5500A phase noise measurement system programming commands comply with the SCPI standard. For information regarding the correct syntax for forming and sending command strings via the GPIB, the IEEE 488.2 standard, or the SCPI standard, refer to the following books:

- Hewlett-Packard Company. *Tutorial Description of Hewlett-Packard Interface Bus*, 1987.
- Hewlett-Packard Company. *A Beginner's Guide to SCPI*, 1991.
- Hewlett-Packard Company. *SCPI—Standard Commands for Programmable Instruments*, 1995.
- International Institute of Electrical and Electronics Engineers. *IEEE Standard 488.1-1987, IEEE Standard Digital Interface for Programmable Instrumentation*. New York, NY, 1987.
- International Institute of Electrical and Electronics Engineers. *IEEE Standard 488.2-1987, IEEE Standard Codes, Formats, Protocols and Common commands For Use with ANSI/IEEE Std 488.1-1987*. New York, NY, 1987.

2

Configuring the SCPI Remote Interface

What You'll Find in This Chapter...

- [Configuring Telnet and Sockets Interfaces](#), page 2-2
- [Configuring the RS-232 Interface](#), page 2-3
- [Configuring the VXI-11 Interface](#), page 2-4

Overview

The E5500 SCPI Interface provides connectivity between the E5500 Phase Noise Subsystem and the user's application programming environment. Multiple types of interfaces and protocols are supported. Before using an interface with the E5500 SCPI software, some system configuration may be required. You only need to configure the interfaces you plan to use.

Configuring Telnet and Sockets Interfaces

These interfaces use TCP/IP protocol via a LAN connection. In order to use these you must have Windows networking installed and configured. An appropriate LAN card must also be installed and configured if you want to control the E5500 Subsystem from a remote computer.

Configuring the RS-232 Interface

This interface provides direct serial port connection using a null-modem cable. COM port settings as set by the Windows operating system configuration utility are used. To view or modify settings click:

Start > Settings > Control Panel > Ports.

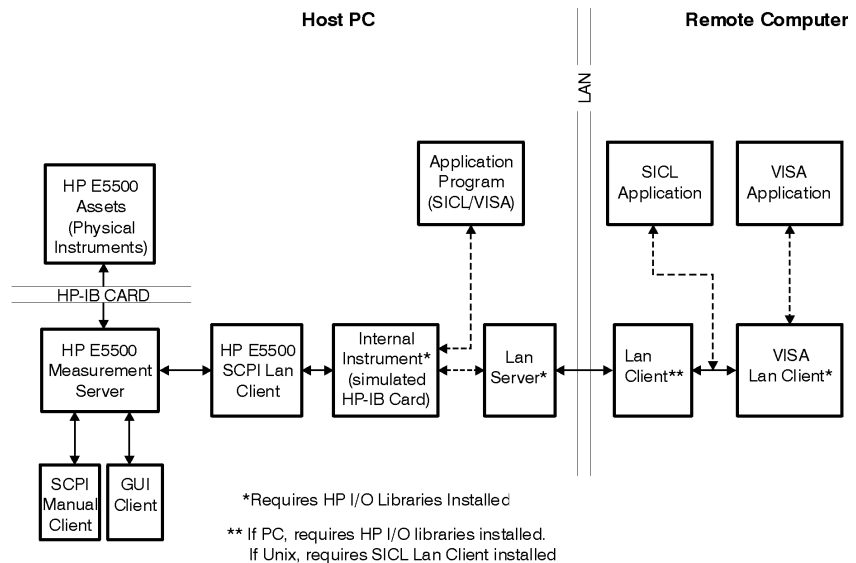
Configuring the VXI-11 Interface

This interface uses VXI-11 protocol which provides full GPIB emulation via a LAN connection. It requires a software simulated GPIB card called an "Internal Instrument". installed and configured in Agilent/HP I/O Libraries. The E5500 SCPI Interface is then used to open this simulated GPIB card and connect it to the Phase Noise Subsystem. Then I/O Libraries software and an upgrade which adds the Internal Instrument interface is included on the E5500 CD-ROM.

NOTE

The Agilent/HP I/O Libraries (version H.01.02 or greater) must be installed (including the Agilent/HP I/O Library Upgrade) before beginning this procedure. Refer to the E5500A Installation Guide for more information.

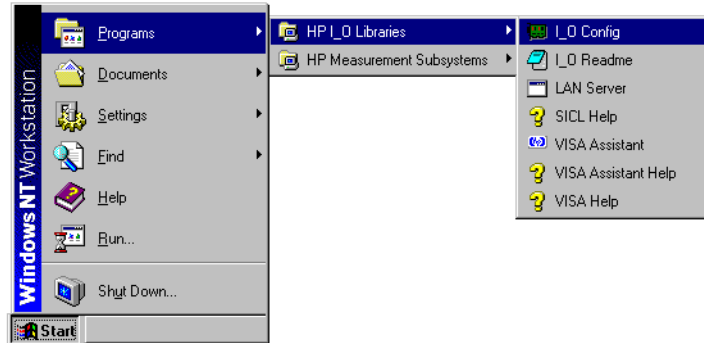
The following diagram shows the relationship between the E5500 Phase Noise Subsystem, the E5500 SCPI Interface, the I/O Libraries Internal Instrument, and the users application program running on either the same or a remote computer.



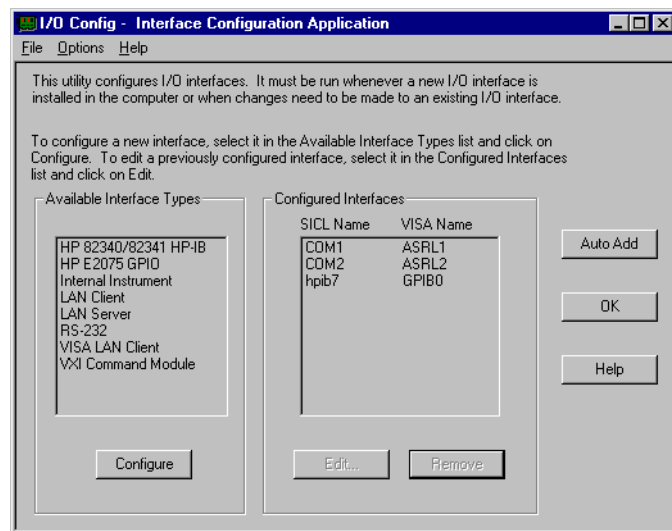
lan.cdr

The Agilent/HP I/O Libraries, I/O Config utility must be used to configure an Internal Instrument interface before attempting to open a VXI-11 interface with the E5500 SCPI Interface. To do so, do the following.

1. Click the **Start** button, point to **Programs**, Point to **Agilent/HP I_O Libraries**, then click **I_O Config**.



2. The following dialog box will appear.

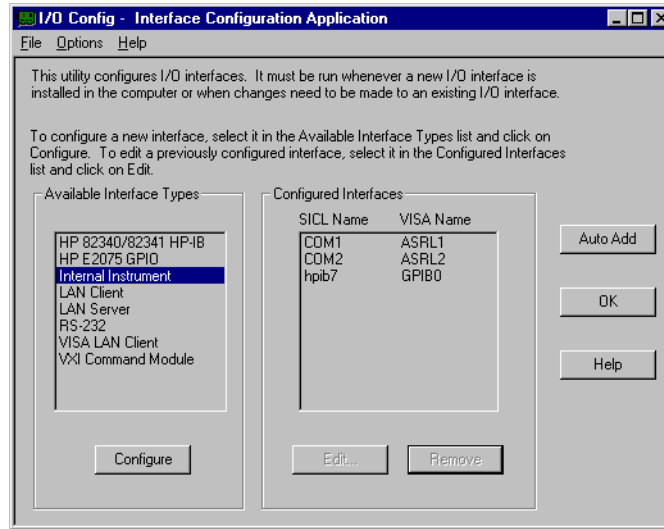


3. Select **Internal Instrument** in the left pane and click the **Configure** button.

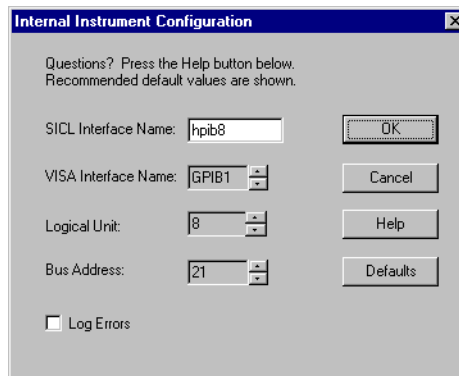
If **Internal Instrument** is not available as a choice in the left pane, refer to your system's "Installation Guide" and install the **Agilent/HP I/O Library Upgrade**.

Configuring the SCPI Remote Interface

Configuring the VXI-11 Interface

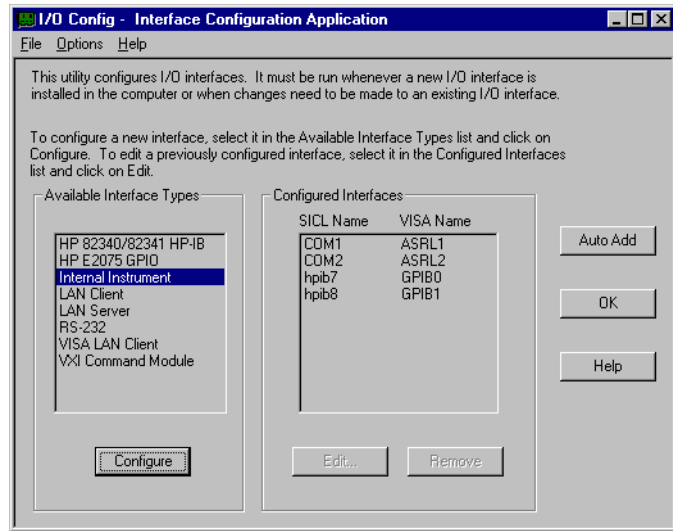


4. The following dialog box will appear. Accept the default settings by clicking the **OK** button. The software will choose the next available “hpib(x)” number (usually hpib8 and GPIB1).



NOTE

The internal instrument required to use the SCPI (LAN) Client has been added to the list in the right pane (Configured Interfaces).



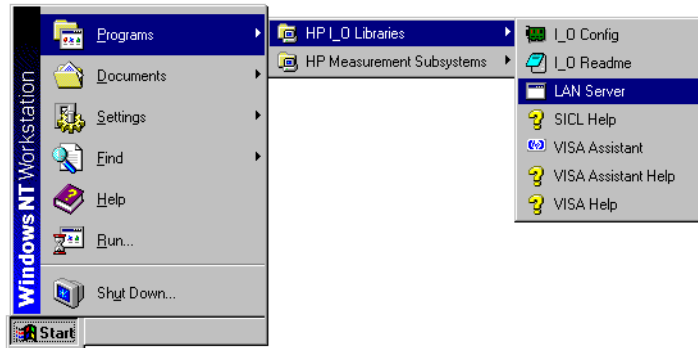
Activating the LAN Server (Optional)

NOTE

Activating the LAN Server is only required if you wish to control the E5500 Phase Noise System using the E5500 SCPI VXI-11 interface from a remote computer.

5. Click the **Start** button, point to **Programs**, Point to **Agilent/HP I_O Libraries**, then click **LAN Server**.

Configuring the SCPI Remote Interface
Configuring the VXI-11 Interface



6. The LAN Server dialog box appears. This dialog box does not display any information, but it needs to be active for the LAN Server to function. The window can be minimized (button in the upper-right corner of the dialog box).



The LAN Server is now ready to communicate with a remote PC or UX workstation.

3

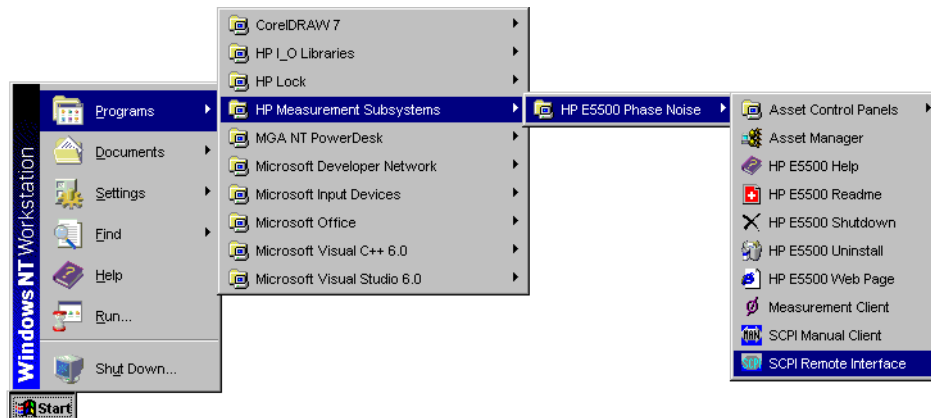
Starting the SCPI Remote Interface

What You'll Find in This Chapter...

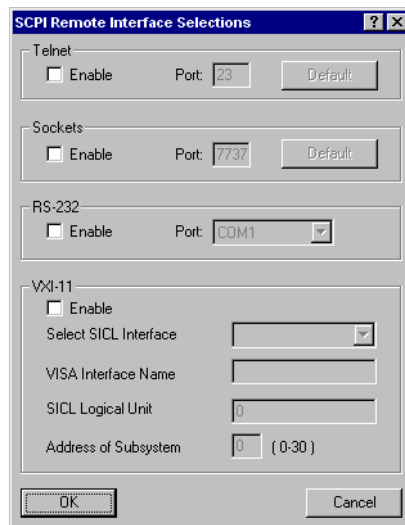
- **Starting the SCPI Remote Interface**, [page 3-2](#)
 - **Click the OK button.**, [page 3-5](#)
- **Starting the SCPI Remote Interface Programmatically**, [page 3-6](#)

Starting the SCPI Remote Interface

1. Click the **Start** button, point to **Programs**, Point to **Agilent Subsystems**, Point to **E5500 Phase Noise**, then click **E5500 SCPI Interface**.

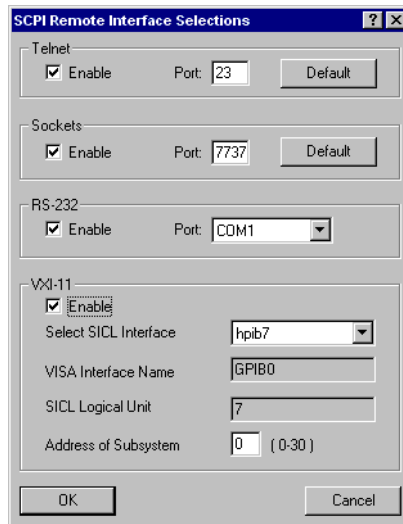


2. The following dialog box appears.



This dialog is used to open one or more interfaces to the SCPI Remote Interface. These interfaces provide connectivity between the Phase Noise Subsystem and the user's application programming environment. Once connected, SCPI commands may be sent to the Subsystem via the SCPI Interface as if it is an instrument on a GPIB bus.

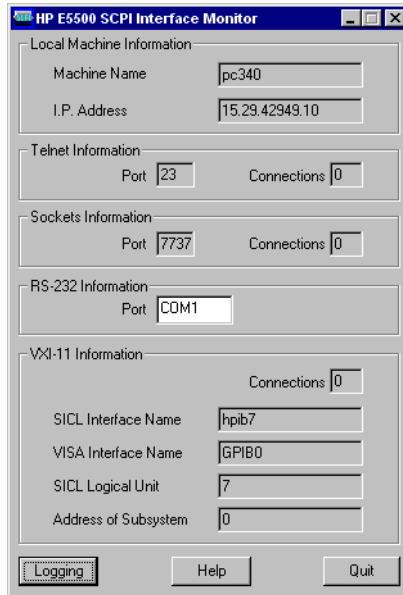
- Telnet - TCP/IP protocol, requires installed and configured networking software.
 - Sockets - TCP/IP protocol, requires installed and configured networking software.
 - RS-232 - Serial COM Port Interface (direct connection)
 - VXI-11 - VXI-11 protocol, requires Agilent/HP I/O Libraries software with a “Internal Instrument” type interface configured, see [Chapter 2, “Configuring the SCPI Remote Interface”](#).
3. Choose which protocol you wish to use by checking either **Telnet Enable**, **Sockets Enable**, **RS-232 Enable** or **VXI-11 Enable**.
 4. If you selected **VXI-11 Enable**, in the **Select SICL Interface** pull-down list, select **hpib (x)** (**hpib8**, for example). Only the Internal Instrument type interfaces defined in [Chapter 2, “Configuring the SCPI Remote Interface”](#), will be listed. You may also change the bus address of the subsystem if desired (default is 0).
 5. Click the **OK** button.



Starting the SCPI Remote Interface

Starting the SCPI Remote Interface

6. Checking all four “enable” check-boxes produces the following dialog box. The Telnet, Sockets, RS-232, and VXI-11 (VISA, SICL, Serial, and GPIB) information appears. Click the **OK** button.



7. The selected interface is now open and connected to the E5500 phase noise measurement server. Local machine (PC) information appears in the top section, the bottom sections contain information about the selected interface(s).

NOTE

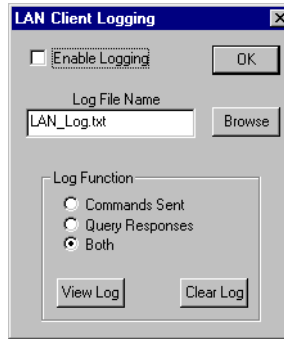
You must open an interface with the E5500 SCPI Interface **before** running any application software.

SCPI Logging Function

1. Click the **Logging** button. The SCPI Logging dialog box appears.

When enabled, the SCPI Interface Logging dialog box allows you to log Commands Sent, or Query Responses, or Both. The logged functions can be saved as a text file (*.txt) using a name and directory of your choice.

2. Click the **OK** button.



Starting the SCPI Remote Interface Programmatically

The SCPI Interface can be started programmatically from a DOS command prompt, a WinNT shortcut, or from a program using the following syntax:

`lanSCPIclient.exe \driver (“name” | telnet | sockets | serial)`

Only one type of interface may be opened via the command line. The \driver option takes one of the following arguments:

“name” Opens a VXI-11 interface where **“name”** is a quoted string SIDL interface name to open.

Example: `lanScpiClient.exe \driver “hpib7”`

telnet Opens a telnet interface. The port setting last opened is used.

Example: `lanScpiClient.exe \driver telnet`

sockets Opens a sockets interface. The port setting last opened is used.

Example: `lanScpiClient.exe \driver sockets`

serial Opens a RS-232 interface. The port setting last opened is used.

Example: `lanScpiClient.exe \driver serial`

DOS Command Prompt Example:

- Telnet:
 - `C:\Program Files\Agilent\Measurement and Stimulus Subsystems\E5500 Phase Noise\lanSCPIclient.exe \driver telnet`
- Sockets:
 - `C:\Program Files\Agilent\Measurement and Stimulus Subsystems\E5500 Phase Noise\lanSCPIclient.exe \driver sockets`
- Serial (RS-232):
 - `C:\Program Files\Agilent\Measurement and Stimulus Subsystems\E5500 Phase Noise\lanSCPIclient.exe \driver serial`
- VXI-11:
 - `C:\Program Files\Agilent\Measurement and Stimulus Subsystems\E5500 Phase Noise\lanSCPIclient.exe \driver "hpib8"`

Windows NT Shortcut Example:

For a Windows NT shortcut you must use quotes due to spaces in the path

- Telnet:
 - **"C:\Program Files\Agilent\Measurement and Stimulus Subsystems\E5500 Phase Noise\lanSCPIclient.exe" \driver telnet**
- Sockets:
 - **"C:\Program Files\Agilent\Measurement and Stimulus Subsystems\E5500 Phase Noise\lanSCPIclient.exe" \driver sockets**
- Serial (RS-232):
 - **"C:\Program Files\Agilent\Measurement and Stimulus Subsystems\E5500 Phase Noise\lanSCPIclient.exe" \driver serial**
- VXI-11:
 - **"C:\Program Files\Agilent\Measurement and Stimulus Subsystems\E5500 Phase Noise\lanSCPIclient.exe" \driver "hpib8"**

4

Starting the SCPI Assistant

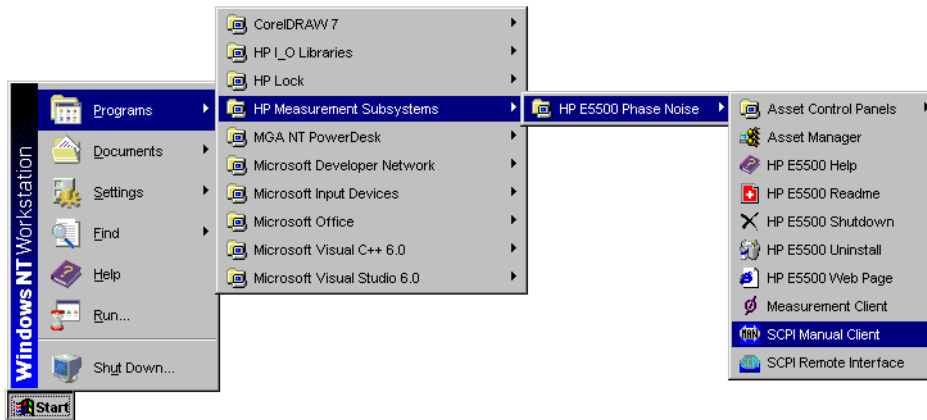
What You'll Find in This Chapter...

- **Starting the SCPI Assistant**, [page 4-2](#)
 - **SCPI Logging Function**, [page 4-3](#)

Starting the SCPI Assistant

The Agilent Technologies E5500 SCPI Assistant is useful for program development, allowing the user to enter one SCPI command at a time and view queries.

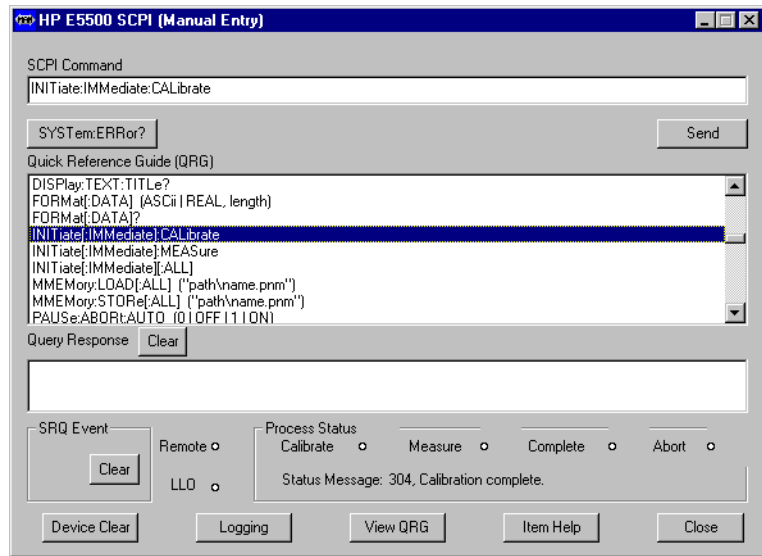
- To view the SCPI Assistant, Click the **Start** button, point to **Programs**, Point to **Agilent Subsystems**, Point to **E5500 Phase Noise**, then click **E5500 SCPI Assistant**.



The SCPI Assistant interface allows the user to enter SCPI commands directly using the keyboard and then send the commands directly using the keyboard's Enter key.

A Quick Reference Guide (QRG) window lists the SCPI command which (by double-clicking) can be placed in the SCPI command window and sent by using the keyboard's Enter key. To view the QRG (Quick Reference Guide) and click the **View QRG** button. The QRG can be printed using the standard Window printing capability.

Query responses appear in the Query Response Window.

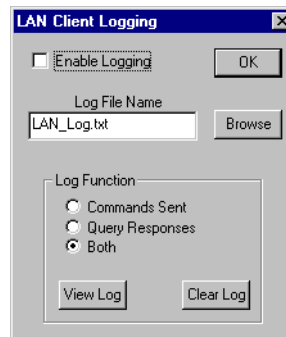


SCPI Logging Function

1. Click the **Logging** button. The SCPI Logging dialog box appears.

When enabled, the SCPI Assistant Logging dialog box allows you to log Commands Sent, or Query Responses, or Both. The logged functions can be saved as a text file (*.txt) using a name and directory of your choice.

2. Click the **OK** button.



5

SCPI Program Examples

What You'll Find in This Chapter...

- **Application Environment Setup**, [page 5-2](#)
- **E5500 SCPI Programming**, [page 5-4](#)
- **HP Basic Program Example**, [page 5-6](#)

Application Environment Setup

Application software communicates with the E5500 Measurement Server through the interface opened with the SCPI Interface. The SCPI Interface must open the selected interface before running your application software. Information about the selected interface and the host PC is displayed in the SCPI Interface Monitor dialog. The information your application needs to connect to this interface depends upon the type of I/O your application supports (SICL or VISA) and if your application is running on the same PC as the E5500 or a remote computer.

The following are language specific examples of application setup. In all cases the E5500 SCPI Interface has opened a VXI-11 interface with the SICL Interface Name of "hpib8" and the address of Subsystem "0".

HP BASIC for Windows on same PC as E5500 or Remote PC

HPBW autost file must use the HPIBS driver to open the interface as follows:

- LOAD BIN "HPIBS; DEV hpib8 ISC 7"! Same PC as E5500
or
- LOAD BIN "HPIBS; DEV lan[pc52]:hpib8 ISC 7"! Remote PC name or IP
 - "pc52" is the name of the PC running the E5500. You may also use an IP address here. "hpib8" is the SICL interface name opened with the SCPI LAN Client.

NOTE

Requires Agilent/HP I/O Libraries LAN CLIENT on Remote PC.

- Your program would then:

```
OUTPUT 700; "*IDN?"  
ENTER 700; Id$
```

For more information on autost file see HPBW "Installing and Using Guide", Chapter 5.

RMBUX on remote workstation

Users RMBUX ~/.rmbrc file must open interface as follows:

- INTERFACE 9="lan[pc52]:hpib8"; normal

- “pc52” is the name of the pc running the E5500. You may also use an IP address here. “hpib8” is the SICL interface name opened with the SCPI LAN Client.

NOTE

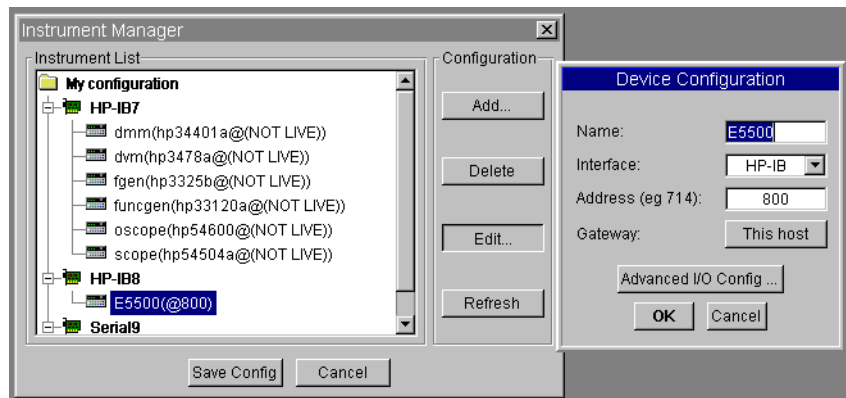
RMBUX Requires that the SICL LAN CLIENT be installed on your UNIX system, see *your* UNIX system administrator.

- Your program would then:

```
OUTPUT 900;"*IDN?"  
ENTER 900;Id$
```

**Agilent/HP Vee on
same PC as E5500 or
Remote PC**

Use the Instrument Manager in the I/O menu to configure an interface and device as defined in the SCPI LAN Client. If on a remote computer add the name or IP address of the PC running the E5500 in the Gateway: pane.



E5500 SCPI Programming

Generally, any E5500 program must perform the following tasks:

1. Setup the measurement definition.
 - * Setup the device under test (DUT).
2. Initiate measurement and determine when complete.
3. Read the data from the measurement.
 - * Process, display and store the data.

* = These tasks are user specific and will not be discussed.

As with most programming tasks there is more than one method to accomplish these tasks. Several methods will be discussed here.

Setup the measurement definition.

Before programming a measurement you should first make the desired measurement manually using the Measurement Client GUI. Once you have the measurement defined and are satisfied with the results, save the measurement setup using the File > Save As menu selection. This saves the complete measurement definition and the data from the last measurement performed to a file. A file should be saved for each Measurement Type (Absolute, Noise, etc.) you wish to make. A few parameters (e.g. carrier freq.) may be adjusted after recalling a measurement setup file.

Your program can then load the desired measurement file using the command `MMEMory:LOAD[:ALL]` (“path\name.pnm”), adjust any parameters as necessary and begin the measurement. This is the simplest method of measurement setup under program control.

NOTE

Do not use special characters, including the underscore (`_`) character, in pathnames.

Users who do not wish to use pre-defined measurement files to setup their measurement should still use the Measurement Client GUI to develop and successfully make their measurement. Start from the File > New menu state and then adjust all required parameters. The command `*RST` is the equivalent of File > New and should begin the setup section of your program followed by the commands for all adjusted parameters.

Initiate measurement and determine when complete.

The SCPI command `INITiate[:IMMediate][:ALL]` begins a calibration and measurement sequence. A wait of ~15 seconds is recommended after issuing this command as the E5500 requires nearly 100% CPU time when initializing the system assets (instruments).

Once a measurement is begun, the SCPI status system may be used to monitor measurement progress and determine when complete. This may be done by polling the Operation Condition register with the query, `STATus:OPERation:CONDition?` and waiting for the “Idle” bit. Another method is to send `*OPC` after the `INITiate` command which will set the “Operation Complete” bit of the Event Status Register (`*ESR?`).

System errors should be checked while measuring or at least when a measurement is complete using the SCPI query `SYSTEM:ERRor?`. The error que should be read until message 0, “No error” is received and appropriate action, if any taken upon non 0 errors. In addition to standard SCPI errors, the following E5500 error messages may occur:

- *100-199 Abort messages* - measurement aborted.
- *200-299 Pause point messages* - measurement paused. This may also trigger an abort if `PAUSE:ABORt:AUTO` is ON(default=ON).
- *300-499 Status messages* - informative only, no action required. Only reported if `STATus:ADVise:ENABLE` is ON (default=OFF).
- *500-599 Warning messages* - advisory but measurement can proceed on its own.

Read the data from the measurement.

After completing a measurement and checking system errors you are ready to read the data. The simplest method of data output uses the SCPI queries `[:SENSE]:DATA:HEADer:POINts?` and `[:SENSE]:DATA?`. The `POINts?` query returns the total number of points measured. The `DATA?` query returns one line per point in the format: frequency, amplitude, spur flag. The amplitude is always in the Spectral density of phase fluctuations (dB/Hz) format and the spur flag (0/1) is 1 if the data point is a spur.

Commands in the `CALCulate: SCPI` tree can be used to output subsets of the data, data in other formats, or processed data.

HP Basic Program Example

The following program example demonstrates use of the above techniques in an HP BASIC program. It will run with your system in Demo Mode. The program recalls a measurement setup file and then makes measurements at several carrier frequencies.

Lines to Note:

- 210 Timeout - a minimum 5 second timeout period is recommended.
- 570 Recall measurement setup file.
- 650 Set carrier frequency.
- 710 Begin measurement sequence.
- 900 Wait for measurement to complete subroutine. This polls the Status Operation Condition register to monitor measurement progress.
- 1050 Check System Errors subroutine.
- 1280 Read data subroutine. This uses SENSE:DATA? output.

```
-----  
10 ! RE-SAVE "DEMO"  
20 ! *****  
30 ! * E5500 Phase Noise Measurement Subsystem  
40 ! *  
50 ! * Demonstration program to setup and initiate a  
60 ! * measurement and then output the data.  
70 ! *  
80 ! * Copyright 1998 Agilent Technologies  
90 ! *  
100 ! *****  
110 !  
120 OPTION BASE 1  
130 INTEGER Isc,Soc,Err,I,J,M,Abort_flg,Points  
140 DIM Msg$(160),Path$(80),File$(5)[30],Freq$(5)[15]  
150 !  
160 Isc=8 ! Interface select code as set in HPBW  
autost file.  
170 ASSIGN @Pn TO 800 ! Address of E5500. Always 00.  
180 !  
190 ! Initialize system.  
200 !  
210 ON TIMEOUT Isc,5 GOTO Time_out ! Recommend 5 second  
timeout.  
220 CLEAR SCREEN  
230 OUTPUT @Pn;"*RST; *CLS; STAT:PRES; *IDN?"
```

```

240 ENTER @Pn;Msg$
250 PRINT Msg$
260 ! OUTPUT @Pn;"STAT:ADVise:ENABLE ON" ! Optional enable
status messages.
270 ! -----
280 ! Load measurement setup file and frequency arrays.
290 ! -----
300 !
310 ! Measurement setup file path and names.
320 Path$="c:\Program Files\Agilent\Measurement and
Stimulus\E5500\"
330 DATA Demomode.pnm, Confidence.pnm, EOD
340 !
350 I=1
360 READ File$(I)
370 WHILE File$(I)<>"EOD"
380 I=I+1
390 READ File$(I)
400 END WHILE
410 !
420 ! Carrier frequencies to measure for each setup.
430 DATA 500E6, 900E6, EOD
440 !
450 I=1
460 READ Freq$(I)
470 WHILE Freq$(I)<>"EOD"
480 I=I+1
490 READ Freq$(I)
500 END WHILE
510 ! -----
520 ! Start of measurement loop.
530 ! -----
540 I=1
550 WHILE File$(I)<>"EOD" ! Load a measurement setup file.
560 !
570 OUTPUT @Pn;"MMEM:LOAD ""&Path$&File$(I)&""
580 PRINT
590 PRINT "Measurement setup: ";File$(I)
600 GOSUB Read_err ! Check for file load errors.
610 WAIT 1
620 J=1
630 WHILE Freq$(J)<>"EOD" ! Set carrier frequency and
measure.
640 !
650 OUTPUT @Pn;"SOUR:CARR:FREQ "&Freq$(J)

```


SCPI Program Examples

HP Basic Program Example

```
660 !
670 ! Code to setup Device Under Test goes here.
680 !
690 Restart: Abort_flg=0
700 !
710 OUTPUT @Pn;"INIT:ALL"    ! Start new calibration &
measurement.
720 !
730 PRINT "      Measuring carrier at: ";Freq$(J);" Hz"
740 GOSUB Wait_until_done    ! Monitor status and errors.
750 !
760 ! Read the data if measurement was sucessful.
770 IF NOT Abort_flg THEN GOSUB Read_data
780 !
790 J=J+1
800 END WHILE
810 I=I+1
820 END WHILE
830 !
840 !
850 LOCAL @Pn
860 DISP "End of Program."
870 STOP ! End of main program.
880 !      -----
890 !
900 Wait_until_done:      !
910 REPEAT
920 DISP ""
930 WAIT 1
940 OUTPUT @Pn;"STAT:OPER:COND?"
950 ENTER @Pn;Soc
960 IF BIT(Soc,0) THEN DISP "Calibrating"
970 IF BIT(Soc,4) THEN DISP "Measuring"
980 IF BIT(Soc,8) THEN DISP "Paused"
990 IF BIT(Soc,9) THEN DISP "Idle"
1000 GOSUB Read_err
1010 WAIT 1
1020 UNTIL BIT(Soc,9)    ! Idle state
1030 RETURN
1040 !
1050 Read_err: !
1060 REPEAT
1070 OUTPUT @Pn;"SYST:ERR?"
1080 ENTER @Pn;Err,Msg$
```

```

1090 IF Err<>0 THEN
1100 BEEP 300,.1
1110 PRINT " ";Err;Msg$
1120 IF Err=-250 THEN ! File error
1130 PRINT
1140 PRINT "File load error! Check path, file name and file
exists."
1150 PRINT Path$&File$(I)
1160 LOCAL @Pn
1170 STOP
1180 END IF
1190 IF Err>=100 AND Err<=199 THEN ! Abort messages.
1200 Abort_flg=1
1210 INPUT "Measurement Aborted! Try again? (Y/N)",Ans$
1220 IF UPC$(Ans$[1,1])="Y" THEN GOTO Restart
1230 END IF
1240 END IF
1250 UNTIL Err=0 ! No error
1260 RETURN
1270 !
1280 Read_data: !
1290 OUTPUT @Pn;"DATA:HEAD:POIN?"
1300 ENTER @Pn;Points
1310 DISP "Reading";Points;" Data Points..."
1320 ALLOCATE Trace_data(Points,3)
1330OUTPUT @Pn;"SENS:DATA?" ! Data triples - freq, amplitude,
spur flag
1340 ENTER @Pn;Trace_data(*)
1350 !
1360 ! Data processing, storage or display code goes here.
1370 !
1380 PRINT " Point 1 =" ;Trace_data(1,2);"db
@" ;Trace_data(1,1);"Hz"
1390 PRINT " Point" ;Points;"=" ;Trace_data(Points,2);"db
@" ;Trace_data(Points,1);"Hz"
1400 !
1410 DEALLOCATE Trace_data(*)
1420 DISP ""
1430 RETURN
1440 !
1450 Time_out: !
1460 BEEP 300,.2
1470 PRINT
1480 PRINT "*** Timeout Error! *** Check system and try
again."
1490 !

```

6

SCPI Command Reference

What You'll Find in This Chapter...

- **Syntax Conventions**, [page 6-2](#)
- **Status Registers**, [page 6-3](#)
- **Full Command List**, [page 6-7](#)

Syntax Conventions

Table 6-1 Notation Conventions and Definitions

Convention	Additional Information
< >	Angle brackets indicate values entered by the programmer
	"Or" indicates a choice of one element from a list. For example, but not both
[]	Square brackets indicate that the enclosed items are optional.
{ }	When several items are enclosed by braces, one, and only one of these elements must be selected.
~	A tilde indicates a valid range of values
Integer	An ASCII string representing an integer. This is defined by the IEEE 488.2 <NR1> format.
Real	An ASCII string representing a real number. This is defined by the IEEE 488.2 <NR2> or <NRf> formats.
NA	not Applicable.

- If "Command Only" is not listed, the command can be issued as a query by including the question mark "?" character. For example,

:CALibrate:DETEctor:CONStant

command can be sent as a query by sending

:CALibrate:DETEctor:CONStant?
- A colon (:) indicates branching points on the command tree
- A semicolon (;) sends multiple commands within a single program message. The command parser assumes the second command comes from the same branch as the preceding command. Use ; to reset the command parser to the base of the command tree.

Status Registers

Event Status Register (*ESR?) Mask is *ESE (value)

- bit 0 = 1 = Operation Complete
- bit 1 = 2 = Request Control
- bit 2 = 4 = Query Error
- bit 3 = 8 = Device Dependent Error. (Summary of QSR bits 9,10,11)
- bit 4 = 16 = Execution Error. All API call error returns.
- bit 5 = 32 = Command Error
- bit 6 = 64 = User Request
- bit 7 = 128 = Power On

Status Byte (*STB?) Mask is *SRE (value)

- bit 0 = 1 = Not Used
- bit 1 = 2 = Not Used
- bit 2 = 4 = Error/Event Queue
- bit 3 = 8 = Summary of Questionable Status Register
- bit 4 = 16 = MAV (Message Available)
- bit 5 = 32 = Summary of Standard Event Status Register
- bit 6 = 64 = RQS (SRQ State)
- bit 7 = 128 = Summary of Operation Status Register

Status Operation Register (STATus:OPERation:CONDition?)

- bit 0 = 1 = Calibrating
- bit 1–3 = Not Used
- bit 4 = 16 = Measuring
- bit 5–7 = Not Used
- bit 8 = 256 = Paused
- bit 9 = 512 = Idle
- bit 10–15 = Not Used

Status Questionable Register (STATus:QUESTionable:CONDition?)

- bit 0–8 = Not Used
- bit 9 = 512 = Warning message available.
- bit 10 = 1024 = Pause message available
- bit 11 = 2048 = Abort message available
- bit 12 = 4096 = Server Status message available.
- bit 13–15 = Not Used

Status Registers

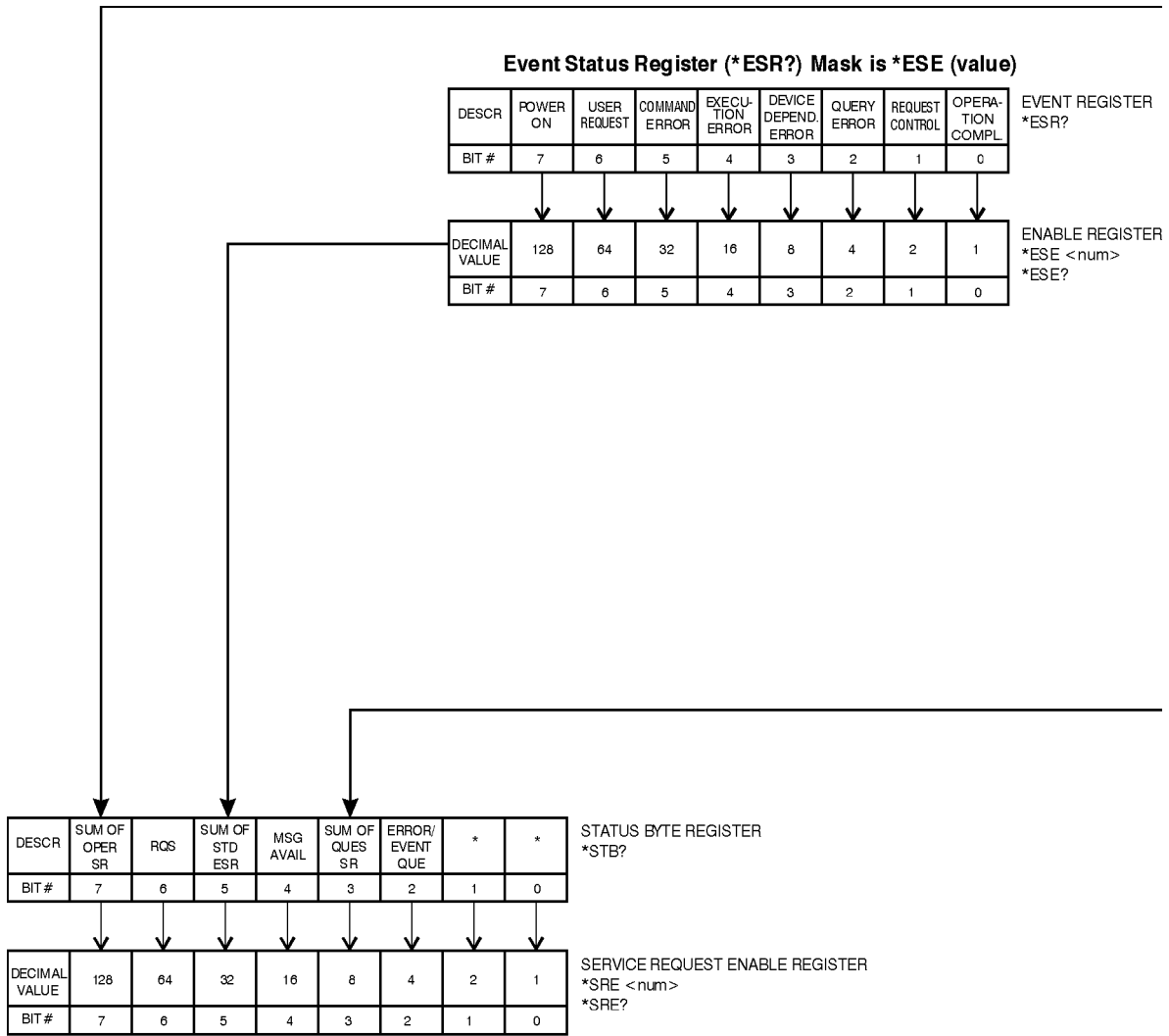
Overlapped commands which start Pending Operations.

These are the only commands to which *OPC, *OPC? and *WAI apply:

INITiate:CALibrate (Calibrate)

INITiate:MEASure (Measure)

INITiate:IMMEDIATE:ALL (Calibrate and Measure)



* NOT USED

statreg1.cdr

SCPI Command Reference

Status Registers

Status Operation Register (STATus:OPERation:CONDition?)

DESCR	*	*	*	*	*	*	IDLE	PAUSED	*	*	*	MEAS-URING	*	*	*	CALI-BRATING
BIT#	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

CONDITION REGISTER
STAT:OPER:COND?

	┌	┌	┌	┌	┌	┌	┌	┌	┌	┌	┌	┌	┌	┌	┌	┌
--	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

NEGATIVE TRANSITION FILTER
STAT:OPER:NTR <num>
STAT:OPER:NTR?

	┐	┐	┐	┐	┐	┐	┐	┐	┐	┐	┐	┐	┐	┐	┐	┐
--	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

POSITIVE TRANSITION FILTER
STAT:OPER:PTR <num>
STAT:OPER:PTR?

DECIMAL VALUE	32,768	16,384	8,192	4,096	2,048	1,024	512	256	128	64	32	16	8	4	2	1
BIT#	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

EVENT REGISTER
STAT:OPER:EVENT?

DECIMAL VALUE	32,768	16,384	8,192	4,096	2,048	1,024	512	256	128	64	32	16	8	4	2	1
BIT#	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

ENABLE REGISTER
STAT:OPER:ENAB <num>
STAT:OPER:ENAB?

← To
*STB OPER
Summary
bit 7

Status Questionable Register (STATus:QUESTionable:CONDition?)

DESCR	*	*	*	SERVER STATUS MSG	ABORT	PAUSE	WARN MSG AVAIL	*	*	*	*	*	*	*	*	*
BIT#	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

CONDITION REGISTER
STAT:QUES:COND?

	┌	┌	┌	┌	┌	┌	┌	┌	┌	┌	┌	┌	┌	┌	┌	┌
--	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

NEGATIVE TRANSITION FILTER
STAT:QUES:NTR <num>
STAT:QUES:NTR?

	┐	┐	┐	┐	┐	┐	┐	┐	┐	┐	┐	┐	┐	┐	┐	┐
--	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

POSITIVE TRANSITION FILTER
STAT:QUES:PTR <num>
STAT:QUES:PTR?

DECIMAL VALUE	32,768	16,384	8,192	4,096	2,048	1,024	512	256	128	64	32	16	8	4	2	1
BIT#	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

EVENT REGISTER
STAT:QUES:EVENT?

DECIMAL VALUE	32,768	16,384	8,192	4,096	2,048	1,024	512	256	128	64	32	16	8	4	2	1
BIT#	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

ENABLE REGISTER
STAT:QUES:ENAB <num>
STAT:QUES:ENAB?

← To
*STB QUES
Summary
bit 3

NOTE:

STAT:PRES
This command presets the following enable and transition registers:
OPER, AND QUES.

* Not used

Full Command List

NOTE

- Characters in lower case may be omitted.
- Commands within square brackets [.] are optional.
- Parentheses mark places where data parameters are required. Choose one from the list provided or enter a numeric value.
- Allowed Terminators: DB, mA, Hz, KHz, MHz, GHz.
- Default Terminators: If none of the above are provided, the entry will be assumed to be in fundamental units of measure, which includes: dB, Hz, etc.
- Numeric values: Exponential notation is accepted along with simple integers. i.e. 234, -139.34E+6

Format Of Measurement Data Output

Measured Data

SENSe:DATA:HEADer:POINts? Returns Number of points.

SENSe:DATA?

Returns triples with each element separated by a comma and terminated by cr/lf.

Each triple consists of the "frequency, amplitude and spur flag".

Processes Data

CALCulate:DATA:HEADer:POINts? Returns the number of points.

CALCulate:DATA? Return varies with CALCulate:VIEW: setting.

CALCulate:VIE:SPURtable Returns "frequency, amplitude" per point.

CALCulate:VIEW:TRACe Returns "amplitude" per point.

CALCulate:VIEW:XYData Returns "frequency, amplitude, spur_flag" per point.

SCPI Commands

*CLS
 *ESE (value)
 *ESE?
 *ESR?
 *IDN?
 *OPC
 *OPC?
 *RST
 *SRE (value)
 *SRE?
 *STB?
 *TST?

Full Command List

*WAI

ABORt

ABORt

CALCulate

CALCulate:ACCumulated[:PHASe]? (tau, start-offset, stop-offset)

CALCulate:AVARiance? (tau, Frequency offset)

CALCulate:CFSCale (value)

CALCulate:CFSCale?

CALCulate:DATA?

CALCulate:DATA:HEADer:POINts?

CALCulate:DIPower (value)

CALCulate:DIPower?

CALCulate:F2OScillator[:COMPare] ("inFileA", "inFileAB", "outFile"[,"title"])

CALCulate:F3OScillator[:COMPare] ("inFileAB", "inFileAC", "inFileBC", "outFileA", "outFileB", "outFileC"[,"titleA", "titleB", "titleC"])

CALCulate:INTegral? (start-offset, stop-offset)

CALCulate:INTegral:TYPE (NORMal | CCITt)

CALCulate:INTegral:TYPE?

CALCulate:LIMit:NOISe:MAXimum (segment#, start_freq, start_amplitude, stop_freq, stop_amplitude)

CALCulate:LIMit:NOISe:MAXimum? (segment_number)

CALCulate:LIMit:NOISe:MAXimum:DELeTe (segment_number)

CALCulate:LIMit:NOISe:MAXimum:DELeTe:ALL

CALCulate:LIMit:NOISe:MAXimum:DISPlay (0 | OFF | 1 | ON)

CALCulate:LIMit:NOISe:MAXimum:DISPlay?

CALCulate:LIMit:NOISe:MAXimum:FAIL? (segment_number)

CALCulate:LIMit:NOISe:MAXimum:FAIL:ALL?

CALCulate:LIMit:NOISe:MINimum (segment#, start_freq, start_amplitude, stop_freq, stop_amplitude)

CALCulate:LIMit:NOISe:MINimum? (segment_number)

CALCulate:LIMit:NOISe:MINimum:DELeTe (segment_number)

CALCulate:LIMit:NOISe:MINimum:DELeTe:ALL

CALCulate:LIMit:NOISe:MINimum:DISPlay (0 | OFF | 1 | ON)

CALCulate:LIMit:NOISe:MINimum:DISPlay?

CALCulate:LIMit:NOISe:MINimum:FAIL? (segment_number)

CALCulate:LIMit:NOISe:MINimum:FAIL:ALL?

CALCulate:LIMit:SPUR:MAXimum (segment#, start_freq, start_amplitude, stop_freq, stop_amplitude)

CALCulate:LIMit:SPUR:MAXimum? (segment_number)

CALCulate:LIMit:SPUR:MAXimum:DELeTe (segment_number)

CALCulate:LIMit:SPUR:MAXimum:DELeTe:ALL

CALCulate:LIMit:SPUR:MAXimum:DISPlay (0 | OFF | 1 | ON)

CALCulate:LIMit:SPUR:MAXimum:DISPlay?

CALCulate:LIMit:SPUR:MAXimum:FAIL? (segment_number)

CALCulate:LIMit:SPUR:MAXimum:FAIL:ALL?

CALCulate:LIMit:SPUR:MINimum (segment#, start_freq, start_amplitude, stop_freq, stop_amplitude)
 CALCulate:LIMit:SPUR:MINimum? (segment_number)
 CALCulate:LIMit:SPUR:MINimum:DELeTe (segment_number)
 CALCulate:LIMit:SPUR:MINimum:DELeTe:ALL
 CALCulate:LIMit:SPUR:MINimum:DISPlay (0 | OFF | 1 | ON)
 CALCulate:LIMit:SPUR:MINimum:DISPlay?
 CALCulate:LIMit:SPUR:MINimum:FAIL? (segment_number)
 CALCulate:LIMit:SPUR:MINimum:FAIL:ALL?
 CALCulate:PBWidth (value)
 CALCulate:PBWidth?
 CALCulate:SELEct (BOTH | NOISe | SPURs)
 CALCulate:SELEct?
 CALCulate:SMOothing (value)
 CALCulate:SMOothing?
 CALCulate:TRANsform (SSBN | SDPF | SDFE | SFFF | NF2P | AMN | BBAN)
 CALCulate:TRANsform?
 CALCulate:TVAriance? (tau, Frequency offset)
 CALCulate:VIEW:SPURtable
 CALCulate:VIEW:TRACe (start-offset, stop-offset, #points, distribution (log/linear))
 CALCulate:VIEW:XYData
 CALCulate:YSHift (value)
 CALCulate:YSHift?

CALibrate

CALibrate:DETEctor:CONStant (value)
 CALibrate:DETEctor:CONStant?
 CALibrate:DETEctor:CONStant:METhod (BEATnote | DCPeak | SSPur | DSPur | FMD | AMIN | PSHift | NONE)
 CALibrate:DETEctor:CONStant:METhod?
 CALibrate:DETEctor:CONStant:SPUR:AMPLitude (value)
 CALibrate:DETEctor:CONStant:SPUR:AMPLitude?
 CALibrate:DETEctor:CONStant:SPUR:OFFSet (value)
 CALibrate:DETEctor:CONStant:SPUR:OFFSet?
 CALibrate:VCO:IREsistance (value)
 CALibrate:VCO:IREsistance?
 CALibrate:VCO:PLLSuppress (0 | OFF | 1 | ON)
 CALibrate:VCO:PLLSuppress?
 CALibrate:VCO:PLLSuppress:APOLe?
 CALibrate:VCO:PLLSuppress:CBWidth?
 CALibrate:VCO:PLLSuppress:PTRange?
 CALibrate:VCO:PLLSuppress:DISPlay[:ALWays] (0 | OFF | 1 | ON)
 CALibrate:VCO:PLLSuppress:DISPlay[:ALWays]?
 CALibrate:VCO:PLLSuppress:ERRor (value)

Full Command List

CALibrate:VCO:PLLSuppress:ERRor?
 CALibrate:VCO:PLLSuppress:ERRor:ACTion (THEoretical | ADJusted | PAUSE)
 CALibrate:VCO:PLLSuppress:ERRor:ACTion?
 CALibrate:VCO:TCONstant:METHod (CURRent | MEASure | CALCulate)
 CALibrate:VCO:TCONstant:METHod?
 CALibrate:VCO:TCONstant:NOMinal (value)
 CALibrate:VCO:TCONstant:NOMinal?
 CALibrate:VCO:TCONstant?
 CALibrate:VCO:TMODE (value)
 CALibrate:VCO:TMODE?
 CALibrate:VCO:VCADjust (value)
 CALibrate:VCO:VCADjust?
 CALibrate:VCO:VCENter (value)
 CALibrate:VCO:VCENter?
 CALibrate:VCO:VRANge (value)
 CALibrate:VCO:VRANge?

DISPlay

DISPlay:GRAPh:BOUNds:AMPLitude (Transform,Min,Max)
 DISPlay:GRAPh:BOUNds:AMPLitude? (SSBN | SDPF | SDFF | SFFF | NF2P | AMN | BBAN)
 DISPlay:GRAPh:BOUNds:FREQuency (Min,Max)
 DISPlay:GRAPh:BOUNds:FREQuency?
 DISPlay:GRAPh:CFSCale (value)
 DISPlay:GRAPh:CFSCale?
 DISPlay:GRAPh:DIPower (value)
 DISPlay:GRAPh:DIPower?
 DISPlay:GRAPh:PBWidth (value)
 DISPlay:GRAPh:PBWidth?
 DISPlay:GRAPh:SMOothing (value)
 DISPlay:GRAPh:SMOothing?
 DISPlay:GRAPh:TRANSform (SSBN | SDPF | SDFF | SFFF | NF2P | AMN | BBAN)
 DISPlay:GRAPh:TRANSform?
 DISPlay:GRAPh:YSHift (value)
 DISPlay:GRAPh:YSHift?
 DISPlay:MEASurement:TIME?
 DISPlay:TEXT:TITLe ("string")
 DISPlay:TEXT:TITLe?

FORMat

FORMat[:DATA] (ASCIi | REAL, length)

INITiate

INITiate[:IMMediate]:CALibrate
 INITiate[:IMMediate]:MEASure
 INITiate[:IMMediate][:ALL]

MMEMory

MMEMory:LOAD[:ALL] ("path\file.pnm")

MMEMory:STORe[:ALL] ("path\file.pnm")

PAUSE

PAUSE:ABORt:AUTO (0 | OFF | 1 | ON)
 PAUSE:ABORt:AUTO?
 PAUSE:ADJust:VCOCenter (value)
 PAUSE:ADJust:VCOCenter?
 PAUSE:ADJust:LNAGain (14 | 28 | 42 | 56)
 PAUSE:ADJust:LNAGain?
 PAUSE:CONNect (0 | OFF | 1 | ON)
 PAUSE:CONNect?
 PAUSE:CONTInue
 PAUSE:RETRy
 PAUSE:SPECIal

SENSE

[:SENSE]:DATA?
 [:SENSE]:DATA:HEADer:POINts?
 [:SENSE]:DCONverter:ATTenuator (0 to 35 dB in 5 dB steps)
 [:SENSE]:DCONverter:ATTenuator?
 [:SENSE]:DCONverter:ATTenuator:AUTO (0 | OFF | 1 | ON)
 [:SENSE]:DCONverter:ATTenuator:AUTO?
 [:SENSE]:DCONverter:ATTenuator:AUTO:IMMEDIATE
 [:SENSE]:DCONverter:BAND (UWAVe | A | K | Q | U | V | W)
 [:SENSE]:DCONverter:BAND?
 [:SENSE]:DCONverter:FREQuency (value)
 [:SENSE]:DCONverter:FREQuency?
 [:SENSE]:DCONverter:IFFREquency?
 [:SENSE]:DCONverter:IFGAIN (-10 to 45 dB in 5 dB steps)
 [:SENSE]:DCONverter:IFGAIN?
 [:SENSE]:DCONverter:IFGAIN:AUTO (0 | OFF | 1 | ON)
 [:SENSE]:DCONverter:IFGAIN:AUTO?
 [:SENSE]:DCONverter:IFGAIN:AUTO:IMMEDIATE
 [:SENSE]:DCONverter:L1BWidth (value)
 [:SENSE]:DCONverter:L1BWidth?
 [:SENSE]:DCONverter:L6BWidth (value)
 [:SENSE]:DCONverter:L6BWidth?
 [:SENSE]:DCONverter:LOPower (value)
 [:SENSE]:DCONverter:LOPower?
 [:SENSE]:DCONverter:LOSelect (value)
 [:SENSE]:DCONverter:LOSelect?
 [:SENSE]:DCONverter:LOSelect:AUTO (0 | OFF | 1 | ON)
 [:SENSE]:DCONverter:LOSelect:AUTO?
 [:SENSE]:DCONverter:MEXT:BIAS (value)
 [:SENSE]:DCONverter:MEXT:BIAS?
 [:SENSE]:DCONverter:MEXT:BIAS:STATe (0 | OFF | 1 | ON)
 [:SENSE]:DCONverter:MEXT:BIAS:STATe?

Full Command List

```

[:SENSe]:DCONverter:MMLO?
[:SENSe]:DCONverter:PLO
[:SENSe]:DCONverter:REFerence (value)
[:SENSe]:DCONverter:REFerence?
[:SENSe]:DCONverter:TSENsitivity?
[:SENSe]:DCONverter:TUNE:PORT (OFF | INTernal | FRONt | REAR)
[:SENSe]:DCONverter:TUNE:PORT?
[:SENSe]:DETector:FREQuency (value)
[:SENSe]:DETector:FREQuency?
[:SENSe]:DETector:SElect (AUTO | EXT | LFR | HFR | UWAVE | TAM |
TINoise | DCAM)
[:SENSe]:DETector:SElect?
[:SENSe]:FFT:INPut:COUPling (AC | DC)
[:SENSe]:FFT:INPut:COUPling?
[:SENSe]:NOISe:BBGain (value)
[:SENSe]:NOISe:BBGain?
[:SENSe]:NOISe:FFT (EXTended | MULTiple)
[:SENSe]:NOISe:FFT?
[:SENSe]:NOISe:MEAStype (ABS | RES | FM | AM | BBAN | NOTS)
[:SENSe]:NOISe:MEAStype?
[:SENSe]:NOISe:PULSed (0 | OFF | 1 | ON)
[:SENSe]:NOISe:PULSed?
[:SENSe]:NOISe:QUADrature[:METHod] (PSHifter | SOURce)
[:SENSe]:NOISe:QUADrature[:METHod]?
[:SENSe]:RANGe:FFT:AVERage:MINimum (value)
[:SENSe]:RANGe:FFT:AVERage:MINimum?
[:SENSe]:RANGe:FFT:SEGTable[:MEASurement][:QUALity] (NORMal |
FAST | HRESolution | CUSTom)
[:SENSe]:RANGe:FFT:SEGTable[:MEASurement][:QUALity]?
[:SENSe]:RANGe:OFFSet (Start frequency,Stop frequency)
[:SENSe]:RANGe:OFFSet?
[:SENSe]:RANGe:SWEPt:SEGTable[:MEASurement][:QUALity] (NORMal |
FAST | HRESolution | CUSTom)
[:SENSe]:RANGe:SWEPt:SEGTable[:MEASurement][:QUALity]?
[:SENSe]:TSET:ATTenuator (0 to 35 dB in 5 dB steps)
[:SENSe]:TSET:ATTenuator?
[:SENSe]:TSET:ATTenuator:AUTO (0 | OFF | 1 | ON)
[:SENSe]:TSET:ATTenuator:AUTO?
[:SENSe]:TSET:DCBlock (0 | OFF | 1 | ON)
[:SENSe]:TSET:DCBlock?
[:SENSe]:TSET:LNAGain (14 | 28 | 42 | 56)
[:SENSe]:TSET:LNAGain?
[:SENSe]:TSET:LNAGain:METHod (AUTO | FIXed | PAUSE)
[:SENSe]:TSET:LNAGain:METHod?

```

[:SENSe]:TSET:LNAGain:MINimum (value)
 [:SENSe]:TSET:LNAGain:MINimum?
 [:SENSe]:TSET:LPF (value)
 [:SENSe]:TSET:LPF?
 [:SENSe]:TSET:LPF:AUTO (0 | OFF | 1 | ON)
 [:SENSe]:TSET:LPF:AUTO?
 [:SENSe]:TSET:PLL:ATTenuator (value)
 [:SENSe]:TSET:PLL:ATTenuator?
 [:SENSe]:TSET:PLL:UNLock:IGNore (0 | OFF | 1 | ON)
 [:SENSe]:TSET:PLL:UNLock:IGNore?
 [:SENSe]:TVCO (REFerence | CARRier | DCONverter | INTernal)
 [:SENSe]:TVCO?
 [:SENSe]:TSET:MLEVel:AMDetector (value)
 [:SENSe]:TSET:MLEVel:AMDetector?
 [:SENSe]:TSET:MLEVel:RFDetector (value)
 [:SENSe]:TSET:MLEVel:RFDetector?
 [:SENSe]:TSET:MLEVel:UWDetector (value)
 [:SENSe]:TSET:MLEVel:UWDetector?
 [:SENSe]:VIEW:DCONverter (BBANd | CARRier | IF)
 [:SENSe]:VIEW:DCONverter?
 [:SENSe]:VIEW:TSET (BBANd | CARRier | IF)
 [:SENSe]:VIEW:TSET?

SOURce

SOURce:CALibration:FM:INTernal:FREQuency (value)
 SOURce:CALibration:FM:INTernal:FREQuency?
 SOURce:CALibration:FM[:DEViation] (value)
 SOURce:CALibration:FM[:DEViation]?
 SOURce:CALibration:FREQuency[:CW|FIXed] (value)
 SOURce:CALibration:FREQuency[:CW|FIXed]?
 SOURce:CALibration:POWer[:LEVel|AMPLitude] (value)
 SOURce:CALibration:POWer[:LEVel|AMPLitude]?
 SOURce:CARRier:FREQuency[:CW|FIXed] (value)
 SOURce:CARRier:FREQuency[:CW|FIXed]?
 SOURce:CARRier:POWer[:LEVel|AMPLitude] (value)
 SOURce:CARRier:POWer[:LEVel|AMPLitude]?
 SOURce:REFerence:FREQuency:DIVisor (value)
 SOURce:REFerence:FREQuency:DIVisor?
 SOURce:REFerence:FREQuency:MULTiplier (value)
 SOURce:REFerence:FREQuency:MULTiplier?
 SOURce:REFerence:POWer[:LEVel|AMPLitude] (value)
 SOURce:REFerence:POWer[:LEVel|AMPLitude]?
 SOURce:RESidual:FREQuency[:CW|FIXed] (value)
 SOURce:RESidual:FREQuency[:CW|FIXed]?

Full Command List

SOURce:RESidual:FREQuency:CALCulate (0 | OFF | 1 | ON)
 SOURce:RESidual:FREQuency:CALCulate?
 SOURce:RESidual:FREQuency:DIVisor (value)
 SOURce:RESidual:FREQuency:DIVisor?
 SOURce:RESidual:FREQuency:DETEctor:COUPled (0 | OFF | 1 | ON)
 SOURce:RESidual:FREQuency:DETEctor:COUPled?
 SOURce:RESidual:FREQuency:MULTiplier (value)
 SOURce:RESidual:FREQuency:MULTiplier?
 SOURce:RESidual:POWer[:LEVel|AMPLitude] (value)
 SOURce:RESidual:POWer[:LEVel|AMPLitude]?

STATus

STATus:ADVise:ENABle (0 | OFF | 1 | ON)
 STATus:ADVise:ENABle?
 STATus:OPERation:CONDition?
 STATus:OPERation:ENABle (value)
 STATus:OPERation:ENABle?
 STATus:OPERation:EVENT?
 STATus:OPERation:NTRansition (value)
 STATus:OPERation:NTRansition?
 STATus:OPERation:PTRansition (value)
 STATus:OPERation:PTRansition?
 STATus:PRESet
 STATus:QUEStionable:CONDition?
 STATus:QUEStionable:ENABle (value)
 STATus:QUEStionable:ENABle?
 STATus:QUEStionable:EVENT?
 STATus:QUEStionable:NTRansition (value)
 STATus:QUEStionable:NTRansition?
 STATus:QUEStionable:PTRansition (value)
 STATus:QUEStionable:PTRansition?

SYSTEM

SYSTem:ASSet:CALSource ("None" | "asset_name")
 SYSTem:ASSet:CALSource?
 SYSTem:ASSet:CARRier ("None" | "asset_name")
 SYSTem:ASSet:CARRier?
 SYSTem:ASSet:COUNter ("None" | "asset_name")
 SYSTem:ASSet:COUNter?
 SYSTem:ASSet:DCONverter ("None" | "asset_name")
 SYSTem:ASSet:DCONverter?
 SYSTem:ASSet:DCONverter:USAGe (NONE | MANual | SYSTem)
 SYSTem:ASSet:DCONverter:USAGe?
 SYSTem:ASSet:FFT ("None" | "asset_name")
 SYSTem:ASSet:FFT?
 SYSTem:ASSet:FFT:CUSTom[:SEGTable] ("path\name.fst")
 SYSTem:ASSet:PSHifter ("None" | "asset_name")

SYSTem:ASSet:PSHifter?
SYSTem:ASSet:REFerence ("None" | "asset_name")
SYSTem:ASSet:REFerence?
SYSTem:ASSet:RESidual ("None" | "asset_name")
SYSTem:ASSet:RESidual?
SYSTem:ASSet:SLAVe:SOURce ("None" | "asset_name")
SYSTem:ASSet:SLAVe:SOURce?
SYSTem:ASSet:SLAVe:SOURce:PORT (LOINput | AUXinput)
SYSTem:ASSet:SLAVe:SOURce:PORT?
SYSTem:ASSet:SWANalyzer ("None" | "asset_name")
SYSTem:ASSet:SWANalyzer?
SYSTem:ASSet:SWANalyzer:CUSTom[:SEGTable] ("path\name.sst")
SYSTem:ASSet:TBASe ("None" | "asset_name")
SYSTem:ASSet:TBASe?
SYSTem:ASSet:TSET ("None" | "asset_name")
SYSTem:ASSet:TSET?
SYSTem:ERRor?
SYSTem:GUI:LLOCKout
SYSTem:GUI:REMote (0 | OFF | 1 | ON)
SYSTem:GUI:REMote?
SYSTem:HELP:HEADers?
SYSTem:PATH:CARRier (TSET | DCONverter)
SYSTem:PATH:CARRier?
SYSTem:PATH:DCBBanalyzer (TSLF | TSHF | TSRF)
SYSTem:PATH:DCBBanalyzer?
SYSTem:PATH:FFTanalyzer (TSLF | TSHF | TSRF | DCRF)
SYSTem:PATH:FFTanalyzer?
SYSTem:PATH:SWANalyzer (TSLF | TSHF | TSRF | DCRF)
SYSTem:PATH:SWANalyzer?
SYSTem:PATH:TVCO (FPANel | RPANel | INTernal)
SYSTem:PATH:TVCO?
SYSTem:VERSion?

7

Required Commands

Refer to Chapter 1 for update information.

The following IEEE 488.2 Required Commands are supported:

- *CLS
- *ESE (Value)
- *ESE?
- *ESR?
- *IDN?
- *OPC
- *OPC?
- *RST
- *SRE (Value)
- *SRE?
- *STB?
- *TST?
- *WAI

Required Commands

***CLS**

***CLS**

Clears the Status Byte by emptying the error queue and clearing all event registers.

Command Syntax

*CLS

Query Syntax

Command Only

Attribute Summary

Synchronization Required: no

Preset (*RST) State: NA

SCPI Compliance: standard

Additional Information

Also see STATus:PRESet command

***ESE**

Specifies which bits in the Event Register (ESR?) set the ESR summary Bit (bit 5) in the Status Byte (*STB?).

Command Syntax

*ESE (0-255)

Example

*ESE 1

Query Syntax

*ESE?

Attribute Summary

Synchronization Required: no

Preset (*RST) State: NA

SCPI Compliance: standard

Additional Information

See *ESR for bit assignments.

***ESR?**

Reads and clears the Standard Event Enable register.

Command Syntax

Query Only

Query Syntax

*ESR?

Return Format

Integer (0-255)

Attribute Summary

Synchronization Required: none

Preset (*RST) State: NA

SCPI Compliance: standard

**Additional
Information**

- bit 0 = 1 = Operation Complete
- bit 1 = 2 = Request Control
- bit 2 = 4 = Query Error
- bit 3 = 8 = Device Dependent Error. (Pause, Abort, or Warning, Summary of QSR bits 9,10,11)
- bit 4 = 16 = Execution Error. (synchronous error)
- bit 5 = 32 = Command Error
- bit 6 = 64 = User Request
- bit 7 = 128 = Power On

***IDN?**

Returns a comma-separated list of arbitrary ASCII response data items that uniquely identifies the system.

Command Syntax

Query Only

Example

E5500A, Phase Noise Measurement Sub system, Version A.01.00

Query Syntax

*IDN?

Return Format

ARB_ASCII

Attribute Summary

Synchronization Required: no

Preset (*RST) State: instrument specific

SCPI Compliance: standard

***OPC**

Causes bit #0 (Operation Complete) of the Event Status Register to be set to 1 when all pending operations are complete.

Command Syntax

*OPC

Query Syntax

*OPC?

Return Format

Integer

Attribute Summary

Synchronization Required: no

Preset (*RST) State: NA

SCPI Compliance: standard

Additional Information

These are the only commands to which *OPC, *OPC? and *WAI apply:

- INITiate:CALibrate
- INITiate:MEASure
- INITiate:IMMediate:ALL

All other commands execute immediately and do not cause pending operations.

***OPC?**

Returns a ASCII 1 when all pending operations are complete.

Command Syntax

Query Only

Query Syntax

OPC?

Return Format

Integer

Attribute Summary

Synchronization Required: no

Preset (*RST) State: NA

SCPI Compliance: standard

Additional Information

These are the only commands to which *OPC, *OPC? and *WAI apply:

- INITiate:CALibrate
- INITiate:MEASure
- INITiate:IMMEDIATE:ALL

All other commands execute immediately and do not cause pending operations.

***RST**

Executes a device reset.

Command Syntax

*RST

Query Syntax

Command Only

Attribute Summary

Synchronization Required: none

Preset (*RST) State: NA

SCPI Compliance: standard

**Additional
Information**

After *RST parameter queries return the following values:

- CALCulate:CFSCale? 1.00000000
- CALCulate:DATA:HEADer:POINts? +0
- CALCulate:DIPower? 0.00000000
- CALCulate:PBWidth? 1.00000000
- CALCulate:SElect?. BOTH
- CALCulate:SMOothing? +0
- CALCulate:TRANsform? SSBNoise
- CALCulate:YSHift?. 0.00000000
- CALibrate:DETEctor:CONStant:METHod? BEATnote
- CALibrate:DETEctor:CONStant:SPUR:AMPLitude? -130.00000000
- CALibrate:DETEctor:CONStant:SPUR:OFFSet? 1.00000000
- CALibrate:DETEctor:CONStant? 1.00000000E-6
- CALibrate:VCO:IRESistance? 1.00000000E+06
- CALibrate:VCO:PLLSuppress:DISPlay:ALWays?. 0
- CALibrate:VCO:PLLSuppress:ERRor:ACTion? ADJusted
- CALibrate:VCO:PLLSuppress:ERRor? 1.00000000
- CALibrate:VCO:PLLSuppress?. 1
- CALibrate:VCO:TCONstant:METHod? MEASure
- CALibrate:VCO:TCONstant:NOMinal? 10.00000000E+03
- CALibrate:VCO:TCONstant? 1.00000000E+06
- CALibrate:VCO:TMODE?. EFC
- CALibrate:VCO:VCADjust?. 1.00000000
- CALibrate:VCO:VCENter?. 0.00000000
- CALibrate:VCO:VRANge?. 1.00000000
- DISPlay:GRAPh:BOUNds:AMPLitude? SSBN -170.00000000,
0.00000000

• DISPLAY:GRAPH:BOUNds:AMPLitude? SDPF	-170.00000000,
0.00000000	
• DISPLAY:GRAPH:BOUNds:AMPLitude? SDFP	1.00000000E-6,
1.00000000E+03	
• DISPLAY:GRAPH:BOUNds:AMPLitude? SFFF	1.00000000E-15,
1.00000000E-6	
• DISPLAY:GRAPH:BOUNds:AMPLitude? NF2P	-170.00000000,
0.00000000	
• DISPLAY:GRAPH:BOUNds:AMPLitude? AMN	-170.00000000,
0.00000000	
• DISPLAY:GRAPH:BOUNds:AMPLitude? BBAN	-170.00000000,
0.00000000	
• DISPLAY:GRAPH:BOUNds:FREQuency?	1.00000000,
100.00000000E+06	
• DISPLAY:GRAPH:CFSCale?	1.00000000
• DISPLAY:GRAPH:DIPower?	0.00000000
• DISPLAY:GRAPH:PBWidth?	1.00000000
• DISPLAY:GRAPH:SMOothing?	+0
• DISPLAY:GRAPH:TRANSform?	SSBNoise
• DISPLAY:GRAPH:YSHift?	0.00000000
• DISPLAY:TEXT:TITLe? E5500 Absolute Phase Noise Measurement	
• FORMat:DATA?	ASCIi,6
• PAUSe:ABORt:AUTO?	1
• PAUSe:CONNect?	0
• SOURce:CALibration:FM:INTernal:FREQuency?	1.00000000E+03
• SOURce:CALibration:FM:DEVIation?	10.00000000E+03
• SOURce:CALibration:FREQuency:FIXed?	500.00000000E+06
• SOURce:CALibration:POWer:AMPLitude?	10.00000000
• SOURce:CARRier:FREQuency:FIXed?	500.00000000E+06
• SOURce:CARRier:POWer:AMPLitude?	10.00000000
• SOURce:REFeRence:POWer:AMPLitude?	10.00000000
• SOURce:RESidual:FREQuency:FIXed?	500.00000000E+06
• SOURce:RESidual:POWer:AMPLitude?	10.00000000
• STATus:ADVise:ENABle?	0
• SYSTem:ASSet:DCONverter:USAGe?	NONE
• SYSTem:ASSet:SLAVe:SOURce:PORT?	LOINput
• SYSTem:PATH:CARRier?	TSET
• SYSTem:PATH:DCBBanalyzer?	TSLF
• SYSTem:PATH:FFTanalyzer?	TSLF
• SYSTem:PATH:SWANalyzer?	TSLF
• SYSTem:PATH:TVCO?	FPANel
• SENSe:DATA:HEADer:POINts?	+0
• SENSe:DCONverter:ATTenuator:AUTO?	1
• SENSe:DCONverter:ATTenuator?	0.00000000
• SENSe:DCONverter:BAND?	UWAVe
• SENSe:DCONverter:FREQuency?	10.00000000E+09
• SENSe:DCONverter:IFFRequency?	0.00000000

Required Commands

*RST

• SENSE:DCONverter:IFGain:AUTO?	1
• SENSE:DCONverter:IFGain?	0.00000000
• SENSE:DCONverter:L1BWidth?	126
• SENSE:DCONverter:L6BWidth?	10e3
• SENSE:DCONverter:LOPower?	10.00000000
• SENSE:DCONverter:LOSelect:AUTO?	1
• SENSE:DCONverter:LOSelect?	1.00000000E+09
• SENSE:DCONverter:MEXT:BIAS:STATe?	0
• SENSE:DCONverter:MEXT:BIAS?	0.00000000
• SENSE:DCONverter:MLEVel:AMDetector?	0.00000000
• SENSE:DCONverter:MLEVel:UWMixer?	0.00000000
• SENSE:DCONverter:MMLO?	0.00000000
• SENSE:DCONverter:REFeRence?	10e+6
• SENSE:DCONverter:TSENsitivity?	0
• SENSE:DCONverter:TUNE:PORT?	0
• SENSE:DETEctor:FREQuency?	500.00000000E+06
• SENSE:DETEctor:SELEct?	AUTO
• SENSE:FFT:INPut:COUPLing?	AC
• SENSE:NOISe:BBGain?	0.00000000
• SENSE:NOISe:FFT?	MULTiple
• SENSE:NOISe:MEAStype?	ABSolute
• SENSE:NOISe:PULSed?	0
• SENSE:NOISe:QUADrature:METhod?	SOURce
• SENSE:RANGe:FFT:AVERage:MINimum?	+4
• SENSE:RANGe:FFT:SEGTable:MEASurement:QUALity?	NORMAL
• SENSE:RANGe:OFFSet?	1.00000000, 100.00000000E+06
• SENSE:RANGe:SWEPT:SEGTable:MEASurement:QUALity?	NORMAL
• SENSE:TSET:ATTenuator:AUTO?	1
• SENSE:TSET:ATTenuator?	0
• SENSE:TSET:DCBlock?	0
• SENSE:TSET:LNAGain:METhod?	AUTO
• SENSE:TSET:LNAGain:MINimum?	14
• SENSE:TSET:LNAGain?	42
• SENSE:TSET:LPF:AUTO?	1
• SENSE:TSET:LPF?	20e+3
• SENSE:TSET:MLEVel:AMDetector?	0.00000000
• SENSE:TSET:MLEVel:RFDetector?	0.00000000
• SENSE:TSET:MLEVel:UWDetector?	0.00000000
• SENSE:TSET:PLL:ATTenuator?	0.0
• SENSE:TSET:PLL:UNLock:IGNore?	0
• SENSE:TVCO?	REFeRence
• SENSE:VIEW:DCONverter?	BBANd
• SENSE:VIEW:TSET?	BBANd

***SRE**

Specify bits in the Service Request Enable register (masks status byte).

Command Syntax

```
*SRE <number>  
<number> ::= an integer (NRf data)  
           limits: 0:191
```

Example

```
*SRE 32
```

Query Syntax

```
*SRE?
```

Return Format

Integer 0-191

Attribute Summary

Synchronization Required: none

Preset (*RST) State:NA

SCPI Compliance: standard

Additional Information

Enabled bits will cause an SRQ to be issued and bit 6 (RQS) of the Status Byte to be set.

Valid range is 0-191 as *SRE bit 6 (value 64) can not be set. See *STB for bit assignments.

***STB?**

Reads the Status Byte register

Command Syntax

Query Only

Query Syntax

*STB?

Return Format

Integer 0-255

Attribute Summary

Synchronization Required: none

Preset (*RST) State: NA

SCPI Compliance: IEEE 488.2

**Additional
Information**

- bit 0 = 1 = Not Used
- bit 1 = 2 = Not Used
- bit 2 = 4 = Device Dependent Warning, Pause, Error Message Available
- bit 3 = 8 = Summary of Questionable Status Register
- bit 4 = 16 = Message Available (MAV)
- bit 5 = 32 = Summary of Standard Event Status Register
- bit 6 = 64 = Request Service (RQS)
- bit 7 = 128 = Summary of Operation Status Register

***TST?**

Performs system self test.

Command Syntax

Query Only

Query Syntax

*TST?

Return Format

Integer

- 0 = Passed Self Test
- non 0 = Failed Self Test

Attribute Summary

Synchronization Required: none

Preset (*RST) State:NA

SCPI Compliance:IEEE 488.2

**Additional
Information**

Failure messages are placed in the error que read by `SYSTEM:ERRor?`

***WAI**

Holds off processing of subsequent commands until all preceding commands have been processed.

Command Syntax

*WAI

Query Syntax

Command Only

Attribute Summary

Synchronization Required: none

Preset (*RST) State: NA

SCPI Compliance: standard

Additional Information

These are the only commands to which *OPC, *OPC? and *WAI apply:

- INITiate:CALibrate
- INITiate:MEASure
- INITiate:IMMEDIATE:ALL

All other commands execute immediately and do not cause pending operations.

8

Abort Commands

Refer to Chapter 1 for update information.

The commands in this subsystem have the following command hierarchy:

- ABORt

ABORt

ABORt

Terminates any measurement, calibration or pause point and causes the system to return to its idle state without changing any measurement definition parameters.

.

Command Syntax

ABORt

Example

ABORt

Query Syntax

Command only

Attribute Summary

Synchronization Required: none

Preset (*RST) State: NA

SCPI Compliance: standard

9

CALCulate Commands

Refer to Chapter 1 for update information.

Use the CALCulate commands to perform post acquisition data processing.

The commands in this subsystem have the following command hierarchy:

- CALCulate:ACCumulated[:PHASe]? (tau, start-offset, stop-offset)
- CALCulate:AVARiance? (tau, Frequency offset)
- CALCulate:CFSCale (value)
- CALCulate:CFSCale?
- CALCulate:DATA?
- CALCulate:DATA:HEADer:POINts?
- CALCulate:DIPower (value)
- CALCulate:DIPower?
- CALCulate:F2OScillator[:COMPare] ("inFileA", "inFileAB", "outFile"[, "title"])
- CALCulate:F3OScillator[:COMPare] ("inFileAB", "inFileAC", "inFileBC", "outFileA", "outFileB", "outFileC"[, "titleA", "titleB", "titleC"])
- CALCulate:INTegral?
- CALCulate:INTegral:TYPE
- CALCulate:LIMit:NOISe:MAXimum (segment#, start_freq, start_amplitude, stop_freq, stop_amplitude)
- CALCulate:LIMit:NOISe:MAXimum:DELeTe (segment_number)
- CALCulate:LIMit:NOISe:MAXimum:DELeTe:ALL
- CALCulate:LIMit:NOISe:MAXimum:DISPlay (0 | OFF | 1 | ON)
- CALCulate:LIMit:NOISe:MAXimum:DISPlay?
- CALCulate:LIMit:NOISe:MAXimum:FAIL? (segment_number)
- CALCulate:LIMit:NOISe:MAXimum:FAIL:ALL?
- CALCulate:LIMit:NOISe:MAXimum? (segment_number)
- CALCulate:LIMit:NOISe:MINimum (segment#, start_freq, start_amplitude, stop_freq, stop_amplitude)
- CALCulate:LIMit:NOISe:MINimum? (segment_number)
- CALCulate:LIMit:NOISe:MINimum:DELeTe (segment_number)
- CALCulate:LIMit:NOISe:MINimum:DELeTe:ALL
- CALCulate:LIMit:NOISe:MINimum:DISPlay (0 | OFF | 1 | ON)
- CALCulate:LIMit:NOISe:MINimum:DISPlay?
- CALCulate:LIMit:NOISe:MINimum:FAIL? (segment_number)
- CALCulate:LIMit:NOISe:MINimum:FAIL:ALL?
- CALCulate:LIMit:SPUR:MAXimum (segment#, start_freq, start_amplitude, stop_freq, stop_amplitude)
- CALCulate:LIMit:SPUR:MAXimum? (segment_number)
- CALCulate:LIMit:SPUR:MAXimum:DELeTe (segment_number)

CALCulate Commands

- CALCulate:LIMit:SPUR:MAXimum:DELeTe:ALL
- CALCulate:LIMit:SPUR:MAXimum:DISPlay (0 | OFF | 1 | ON)
- CALCulate:LIMit:SPUR:MAXimum:DISPlay?
- CALCulate:LIMit:SPUR:MAXimum:FAIL? (segment_number)
- CALCulate:LIMit:SPUR:MAXimum:FAIL:ALL?
- CALCulate:LIMit:SPUR:MINimum (segment#, start_freq, start_amplitude, stop_freq, stop_amplitude)
- CALCulate:LIMit:SPUR:MINimum:DELeTe (segment_number)
- CALCulate:LIMit:SPUR:MINimum:DELeTe:ALL
- CALCulate:LIMit:SPUR:MINimum:DISPlay (0 | OFF | 1 | ON)
- CALCulate:LIMit:SPUR:MINimum:DISPlay?
- CALCulate:LIMit:SPUR:MINimum:FAIL? (segment_number)
- CALCulate:LIMit:SPUR:MINimum:FAIL:ALL?
- CALCulate:LIMit:SPUR:MINimum? (segment_number)
- CALCulate:PBWidth (value)
- CALCulate:PBWidth?
- CALCulate:SElect (BOTH | NOISe | SPURs)
- CALCulate:SElect?
- CALCulate:SMOothing (value)
- CALCulate:SMOothing?
- CALCulate:TRANsform (SSBN | SDPF | SDFE | SFFF | NF2P | AMN | BBAN)
- CALCulate:TRANsform?
- CALCulate:TVAriance? (tau, Frequency offset)
- CALCulate:VIEW:SPURtable
- CALCulate:VIEW:TRACe (start-offset, stop-offset, #points, distribution (log/linear))
- CALCulate:VIEW:XYData
- CALCulate:YSHift (value)
- CALCulate:YSHift?

CALCulate:ACCumulated[PHASe]?

Accumulated phase noise integration.

NOTE

This optional command requires users to purchase a license to use. If the user attempts to use the command without a license, an error message is generated in the SYST:ERR? que.

Command Syntax

Query Only

Query Syntax

```
CALCulate:Accumulated? <tau>,<offset upper value>,  
<stop offset>
```

Example

```
CALCulate:Accumulated? 1,100,100E9
```

Return Format

Real

Attribute Summary

Synchronization Required:none

Preset (*RST) State:NA

SCPI Compliance:Instrument Specific.

Additional Information

Performs calculation on currently selected data as specified by CALCulate:SElect (BOTH|NOISE|SPURs)

CALCulate:AVARiance?

Allan Variance calculation.

Command Syntax

Query Only

Query Syntax

`CALCulate:AVARiance? <tau>,<offset upper value>`

Example

`CALCulate:AVARiance? 1,10E3`

Return Format

Real

Attribute Summary

Synchronization Required: none

Preset (*RST) State: NA

SCPI Compliance: Instrument Specific.

Additional Information

Performs calculation on currently selected data as specified by
`CALCulate:SElect (BOTH|NOISE|SPURs)`

CALCulate:CFSCale

Carrier frequency scale multiplier.

Command Syntax

```
CALCulate:CFSCale <number>  
<number> ::= a real number (NRf data)
```

Example

```
CALCulate:CFSCale 2
```

Query Syntax

```
CALCulate:CFSCale?
```

Return Format

Real

Attribute Summary

Synchronization Required: none

Preset (*RST) State: 1.0

SCPI Compliance: Instrument Specific.

Additional Information

Applied to data output by CALCulate:DATA? when outputting:

- CALCulate:VIEW:TRACe (start, stop, #points, distribute (log/linear) or
- CALCulate:VIEW:XYData.

CALCulate:DATA?

Provides access to measurement data as processed by the CALCulate subsystem.

Command Syntax

Query Only

Query Syntax

CALCulate:DATA?

Return Format

Varies with CALCulate:VIEW. See Additional Information below. Frequencies and Amplitudes are Reals. Spurs and Flags are integers.

Attribute Summary

Synchronization Required: none

Preset (*RST) State:

- ▲ CALCulate:VIEW:XYData
- ▲ CALCulate:CFScale 0
- ▲ CALCulate:DIPower 0
- ▲ CALCulate:PBWidth 1
- ▲ CALCulate:SElect BOTH
- ▲ CALCulate:SMOothing 0
- ▲ CALCulate:TRANSform SSBN
- ▲ CALCulate:YSHift 0

SCPI Compliance: Instrument Specific.

Additional Information

Returns a one line per data point as indicated by CALCulate:DATA:HEADer:POINts. Data type returned is last sent CALCulate:VIEW command which in turn is affected by the current setting of the commands listed under each CALCulate:VIEW command below. The number of significant digits for frequency and amplitude is controlled by the FORMat[:DATA] command.

- CALCulate:VIEW:SPURtableReturns “frequency, amplitude” per point.
 - CALCulate:TRANSform (SSBN | SDPF | SDFF | SFFF | NF2P | AMN | BBAN)
 - CALCulate:YSHift (value)
- CALCulate:VIEW:TRACe (start, stop, #points, distribution (log/linear)) Returns "amplitude" per point.
 - CALCulate:CFScale (value)
 - CALCulate:DIPower (value)

- CALCulate:PBWidth (value)
- CALCulate:SElect (BOTH | NOISe | SPURs)
- CALCulate:SMOothing (value)
- CALCulate:TRANSform (SSBN | SDPF | SDFF | SFFF | NF2P | AMN | BBAN)
- CALCulate:YSHift (value)
- CALCulate:VIEW:XYData Returns “frequency, Amplitude, spur_flag (1=spur 0=noise)” per point.
 - CALCulate:CFSCale (value)
 - CALCulate:DIPower (value)
 - CALCulate:PBWidth (value)
 - CALCulate:SElect (BOTH | NOISe | SPURs)
 - CALCulate:SMOothing (value)
 - CALCulate:TRANSform (SSBN | SDPF | SDFF | SFFF | NF2P | AMN | BBAN)
 - CALCulate:YSHift (value)

CALCulate:DATA:HEADer:POINts?

Returns the number of data points which will be returned by
CALCulate:DATA?

Command Syntax

Query Only

Query Syntax

CALCulate:DATA:HEADer:POINts?

Example

CALCulate:DATA:HEADer:POINts?

Return Format

Integer

Attribute Summary

Synchronization Required: none

Preset (*RST) State: 0

SCPI Compliance: Instrument Specific.

CALCulate:DIpower

DUT input power adjustment.

Command Syntax

```
CALCulate:DIpower <number>  
<number> ::= a real number (NRf data)
```

Example

```
CALCulate:DIpower -10
```

Query Syntax

```
CALCulate:DIpower?
```

Return Format

Real

Attribute Summary

Synchronization Required: none

Preset (*RST) State: 0.0

SCPI Compliance: Instrument Specific.

Additional Information

Applied to data output by CALCulate:DATA? when outputting:

CALCulate:VIEW:TRACe (start, stop, #points, distribution(log/linear))
or CALCulate:VIEW:XYData

CALCulate:F2OScillator[:COMPare]

Function compares 2 oscillator measurement files and outputs a new measurement file of the result.

Command Syntax

```
CALCulate:F2OScillator[:COMPare] <"InFileA.pnm",  
"InFileAB.pnm", "OutFileB.pnm" [, "Title"]>
```

"InFileA.pnm" ::= Input file of DUT A oscillator measurement from the result of a previous 3 Oscillator comparison.

"InFileAB.pnm" ::= Input file of DUT A vs DUT B oscillator measurement.

"OutFileB.pnm" ::= Output file of calculated DUT B oscillator. File names are quoted strings (full or relative path and filename).

"Title" ::= Optional quoted string title for graph displayed by output file. Defaults to file name of output file.

Example

```
CALCulate:F2OScillator "Ref.pnm", "RefvsDut.pnm", "Dut.pnm", "2 Osc  
Result"
```

Query Syntax

Command Only

Attribute Summary

Synchronization Required: none

Preset (*RST) State: NA

SCPI Compliance: Standard

Additional Information

Input files must exist. Input files must have identical segment tables and measurement ranges.

Note that there is no format specification for files. The files are in a format relevant only to the Phase Noise GUI or SCPI clients (.pnm files)

The 2 Oscillator Comparison determines the actual noise level of the unit under test (UUT) by subtracting the known noise level of the reference source device. This procedure is almost identical to the 3 Oscillator Comparison, except that only one measurement from a pair of devices is necessary.

3 Oscillator Comparison must first be done to determine the noise level of the reference device. See CALCulate:F3OScillator[:COMPare]. Once a known reference level is established, the software subtracts the reference noise from the measurement results. The 2 Oscillator Comparison is valid

only if the Measurement Definition parameters and the segment table parameters are kept the same for the Reference device versus the DUT noise measurement as they were when the Reference device was measured using the 3 Oscillator Comparison.

Measurement Uncertainty

To insure accurate results, the noise levels of the three devices should be within 3 to 6 dB of each other at the frequency offsets of interest. Computation uncertainty increases when the phase noise differs by more than 3 dB between the three devices. A minimum of 20 averages, defined in the Type/Range tab of Measurement Definition menu, is recommended when making the noise measurements.

Considerations for the 2 Oscillator Comparison

Unmarked spurs can cause erroneous results for the 2 Oscillator Comparison. Marked spurs are automatically stripped out during the computation, unmarked spurs are not. Unmarked spurs prevent the test system from performing a proper subtraction since spurs cannot be subtracted on a power basis. Observe the measurement results to insure that a minimum of unmarked spurs are present around the offsets of interest. The number of unmarked spurs within the measurement can be reduced by either reducing the measurement bandwidth, or by increasing the number of averages specified in the Type/Range section of Measurement Definition menu and in the FFT and Swept Segment tables.

CALCulate:F3Oscillator[:COMPare]

Function compares 3 oscillator measurement files and outputs 3 new measurement files of the results.

Command Syntax

```
CALCulate:F3Oscillator[:COMPare] <"InFileAB.pnm",  
"InFileAC.pnm", "InFileBC.pnm", "OutFileA.pnm",  
"OutFileB.pnm", "OutFileC.pnm" [, "Title A","Title B" "Title  
C"]>
```

"InFileAB.pnm" ::= Input file of DUT A vs DUT B oscillator measurement.

"InFileAC.pnm" ::= Input file of DUT A vs DUT C oscillator measurement.

"InFileBC.pnm" ::= Input file of DUT B vs DUT C oscillator measurement.

"OutFileA.pnm" ::= Output file of calculated DUT A oscillator.

"OutFileB.pnm" ::= Output file of calculated DUT B oscillator.

"OutFileC.pnm" ::= Output file of calculated DUT C oscillator.

File names are quoted strings (full or relative path and filename).

"Title (ABC)" ::= Optional quoted string titles for graphs displayed by output files. Defaults to file name of corresponding output file.

Example

```
CALCulate:F3Oscillator "DutAvsDutB.pnm", "DutAvsDutC.pnm",  
"DutBvsDutC.pnm", "DutA.pnm", "DutB.pnm", "DutC.pnm"  
"Oscillator A", "Oscillator B", "Oscillator C"
```

Query Syntax

Command Only

Attribute Summary

Synchronization Required: none

Preset (*RST) State: NA

SCPI Compliance: Standard

Additional Information

Input files must exist. Input files must have identical segment tables and measurement ranges.

Note that there is no format specification for files. The files are in a format relevant only to the Phase Noise GUI or SCPI clients (.pnm files)

The 3 Oscillator Comparison determines the noise level for each of three similar sources or devices. This comparison is useful when a reference source with a known lower noise level is not available for a direct

comparison and is the only practical way to determine the exact noise level of state-of-the-art sources. The 3 Oscillator Comparison computes the noise level of three separate devices by measuring them in pairs (A vs. B, A vs. C, and then B vs. C). To achieve accurate results, each of the three measurements must be performed under the same conditions (frequency, measurement range, and segment definition). The test system computes the noise level of each device by comparing and analyzing the measurement results for each of the three noise measurements. The SOFTWARE stores the computed noise level for each of the three devices in separate Result (.pnm) files. The following expressions are used to evaluate the three devices:

- A vs. B = XX = The results of device A vs. B without spurs
- A vs. C = YY = The results of device A vs. C without spurs
- B vs. C = ZZ = The results of device B vs. C without spurs

Therefore:

- $A = (X + Y - Z)/2A$ = The results of device A only
- $B = (X + Z - Y)/2B$ = The results of device B only
- $C = (Y + Z - X)/2C$ = The results of device C only

Measurement Uncertainty

To insure accurate results, the noise levels of the three devices should be within 3 to 6 dB of each other at the frequency offsets of interest. Computation uncertainty increases when the phase noise differs by more than 3 dB between the three devices. A minimum of 20 averages, defined in the Type/Range tab of Measurement Definition menu, is recommended when making the noise measurements. A good test to use for verifying the validity of the noise measurements before performing the 3 Oscillator Comparison is to check that the combined level of the two lowest measurements is greater than or equal to the level of the highest measurement

Considerations for the 3 Oscillator Comparison

Unmarked spurs can cause erroneous results for the 3 Oscillator Comparison. Marked spurs are automatically stripped out during the computation, unmarked spurs are not. Unmarked spurs prevent the test system from performing a proper subtraction since spurs cannot be subtracted on a power basis. Observe the measurement results to insure that a minimum of unmarked spurs are present around the offsets of interest. The number of unmarked spurs within the measurement can be reduced by either reducing the measurement bandwidth, or by increasing the number of averages specified in the Type/Range section of Measurement Definition menu and in the FFT and Swept Segment tables.

CALCulate:INTEgral?

Trace integration.

Command Syntax

Query Only

Query Syntax

```
CALCulate:INTEgral? <start_offset> [<frequency_suffix>],  
                    <stop_offset> [<frequency_suffix>]  
<start_offset>      ::= a real number (NRf data)  
<stop_offset>       ::= a real number (NRf data)  
<frequency_suffix> ::= Hz | kHz | MHz | GHz
```

Example

```
CALCulate:INTEgral? 10,100E3
```

Return Format

Real

Attribute Summary

Synchronization Required: none

Preset (*RST) State: NA

SCPI Compliance: Instrument Specific

Additional Information

Performs calculation on currently selected data as specified by:

- CALCulate:SElect (BOTH | NOISe | SPURs)
- CALCulate:TRANsform (SSBN | SDPF | SDFF | SFFF | NF2P | AMN | BBAN)

CALCulate:INTEgral:TYPE

Specify the type of integral calculation used by the CALCulate:INTEgral command.

NORMal:

CCITt:

Command Syntax

CALCulate:INTEgral:TYPE NORMal|CCITt

Example

CALCulate:INTEgral:TYPE CCITt

Query Syntax

CALCulate:INTEgral:TYPE?

Return Format

string

Attribute Summary

Synchronization Required: none

Preset (*RST) State: NORMal

SCPI Compliance: Instrument Specific

Additional Information

When type is set to CCITt, the weighting over frequency is as specified in ITU-T Recommendation 0.41 (10/94), Table 1/0.41.

CALCulate:LIMit:NOISe:MAXimum

Specifies a limit line segment for the maximum amplitude limit for noise data.

Command Syntax

```
CALCulate:LIMit:NOISe:MAXimum <segment_number>,  
<start_offset> [<frequency_suffix>],<start_power>  
[<power_suffix>],<stop_offset> [<frequency_suffix>],  
<stop_power> [<power_suffix>]  
  
<segment_number> ::= an integer in the range 1-100  
<start_offset> ::= a real number (NRf data)  
<start_power> ::= a real number (NRf data)  
<stop_offset> ::= a real number (NRf data)  
<stop_power> ::= a real number (NRf data)
```

Example

```
CALC:LIMit:NOISe:MAX 1, 100, -50, 10e3, -80
```

Query Syntax

```
CALCulate:LIMit:NOISe:MAXimum? <segment_number>  
CALC:LIMit:NOISe:MAX? 1
```

Return Format

Real, Real, Real, Real

Attribute Summary

Synchronization Required: none
Preset (*RST) State: (all segments deleted)
SCPI Compliance: Instrument Specific.

Additional Information

Segment amplitude units are those of the current display graph type.

CALCulate:LIMit:NOISe:MAXimum:DELeTe

Delete the specified limit line segment for the maximum amplitude limit for noise data.

Command Syntax

```
CALCulate:LIMit:NOISe:MAXimum:DELeTe <segment_number>  
<segment_number> ::= an integer in the range 1-100
```

Example

```
CALC:LIMit:NOISe:MAX:DELeTe 1
```

Query Syntax

Command Only

Attribute Summary

Synchronization Required: none

Preset (*RST) State: NA

SCPI Compliance: Instrument Specific.

CALCulate:LIMit:NOISe:MAXimum: DELeTe:ALL

Delete all limit line segments for the maximum amplitude limit for noise data.

Command Syntax

CALCulate:LIMit:NOISe:MAXimum:DELeTe:ALL

Example

CALC:LIMit:NOISe:MAX:DELeTe:ALL

Query Syntax

Command Only

Attribute Summary

Synchronization Required: none

Preset (*RST) State: NA

SCPI Compliance: Instrument Specific.

CALCulate:LIMit:NOISe:MAXimum:DISPlay

Display all limit line segments for the maximum amplitude limit for noise data.

Command Syntax

CALCulate:LIMit:NOISe:MAXimum:DISPlay <boolean>

<boolean> ::= OFF|0|ON|1

Example

CALC:LIMit:NOISe:MAX:DISPlay ON

Query Syntax

CALC:LIMit:NOISe:MAX:DISPlay?

Return Format

Integer 0=OFF 1=ON

Attribute Summary

Synchronization Required: none

Preset (*RST) State: OFF

SCPI Compliance: Instrument Specific.

CALCulate:LIMit:NOISe:MAXimum:FAIL?

Query if the data type exceeds the limit type for the indicated segment.

Command Syntax

Query Only

Example

CALCulate:LIMit:NOISe:MAXimum:FAIL? 2

Query Syntax

CALCulate:LIMit:NOISe:MAXimum:FAIL? <segment_number>

<segment_number> ::= an integer in the range 1-100

Return Format

Boolean 0=Pass 1=Fail

Attribute Summary

Synchronization Required: none

Preset (*RST) State: N/A

SCPI Compliance: Instrument Specific.

Additional Information

Returns 0 and an error message is generated if requested limit is not defined.
Limit does not have to be displayed.

CALCulate:LIMit:NOISe:MAXimum:FAIL: ALL?

Query if the data type exceeds the limit type for all defined segments.

Command Syntax

Query Only

Example

```
CALCulate:LIMit:NOISe:MAXimum:FAIL:ALL?
```

Query Syntax

```
CALCulate:LIMit:NOISe:MAXimum:FAIL:ALL?
```

Return Format

Boolean 0=Pass 1=Fail

Attribute Summary

Synchronization Required: none

Preset (*RST) State: N/A

SCPI Compliance: Instrument Specific.

Additional Information

Limit does not have to be displayed to query.

CALCulate:LIMit:NOISe:MINimum

Specifies a limit line segment for the minimum amplitude limit for noise data.

Command Syntax

```
CALCulate:LIMit:NOISe:MINimum <segment_number>,  
<start_offset> [<frequency_suffix>], <start_power>  
[<power_suffix>], <stop_offset> [<frequency_suffix>],  
<stop_power> [<power_suffix>]
```

```
<segment_number> ::= an integer in the range 1-100  
<start_offset> ::= a real number (NRF data)  
<start_power> ::= a real number (NRF data)  
<stop_offset> ::= a real number (NRF data)  
<stop_power> ::= a real number (NRF data)
```

Example

```
CALC:LIMit:NOISe:MIN 1, 100, -50, 10e3, -80
```

Query Syntax

```
CALCulate:LIMit:NOISe:MINimum? <segment_number>  
CALC:LIMit:NOISe:MIN? 1
```

Return Format

Real, Real, Real, Real

Attribute Summary

Synchronization Required: none
Preset (*RST) State: (all segments deleted)
SCPI Compliance: Instrument Specific.

Additional Information

Segment amplitude units are those of the current display graph type.

CALCulate:LIMit:NOISe:MINimum:DELeTe

Delete the specified limit line segment for the minimum amplitude limit for noise data.

Command Syntax

```
CALCulate:LIMit:NOISe:MINimum:DELeTe <segment_number>  
<segment_number> ::= an integer in the range 1-100
```

Example

```
CALC:LIMit:NOISe:MIN:DELeTe 1
```

Query Syntax

Command Only

Attribute Summary

Synchronization Required: none

Preset (*RST) State: NA

SCPI Compliance: Instrument Specific.

CALCulate:LIMit:NOISe:MINimum: DELeTe:ALL

Delete all limit line segments for the minimum amplitude limit for noise data.

Command Syntax

CALCulate:LIMit:NOISe:MINimum:DELeTe:ALL

Example

CALC:LIMit:NOISe:MIN:DELeTe:ALL

Query Syntax

Command Only

Attribute Summary

Synchronization Required: none

Preset (*RST) State: NA

SCPI Compliance: Instrument Specific.

CALCulate:LIMit:NOISe:MINimum:DISPlay

Display all limit line segments for the minimum amplitude limit for noise data.

Command Syntax

CALCulate:LIMit:NOISe:MINimum:DISPlay <boolean>

<boolean>: := OFF|0|ON|1

Example

CALC:LIMit:NOISe:MIN:DISPlay ON

Query Syntax

CALC:LIMit:NOISe:MIN:DISPlay?

Return Format

Integer 0=OFF 1=ON

Attribute Summary

Synchronization Required: none

Preset (*RST) State: OFF

SCPI Compliance: Instrument Specific.

CALCulate:LIMit:NOISe:MINimum:FAIL?

Query if the data type exceeds the limit type for the indicated segment.

Command Syntax

Query Only

Example

CALCulate:LIMit:NOISe:MINimum:FAIL? 2

Query Syntax

CALCulate:LIMit:NOISe:MINimum:FAIL? <segment_number>

<segment_number> ::= an integer in the range 1-100

Return Format

Boolean 0=Pass 1=Fail

Attribute Summary

Synchronization Required: none

Preset (*RST) State: N/A

SCPI Compliance: Instrument Specific.

Additional Information

Returns 0 and an error message is generated if requested limit is not defined. Limit does not have to be displayed.

CALCulate:LIMit:NOISe:MINimum: FAIL:ALL?

Query if the data type exceeds the limit type for all defined segments.

Command Syntax

Query Only

Example

```
CALCulate:LIMit:NOISe:MINimum:FAIL:ALL?
```

Query Syntax

```
CALCulate:LIMit:NOISe:MINimum:FAIL:ALL?
```

Return Format

Boolean 0=Pass 1=Fail

Attribute Summary

Synchronization Required: none

Preset (*RST) State: N/A

SCPI Compliance: Instrument Specific.

Additional Information

Limit does not have to be displayed to query.

CALCulate:LIMit:SPUR:MAXimum

Specifies a limit line segment for the maximum amplitude limit for spur data.

Command Syntax

```
CALCulate:LIMit:SPUR:MAXimum <segment_number>, <start_offset>  
[<frequency_suffix>], <start_power> [power_suffix],  
<stop_offset> [<frequency_suffix>], <stop_power>  
[power_suffix]
```

<segment_number> ::= an integer in the range 1-100

<start_offset> ::= a real number (NRF data)

<start_power> ::= a real number (NRF data)

<stop_offset> ::= a real number (NRF data)

<stop_power> ::= a real number (NRF data)

Example

```
CALC:LIMit:SPUR:MAX 1, 100, -50, 10e3, -80
```

Query Syntax

```
CALCulate:LIMit:SPUR:MAXimum? <segment_number>  
CALC:LIMit:SPUR:MAX? 1
```

Return Format

Real, Real, Real, Real

Attribute Summary

Synchronization Required: none

Preset (*RST) State: (all segments deleted)

SCPI Compliance: Instrument Specific.

Additional Information

Segment amplitude units are those of the current display graph type.

CALCulate:LIMit:SPUR:MAXimum:DELeTe

Delete the specified limit line segment for the maximum amplitude limit for spur data.

Command Syntax

```
CALCulate:LIMit:SPUR:MAXimum:DELeTe <segment_number>
```

```
<segment_number> ::= an integer in the range 1-100
```

Example

```
CALC:LIMit:SPUR:MAX:DELeTe 1
```

Query Syntax

Command Only

Attribute Summary

Synchronization Required: none

Preset (*RST) State: NA

SCPI Compliance: Instrument Specific.

CALCulate:LIMit:SPUR:MAXimum: DElete:ALL

Delete all limit line segments for the maximum amplitude limit for spur data.

Command Syntax

CALCulate:LIMit:SPUR:MAXimum:DElete:ALL

Example

CALC:LIMit:SPUR:MAX:DElete:ALL

Query Syntax

Command Only

Attribute Summary

Synchronization Required: none

Preset (*RST) State: NA

SCPI Compliance: Instrument Specific.

CALCulate:LIMit:SPUR:MAXimum:DISPlay

Display all limit line segments for the maximum amplitude limit for spur data.

Command Syntax

```
CALCulate:LIMit:SPUR:MAXimum:DISPlay <boolean>  
<boolean> ::= OFF|0|ON|1
```

Example

```
CALC:LIMit:SPUR:MAX:DISPlay ON
```

Query Syntax

```
CALC:LIMit:SPUR:MAX:DISPlay?
```

Return Format

Integer 0=OFF 1=ON

Attribute Summary

Synchronization Required: none

Preset (*RST) State: OFF

SCPI Compliance: Instrument Specific.

CALCulate:LIMit:SPUR:MAXimum:FAIL?

Query if the data type exceeds the limit type for the indicated segment.

Command Syntax

Query Only

Example

CALCulate:LIMit:SPUR:MAXimum:FAIL? 2

Query Syntax

CALCulate:LIMit:SPUR:MAXimum:FAIL? <segment_number>

<segment_number> ::= an integer in the range 1-100

Return Format

Boolean 0=Pass 1=Fail

Attribute Summary

Synchronization Required: none

Preset (*RST) State: N/A

SCPI Compliance: Instrument Specific.

Additional Information

Returns 0 and an error message is generated if requested limit is not defined.
Limit does not have to be displayed.

CALCulate:LIMit:SPUR:MAXimum: FAIL:ALL?

Query if the data type exceeds the limit type for all defined segments.

Command Syntax

Query Only

Example

```
CALCulate:LIMit:SPUR:MAXimum:FAIL:ALL?
```

Query Syntax

```
CALCulate:LIMit:SPUR:MAXimum:FAIL:ALL?
```

Return Format

Boolean 0=Pass 1=Fail

Attribute Summary

Synchronization Required: none

Preset (*RST) State: N/A

SCPI Compliance: Instrument Specific.

Additional Information

Limit does not have to be displayed to query.

CALCulate:LIMit:SPUR:MINimum

Specifies a limit line segment for the minimum amplitude limit for noise data.

Command Syntax

```
CALCulate:LIMit:SPUR:MINimum <segment_number>, <start_offset>  
[<frequency_suffix>], <start_power> [power_suffix],  
<stop_offset> [<frequency_suffix>], <stop_power>  
[power_suffix]
```

<segment_number> ::= an integer in the range 1-100

<start_offset> ::= a real number (NRF data)

<start_power> ::= a real number (NRF data)

<stop_offset> ::= a real number (NRF data)

<stop_power> ::= a real number (NRF data)

Example

```
CALC:LIMit:SPUR:MIN 1, 100, -50, 10e3, -80
```

Query Syntax

```
CALCulate:LIMit:SPUR:MINimum? <segment_number>  
CALC:LIMit:SPUR:MIN? 1
```

Return Format

Real, Real, Real, Real

Attribute Summary

Synchronization Required: none

Preset (*RST) State: (all segments deleted)

SCPI Compliance: Instrument Specific.

Additional Information

Segment amplitude units are those of the current display graph type.

CALCulate:LIMit:SPUR:MINimum:DElete

Delete the specified limit line segment for the minimum amplitude limit for spur data.

Command Syntax

```
CALCulate:LIMit:SPUR:MINimum:DElete <segment_number>  
<segment_number> ::= an integer in the range 1-100
```

Example

```
CALC:LIMit:SPUR:MIN:DElete 1
```

Query Syntax

Command Only

Attribute Summary

Synchronization Required: none

Preset (*RST) State: NA

SCPI Compliance: Instrument Specific.

CALCulate:LIMit:SPUR:MINimum: DELeTe:ALL

Delete all limit line segments for the minimum amplitude limit for spur data.

Command Syntax

CALCulate:LIMit:SPUR:MINimum:DELeTe:ALL

Example

CALC:LIMit:SPUR:MIN:DELeTe:ALL

Query Syntax

Command Only

Attribute Summary

Synchronization Required: none

Preset (*RST) State: NA

SCPI Compliance: Instrument Specific.

CALCulate:LIMit:NOISe:MINimum:DISPlay

Display all limit line segments for the minimum amplitude limit for spur data.

Command Syntax

```
CALCulate:LIMit:SPUR:MINimum:DISPlay <boolean>  
<boolean> ::= OFF|0|ON|1
```

Example

```
CALC:LIMit:SPUR:MIN:DISPlay ON
```

Query Syntax

```
CALC:LIMit:SPUR:MIN:DISPlay?
```

Return Format

Integer 0=OFF 1=ON

Attribute Summary

Synchronization Required: none

Preset (*RST) State: OFF

SCPI Compliance: Instrument Specific.

CALCulate:LIMit:SPUR:MINimum:FAIL?

Query if the data type exceeds the limit type for the indicated segment.

Command Syntax

Query Only

Example

```
CALCulate:LIMit:SPUR:MINimum:FAIL? 2
```

Query Syntax

```
CALCulate:LIMit:SPUR:MINimum:FAIL? <segment_number>  
<segment_number> ::= an integer in the range 1-100
```

Return Format

Boolean 0=Pass 1=Fail

Attribute Summary

Synchronization Required: none
Preset (*RST) State: N/A
SCPI Compliance: Instrument Specific.

Additional Information

Returns 0 and an error message is generated if requested limit is not defined.
Limit does not have to be displayed.

CALCulate:LIMit:SPUR:MINimum: FAIL:ALL?

Query if the data type exceeds the limit type for all defined segments.

Command Syntax

Query Only

Example

CALCulate:LIMit:SPUR:MINimum:FAIL:ALL?

Query Syntax

CALCulate:LIMit:SPUR:MINimum:FAIL:ALL?

Return Format

Boolean 0=Pass 1=Fail

Attribute Summary

Synchronization Required: none

Preset (*RST) State: N/A

SCPI Compliance: Instrument Specific.

Additional Information

Limit does not have to be displayed to query.

CALCulate:PBWidth

Power Bandwidth adjustment.

Command Syntax

```
CALCulate:PBWidth <number>  
<number> ::= a real number (NRf data)
```

Query Syntax

```
CALCulate:PBWidth?
```

Example

```
CALCulate:PBWidth 100
```

Return Format

Real

Attribute Summary

Synchronization Required: none

Preset (*RST) State: 1 Hz

SCPI Compliance: Instrument Specific.

Additional Information

Applied to data output by CALCulate:DATA? when outputting:

CALCulate:VIEW:TRACe (start, stop, #points, distribution(log/linear))
or CALCulate:VIEW:XYData

CALCulate:SElect

Specify which data set is selected for return by:

- CALCulate:DATA?
- CALCulate:ACCumulated[:PHASe]? (tau, start-offset, stop-offset)
- CALCulate:AVARiance? (tau, Frequency offset)
- CALCulate:TVARiance? (tau, Frequency offset)

Command Syntax

CALCulate:SElect BOTH|NOISe|SPURs

Example

CALCulate:SElect NOISe

Query Syntax

CALCulate:SElect?

Return Format

String

Attribute Summary

Synchronization Required: none

Preset (*RST) State: BOTH

SCPI Compliance: Instrument Specific.

Additional Information

- BOTH = return both noise and spur data
- NOISe = return noise data only
- SPURs = return spur data only

CALCulate:SMOothing

Specify the degree of smoothing to be applied to data.

Command Syntax

```
CALCulate:SMOothing <number>  
<number> ::= an integer (NRf data)
```

Example

```
CALCulate:SMOothing 2
```

Query Syntax

```
CALCulate:SMOothing?
```

Return Format

Real

Attribute Summary

Synchronization Required: none

Preset (*RST) State: 0

SCPI Compliance: Instrument Specific.

Additional Information

Smoothing performs a running average on each data point. The smoothing value is the number of points to average on each side of the current data point. Thus, the number of points averaged is:

$2*n+1$ where n is the value used for smoothing

Applied to data output by CALCulate:DATA? when outputting:

CALCulate:VIEW:TRACe (start, stop, #points, distribution(log/linear))
or CALCulate:VIEW:XYData

CALCulate:TRANSform

Specify the graph type transform for CALCulate:DATA?

Command Syntax

```
CALCulate:TRANSform
SSBNoise | SDPF | SDFE | SFFF | NF2Port | AMNoise | BBAND
```

```
SSBNoise = Single Sideband Noise (dBc/Hz)
SDPF      = Spectral Density of Phase Fluctuations (dB/Hz)
SDFE      = Spectral Density of Frequency Fluctuations
            (Hz/Sqrt(Hz))
SFFF      = Spectral Density of Fractional Frequency
            Fluctuations (1/Sqrt(Hz))
NF2Port   = Noise Figure for a 2-Port device (dB)
AMNoise   = AM Noise (dB)
BBAND     = Base Band Noise (dB)
```

Example

```
CALCulate:TRANSform BBAND
```

Query Syntax

```
CALCulate:TRANSform?
```

Return Format

String

Attribute Summary

```
Synchronization Required: none
Preset (*RST) State: SSBNoise
SCPI Compliance: Instrument Specific.
```

CALCulate:TVARiance?

Jitter.

Command Syntax

Query Only

Query Syntax

```
CALCulate:TVARiance? <tau>,<upper_offset>  
<tau> ::= a real number (NRf data)  
<upper_offset> ::= a real number (NRf data)
```

Return Format

Real

Attribute Summary

Synchronization Required: none
Preset (*RST) State: NA
SCPI Compliance: Instrument Specific.

Additional Information

Performs calculation on currently selected data as specified by:
CALCulate:SElect (BOTH | NOISe | SPURs)

CALCulate:VIEW:SPURtable

Specifies the list of spurs returned by CALCulate:DATA?. The list of spurs is returned as a list of frequency/amplitude pairs.

Command Syntax

CALCulate:VIEW:SPURtable

Example

CALCulate:VIEW:SPURtable

Query Syntax

Command Only

Attribute Summary

Synchronization Required: none

Preset (*RST) State: NA

SCPI Compliance: Instrument Specific.

Additional Information

When this view is specified, results returned via CALCulate:DATA? are affected by the following:

- CALCulate:TRANSform
- CALCulate:YShif

CALCulate:VIEW:TRACe

Specifies that trace data be returned by CALCulate:DATA?.

Command Syntax

```
CALCulate:VIEW:TRACe <start_offset> [<frequency_suffix>],  
                    <stop_offset> [<frequency_suffix>],  
                    <points>,  
                    <distribution>
```

<start_offset> ::= a real number (NRf data)

<stop_offset> ::= a real number (NRf data)

<frequency_suffix> ::= Hz|kHz|MHz|GHz

<points> ::= an integer (NRf data)

<distribution> ::= LOG|LINear

Example

```
CALCulate:VIEW:TRACe 10Hz, 100kHz, 1000, LOG
```

Query Syntax

```
CALCulate:VIEW:TRACe
```

Return Format

Frequency/amplitude pairs.

Attribute Summary

Synchronization Required: none

Preset (*RST) State: NA

SCPI Compliance: Instrument Specific.

Additional Information

CALCulate:VIEW:TRACe data is interpolated between start and stop offsets to provide data for the requested distribution (log or linear) and number of points. If you want the actual measured data points use CALCulate:VIEW:XYData.

CALCulate:VIEW:XYData

Specifies that data triples be returned by CALCulate:DATA? for the user-defined measurement range.

Command Syntax

CALCulate:VIEW:XYData

Example

CALCulate:VIEW:XYData

Query Syntax

Command Only

Attribute Summary

Synchronization Required: none

Preset (*RST) State: NA

SCPI Compliance: Instrument Specific.

Additional Information

Returns measured data points.

CALCulate:YShifT

The specified amplitude adjustment is applied to data returned by CALCulate:DATA?.

Command Syntax

```
CALCulate:YShifT <number> [dB]
<number> ::= a real number (NRf data)
```

Example

```
CALCulate:YShifT 2
```

Query Syntax

```
CALCulate:YShifT?
```

Return Format

Real

Attribute Summary

Synchronization Required: none

Preset (*RST) State: 0.0

SCPI Compliance: Instrument Specific.

Additional Information

Applied to data output by CALCulate:DATA? when outputting:

- CALCulate:VIEW:SPURtable
- CALCulate:VIEW:TRACe (start, stop, #points, distribution (log/linear))
or
- CALCulate:VIEW:XYData

Refer to Chapter 1 for update information.

The calibration process for a phase lock loop measurement involves determining the phase Detector Constant, the VCO Tune Constant, and verifying the Loop Suppression. The phase Detector Constant is the sensitivity of the phase detector expressed in volts per radian (V/Rad). The VCO Tune Constant is the sensitivity of the tuning input of the Voltage Controlled Oscillator (VCO) expressed in Hertz per volt (Hz/V). The Loop Suppression verification characterizes the phase lock loop configured for the measurement and then compares the measured loop response with a theoretical loop suppression curve.

The commands in this subsystem have the following command hierarchy:

- CALibrate:DETEctor:CONStant (value)
- CALibrate:DETEctor:CONStant:METhod (BEATnote | DCPeak | SSPur | DSPur | FMD | AMIN | PSHift | NONE)
- CALibrate:DETEctor:CONStant:SPUR:AMPLitude (value)
- CALibrate:DETEctor:CONStant:SPUR:OFFSet (value)
- CALibrate:VCO:IRESistance (value)
- CALibrate:VCO:PLLSuppress?
- CALibrate:VCO:PLLSuppress:APOLe?
- CALibrate:VCO:PLLSuppress:CBWidth?
- CALibrate:VCO:PLLSuppress (0 | OFF | 1 | ON)
- CALibrate:VCO:PLLSuppress:DISPlay[:ALWays] (0 | OFF | 1 | ON)
- CALibrate:VCO:PLLSuppress:ERRor (value)
- CALibrate:VCO:PLLSuppress:ERRor:ACTion (THEoretical | ADJusted | PAUSE)
- CALibrate:VCO:PLLSuppress:PTRange?
- CALibrate:VCO:TCONstant?
- CALibrate:VCO:TCONstant:METhod (CURRENT | MEASURE | CALCulate)
- CALibrate:VCO:TCONstant:NOMinal (value)
- CALibrate:VCO:TMODE (EFC | DCFM)
- CALibrate:VCO:VCADjust (value)
- CALibrate:VCO:VCENter (value)
- CALibrate:VCO:VRANge (value)

CALibrate:DETECTOR:CONSTant

Specify the detector constant in Volts / Radian.

Command Syntax

```
CALibrate:DETECTOR:CONSTant <number>  
<number> ::= a real number (NRf data)
```

Example

```
CALibrate:DETECTOR:CONSTant 1.5E-6
```

Query Syntax

```
CALibrate:DETECTOR:CONSTant?
```

Return Format

Real

Attribute Summary

Synchronization Required: none

Preset (*RST) State: 1E-6

SCPI Compliance:Instrument Specific.

Additional Information

The phase detector translates relative phase fluctuations between the two phase detector inputs to voltage fluctuations. The amplitude of the phase detector's output voltage is proportional to the amount of relative phase fluctuation at its two inputs scaled by and the sensitivity of the detector. The phase detector's sensitivity is the phase Detector Constant. The phase Detector Constant determines the absolute noise floor of the test system. The accuracy of the Phase Lock Loop measurement is dependent on the accuracy of the phase Detector Constant.

Use the Current Detector Constant and Measure the Detector Constant are the two methods available for determining the phase Detector Constant.

CALibrate:DETEctor:CONStant:METhod

This command is used to specify the method by which to measure the detector constant.

Command Syntax

```
CALibrate:DETEctor:CONStant:METhod BEATnote|DCPeak|SSPur  
|DSPur|FMDeviation|AMINternal|PSHift|NONE
```

Example

```
CALibrate:DETEctor:CONStant:METhod FM Deviation
```

Query Syntax

```
CALibrate:DETEctor:CONStant:METhod?
```

Return Format

String

Attribute Summary

Synchronization Required: none

Preset (*RST) State: BEATnote

SCPI Compliance: Instrument Specific.

Additional Information

- BEATnote Derive from measured beatnote
- DCPeak Derive from measured +/- DC peak voltage
- SSPur Derive from single-sided spur
- DSPur Derive from double-sided spur
- FMDeviation Derive from FM rate and deviation
- AMINternal Use AM internal automatic self-calibration.
- NONE Use the current detector constant

A subset of the detector constant methods is valid for each measurement type as follows:

- ABSolute(Phase Noise using a Phase Locked Loop)
 - NONE Use the current detector constant
 - BEATnote Derive from measured beatnote
- RESidual(Phase Noise without using a Phase Locked Loop)
 - NONE Use the current detector constant
 - BEATnote Derive from measured beatnote
 - DCPeak Derive from measured +/- DC peak voltage
 - SSPur Derive from single-sided spur
 - DSPur Derive from double-sided spur
- FMDiscriminator (Phase Noise using an FM Discriminator)

CALibrate:DETEctor:CONStant:METhod

- NONE Use the current detector constant
- DSPur Derive from double-sided spur
- FMDeviation Derive from FM rate and deviation
- AM (AM Noise Measurement)
 - NONE Use the current detector constant
 - AMINternal Use AM internal automatic self-calibration.
 - SSPur Derive from single-sided spur
 - DSPur Derive from double-sided spur
- BBANd. Base Band Noise Measurement
 - The detector constant method is not applicable for baseband noise measurements.
- NOTSet (Base Band Noise Measurement without using the Test Set)
 - The detector constant method is not applicable for baseband noise measurements.

CALibrate:DETEctor:CONStant:SPUR: AMPLitude

Specify the spur amplitude used to measure the detector constant when the detector constant method is set to SSPUR or DSPUR.

Command Syntax

```
CALibrate:DETEctor:CONStant:SPUR:AMPLitude <number> [dB]  
<number> ::= a real number (NRF data)
```

Example

```
CALibrate:DETEctor:CONStant:SPUR:AMPLitude -100
```

Query Syntax

```
CALibrate:DETEctor:CONStant:SPUR:AMPLitude?
```

Return Format

Real

Attribute Summary

Synchronization Required: none

Preset (*RST) State: -130

SCPI Compliance: Instrument Specific.

CALibrate:DETEctor:CONStant:SPUR:OFFSet

Specify the spur offset frequency used to measure the detector constant when the detector constant method is set to SSPUR or DSPUR.

Command Syntax

```
CALibrate:DETEctor:CONStant:SPUR:OFFSet <number>  
[Hz|kHz|MHz|GHz]  
<number> ::= a real number (NRf data)
```

Example

```
CALibrate:DETEctor:CONStant:SPUR:OFFSet 10
```

Query Syntax

```
CALibrate:DETEctor:CONStant:SPUR:OFFSet?
```

Return Format

Real

Attribute Summary

Synchronization Required: none

Preset (*RST) State: 1 Hz

SCPI Compliance: Instrument Specific.

CALibrate:VCO:IRESistance

Specify the input resistance of the VCO.

Command Syntax

```
CALibrate:VCO:IRESistance <number> [OHMS|KOHMS|MOHMS]  
<number> ::= a real number (NRf data)
```

Example

```
CALibrate:VCO:IRESistance 500E3
```

Query Syntax

```
CALibrate:VCO:IRESistance?
```

Return Format

Real

Attribute Summary

Synchronization Required: none

Preset (*RST) State: 1E6 Ohms

SCPI Compliance: Instrument Specific.

Additional Information

VCO Tune Port Input Resistance is the input impedance of the VCO's tune port. This entry must be accurate to within +/-5% when the VCO Tune Constant is not going to be measured. When the Tune Constant is going to be measured by the software, the actual value of the VCO Tune Port Input Resistance is not critical, and an entry of 1E+6 is recommended.

CALibrate:VCO:PLLSuppress

Enable/disable PLL suppression verification.

Phase Lock Loop Suppression verification can be turned on or off. Verification of the Phase Lock Loop Suppression insures that the Phase Lock Loop Bandwidth (PLL BW), the Phase Detector Constant, and the VCO Tune Constant are all accurate for the measurement. (Note that the software always corrects for the loop suppression whether or not verification is selected.)

Command Syntax

```
CALibrate:VCO:PLLSuppression <boolean>  
<boolean> ::= OFF|0|ON|1
```

Example

```
CALibrate:VCO:PLLSuppression OFF
```

Query Syntax

```
CALibrate:VCO:PLLSuppression?
```

Return Format

Integer 0=OFF 1=ON

Attribute Summary

Synchronization Required: none

Preset (*RST) State: ON

SCPI Compliance: Instrument Specific.

Additional Information

Verification of the Phase Lock Loop Suppression insures that the Phase Lock Loop Bandwidth (PLL BW), the Phase Detector Constant, and the VCO Tune Constant are all accurate for the measurement. (Note that the software always corrects for the loop suppression whether or not verification is selected.)

The software verifies Loop Suppression before beginning the measurement by closing the phase lock loop and injecting noise into the loop. The response of the loop is measured and recorded. The loop is then opened, and a measurement is made of the open loop response. The suppression of the PLL is the difference between the closed loop response and open loop response to the injected noise. The measured Loop Suppression curve is smoothed and then four points are selected along the curve including the peak and the 8 dB point. The measured curve is then compared to the theoretical curve at the four selected points.

The amount of adjustment required to fit the theoretical curve to the measured curve is recorded as the PLL GAIN CHANGE on the Suppression Graph and the frequency of the Assumed Pole. The Assumed Pole is a software controlled term used to fit the Adjusted Theoretical loop response to the smoothed measured response of the phase lock loop. The MAX ERROR is the final amount of error existing between the adjusted theoretical curve and the measured curve of the phase lock loop suppression. If the PLL GAIN CHANGE or MAX ERROR is greater than 1 dB, the test system indicates that an Accuracy Specification Degradation has been detected.

Verification of the loop suppression may be prevented by excessive noise close to the carrier, by a large spur within the bandwidth of the PLL, or by injection locking.

It should be assumed that an indicated Accuracy Specification Degradation applies at all offset frequencies (both inside and outside of the loop bandwidth).

If the Detector Constant and VCO Tune Constant are known to be accurate, the PTR is less than 10 MHz, and the VCO tuning port bandwidth is greater than PLL BW X 10, then a measurement without Loop Suppression Verification will be accurate.

CALibrate:VCO:PLLSuppress:APOLe?

This is the frequency of the Assumed Pole required to adjust the theoretical loop suppression to match the smoothed measured Loop Suppression. The Assumed Pole frequency is normally much greater than the PLL BW.

Query Syntax

`CALibrate:VCO:PLLSuppress:APOLe?`

Return Format

Real (Hz)

Attribute Summary

Synchronization Required: none

Preset (*RST) State: n/a

SCPI Compliance: Instrument Specific.

Additional Information

If the Assumed Pole is adjusted to a frequency of less than 10 X PLL BW, peaking in the PLL suppression is indicated. For PLL BWs less than 20 kHz, an Assumed Pole of less than 10 X PLL BW indicates a delay or phase shift in the VCO Tune Port. For PLL BWs greater than 20 kHz, the Assumed Pole may be adjusted to less than 10 X PLL BW to account for phase shifts in the test set.

Also see: `CALibrate:VCO:PLLSuppress`

CALibrate:VCO:PLLSuppress:CBWidth?

This is the predicted Phase Lock Loop Bandwidth for the measurement. The predicted PLL BW is based on the predicted PTR. The Closed PLL BW will not be adjusted as a result of an accuracy degradation. If an accuracy degradation is detected, the amount of error is determined from either the PLL Gain Change or the Maximum Error, which ever is larger. The degradation itself if 1 dB less than the greater of these.

Query Syntax

CALibrate:VCO:PLLSuppress:CBWidth?

Return Format

Real (Hz)

Attribute Summary

Synchronization Required: none

Preset (*RST) State: n/a

SCPI Compliance: Instrument Specific.

Additional Information

Also see: CALibrate:VCO:PLLSuppress

CALibrate:VCO:PLLSuppress:DISPlay [:ALWays]

When turned ON, the PLL suppression graph will be displayed each time PLL suppression verification is performed. When turned OFF, the PLL suppression graph is displayed only when the suppression error is out of bounds.

Command Syntax

```
CALibrate:VCO:PLLSuppress:DISPlay[:ALWays] <boolean>  
<boolean> ::= OFF|0|ON|1
```

Example

```
CALibrate:VCO:PLLSuppress:DISPlay ON
```

Query Syntax

```
CALibrate:VCO:PLLSuppress:DISPlay?
```

Return Format

Integer 0=OFF 1=ON

Attribute Summary

Synchronization Required: none
Preset (*RST) State: OFF
SCPI Compliance: Instrument Specific.

CALibrate:VCO:PLLSuppress:ERRor

Specify the acceptable error for PLL suppression verification.

The Maximum Suppression Error Limit is the final amount of error existing between the adjusted theoretical curve and the measured curve of the phase lock loop suppression.

Command Syntax

```
CALibrate:VCO:PLLSuppress:ERRor <number> [dB]
```

Example

```
CALibrate:VCO:PLLSuppress:ERRor .5
```

Query Syntax

```
CALibrate:VCO:PLLSuppress:ERRor?
```

Return Format

Real

Attribute Summary

Synchronization Required: none

Preset (*RST) State: 1 dB

SCPI Compliance: Instrument Specific.

CALibrate:VCO:PLLSuppress:ERRor:ACTion

Specify which PLL suppression curve to use in the case that PLL suppression verification fails.

Command Syntax

CALibrate:VCO:PLLSuppress:ERRor:ACTion
THEoretical|ADJusted|PAUSE

Example

CALibrate:VCO:PLLSuppress:ERRor:ACTion THEoretical

Query Syntax

CALibrate:VCO:PLLSuppress:ERRor:ACTion?

Return Format

String

Attribute Summary

Synchronization Required: none
Preset (*RST) State: THEoretical
SCPI Compliance: Instrument Specific.

Additional Information

See CALibrate:VCO:PLLSuppress:ERRor.

CALibrate:VCO:PLLSuppress:PTRange?

This is the Peak Tuning Range (PTR) for the measurement determined from the VCO Tune Constant and the Tune Range of VCO. This is the key parameter in determining the PLL properties, the Drift tracking Range, and the ability to phase lock sources with high close in noise.

Query Syntax

CALibrate:VCO:PLLSuppress:PTRange?

Return Format

Real (Hz)

Attribute Summary

Synchronization Required: none

Preset (*RST) State: n/a

SCPI Compliance: Instrument Specific.

Additional Information

The PTR value displayed should be approximately equal to the product of the VCO Tune Constant times the Tune Range of VCO. This will not be the case when a significant Accuracy Specification Degradation is detected (> 4 db) by the Loop Suppression Verification. For Accuracy Specification Degradations of >4 db, the PTR and Assumed Pole are adjusted when the theoretical loop suppression is fit to the smoothed measured Loop Suppression. In these cases, the test system displays the adjusted PTR. Any time the PTR must be adjusted by more than 1 db (as indicated by an Accuracy Specification Degradation of greater than 0 db) it is an indication that the phase Detector Constant or the VCO Tune Constant will be in error at frequency offsets near the PLL BW, or that the PLL BW is being affected by some other problem such as injection locking.

Also see: CALibrate:VCO:PLLSuppress

CALibrate:VCO:TCONstant?

Query the VCO tune constant (which was determined by the calibration method).

Command Syntax

Query Only

Query Syntax

`CALibrate:VCO:TCONstant?`

Return Format

Real (Hz/Volt)

Attribute Summary

Synchronization Required: none

Preset (*RST) State: 1E6

SCPI Compliance: Instrument Specific.

Additional Information

The VCO Tune Constant is the frequency sensitivity of the VCO source to voltage changes at its control voltage input. This constant is measured in Hz/V. The software uses the VCO Tune Constant along with the defined Tune Range of VCO and the VCO's Tune Port Input Resistance to determine the Peak Tuning Range for the measurement. The Bandwidth of the Phase Lock Loop, the Lock Capture Range, and the Drift Tracking Range for the measurement are all derived from the Peak Tuning Range. Three calibration techniques are available for determining the VCO Tune Constant for the measurement; Use current VCO tuning sensitivity, Measure the VCO tuning sensitivity, Calculate from the expected VCO tuning sensitivity using the Tune Port Resistance.

CALibrate:VCO:TCONstant:METHOD

Specify the method used to determine the VCO tune constant.

Command Syntax

CALibrate:VCO:TCONstant:METHOD CURRENT|MEASure|CALCulate

Example

CALibrate:VCO:TCONstant:METHOD?

Query Syntax

CALibrate:VCO:TCONstant:METHOD MEASure

Return Format

String

Attribute Summary

Synchronization Required: none

Preset (*RST) State: MEASure

SCPI Compliance: Instrument Specific.

Additional Information

CURRENT

This calibration method uses the value of the VCO Tune Constant currently in memory. Selecting this method saves time for measuring the VCO Tune Constant without any degradation in accuracy, providing either the same VCO source is being used or a source with an identical tuning constant is being used (within +/- 5%).

MEASure

Selecting this calibration method causes the test system to automatically measure the tuning sensitivity and tuning linearity of the VCO source. Measuring the VCO Tune Constant is the recommended method under most circumstances for ensuring measurement accuracy. The test system cannot measure the VCO Tune Constant for an entered Peak Tuning Range greater than 500 kHz for test systems configured without an RF Analyzer, and 200 MHz for test systems configured with an RF Analyzer.

When measuring the VCO Tune Constant, the test system requires that the VCO Tune Constant defined in the Measurement Definition, Sources tab window be within a factor of two of the VCO's actual sensitivity. This approximation is required in order for the software to establish a beatnote. The frequency response of the VCO source must remain flat (+/- 3 dB) from dc to the phase lock loop bandwidth (PLL BW).

CALCulate

Selecting the Calculate from Expected Tune Constant method causes the test system to use the entered VCO Tune Constant and VCO Tune Port Input Resistance that have been entered in the Measurement Definition, Sources tab window. This method requires that the entered VCO Tune Constant is within +/- 5% of the VCO's actual sensitivity. If it is not, the test system reports an Accuracy Specification Degradation after verifying the Phase Lock Loop Suppression.

$$\text{VCO Tuning Constant} = \text{Expected Tuning Constant} * (\text{Rin} / (\text{Rin} + 50))$$

Where: Rin = VCO Tune Port Input Resistance

CALibrate:VCO:TCONstant:NOMinal

Specify the nominal value used to calculate the VCO tune constant.

Command Syntax

```
CALibrate:VCO:TCONstant:NOMinal <number>  
<number> ::= a real number (Nrf data)
```

Example

```
CALibrate:VCO:TCONstant:NOMinal 2e3
```

Query Syntax

```
CALibrate:VCO:TCONstant:NOMinal?
```

Return Format

Real

Attribute Summary

Synchronization Required: none

Preset (*RST) State: 10E3

SCPI Compliance: Instrument Specific.

Additional Information

VCO Tune Constant is the resulting frequency deviation of the Voltage Controlled Oscillator (VCO) for a 1 volt change at its tune port. This entry must be accurate to within +/-5% when the phase noise software is not going to measure the VCO Tune Constant. The entry must be accurate to within a factor of 2 when the VCO Tune Constant is going to be measured by the software.

CALibrate:VCO:TMODe

Specify the VCO tune mode. The VCO is the source to which the tune voltage is applied as specified in [SENSe:]TVCO REFerence|CARRier.

Command Syntax

`CALibrate:VCO:TMODe EFC|DCFM`

Example

`CALibrate:VCO:TMODe EFC`

Query Syntax

`CALibrate:VCO:TMODe?`

Return Format

String

Attribute Summary

Synchronization Required: none
Preset (*RST) State: EFC
SCPI Compliance: Instrument Specific.

Additional Information

- EFC - Electronic Frequency Control
- DCFM - DC Frequency Modulation

NOTE

Not all VCO's support tuning via DC FM.

CALibrate:VCO:VCADjust

Specify the allowable adjustment range (from the VCO's center voltage).

Command Syntax

```
CALibrate:VCO:VCADjust <number> [V]  
<number> ::= a real number (NRf data)
```

Example

```
CALibrate:VCO:VCADjust 2.5
```

Query Syntax

```
CALibrate:VCO:VCADjust?
```

Return Format

Real

Attribute Summary

Synchronization Required: none

Preset (*RST) State: 1 V

SCPI Compliance: Instrument Specific.

CALibrate:VCO:VCENter

Specifies the center voltage for the VCO.

Command Syntax

```
CALibrate:VCO:VCENter <number> [V]  
<number> ::= a real number (NRf data)
```

Example

```
CALibrate:VCO:VCENter 1.5
```

Query Syntax

```
CALibrate:VCO:VCENter?
```

Return Format

Real

Attribute Summary

Synchronization Required: none

Preset (*RST) State: 0 Volts

SCPI Compliance:Instrument Specific.

Additional Information

Center Voltage of VCO Tune Curve is the center voltage of the tuning curve of the VCO, that is, the voltage required at the tune port input to set the VCO to the desired carrier frequency for the measurement. A 0 volt entry allows the maximum Voltage Tune Range available from the test set. Entries other than zero restrict the Voltage Tune Range as described below:

Table 10-1 Voltage Tune Range

Center Voltage	Maximum Voltage Tuning Range
$\leq 2V$	$\pm 10V$
$> 2V$	$\pm 12V - \text{Center Voltage}$

CALibrate:VCO:VRANge

Specifies the tuning voltage range for the VCO.

Command Syntax

```
CALibrate:VCO:VRANge <number> [V]  
<number> ::= a real number (NRf data)
```

Example

```
CALibrate:VCO:VRANge 4.5
```

Query Syntax

```
CALibrate:VCO:VRANge?
```

Return Format

Real

Attribute Summary

Synchronization Required: none

Preset (*RST) State: 1 Volts

SCPI Compliance: Instrument Specific.

Additional Information

This is the defined Tune Range of VCO through which the test system may tune the VCO around the Center Voltage. The VCO output must not change in amplitude by greater than 1 dB, show any reversal in frequency tuning, or be susceptible to damage within the defined Tune Range around the Center Voltage.

For the phase noise software to measure the VCO Tune Constant accurately, the VCO must tune monotonically across +/-45% of the Tune Range of VCO around the Center Voltage, and the slope must not go to zero. If the VCO Tune Constant is not measured by the test system, the VCO must tune to within +/-10%% of the Tune Range of VCO around the Center Voltage using the defined VCO Tune Constant.

To protect the VCO from possible damage, the Tune Voltage output port should not be connected to the VCO's input port until the Connect diagram is displayed.

VCO Tuning Constant * Voltage Tuning Range = Peak Tuning Range

11

DISPlay Commands

Refer to Chapter 1 for update information.

Use the DISPLAY commands to control the display of data in the Measurement Development Client GUI. These commands only effect the display presented to the user and do NOT effect data output by CALCulate:DATA? or DATA? queries. Similar commands are available in the CALCulate tree.

selection and presentation of textual, graphical, and TRACe information. This information includes measurement data, user-interaction display, and data presented to the instrument by the controller. DISPlay is independent of, and does not modify, how data is returned to the controller.

The commands in this subsystem have the following command hierarchy:

- DISPlay:GRAPh:BOUNds:AMPLitude? (SSBN | SDPF | SDFF | SFFF | NF2P | AMN | BBAN)
- DISPlay:GRAPh:BOUNds:AMPLitude (Transform,Min,Max)
- DISPlay:GRAPh:BOUNds:FREQuency (Min,Max)
- DISPlay:GRAPh:CFSCale (value)
- DISPlay:GRAPh:DIPower (value)
- DISPlay:GRAPh:PBWidth (value)
- DISPlay:GRAPh:SMOothing (value)
- DISPlay:GRAPh:TRANsform (SSBN | SDPF | SDFF | SFFF | NF2P | AMN | BBAN)
- DISPlay:GRAPh:YSHift (value)
- DISPlay:MEASurement:TIME?
- DISPlay:TEXT:TITLe ("string")

DISPlay:GRAPh:BOUNds:AMPLitude

Amplitude coordinate units and bounds of graphical display.

Command Syntax

```
DISPlay:GRAPh:BOUNds:AMPLitude <transform>,<Ymin>,<Ymax>  
<transform> ::= SSBNoise|SDPF|SDFF|SFFF|NF2Port|AMNoise|BBAND  
• SSBNoise = Single Sideband Noise (dBc/Hz)  
• SDPF = Spectral Density of Phase Fluctuations (dB/Hz)  
• SDFF = Spectral Density of Frequency Fluctuations (Hz/Sqrt(Hz))  
• SFFF = Spectral Density of Fractional Frequency Fluctuations  
  (Hz/Sqrt(Hz))  
• NF2Port = Noise Figure for a 2-Port device (dB)  
• AMNoise = AM Noise (dB)  
• BBAND = Base Band Noise (dB)  
<Ymin> ::= a real number (NRf data)  
          limits:  
<Ymax> ::= a real number (NRf data)  
          limits:
```

Command Example

```
DISPlay:GRAPh:BOUNds:AMPLitude -150,0
```

Query Syntax

```
DISPlay:GRAPh:BOUNds:AMPLitude? <transform>  
<transform> ::= SSBNoise|SDPF|SDFF|SFFF|NF2Port|AMNoise|BBAND
```

Query Example

```
DISPlay:GRAPh:BOUNds:AMPLitude? SDFF
```

Return Format

Real, Real

Attribute Summary

Synchronization Required: none

Preset (*RST) State: SSBNoise, -170, 0

SCPI Compliance: Instrument Specific.

DISPlay:GRAPh:BOUNds:AMPLitude?

Amplitude coordinate units and bounds of graphical display.

Command Syntax

Query Only

Query Syntax

```
DISPlay:GRAPh:BOUNds:AMPLitude? <transform>,<Ymin>,<Ymax>  
<transform> ::= SSBNoise|SDPF|SDFF|SFFF|NF2Port|AMNoise|BBAND
```

Example

```
DISPlay:GRAPh:BOUNds:AMPLitude SDFF
```

Return Format

Real, Real

Attribute Summary

Synchronization Required: none

Preset (*RST) State: SSBNoise, -170, 0

SCPI Compliance: Instrument Specific.

DISPlay:GRAPh:BOUNds:FREQuency

Frequency coordinate bounds of graphical display.

Command Syntax

```
DISPlay:GRAPh:BOUNds:FREQuency <Xmin>,<Xmax>  
<Xmin> ::= a real number (NRf data)  
<Xmax> ::= a real number (NRf data)
```

Example

```
DISPlay:GRAPh:BOUNds:FREQuency 10, 100E3
```

Query Syntax

```
DISPlay:GRAPh:BOUNds:FREQuency?
```

Return Format

Real, Real

Attribute Summary

Synchronization Required: none
Preset (*RST) State: 1, 10E6
SCPI Compliance: Instrument Specific.

DISPlay:GRAPh:CFSCale

Specify the Carrier Frequency SCale multiplier used for displaying the graph.

Command Syntax

```
DISPlay:GRAPh:CFSCale <number>  
<number> ::= a real number (NRf data)
```

Example

```
DISPlay:GRAPh:CFSCale 2.0  
DISPlay:GRAPh:CFSCale 0.5
```

Query Syntax

```
DISPlay:GRAPh:CFSCale?
```

Return Format

Real

Attribute Summary

Synchronization Required: none

Preset (*RST) State: 1.0

SCPI Compliance: Instrument Specific.

Additional Information

The carrier frequency scaling function allows you to plot results based on frequency changes in the carrier, and to plot the noise of the measured source to a translated frequency. The scale factor can be any number greater than zero. Entering a factor-of-2 doubles the frequency, a factor-of-3 triples the frequency, and so on. A number less than one corresponds to a scaling to a lower carrier frequency.

Plots made after the scale factor has been entered indicate noise for the scaled frequency. Noise levels increase for scale factors > 1 and decrease for factors <1.

DISPlay:GRAPh:DIPower

Specify the power present at the input of the DUT in dBm.

Command Syntax

```
DISPlay:GRAPh:DIPower <number>  
<number> ::= a real number (NRf data)
```

Example

```
DISPlay:GRAPh:DIPower -10
```

Query Syntax

```
DISPlay:GRAPh:DIPower?
```

Return Format

Real

Attribute Summary

Synchronization Required: none
Preset (*RST) State: 0
SCPI Compliance: Instrument Specific.

DISPlay:GRAPh:PBWidth

Specify the trace power bandwidth adjustment used in displaying the graph.

Command Syntax

```
DISPlay:GRAPh:PBWidth <number>  
<number> ::= a real number (Nrf data)
```

Example

```
DISPlay:GRAPh:PBWidth 100 Hz
```

Query Syntax

```
DISPlay:GRAPh:PBWidth?
```

Return Format

Real

Attribute Summary

Synchronization Required: none

Preset (*RST) State: 1.0 Hz

SCPI Compliance: Instrument Specific.

Additional Information

This function specifies the bandwidth frequency the trace data will be normalized to. Because phase noise is usually specified in a 1 Hz bandwidth, the measurement result must also be corrected for the equivalent noise bandwidth for the analyzer. This bandwidth normalization process requires subtracting $10 \log$ (equivalent noise in Hz) from the measured value. For example, if a value of -123 dBc is obtained from a measurement with an analyzer equivalent noise bandwidth of 1.2 kHz, this value must be corrected by subtracting $10 \log(1200)$, yielding -153.8 dBc/Hz.

DISPlay:GRAPh:SMOothing

Specify the degree of smoothing used in displaying the graph.

Command Syntax

```
DISPlay:GRAPh:SMOothing <number>  
<number> ::= an integer (NRf data)
```

Example

```
DISPlay:GRAPh:SMOothing 5
```

Query Syntax

```
DISPlay:GRAPh:SMOothing?
```

Return Format

Integer

Attribute Summary

Synchronization Required: none

Preset (*RST) State: 0

SCPI Compliance: Instrument Specific.

Additional Information

Smoothing performs a running average on each data point. The smoothing value is the number of points to average on each side of the displayed data point. Thus, the number of points averaged is:

$2*n+1$ where n is the value used for smoothing

DISPlay:GRAPh:TRANSform

Specify the transformation used to display data on the graph.

Command Syntax

```
DISPlay:GRAPh:TRANSform  
SSBNoise | SDPF | SDFP | SFFF | NF2Port | AMNoise | BBAND
```

Example

```
DISPlay:GRAPh:TRANSform SFFF
```

Query Syntax

```
DISPlay:GRAPh:TRANSform?
```

Return Format

String

Attribute Summary

Synchronization Required: none

Preset (*RST) State: SSBNoise

SCPI Compliance: Instrument Specific.

Additional Information

- SSBNoise = Single Sideband Noise (dBc/Hz)
- SDPF = Spectral Density of Phase Fluctuations (dB/Hz)
- SDFP = Spectral Density of Frequency Fluctuations (Hz/Sqrt(Hz))
- SFFF = Spectral Density of Fractional Frequency Fluctuations 1/Sqrt(Hz)
- NF2Port = Noise Figure for a 2-Port device (dB)
- AMNoise = AM Noise (dB)
- BBAND = Base Band Noise (dB)

DISPlay:GRAPh:YSHift

Specify the amplitude offset used in displaying the graph.

Command Syntax

```
DISPlay:GRAPh:YSHift <number> [dBc]  
<number> ::= a real number (NRf data)
```

Example

```
DISPlay:GRAPh:YSHift 2.0
```

Query Syntax

```
DISPlay:GRAPh:YSHift?
```

Return Format

Real

Attribute Summary

Synchronization Required: none
Preset (*RST) State:0.0 dBm
SCPI Compliance: Instrument Specific.

DISPlay:MEASurement:TIME?

The date, start time and stop time of the current measurement.

Query Syntax

DISPlay:MEASurement:TIME?

Return Format

String

Attribute Summary

Synchronization Required: none

Preset (*RST) State: n/a

SCPI Compliance: Instrument Specific.

Additional Information

Example: 29 Jul 1997 16:51:57 - 16:53:19

DISPlay:TEXT:TITLe

Title of the measurement.

Command Syntax

DISPlay:TEXT:TITLe <“string”>

Query Syntax

DISPlay:TEXT:TITLe?

Example

DISPlay:TEXT:TITLe “Absolute Phase Noise for Device #1234”

Return Format

String

Attribute Summary

Synchronization Required: none

Preset (*RST) State: “E5500 Absolute Phase Noise Measurement.”

SCPI Compliance: Instrument Specific.

Refer to Chapter 1 for update information.

Use the FORMat command to set a data format for transferring numeric and array information. This data format is used for both command and response data by those commands that are specifically designated to be affected by the FORMat function.

The commands in this subsystem have the following command hierarchy:

FORMat[:DATA] (ASCii, length)

FORMat[:DATA]

Specifies the format (ASCII) and number of significant digits for block data transferred across the GPIB. Only ASCII format is currently supported.

Command Syntax

```
FORMat[:DATA] ASCii, <length>  
  <length> ::= an integer (NRf data)  
  Where <length>: number of significant digits
```

Example

```
FORMat ASCii, 9
```

Query Syntax

```
FORMat?
```

Return Format

```
CHAR, Integer
```

Attribute Summary

Synchronization Required: none
Preset (*RST) State: ASCii,6
SCPI Compliance: Instrument Specific.

Additional Information

FORMat[:DATA] affects transfers initiated with the following commands:

- CALC:DATA?
- [SENSe:]DATA?

ASCII data consists of one line per data point separated by CR/LF. See CALC:DATA? and [SENSe:]DATA? pages for line format information.

Refer to Chapter 1 for update information.

Use the INITiate commands to control when measurement acquisition occurs.

The commands in this subsystem have the following command hierarchy:

- INITiate[:IMMEDIATE][:ALL]
- INITiate[:IMMEDIATE]:CALibrate
- INITiate[:IMMEDIATE]:MEASure

Also see related PAUSE:... commands

INITiate[:IMMediate][:ALL]

Initiates a system calibration and measurement.

Command Syntax

INITiate[:IMMediate][:ALL]

Example

INIT

Query Syntax

Command Only

Attribute Summary

Synchronization Required: none

Preset (*RST) State: NA

SCPI Compliance: Standard

INITiate[:IMMediate]:CALibrate

Initiate system calibration.

Command Syntax

`INITiate[:IMMediate]:CALibrate`

Example

`INITiate:CALibrate`

Query Syntax

Command Only

Attribute Summary

Synchronization Required: none

Preset (*RST) State: NA

SCPI Compliance: Instrument Specific.

Additional Information

Calibration parameters are set using the `CALibrate: tree`.

INITiate[:IMMediate]:MEASure

Initiate measurement only – no calibration. Used for repeat measurements. If the measurement definition changes a new calibration is required.

Command Syntax

INITiate[:IMMediate]:MEASure

Example

INITiate:MEASure

Query Syntax

Command Only

Attribute Summary

Synchronization Required: none

Preset (*RST) State: NA

SCPI Compliance: Instrument Specific.

Refer to Chapter 1 for update information.

The MMEMory commands provide mass storage capabilities for the system.

The commands in this subsystem have the following command hierarchy:

- MMEMory:LOAD[:ALL] ("path\name.pnm")
- MMEMory:STORe[:ALL] ("path\name.pnm")

MMEMemory:LOAD[:ALL]

Load a measurement definition and data file.

Command Syntax

```
MMEMemory:LOAD[:ALL] <"path\file_name.pnm">  
<path\file_name.pnm> ::= quoted string (full or relative path  
and filename)
```

Example

```
MMEMemory:LOAD "Measfiles\Phase1.pnm"
```

Attribute Summary

Synchronization Required: none

*RST State: NA

SCPI Compliance: Standard

Additional Information

Use the MMEMemory:STORE command to store a file to load via MMEMemory:LOAD.

Note that there is no format specification. This file is stored in a format (.pnm files) relevant only to the Phase Noise Graphical User Interface (Phase.exe) or SCPI clients

MMEMory:STORe

Stores the current measurement definition and data to a file.

Command Syntax

MMEMory:STORe <“path\file_name.pnm”>

<path\file_name.pnm> :: = quoted string (full or relative path and filename)

Example

MMEMory:STORe “.\measurements\meas1.pnm”

Query Syntax

Command Only

Return Format

Attribute Summary

Synchronization Required: none

*RST State: NA

SCPI Compliance: Standard

Additional Information

Use the MMEMory:LOAD command to load a file stored via MMEMory:STORe.

Note that there is no format specification. This file is stored in a format relevant only to the Phase Noise GUI or SCPI clients (.pnm files)

Refer to Chapter 1 for update information.

The commands in this subsystem have the following command hierarchy:

- PAUSE:ABORt:AUTO (0 | OFF | 1 | ON)
- PAUSE:ADJust:LNAGain
- PAUSE:ADJust:VCOCenter
- PAUSE:CONNect (0 | OFF | 1 | ON)
- PAUSE:CONTinue
- PAUSE:RETRy
- PAUSE:SPECial

PAUSE:ABORT:AUTO

Automatically abort a calibration or measurement if and when a pause event occurs.

Command Syntax

```
PAUSE:ABORT:AUTO <boolean>  
<boolean> ::= OFF|0|ON|1
```

Example

```
PAUSE:ABORT:AUTO OFF
```

Query Syntax

```
PAUSE:ABORT:AUTO?
```

Return Format

Integer 0=OFF 1=ON

Attribute Summary

Synchronization Required: none

Preset (*RST) State: 0

SCPI Compliance: Instrument Specific.

Additional Information

Pause events include PAUSE:CONNECT ON, PAUSE:SPECIAL and any other events which interrupt the progression of a calibration or measurement.

PAUSE:ADJust:LNAGain

Adjust the nominal gain for the low noise amplifier in the test set after autorange when the LNA gain adjust pause point is enabled.

Command Syntax

```
PAUSE:ADJust:LNAGain <number> [dB]
```

<number> ::= a real number (NRf data)
Accepts numeric input but snaps to nearest of the following values:

```
14  
28  
42  
56
```

Example

```
PAUSE:ADJust:LNAGain 28
```

Query Syntax

```
PAUSE:ADJust:LNAGain?
```

Return Format

Real

Attribute Summary

Synchronization Required: none

Preset (*RST) State: 42 dB

SCPI Compliance: Instrument Specific.

Additional Information

[SENSe:]TSET:LNAGain:METHOD PAUSE must be set for PAUSE:ADJust:LNAGain to be in effect.

Also see: [SENSe:]TSET:LNAGain:METHOD

PAUSE:ADJust:VCOCenter

Adjust the VCO tuning center voltage when paused.

Command Syntax

```
PAUSE:ADJust:VCOCenter <number> [dB]  
<number> ::= a real number (NRf data)
```

Example

```
PAUSE:ADJust:VCOCenter .5
```

Query Syntax

```
PAUSE:ADJust:VCOCenter?
```

Return Format

Real

Attribute Summary

Synchronization Required: none

Preset (*RST) State: 0 V

SCPI Compliance: Instrument Specific.

Additional Information

The Tune Voltage allows you to change the tune voltage. IF the tuning voltage exceeds 10% of the peak tuning range during system calibration, the phase noise software stops the procedure and informs the user that the source needs to be retuned before the measurement can begin.

PAUSE:CONNEct

Enable or disable the presentation of the connect diagram.

Command Syntax

PAUSE:CONNEct <boolean>

<boolean> ::= OFF|0|ON|1

Example

PAUSE:CONNEct ON

Query Syntax

PAUSE:CONNEct?

Return Format

Integer 0=OFF 1=ON

Attribute Summary

Synchronization Required: none

Preset (*RST) State: OFF

SCPI Compliance: Instrument Specific.

Additional Information

When enabled, the system will always stop at this fixed point and wait for a INITiate:PAUSE:CONTInue command before proceeding. A message will be placed in the message queue.

If PAUSE:ABORT:AUTO is ON and PAUSE:CONNEct is ON the measurement or calibration will abort when the connect diagram is presented. Normally, this should be avoided.

PAUSE:CONTINUE

Continue from a pause point.

Command Syntax

PAUSE:CONTINUE

Example

PAUSE:CONT

Query Syntax

Command Only

Attribute Summary

Synchronization required: none

Preset (*RST) State: NA

SCPI Compliance: Instrument Specific.

PAUSE:RETRy

Retry the last operation prior to a pause point.

Command Syntax

PAUSE:RETRy

Query Syntax

Command Only

Example

PAUSE:RETRy

Attribute Summary

Synchronization Required: none

Preset (*RST) State:NA

SCPI Compliance: Instrument Specific.

PAUSE:SPEcIal

Specify a branch to be taken following a pause point.

Command Syntax

PAUSE:SPEcIal <response>

<response> : := a real number (NRf data)

limits: see discussion

Query Syntax

Example

Return Format

Attribute Summary

Synchronization Required: none

Preset (*RST) State: NULL

SCPI Compliance: Instrument Specific.

Refer to Chapter 1 for update information.

The commands in this subsystem have the following command hierarchy:

- [:SENSE]:DATA?
- [:SENSE]:DATA:HEADer:POINts?
- [:SENSE]:DCONverter:ATTenuator (0 to 35 dB in 5 dB steps)
- [:SENSE]:DCONverter:ATTenuator:AUTO (0 | OFF | 1 | ON)
- [:SENSE]:DCONverter:ATTenuator:AUTO:IMMEDIATE
- [:SENSE]:DCONverter:BAND (UWAVE | A | K | Q | U | V | W)
- [:SENSE]:DCONverter:FREQuency (value)
- [:SENSE]:DCONverter:IFFREQuency?
- [:SENSE]:DCONverter:IFGain (-10 to 45 dB in 5 dB steps)
- [:SENSE]:DCONverter:IFGain:AUTO (0 | OFF | 1 | ON)
- [:SENSE]:DCONverter:IFGain:AUTO:IMMEDIATE
- [:SENSE]:DCONverter:L1BWidth (value)
- [:SENSE]:DCONverter:L6BWidth (value)
- [:SENSE]:DCONverter:LOPower (value)
- [:SENSE]:DCONverter:LOSelect (value)
- [:SENSE]:DCONverter:LOSelect:AUTO (0 | OFF | 1 | ON)
- [:SENSE]:DCONverter:MEXT:BIAS (value)
- [:SENSE]:DCONverter:MEXT:BIAS:STATe (0 | OFF | 1 | ON)
- [:SENSE]:DCONverter:MMLO?
- [:SENSE]:DCONverter:PLO
- [:SENSE]:DCONverter:REFerence (value)
- [:SENSE]:DCONverter:TSENSitivity?
- [:SENSE]:DCONverter:TUNE:PORT (OFF | INTernal | FRONT | REAR)
- [:SENSE]:DETector:FREQuency (value)
- [:SENSE]:DETector:SElect (AUTO | EXT | LFR | HFR | UWAVE | TAM | TINoise | DCAM | DIAM)
- [:SENSE]:FFT:INPut:COUPling (AC | DC)
- [:SENSE]:NOISE:BBGain (value)
- [:SENSE]:NOISE:FFT (EXTended | MULTiple)
- [:SENSE]:NOISE:MEAStype (ABS | RES | FM | AM | BBAN | NOTS)
- [:SENSE]:NOISE:PULSed (0 | OFF | 1 | ON)
- [:SENSE]:NOISE:QUADrature[:METHod] (PSHifter | SOURCE)
- [:SENSE]:RANGE:FFT:AVERage:MINimum (value)

SENSe Commands

- [:SENSe]:RANGe:FFT:SEGTable[:MEASurement][:QUALity]
(NORMal | FAST | ACCurate | CUSTom)
- [:SENSe]:RANGe:OFFSet (Start frequency, Stop frequency)
- [:SENSe]:RANGe:SWEPT:SEGTable[:MEASurement][:QUALity]
(NORMal | FAST | ACCurate | CUSTom)
- [:SENSe]:TSET:ATTenuator (value)
- [:SENSe]:TSET:ATTenuator:AUTO (0 | OFF | 1 | ON)
- [:SENSe]:TSET:DCBlock (0 | OFF | 1 | ON)
- [:SENSe]:TSET:LNAgain (value)
- [:SENSe]:TSET:LNAgain:METHOD (AUTO | FIXed | PAUSE)
- [:SENSe]:TSET:LNAgain:MINimum (value)
- [:SENSe]:TSET:LPF (value)
- [:SENSe]:TSET:LPF:AUTO (0 | OFF | 1 | ON)
- [:SENSe]:TSET:PLL:ATTenuator (value)
- [:SENSe]:TSET:PLL:UNLock:IGNore (0 | OFF | 1 | ON)
- [:SENSe]:TVCO (REFerence | CARRier | DCONverter | INTernal)

[SENSe:]DATA?

Retrieves the raw S-phi data for the user-defined measurement range.

Command Syntax

Query Only

Query Syntax

[SENSe:]DATA?

Return Format

Returns triples with each element separated by a comma and terminated by cr/lf. Each triple consists of the frequency, amplitude and spur flag.

Frequency, Amplitude, Spur_Flag

Spur_Flag: 1=spur 0=noise

Attribute Summary

Synchronization Required: none

Preset (*RST) State: NA

SCPI Compliance: Standard

Additional Information

S-phi data corresponds to data displayed by DISPLAY:GRAPH:TRANSform SDPF.

Returns one line per data point as indicated by [SENSe:]DATA:HEADer:POINts?. Number of significant digits for frequency and amplitude is controlled by FORMat[:DATA] command.

[SENSe:]DATA:HEADer:POINts?

Returns the number of data points which will be returned by [SENSe:]DATA?

Command Syntax

Query Only

Query Syntax

[SENSe:]DATA:HEADer:POINts?

Return Format

Integer

Attribute Summary

Synchronization Required: none

Preset (*RST) State: 0

SCPI Compliance: Standard

[SENSe:]DCONverter:ATTenuator

Specify the amount of input attenuation for the downconverter.

Command Syntax

```
[SENSe:]DCONverter:ATTenuator <number> [dB]
```

<number> ::= a real number (NRf data)

Accepts numeric inputs, but snaps to nearest of the following values: 0, 5, 10, 15, 20, 25, 30, 35

Example

```
DCONverter:ATTenuator 15
```

Query Syntax

```
[SENSe:]DCONverter:ATTenuator?
```

Return Format

Real

Attribute Summary

Synchronization Required: none

Preset (*RST) State: 0 dBm

SCPI Compliance: Instrument Specific.

Additional Information

This command is significant only if an Agilent/HP 70427A downconverter is used. The 70427A is the only downconverter available with an input attenuator. The value specified here will be used by the system if autoranging has been disabled ([SENSe:]DCONverter:ATTenuator:AUTO OFF).

[SENSe:]DCONverter:ATTenuator:AUTO

When enabled, the system will perform automatic autoranging of the downconverter input attenuator.

Command Syntax

```
[SENSe:]DCONverter:ATTenuator:AUTO <boolean>  
<boolean> ::= OFF|0|ON|1
```

Example

```
DCONverter:ATTenuator:AUTO ON
```

Query Syntax

```
DCONverter:ATTenuator:AUTO?
```

Return Format

Integer 0=OFF, 1=ON

Attribute Summary

Synchronization Required: none

Preset (*RST) State: ON

SCPI Compliance: Instrument Specific.

Additional Information

If auto input attenuation is enabled, at the appropriate time in the measurement the measurement engine will instruct the downconverter to do an autorange immediate on the input attenuator. If auto is disabled, the value set by [SENSe:]DCONverter:ATTenuator will be used.

If both an Agilent/HP 70420A OPT 001 and 70427A are in the system, and both attenuators have been set to AUTO, then the test set attenuator is autoranged first, then the downconverter attenuator, then the IF gain. That is, always perform autoranges in order from the beginning of the signal path, paying attention to the auto/manual switches set by the user. The philosophy is to never override a user's value if manual is selected.

[SENSe:]DCONverter:ATTenuator:AUTO: IMMEDIATE

Perform downconverter attenuator autorange.

Command Syntax

[SENSe:]DCONverter:ATTenuator:AUTO:IMMEDIATE

Attribute Summary

Synchronization Required: none

Preset (*RST) State: n/a

SCPI Compliance: Instrument Specific.

Additional Information

Normally issued by the driver automatically. See
[SENSe:]DCONverter:Attenuator:AUTO

[SENSe:]DCONverter:BAND

Specify the input frequency band of the downconverter.

Command Syntax

[SENSe:]DCONverter:BAND UWAVE|A|K|Q|U|V|W

Example

DCONverter:BAND K

Query Syntax

DCONverter:BAND?

Return Format

String

Attribute Summary

Synchronization Required: none

Preset (*RST) State: UWAVE

SCPI Compliance: Instrument Specific.

Additional Information

Selects either microwave or millimeter-wave mode. In millimeter mode an external harmonic mixer is used to convert the signal to a frequency within the downconverter's IF range.

[SENSe:]DCONverter:FREQuency

Specify the downconverter input frequency.

Command Syntax

```
[SENSe:]DCONverter:FREQuency <number> [Hz|kHz|MHz|GHz]  
<number> ::= a real number (NRf data)
```

Example

```
DCONverter:FREQuency 8E9
```

Query Syntax

```
DCONverter:FREQuency?
```

Return Format

Real

Attribute Summary

Synchronization Required: none

Preset (*RST) State: 10E9 Hz

SCPI Compliance: Instrument Specific.

[SENSe:]DCONverter:IFFRequency?

Query the IF frequency of the downconverter.

Command Syntax

Query Only

Query Syntax

[SENSe:]DCONverter:IFFRequency?

Return Format

Real

Attribute Summary

Synchronization Required: none

Preset (*RST) State: NA

SCPI Compliance: Instrument Specific.

**Additional
Information**

Displays the downconverter's IF frequency. When an Input Frequency is entered, the downconverter computes the IF frequency and displays it. The IF frequency can be changed by changing the LO Frequency. The new IF frequency will be calculated and displayed even if it is out of range. The IF frequency is automatically set when the Auto is selected.

[SENSe:]DCONverter:IFGain

Specify the gain for the IF amplifier in the downconverter.

Command Syntax

```
[SENSe:]DCONverter:IFGain <number> [dB]
<number> ::= a real number (NRf data)
           limits: -10:+45 in 5dB steps
```

Example

```
DCONverter:IFGain
```

Query Syntax

```
DCONverter:IFGain?
```

Return Format

Real

Attribute Summary

Synchronization Required: none

Preset (*RST) State: 0 dBm

SCPI Compliance: Instrument Specific.

[SENSe:]DCONverter:IFGain:AUTO

Turn downconverter IF Gain auto ranging on or off.

Command Syntax

```
[SENSe:]DCONverter:IFGain:AUTO <boolean>  
<boolean> ::= OFF|0|ON|1
```

Example

```
DCONverter:IFGain:AUTO OFF
```

Query Syntax

```
DCONverter:IFGain:AUTO?
```

Return Format

Integer 0=OFF, 1=ON

Attribute Summary

Synchronization Required: none

Preset (*RST) State: ON

SCPI Compliance: Instrument Specific.

Additional Information

This command does not turn on or off an autoranging function in the downconverter. If autoranging is enabled, at the appropriate time in the measurement, the phase noise measurement software sends a command to the downconverter to have it do a one-time autorange.

The IF gain is automatically set by triggering a one-time autoranging when the Auto button is clicked. The downconverter autoranges the IF amplifier's gain by first setting the gain to its minimum (-10 dB) then stepping the gain up in 5 dB steps until the signal level at the IF amplifier's output is between 0 and +6 dBm.

[SENSe:]DCONverter:IFGain:AUTO: IMMEDIATE

Perform downconverter IF gain autorange.

Command Syntax

[SENSe:]DCONverter:IFGain:AUTO:IMMEDIATE

Attribute Summary

Synchronization Required: none

Preset (*RST) State: n/a

SCPI Compliance: Instrument Specific.

Additional Information

Normally issued by the driver automatically. See
[SENDe:]DCONverter:IFGain:AUTO

[SENSe:]DCONverter:L1BWidth

Sets or queries the 100 MHz PLL bandwidth value.

Command Syntax

[SENSe:]DCONverter:L1BWidth <value>
<value> ::= a real number (NRf data)

Accepts numeric input but snaps to nearest of the following values:

- 25 Hz
- 53 Hz
- 126 Hz
- 300 Hz
- 650 Hz
- 1500 Hz
- 3600 Hz
- 10000 Hz

Example

DCONverter:L1BWidth 300

Query Syntax

DCONverter:L1BWidth?

Return Format

Real

Attribute Summary

Synchronization Required: none

Preset (*RST) State: 3

SCPI Compliance: Instrument Specific.

[SENSe:]DCONverter:L6BWidth

Specify the downconverter's 600 MHz PLL bandwidth.

Command Syntax

[SENSe:]DCONverter:L6BWidth <value>
<value> ::= a real number (NRf data)

Accepts numeric input but snaps to nearest of the following values:

- 4 kHz
- 10 kHz
- 17 kHz
- 30 kHz

Example

```
DCONverter:L6BWidth 17E6
```

Query Syntax

```
DCONverter:L6BWidth?
```

Return Format

Real

Attribute Summary

Synchronization Required: none

Preset (*RST) State: 10E3

SCPI Compliance: standard

[SENSe:]DCONverter:LOPower

Specify downconverter LO power.

Command Syntax

```
[SENSe:]DCONverter:LOPower <number> [dBm]  
<number> ::= a real number (NRf data)
```

Example

```
DCONverter:LOPower 20
```

Query Syntax

```
DCONverter:LOPower?
```

Return Format

Attribute Summary

Synchronization Required: none
Preset (*RST) State: 20 dBm
SCPI Compliance: Instrument Specific.

[SENSe:]DCONverter:LOSelect

Set the frequency to steer the YIG to one of the SRD comb frequencies.

Command Syntax

```
[SENSe:]DCONverter:LOSelect <frequency> [<frequency_suffix>]  
<frequency> ::= a real number (NRf data)  
<frequency_suffix> ::= MHz|GHz
```

Example

```
DCONverter:LOSelect 1.5E9
```

Query Syntax

```
DCONverter:LOSelect?
```

Return Format

Real

Attribute Summary

Synchronization Required: none

Preset (*RST) State: 1E9 Hz

SCPI Compliance: Instrument Specific.

[SENSe:]DCONverter:LOSelect:AUTO

Turn downconverter frequency auto selection on or off.

Command Syntax

```
[SENSe:]DCONverter:LOSelect:AUTO <boolean>  
<boolean> ::= OFF|0|ON|1
```

Example

```
DCONverter:LOSelect:AUTO OFF
```

Query Syntax

```
DCONverter:LOSelect:AUTO?
```

Return Format

Integer 0=OFF 1=ON

Attribute Summary

Synchronization Required: none

Preset (*RST) State: ON

SCPI Compliance: Instrument Specific.

[SENSe:]DCONverter:MEXT:BIAS

Sets or queries the external mixer DC bias current.

Command Syntax

```
[SENSe:]DCONverter:MEXT:BIAS <number> [mA]  
<number> ::= a real number (NRf data)  
limits: -10:+10
```

Example

```
DCONverter:MEXT:BIAS.002
```

Query Syntax

```
DCONverter:MEXT:BIAS?
```

Return Format

Real

Attribute Summary

Synchronization Required: none

Preset (*RST) State: 0.0 mA

SCPI Compliance: Instrument Specific.

Additional Information

The millimeter mixer is external to the downconverter and may require DC bias for proper operation with odd harmonics of the LO. The DC bias current will be present at the signal input of the downconverter when enabled.

[SENSe:]DCONverter:MEXT:BIAS:STATE

Turns the external mixer bias voltage ON|OFF.

Command Syntax

```
[SENSe:]DCONverter:MEXT:BIAS:STATE <boolean>  
<boolean> ::= OFF|0|ON|1
```

Example

```
DCONverter:MEXT:BIAS:STATE ON
```

Query Syntax

```
DCONverter:MEXT:BIAS:STATE?
```

Return Format

Integer 0=OFF 1=ON

Attribute Summary

Synchronization Required: none

Preset (*RST) State: OFF

SCPI Compliance: Instrument Specific.

Additional Information

The millimeter mixer is external to the downconverter and may require DC bias for proper operation with odd harmonics of the LO. The DC bias voltage will be present at the signal input of the downconverter when enabled.

[SENSe:]DCONverter:MMLO?

Query the external Millimeter mixer's LO frequency.

Command Syntax

Query Only

Query Syntax

[SENSe:]DCONverter:MMLO?

Return Format

Real

Attribute Summary

Synchronization Required: none

Preset (*RST) State: N\A

SCPI Compliance: Instrument Specific.

[SENSe:]DCONverter:PLO

Peak Downconverter LO.

Command Syntax

[SENSe:]DCONverter:PLO

Attribute Summary

Synchronization Required: none

Preset (*RST) State: NA

SCPI Compliance: Instrument Specific.

Additional Information

Peak LO command - 70427A Option H16 version. Normally issued by the driver automatically. May also be issued by downconverter asset control panel.

[SENSe:]DCONverter:REFerence

Select downconverter reference oscillator(s)

Command Syntax

```
[SENSe:]DCONverter:REFerence <number>  
<value> ::= a real number (NRf data)
```

Accepts numeric input but snaps to nearest of the following values:

- 10 E6
- 100 E6
- 600 E6

Example

```
[SENSe:]DCONverter:REFerence 100E6
```

Query Syntax

```
[SENSe:]DCONverter:REFerence?
```

Return Format

Real

Attribute Summary

Synchronization Required: none

Preset (*RST) State: 10.000000 E+6

SCPI Compliance: Instrument Specific.

Additional Information

- 10E6 selects the 10, 100, and 600 MHz oscillators
- 100E6 selects the 100 and 600 MHz oscillators
- 600E6 selects the 600 MHz oscillator

[SENSe:]DCONverter:TSENsitivity?

Returns the downconverter tuning sensitivity.

Command Syntax

Query Only

Query Syntax

[SENSe:]DCONverter:TSENsitivity?

Return Format

- 0
- 0.05
- 1
- 20

Attribute Summary

Synchronization Required: none

Preset (*RST) State: 1

SCPI Compliance: Instrument Specific.

Additional Information

The response to this query is controlled by the setting of these commands:

- [SENSe:]DCONverter:TUNE:PORT
- [SENSe:]DCONverter:REference

0 corresponds to: 10MHz No Tune, All oscillators phase locked, 10 MHz tune input grounded.

- [SENSe:]DCONverter:TUNE:PORT OFF
- [SENSe:]DCONverter:REference 10E6

0.05 corresponds to: All oscillators phase locked, 10 MHz tune enabled.

- [SENSe:]DCONverter:TUNE:PORT INTernal|FRONT|REAR
- [SENSe:]DCONverter:REference 10E6

1 corresponds to: 10 MHz out of chain, 600 MHz phase locked to 100 MHz, 100 MHz tune enabled.

- [SENSe:]DCONverter:TUNE:PORT INTernal|FRONT|REAR
- [SENSe:]DCONverter:REference 100E6

20 corresponds to: 10 MHz and 100 MHz out of chain, 600 MHz tune enabled.

- [SENSe:]DCONverter:TUNE:PORT INTernal|FRONT|REAR

- [SENSe:]DCONverter:REFerence 600E6

For more information, refer to the 70427A/70428A Microwave Downconverter / Microwave Source User's Guide.

[SENSe:]DCONverter:TUNE:PORT

Select downconverter tuning port

Command Syntax

[SENSe:]DCONverter:TUNE:PORT INTernal|OFF|FRONT|REAR

Example

DCONverter:TUNE:PORT OFF

Query Syntax

[SENSe:]DCONverter:TUNE:PORT?

Return Format

String

Attribute Summary

Synchronization Required: none

Preset (*RST) State: OFF

SCPI Compliance: Instrument Specific.

[SENSe:]DETECTOR:FREQUENCY

Specify detector input frequency.

Command Syntax

```
[SENSe:]DETECTOR:FREQUENCY <number>
<number> ::= a real number (Nrf data)
```

Example

```
DETECTOR:FREQUENCY 600E6
```

Query Syntax

```
DETECTOR:FREQUENCY?
```

Return Format

Real

Attribute Summary

Synchronization Required: none

Preset (*RST) State: 500E6 Hz

SCPI Compliance: Instrument Specific.

Additional Information

The Detector Input Frequency is the frequency of the input signal connected to the system's phase detector. The Detector Input Frequency is used to:

Select the appropriate Low Pass Filter (LPF) in the phase noise system. For frequencies < 95 MHz, the test system enables the 2 MHz LPF to eliminate feed through of the fundamental signal and the additive products out of the phase detector. When the 2 MHz LPF is enabled, the maximum offset frequency the test system can measure is 2 MHz.

Set the frequency of the reference source when the reference source is under system control.

[SENSe:]DETEctor:SElect

Specify the input for the detector.

Command Syntax

```
[SENSe:]DETEctor:SElect AUTO|EXTErnal|LFRequency|HFRequency  
|UWAVE|TAM|TINoise|DCAM
```

Example

```
DETEctor:SElect LFR
```

Query Syntax

```
DETEctor:SElect?
```

Return Format

String

Attribute Summary

Synchronization Required: none

Preset (*RST) State: AUTO

SCPI Compliance: Instrument Specific.

Additional Information

- AUTO specifies auto selection of the detector. Note that this will auto select only between the LFRequency, HFRequency, and UWAVE detectors based upon detector input frequency.
- LFRequency selects the low frequency phase detector.
- HFRequency selects the high frequency phase detector.
- UWAVE selects the microwave phase detector.
- TAM selects the test set AM detector.
- DCAM selects the downconverter AM detector.
- TINoise selects the test set internal calibration noise source.

Connect diagram implications:

- AUTO - one of the two REF INPUT ports of test set is connected.
- EXTErnal or TAM - the NOISE INPUT port of the test set is connected.
- LFRequency or HFRequency - REF INPUT 500kHz-1600MHz is connected.
- UWAVE - REF INPUT 1.2-26.5 REF INPUT is connected.
- DCAM - SIGNAL INPUT for the downconverter (note that there is no AM detector in the Agilent/HP 70422A).

[SENSe:]FFT:INPut:COUPling

Selects AC or DC coupling for the FFT Analyzer Input Coupling.

Command Syntax

[SENSe:]FFT:INPut:COUPling AC|DC

Example

FFT:INPut:COUPling DC

Query Syntax

FFT:INPut:COUPling?

Return Format

String

Attribute Summary

Synchronization Required: none

Preset (*RST) State: AC

SCPI Compliance: Instrument Specific.

[SENSe:]NOISe:BBGain

Specify the compensation for an external amplifier's gain.

Command Syntax

```
[SENSe:]NOISe:BBGain <number> [dB]  
<number> ::= a real number (NRf data)
```

Example

```
NOISe:BBGain 10
```

Query Syntax

```
NOISe:BBGain?
```

Return Format

Real

Attribute Summary

Synchronization Required: none

Preset (*RST) State: 0 dBm

SCPI Compliance: Instrument Specific.

Additional Information

This is used when an amplifier is inserted between the DUT (device under test) and the 'Noise' input to the test set and the engineer wants the theoretical noise of the amplifier to be accounted for and removed from the measurement results.

This function is used to offset the plotted measurement data. It compensates for any additional amplifiers or attenuators being used when making the measurement.

- Offsets > 0 compensate for known gains.
- Offsets < 0 compensate for known losses.

[SENSe:]NOISe:FFT

Specify whether multiple time samples or an extended time sample should be used by the FFT analyzer.

Command Syntax

[SENSe:]NOISe:FFT EXTended|MULTiple

Query Syntax

NOISe:FFT MULT

Example

NOISe:FFT?

Return Format

String

Attribute Summary

Synchronization Required: none

Preset (*RST) State: MULTiple

SCPI Compliance: Instrument Specific.

Additional Information

EXTended refers to the ability of the E1430 Digitizer to record a single extended time segment from which multiple FFT's are derived. This feature should greatly reduce measurement time. MULTiple is the current method of recording smaller time segments, which produce many FFT's for averaging.

[SENSe:]NOISe:MEAStype

Specify the measurement type.

Command Syntax

[SENSe:]NOISe:MEAStype
ABSolute|RESidual|FMDiscriminator|AM|BBANd|NOTSet

Example

NOISe:MEAStype BBANd

Query Syntax

NOISe:MEAStype?

Return Format

String

Attribute Summary

Synchronization Required: none

Preset (*RST) State: ABSolute

SCPI Compliance: Instrument Specific.

Additional Information

- ABSolute Phase Noise using a Phase Locked Loop
- RESidual Phase Noise without using a Phase Locked Loop
- FMDiscriminator Phase Noise using an FM Discriminator
- AM AM Noise Measurement
- BBANd Base Band Noise Measurement
- NOTSet Base Band Noise Measurement without using the Test Set

[SENSe:]NOISe:PULSed

Turn source signal pulsed mode on or off.

Command Syntax

```
[SENSe:]NOISe:PULSed <boolean>
<boolean> ::= OFF|0|ON|1
```

Example

```
NOISe:PULSed?
```

Query Syntax

```
NOISe:PULSed ON
```

Return Format

Integer 0=OFF 1=ON

Attribute Summary

Synchronization Required: none

Preset (*RST) State: OFF

SCPI Compliance: Instrument Specific.

Additional Information

The FREQuency mode is always CW and therefore does not have to be specified. The sources can be pulsed, however.

[SENSe:]NOISe:QUADrature[:METHod]

Specify the method used to achieve quadrature for a residual measurement.

Command Syntax

[SENSe:]NOISe:QUADrature[:METHod] PSHifter|SOURce

Example

NOISe:QUADrature PSH

Query Syntax

NOISe:QUADrature?

Return Format

String

Attribute Summary

Synchronization Required: none

Preset (*RST) State: SOURce

SCPI Compliance: Instrument specific

[SENSe:]RANGe:FFT:AVERage:MINimum

Specify the minimum number of averages for the measurement range.

Command Syntax

```
[SENSe:]RANGe:FFT:AVERage:MINimum <number>  
<number> ::= a real number (Nrf data)
```

Example

```
RANGe:FFT:AVERage:MINimum 5
```

Query Syntax

```
RANGe:FFT:AVERage:MINimum?
```

Return Format

Integer

Attribute Summary

Synchronization Required: none

Preset (*RST) State: 4

SCPI Compliance: Instrument Specific.

Additional Information

The minimum number of averages defined for the measurement determines the minimum number of noise level measurements the phase noise software will make for each segment of the specified offset range. The phase noise software actually takes more averages if the minimum number of averages specified in the segment table is greater than the number of averages specified here.

[SENSe:]RANGe:FFT:SEGTable [:MEASurement][:QUALity]

Specify which FFT segment table to use for the specified range.

Command Syntax

```
[SENSe:]RANGe:FFT:SEGTable[:MEASurement][:QUALity]
NORMal|FAST|HRESolution|CUSTom
```

Example

```
RANGe:FFT:SEGTable:MEASurement:QUALity HRESolution
```

Query Syntax

```
RANGe:FFT:SEGTable:MEASurement:QUALity?
```

Return Format

String

Attribute Summary

Synchronization Required: none

Preset (*RST) State: NORMal

SCPI Compliance: Instrument Specific.

Additional Information

- NORMal
- FAST - quickest, but the least accurate
- HRESolution - slowest and most accurate
- CUSTom- speed and accuracy are user defined

[SENSe:]RANGe:OFFSet

Specify the offsets for a measurement range.

Command Syntax

```
[SENSe:]RANGe:OFFSet <start_offset> [<frequency_suffix>]
                                ,<stop_offset> [<frequency_suffix>]
<start_offset>                 ::= a real number (NRf data)
<stop_offset>                  ::= a real number (NRf data)
<frequency_suffix>             ::= Hz | kHz | MHz | GHz
```

Example

```
SENSe:RANGe:OFFSet 10Hz,500Hz
```

Query Syntax

```
SENSe:RANGe:OFFSet?
```

Return Format

Real, Real

Attribute Summary

Synchronization Required: none

Preset (*RST) State: 1Hz,100MHz

SCPI Compliance: Instrument Specific.

**[SENSe:]RANGe:SWEPT:SEGTable
[:MEASurement][:QUALity]**

Specify which Swept segment table to use for the specified range.

Command Syntax

```
[SENSe:]RANGe:SWEPT:SEGTable[:MEASurement][:QUALity]  
NORMal|FAST|HRESolution |CUSTom
```

Example

```
RANGe:SWEPT:SEGTable:MEASurement:QUALity HRESolution
```

Query Syntax

```
RANGe:SWEPT:SEGTable:MEASurement:QUALity?
```

Return Format**Attribute Summary**

Synchronization Required: none

Preset (*RST) State: NORMal

SCPI Compliance: Instrument Specific.

**Additional
Information**

- NORMal
- FAST - quickest, but the least accurate
- HRESolution - slowest and most accurate
- CUSTom- speed and accuracy are user defined

[SENSe:]TSET:ATTenuator

Specify the amount of attenuation applied at the signal input of the test set.

Command Syntax

```
[SENSe:]TSET:ATTenuator <number> [dB]  
<number> ::= a real number (NRf data)
```

Accepts numeric input but snaps to nearest of the following values:

0, 5, 10, 15, 20, 25, 30, 35

Example

```
TSET:ATTenuator 15
```

Query Syntax

```
TSET:ATTenuator?
```

Return Format

Real

Attribute Summary

Synchronization Required: none

Preset (*RST) State: 0 dBm

SCPI Compliance: Instrument Specific.

Additional Information

This command is significant only if a 70420A OPT 001 test set is used. The 70420A OPT 001 is the only test set available with an input attenuator.

[SENSe:]TSET:ATTenuator:AUTO

When enabled, the system will perform automatic autoranging of the test set input attenuator.

Command Syntax

```
[SENSe:]TSET:ATTenuator:AUTO <boolean>  
<boolean> ::= OFF|0|ON|1
```

Example

```
TSET:ATTenuator:AUTO OFF
```

Query Syntax

```
TSET:ATTenuator:AUTO?
```

Return Format

Integer 0=OFF 1=ON

Attribute Summary

Synchronization Required: none

Preset (*RST) State: ON

SCPI Compliance: Instrument Specific.

Additional Information

This command is significant only if an 70420A OPT 001 test set is used. The 70420A OPT 001 is the only test set available with an input attenuator.

If auto input attenuation is enabled, at the appropriate time in the measurement the measurement engine will instruct the test set to do an autorange immediate on the input attenuator. If auto is disabled, the value set by [SENSe:]TSET:ATTenuator will be used.

If both a 70420A OPT 001 and an 70427A are in the system, and both attenuators have been set to AUTO, then the test set attenuator is autoranged first, then the downconverter attenuator, then the IF gain. That is, always perform autoranges in order from the beginning of the signal path, paying attention to the auto/manual switches set by the user. The philosophy is to never override a user's value if manual is selected.

[SENSe:]TSET:DCBLock

Turn the DC high-pass filter ON or OFF.

Command Syntax

```
[SENSe:]TSET:DCBLock <boolean>
<boolean> ::= OFF|0|ON|1
```

Example

```
TSET:DCBLock ON
```

Query Syntax

```
TSET:DCBLock?
```

Return Format

Integer 0=OFF 1=ON

Attribute Summary

Synchronization Required: none

Preset (*RST) State: OFF

SCPI Compliance: Instrument Specific.

Additional Information

Check the DC block when performing AM noise measurements. The 70420A Phase Noise Test Set must be DC blocked when using its Noise Input. The test set will not tolerate more than +/- mV DC input without overloading the Low Noise Amplifier (LNA).

This filter is marked “AM BLOCK” on the block diagrams. It follows the phase detector LPF.

[SENSe:]TSET:LNAGain

Specify the nominal gain for the low noise amplifier in the test set.

Command Syntax

```
[SENSe:]TSET:LNAGain <number> [dB]  
<number> ::= a real number (NRf data)
```

Accepts numeric inputs, but snaps to nearest following values:

14, 28, 42, 56

Example

```
TSET:LNAGain 28
```

Query Syntax

```
TSET:LNAGain?
```

Return Format

Real

Attribute Summary

Synchronization Required: none

Preset (*RST) State: 42 dB

SCPI Compliance: Instrument Specific.

Additional Information

[SENSe:]TSET:LNAGain:METHOD FIXEd must be set for TSET:LNAGain to be in effect.

[SENSe:]TSET:LNAGain:METhod

Specify the method for applying LNA gain.

Command Syntax

[SENSe:]TSET:LNAGain:METhod AUTO|FIXed|PAUSE

Example

TSET:LNAGain:METhod FIXed

Query Syntax

TSET:LNAGain:METhod?

Return Format

String

Attribute Summary

Synchronization Required: none

Preset (*RST) State: AUTO

SCPI Compliance: Instrument Specific.

Additional Information

- AUTO = Auto Gain performs a one-time autoranging of the test set's LNA gain. The minimum gain selected is limited by [SENSe:]TSET:LNAGain:MINimum
- FIXed = use value set by [SENSe:]TSET:LNAGain
- PAUSE = Same as AUTO but generates a pause point after setting the gain.

[SENSe:]TSET:LNAGain:MINimum

Specify the LNA gain minimum limit.

Command Syntax

```
[SENSe:]TSET:LNAGain:MINimum <number> [dB]  
<number> ::= a real number (NRf data)  
limits: 14|28|42|56
```

Example

```
TSET:LNAGain:MINimum 28
```

Query Syntax

```
TSET:LNAGain:MINimum?
```

Return Format

Real

Attribute Summary

Synchronization Required: none

Preset (*RST) State: 14 dBm

SCPI Compliance: Instrument Specific.

Additional Information

Only applies if [SENSe:]TSET:LNAGain:METHOD = AUTO | PAUSE

[SENSe:]TSET:LPF

Specify the cutoff frequency for the input LPF of the Test Set.

Command Syntax

```
[SENSe:]TSET:LPF <value>  
<value> ::= a real number (NRf data)
```

Accepts numeric input but snaps to nearest of the following values:

- 20 kHz
- 200 kHz
- 2 MHz
- 20 MHz
- 100 MHz

Query Syntax

```
TSET:LPF?
```

Example

```
TSET:LPF 2E6
```

Return Format

Real

Attribute Summary

Synchronization Required: none

Preset (*RST) State:

SCPI Compliance: Instrument Specific.

Additional Information

Requires [SENSe:]TSET:LPF:AUTO OFF

The lowpass filters follow the RF phase detectors (LF & HF) in the signal path. The block diagram shows the first filter as a 140 MHz LPF, but the system can only measure up to a 100 MHz offset, so functionally, this filter is a 100 MHz LPF. From the user's standpoint and from the measurement engine's standpoint, this is a 100 MHz LPF.

[SENSe:]TSET:LPF:AUTO

Turn test set LPF automatic selection on or off.

Command Syntax

```
[SENSe:]TSET:LPF:AUTO <boolean>  
<boolean> ::= OFF|0|ON|1
```

Example

```
TSET:LPF:AUTO
```

Query Syntax

```
TSET:LPF:AUTO?
```

Return Format

Integer 0=OFF 1=ON

Attribute Summary

Synchronization Required: none

Preset (*RST) State: ON

SCPI Compliance: Instrument Specific.

Additional Information

When LPF automatic selection is ON, the system will select an appropriate cutoff frequency based on the measurement type and the detector input frequency. When LPF automatic selection is OFF, the system will use the cutoff specified by [SENSe:]TSET:LPF.

[SENSe:]TSET:PLL:ATTenuator

Specify the attenuation setting for the PLL input attenuator of the test set.

Command Syntax

```
[SENSe:]TSET:PLL:ATTenuator <enum> [dB]
<enum> ::= a real number (NRf data)
```

Accepts numeric input but snaps to nearest of the following values:

- 0.00 dB
- 6.28 dB
- 12.16 dB
- 18.12 dB
- 24.18 dB
- 30.48 dB
- 36.41 dB
- 42.43 dB

Query Syntax

TSET:PLL:ATTenuator?

Example

TSET:PLL:ATTenuator 12.16

Return Format

Real

Attribute Summary

Synchronization Required: none

Preset (*RST) State: 1

SCPI Compliance: Instrument Specific.

Additional Information

The PLL integrator attenuation can be switched in to provide for narrower voltage tune ranges.

The PLL input attenuator is the programmable attenuator located in front of the integrator on the PLL board. The associated test set command for this parameter is CTS:PLL:ATT3.

This attenuator changes the loop gain without affecting the drift tracking range (since it is ahead of the integrator). It has some significant effects on the ability to acquire lock under some conditions.

The direct affect from the user's point of view is on the loop bandwidth, but it is not necessarily a monotonic reduction of loop bandwidth (there are

SENSe Commands

[SENSe:]TSET:PLL:ATTenuator

some sawteeth in the curve), and it takes some trial and error to figure out the optimal setting.

[SENSe:]TSET:PLL:UNLock:IGNore

If ON, ignore PLL Unlocked errors and continued with the measurement.

Command Syntax

[SENSe:]TSET:PLL:UNLock:IGNore <boolean>

Query Syntax

TSET:UNLock:IGNore OFF

Example

TSET:UNL:IGN?

Return Format

Integer 0=OFF 1=ON

Attribute Summary

Synchronization Required: none
Preset (*RST) State:OFF
SCPI Compliance: Instrument Specific.

Additional Information

[SENSe:]TVCO

Specifies where the tune voltage output is connected to.

Command Syntax

```
[SENSe:]TVCO REFerence|CARRier|DCONverter|INTernal
```

Query Syntax

```
SENSe:TVCO?
```

Example

```
SENSe:TVCO DCONverter
```

Return Format

String

Attribute Summary

Synchronization Required: none

Preset (*RST) State: REFerence

SCPI Compliance: Instrument Specific.

Additional Information

- REFerence specifies that the tune voltage output is connected to the Modulation Input of the reference source.
- CARRier specifies that the tune voltage output is connected to the Modulation Input of the carrier source.
- DCONverter specifies that the tune voltage output is connected to the Voltage Control Input port of the downconverter
- INTernal specifies that the tune voltage is routed internally to the test

Refer to Chapter 1 for update information.

The SOURce subsystem is used to control source frequency and power parameters.

- SOURce:CALibration:FM[:DEViation] (value)
- SOURce:CALibration:FM:INTernal:FREQuency (value)
- SOURce:CALibration:FREQuency[:CW|FIXED] (value)
- SOURce:CALibration:POWer[:LEVel|AMPLitude] (value)
- SOURce:CARRier:FREQuency[:CW|FIXED] (value)
- SOURce:CARRier:POWer[:LEVel|AMPLitude] (value)
- SOURce:REFeRence:FREQuency:DIVisor
- SOURce:REFeRence:FREQuency:MULTIplier
- SOURce:REFeRence:POWer[:LEVel|AMPLitude] (value)
- SOURce:RESidual:FREQuency[:CW|FIXED] (value)
- SOURce:RESidual:FREQuency:CALCulate
- SOURce:RESidual:FREQuency:DETEctor:COUPled
- SOURce:RESidual:FREQuency:DIVisor
- SOURce:RESidual:FREQuency:MULTIplier
- SOURce:RESidual:POWer[:LEVel|AMPLitude] (value)

SOURce:CALibration:FM[:DEVIation]

Specify the calibration source's modulation deviation of a FM signal.

Command Syntax

```
SOURce:CALibration:FM[:DEVIation] <number> <frequency_suffix>  
<number> ::= a real number (NRf data)  
<frequency_suffix> ::= Hz|kHz|MHz|GHz
```

Example

```
SOURce:CALibration:FM 8E3
```

Query Syntax

```
SOURce:CALibration:FM?
```

Return Format

Real

Attribute Summary

Synchronization Required: none

Preset (*RST) State: 10 kHz

SCPI Compliance: Instrument Specific.

SOURce:CALibration:FM:INTernal: FREQUENCY

Specify the calibration source's FM frequency.

Command Syntax

```
SOURce:CALibration:FM:INTernal:FREQUENCY <number>  
<frequency_suffix>  
  
<number> ::= a real number (NRf data)  
<frequency_suffix> ::= Hz | kHz | MHz | GHz
```

Example

```
SOURce:CALibration:FM:INTernal:FREQUENCY?
```

Query Syntax

```
SOURce:CALibration:FM:INTernal:FREQUENCY 1.5E3
```

Return Format

Real

Attribute Summary

Synchronization Required: none

Preset (*RST) State: 10 kHz

SCPI Compliance: Instrument Specific.

SOURce:CALibration:FREQuency[:CW|FIXed]

Specify the calibration source frequency.

Command Syntax

```
SOURce:CALibration:FREQuency[:CW|FIXed] <number>  
[<frequency_suffix>]  
<number> ::= a real number (NRf data)  
<frequency_suffix> ::= Hz|kHz|MHz|GHz
```

Example

```
SOURce:CALibration:FREQuency 600e6
```

Query Syntax

```
SOURce:CALibration:FREQuency?
```

Return Format

Real

Attribute Summary

Synchronization Required: none

Preset (*RST) State: 500 MHz

SCPI Compliance: Instrument Specific.

SOURce:CALibration:POWer [:LEVel|AMPLitude]

Specify the calibration source power.

Command Syntax

```
SOURce:CALibration:POWer[:LEVel][:AMPLitude] <number> [dBm]  
<number> ::= a real number (NRf data)
```

Example

```
SOURce:CALibration:POWer 0
```

Query Syntax

```
SOURce:CALibration:POWer?
```

Return Format

Real

Attribute Summary

Synchronization Required: none

Preset (*RST) State: +10 dBm

SCPI Compliance: Instrument Specific.

SOURce:CARRier:FREQuency[:CW|FIXed]

Specify the carrier source frequency.

Command Syntax

```
SOURce:CARRier:FREQuency[:CW|FIXed] <number>
[<frequency_suffix>]
<number> ::= a real number (NRf data)
<frequency_suffix> ::= Hz|kHz|MHz|GHz
```

Example

```
SOURce:CARRier:FREQuency 10 GHz
```

Query Syntax

```
SOURce:CARRier:FREQuency?
```

Return Format

Real

Attribute Summary

Synchronization Required: none
Preset (*RST) State: 500 MHz
SCPI Compliance: Instrument Specific.

SOURce:CARRier:POWer [:LEVel|AMPLitude]

Specify the carrier source power.

Command Syntax

```
SOURce:CARRier:POWer[:LEVel][:AMPLitude] <number> [dBm]  
<number> ::= a real number (NRf data)
```

Example

```
SOURce:CARRier:POWer 10
```

Query Syntax

```
SOURce:CARRier:POWer?
```

Return Format

Real

Attribute Summary

Synchronization Required: none

Preset (*RST) State: +10 dBm

SCPI Compliance: Instrument Specific.

Additional Information

The carrier power is the input power to the test set or the downconverter. The power must meet the minimum requirements of the test set's Signal Input.

SOURce:REFeRence:FREQuency:DIVisor

Specify the reference source power.

Command Syntax

```
SOURce:REFeRence:FREQuency:DIVisor <number>  
<number> ::= a real number (NRf data)
```

Example

```
SOURce:REFeRence:FREQuency:DIVisor 3.0
```

Query Syntax

```
SOURce:REFeRence:FREQuency:DIVisor?
```

Return Format

Real

Attribute Summary

Synchronization Required: none
Preset (*RST) State: 1.0
SCPI Compliance: Instrument Specific.

Additional Information

The Reference Source frequency is set to:

```
SENSE:DETeCTOR:FREQuency *
```

```
(SOUeCe:REFeRence:FREQuency:MULTiplier / SOURce:REFeRence:  
FREQuency:DIVisor)
```

SOURce:REfERENCE:FREQuency:MULTIplier

Specify the reference source power.

Command Syntax

```
SOURce:REfERENCE:FREQuency:MULTIplier <number>  
<number> ::= a real number (Nrf data)
```

Example

```
SOURce:REfERENCE:FREQuency:MULTIplier 2.0
```

Query Syntax

```
SOURce:REfERENCE:FREQuency:MULTIplier?
```

Return Format

Real

Attribute Summary

Synchronization Required: none
Preset (*RST) State: 1.0
SCPI Compliance: Instrument Specific.

Additional Information

The Reference Source frequency is set to:

```
SENSE:DETECTOR:FREQuency *
```

```
(SOURce:REfERENCE:FREQuency:MULTIplier / SOURce:REfERENCE:  
FREQuency:DIVisor)
```


SOURce:REFerence:POWer [:LEVel|AMPLitude]

Specify the reference source power.

Command Syntax

```
SOURce:REFerence:POWer[:LEVel][:AMPLitude] <number> [dBm]  
<number> ::= a real number (NRf data)
```

Example

```
SOURce:REFerence:POWer 10
```

Query Syntax

```
SOURce:REFerence:POWer?
```

Return Format

Real

Attribute Summary

Synchronization Required: none
Preset (*RST) State: +16 dBm
SCPI Compliance: Instrument Specific.

Additional Information

The reference source power is the reference input power to the test set. The power must meet the minimum requirements of the test set's Reference Input.

SOURce:RESidual:FREQuency[:CW|FIXed]

Specify the residual source frequency.

Command Syntax

```
SOURce:RESidual:FREQuency[:CW|FIXed] <number>
[<frequency_suffix>]
<number> ::= a real number (NRf data)
<frequency_suffix> ::= Hz|kHz|MHz|GHz
```

Example

```
SOURce:RESidual:FREQuency:CW 1E6
```

Query Syntax

```
SOURce:RESidual:FREQuency?
```

Return Format

Real

Attribute Summary

Synchronization Required: none
Preset (*RST) State: 500 MHz
SCPI Compliance: Instrument Specific.

SOURce:RESidual:FREQuency:CALCulate

When enabled, the Residual source frequency is coupled to the Carrier frequency.

Command Syntax

```
SOURce:RESidual:FREQuency:CALCulate <boolean>  
<boolean> ::= OFF|0|ON|1
```

Example

```
SOURce:RESidual:FREQuency:CALCulate OFF
```

Query Syntax

```
SOURce:RESidual:FREQuency:CALCulate?
```

Return Format

Integer 0=OFF 1=ON

Attribute Summary

Synchronization Required: none
Preset (*RST) State: ON
SCPI Compliance: Instrument Specific.

Additional Information

When SOURce:RESidual:FREQuency:CALCulate is set to ON (the default) the Residual source frequency is automatically set based on the Carrier frequency and the Residual source Multiplier and Divisor. The following equation shows the relationship between the Carrier and Residual frequency, and the Multiplier and Divisor:

$$\text{Carrier} = \text{Residual} * (\text{Multiplier} / \text{Divisor})$$

Thus, for a given Carrier frequency, the Residual source is automatically set as follows:

$$\text{Residual} = \text{Carrier} / (\text{Multiplier} / \text{Divisor})$$

When SOURce:RESidual:FREQuency:CALCulate is set to OFF, the Residual source frequency may be set independently of the Carrier frequency.

Also See:

```
SOURce:RESidual:FREQuency  
SOURce:RESidual:FREQuency:DIVisor  
SOURce:RESidual:FREQuency:MULTiplier
```

SOURce:RESidual:FREQuency:DETEctor: COUPled

When enabled, the Residual source frequency is coupled to the Carrier frequency.

Command Syntax

```
SOURce:RESidual:FREQuency:DETEctor:COUPled <boolean>
<boolean> ::= OFF|0|ON|1
```

Example

```
SOURce:RESidual:FREQuency:DETEctor:COUPled OFF
```

Query Syntax

```
SOURce:RESidual:FREQuency:DETEctor:COUPled?
```

Return Format

Intriguer 0=OFF 1=ON

Attribute Summary

Synchronization Required: none
Preset (*RST) State: ON
SCPI Compliance: Instrument Specific.

Additional Information

When SOURce:RESidual:FREQuency:DETEctor:COUPled is set to ON (the default) the Residual source frequency is automatically set based on the Carrier frequency and the Residual source Multiplier and Divisor. The following equation shows the relationship between the Carrier and Residual frequency, and the Multiplier and Divisor:

$$\text{Carrier} = \text{Residual} * (\text{Multiplier} / \text{Divisor})$$

Thus, for a given Carrier frequency, the Residual source is automatically set as follows:

$$\text{Residual} = \text{Carrier} / (\text{Multiplier} / \text{Divisor})$$

When SOURce:RESidual:FREQuency:DETEctor:COUPled is set to OFF, the Residual source frequency may be set independently of the Carrier frequency.

Also See:

SOURce:RESidual:FREQuency
SOURce:RESidual:FREQuency:DIVisor
SOURce:RESidual:FREQuency:MULTIplier

SOURce:REference:FREQuency:DIVisor

Specify the residual source divisor.

Command Syntax

```
SOURce:RESidual:FREQuency:DIVisor <number>  
<number> ::= a real number (Nrf data)
```

Example

```
SOURce:RESidual:FREQuency:DIVisor 3
```

Query Syntax

```
SOURce:RESidual:FREQuency:DIVisor?
```

Return Format

Real

Attribute Summary

Synchronization Required: none
Preset (*RST) State: 1
SCPI Compliance: Instrument Specific.

Additional Information

The value of this parameter is only applied when `SOURce:RESidual:FREQuency:DETECTOR:COUPled` is set to ON (the default). When this is true, the following equation shows the relationship between the Carrier and Residual frequency and Multiplier and Divisor:

$$\text{Carrier} = \text{Residual} * (\text{Multiplier} / \text{Divisor})$$

Thus, for a given Carrier frequency, the Residual source is automatically set as follows:

$$\text{Residual} = \text{Carrier} / (\text{Multiplier} / \text{Divisor})$$

SOURce:RESidual:FREQuency:MULTiplier

Specify the Residual source multiplier.

Command Syntax

```
SOURce:RESidual:FREQuency:MULTiplier <number>  
<number> ::= a real number (Nrf data)
```

Example

```
SOURce:RESidual:FREQuency:MULTiplier 2
```

Query Syntax

```
SOURce:RESidual:FREQuency:MULTiplier?
```

Return Format

Real

Attribute Summary

Synchronization Required: none
Preset (*RST) State: +16 dBm
SCPI Compliance: Instrument Specific.

Additional Information

The value of this parameter is only applied when SOURce:RESidual:FREQuency:DETECTOR:COUPled is set to ON (the default). When this is true, the following equation shows the relationship between the Carrier and Residual frequency and the Multiplier and Divisor:

$$\text{Carrier} = \text{Residual} * (\text{Multiplier} / \text{Divisor})$$

Thus, for a given Carrier frequency, the Residual source is automatically set as follows:

$$\text{Residual} = \text{Carrier} / (\text{Multiplier} / \text{Divisor})$$

SOURce:RESidual:POWer [:LEVel|AMPLitude]

Specify the residual source power.

Command Syntax

```
SOURce:RESidual:POWer[:LEVel][:AMPLitude] <number> [dBm]  
<number> ::= a real number (NRf data)
```

Example

```
SOURce:RESidual:POWer 0
```

Query Syntax

```
SOURce:RESidual:POWer?
```

Return Format

Integer 0=OFF 1=ON

Attribute Summary

Synchronization Required: none

Preset (*RST) State: +10 dBm

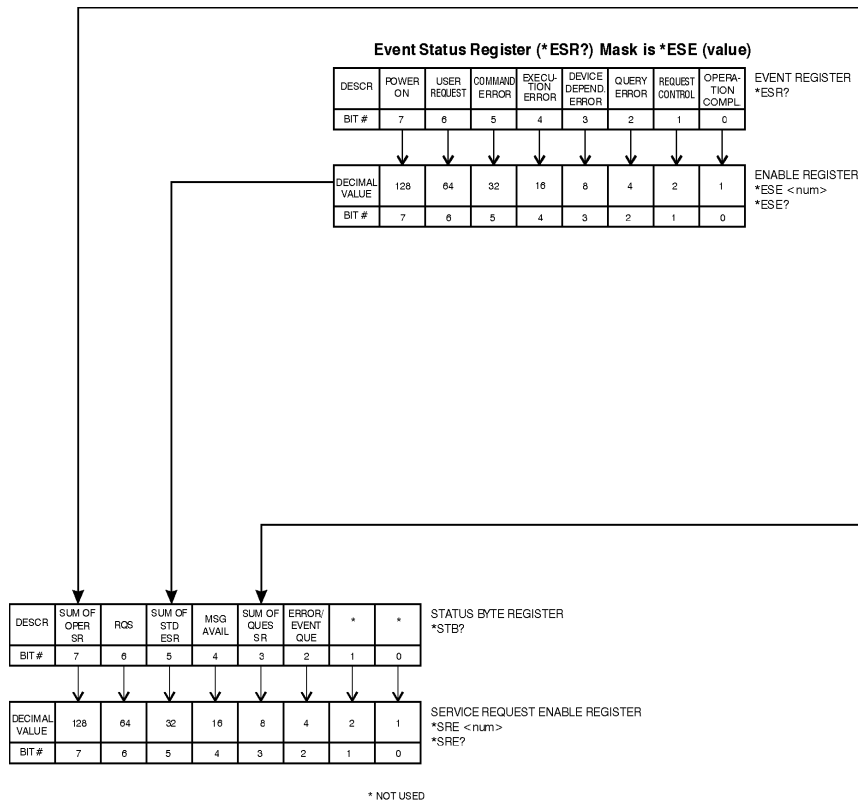
SCPI Compliance: Instrument Specific.

Refer to Chapter 1 for update information.

- STATUS:ADVise:ENABLE (0 | OFF | 1 | ON)
- STATUS:OPERation:CONDition?
- STATUS:OPERation:ENABLE (value)
- STATUS:OPERation:EVENT?
- STATUS:OPERation:NTRansition (value)
- STATUS:OPERation:PTRansition (value)
- STATUS:PRESet
- STATUS:QUEStionable:CONDition?
- STATUS:QUEStionable:ENABLE (value)
- STATUS:QUEStionable:EVENT?
- STATUS:QUEStionable:NTRansition (value)
- STATUS:QUEStionable:PTRansition (value)

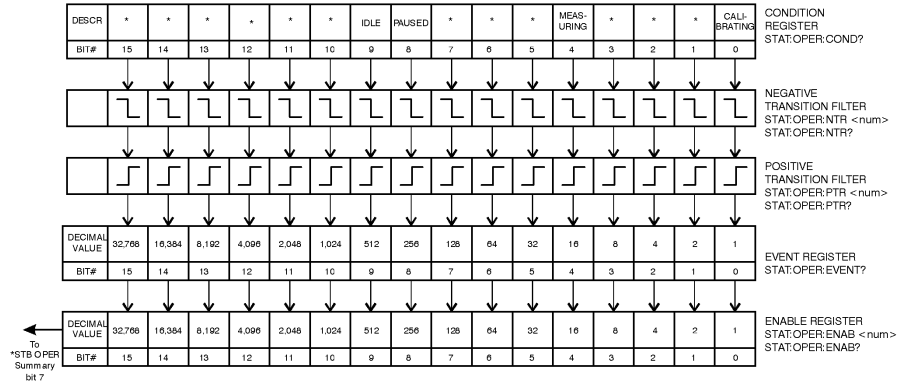
Also see *ESR? *ESE *STB? *SRE commands.

STATUS Commands

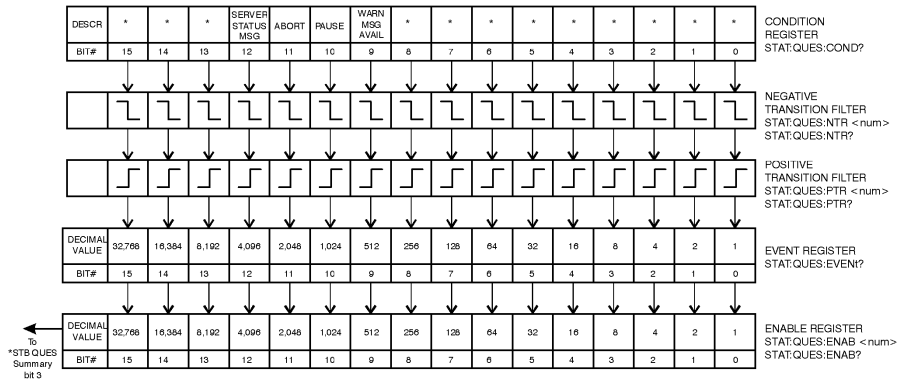


statreg1.cdr

Status Operation Register (STATus:OPERation:CONDition?)



Status Questionable Register (STATus:QUESTionable:CONDition?)



NOTE:

STAT:PRES
This command presets the following
enable and transition registers:
OPER, AND QUES.

* Not used

STATus:ADVise:ENABle

When set to ON, server status messages (non-error) are also placed in the scpi error que and bit 12 of the Questionable Status Register is activated to report when messages are available.

Command Syntax

STATus:ADVise:ENABle OFF|0|ON|1

Example

STATus:ADVise:ENABle ON

Query Syntax

STATus:ADVise:ENABle?

Return Format

Integer

Attribute Summary

Synchronization Required: none

Preset (*RST) State: OFF

SCPI Compliance: Standard

STATus:OPERation:CONDition?

Provides information about the state of the measurement system. Condition registers continually monitor the state of the system. There is no buffering for this register, its status is updated in real time.

Command Syntax

Query Only

Query Syntax

STATus:Operation:CONDition?

Return Format

Integer

Attribute Summary

Synchronization Required: none

Preset (*RST) State: NA

SCPI Compliance: Standard

Additional Information

- bit 0 = 1 = Calibrating
- bit 4 = 16 = Measuring
- bit 8 = 256 = Paused
- bit 9 = 512 = Idle
- All other bits are not used.

Reading the Condition register is nondestructive.

STATus:OPERation:ENABLE

Specifies which bits in the STATus:OPERation:EVENT register set the Operation Register Summary Bit (bit 7) in the Status Byte (*STB?).

Command Syntax

STATus:OPERation:ENABLE <num>

Example

STATus:OPERation:ENABLE

Query Syntax

STATus:OPERation:ENABLE?

Return Format

Integer

Attribute Summary

Synchronization Required: none

Preset (*RST) State: 0

SCPI Compliance: Standard

Additional Information

STATus:PREset clears this enable register.

STATus:OPERation:EVENT?

EVENT registers latch events from the CONDition register as specified by the PTRansition and NTRansition registers.

Command Syntax

Query Only

Query Syntax

STATus:OPERation:EVENT?

Return Format

Integer

Attribute Summary

Synchronization Required: none

Preset (*RST) State: NA

SCPI Compliance: Standard

Additional Information

- STATus: QUEStionable:EVENT? Clears the event register (destructive read).
- *CLS clears the event register.

STATus:OPERation:NTRansition

Transition filters specify what type of bit state transition in the :CONDition register will set corresponding bits in the :EVENT register. Transition filter bits may be set for positive transitions :PTRansition (0 to 1), negative transitions :NTRansition (1 to 0) or both.

Command Syntax

STATus:OPERation:NTRansition

Example

STATus:OPERation:NTRansition

Query Syntax

STATus:OPERation:NTRansition?

Return Format

Integer

Attribute Summary

Synchronization Required: none

Preset (*RST) State: NA

SCPI Compliance: Standard

Additional Information

- STATus:PREset sets all PTRansition bits to 1 and NTRansition bits to 0
- *CLS does not affect transition registers

STATus:OPERation:PTRansition

Transition filters specify what type of bit state transition in the :CONDition register will set corresponding bits in the :EVENT register. Transition filter bits may be set for positive transitions :PTRansition (0 to 1), negative transitions :NTRansition (1 to 0) or both.

Command Syntax

STATus:OPERation:PTRansition

Example

STATus:OPERation:PTRansition

Query Syntax

STATus:OPERation:PTRansition?

Return Format

Integer

Attribute Summary

Synchronization Required: none

Preset (*RST) State: NA

SCPI Compliance: Standard

Additional Information

- STATus:PREset sets all PTRansition bits to 1 and NTRansition bits to 0
- *CLS does not affect transition registers

STATus:PRESet

Presets all SCPI Status :ENABLE, :PTRANSition and :NTRansition registers.

Command Syntax

STATus:PRESet

Query Syntax

Command Only

Example

STATus:PRESet

Attribute Summary

Synchronization Required: none

Preset (*RST) State: NA

SCPI Compliance: Standard

Additional Information

- :ENABLE registers bits are set to 0.
- :PTRANSition register bits are set to 1.
- :NTRANSition register bits are set to 0.
- Other registers are not affected. See *CLS for clearing other registers.

STATus:QUEStionable:CONDition?

Provides information about measurement warnings or errors. Condition registers continually monitor the state of the system. There is no buffering for this register, its status is updated in real time.

Command Syntax

Query Only

Query Syntax

STATus:QUEStionable:CONDition?

Return Format

Integer

Attribute Summary

Synchronization Required: none

Preset (*RST) State: NA

SCPI Compliance: Standard

Additional Information

- bit 9 = 512 = Warning message available.
- bit 10 = 1024 = Pause " "
- bit 11 = 2048 = Abort " "
- bit 12 = 4096 = Server Status message available.
- All other bits are not used.

Reading the Condition register is nondestructive.

STATus:QUEStionable:ENABLE

Specifies which bits in the STATus: QUEStionable:EVENT register set the Questionable Register Summary Bit (bit 3) in the Status Byte (*STB?).

Command Syntax

STATus:QUEStionable:ENABLE

Example

STATus:QUEStionable:ENABLE 3584

Query Syntax

STATus:QUEStionable:ENABLE?

Return Format

Integer

Attribute Summary

Synchronization Required: none

Preset (*RST) State: 0

SCPI Compliance: Standard

Additional Information

STATus:PREset clears this enable register.

STATus:QUEStionable:[EVENT]?

EVENT registers latch events from the CONDition register as specified by the PTRansition and NTRansition registers.

Command Syntax

Query Only

Query Syntax

STATus:QUEStionable:[EVENT]?

Return Format

Integer

Attribute Summary

Synchronization Required: none

Preset (*RST) State: NA

SCPI Compliance: Standard

Additional Information

- STATus: QUEStionable:EVENT? Clears the event register (destructive read).
- *CLS clears the event register.

STATus:QUEStionable:NTRansition

Transition filters specify what type of bit state transition in the :CONDition register will set corresponding bits in the :EVENT register. Transition filter bits may be set for positive transitions :PTRansition (0 to 1), negative transitions :NTRansition (1 to 0) or both.

Command Syntax

STATus:QUEStionable:NTRansition

Example

STATus:QUEStionable:NTRansition 2560

Query Syntax

STATus:QUEStionable:NTRansition?

Return Format

Integer

Attribute Summary

Synchronization Required: none

Preset (*RST) State: NA

SCPI Compliance: Standard

Additional Information

- STATus:PREset sets all PTRansition bits to 1 and NTRansition bits to 0
- *CLS does not affect transition registers

STATus:QUEStionable:PTRansition

Transition filters specify what type of bit state transition in the :CONDition register will set corresponding bits in the :EVENT register. Transition filter bits may be set for positive transitions :PTRansition (0 to 1), negative transitions :NTRansition (1 to 0) or both.

Command Syntax

STATus:QUEStionable:PTRansition

Example

STATus:QUEStionable:PTRansition 2560

Query Syntax

STATus:QUEStionable:PTRansition?

Return Format

Integer

Attribute Summary

Synchronization Required: none

Preset (*RST) State: NA

SCPI Compliance: Standard

Additional Information

- STATus:PREset sets all PTRansition bits to 1 and NTRansition bits to 0
- *CLS does not affect transition registers

Refer to Chapter 1 for update information.

Use the SYSTem commands to select system assets and perform miscellaneous system functions.

The commands in this subsystem have the following command hierarchy:

- SYSTem:ASSet:CALSource ("None" | "asset_name")
- SYSTem:ASSet:CARRier ("None" | "asset_name")
- SYSTem:ASSet:COUNter ("None" | "asset_name")
- SYSTem:ASSet:DCONverter ("None" | "asset_name")
- SYSTem:ASSet:DCONverter:USAGe (NONE | MANual | SYSTem)
- SYSTem:ASSet:FFT ("None" | "asset_name")
- SYSTem:ASSet:FFT:CUSTom:[SEGTaBle]
- SYSTem:ASSet:PSHifter ("None" | "asset_name")
- SYSTem:ASSet:REFerence ("None" | "asset_name")
- SYSTem:ASSet:RESidual ("None" | "asset_name")
- SYSTem:ASSet:SLAVe:SOURce ("None" | "asset_name")
- SYSTem:ASSet:SLAVe:SOURce:PORT (LOINput | AUXinput)
- SYSTem:ASSet:SWANalyzer (value)
- SYSTem:ASSet:SWANalyzer:CUSTom:[SEGTaBle]
- SYSTem:ASSet:TBASe ("None" | "asset_name")
- SYSTem:ASSet:TSET ("None" | "asset_name")
- SYSTem:ERRor?
- SYSTem:GUI:LLOCkout[:STATe]
- SYSTem:GUI:REMote[:STATe] (0 | OFF | 1 | ON)
- SYSTem:HELP:HEADers?
- SYSTem:PATH:CARRier (TSET | DCONverter)
- SYSTem:PATH:DCBBanalyzer (TSLF | TSHF | TSRF)
- SYSTem:PATH:FFTanalyzer (TSLF | TSHF | TSRF | DCRF)
- SYSTem:PATH:SWANalyzer (TSLF | TSHF | TSRF | DCRF)
- SYSTem:PATH:TVCO (FPANel | RPANel | INTernal)
- SYSTem:TSTart?
- SYSTem:VERSion?

SYSTem:ASSet:CALSource

Specify the calibration source.

Command Syntax

SYSTem:ASSet:CALSource <“None”|”asset_name”>

Asset_name = quoted string as defined in the Asset Manager

Example

SYSTem:ASSet: CALSource”HPXXXXX “

SYSTem:ASSet: CALSource “NONE”

Query Syntax

SYSTem:ASSet: CALSource?

Return Format

String

Attribute Summary

Synchronization Required: none

Preset (*RST) State: NULL

SCPI Compliance: Instrument Specific.

Additional Information

Invalid asset names will be accepted but will cause a server error message when a measurement is initialized.

SYSTem:ASSet:CARRier

Specify the carrier source.

Command Syntax

```
SYSTem:ASSet:CARRier <“None”|”asset_name”>
```

Asset_name = quoted string as defined in the Asset Manager

Example

```
SYSTem:ASSet:CARRier “HPXXXX “
```

```
SYSTem:ASSet:CARRier “NONE”
```

Query Syntax

```
SYSTem:ASSet:CARRier?
```

Return Format

Attribute Summary

Synchronization Required: none

Preset (*RST) State: NULL

SCPI Compliance: Instrument Specific.

Additional Information

Invalid asset names will be accepted but will cause a server error message when a measurement is initialized.

SYSTem:ASSet:COUNter

Specify the counter.

Command Syntax

```
SYSTem:ASSet:CARRier <"None" | "asset_name">
```

Asset_name = quoted string as defined in the Asset Manager

Example

```
SYSTem:ASSet:COUNter AUTO,"HPXXXXXA-001,GPIB,11,SN12345"
```

```
SYSTem:ASSet:COUNter NONE
```

Query Syntax

```
SYSTem:ASSet:COUNter?
```

Return Format

String

Attribute Summary

Synchronization Required: none

Preset (*RST) State: VFRMS (NULL)

SCPI Compliance: Instrument Specific.

Additional Information

Invalid asset names will be accepted but will cause a server error message when a measurement is initialized.

SYSTem:ASSet:DCONverter

Specify the downconverter.

Command Syntax

SYSTem:ASSet:COUNter <"None" | "asset_name">

Asset_name = quoted string as defined in the Asset Manager

Example

```
SYSTem:ASSet:DCONverter  
AUTO,"HPXXXXXA-001,GPIB,11,SN12345"
```

Query Syntax

SYSTem:ASSet:DCONverter?

Return Format

String

Attribute Summary

Synchronization Required: none

Preset (*RST) State: NULL

SCPI Compliance: Instrument Specific.

Additional Information

Invalid asset names will be accepted but will cause a server error message when a measurement is initialized.

SYSTem:ASSet:DCONverter:USAGe

Specify if and how the down converter is to be used in the current measurement.

Command Syntax

SYSTem:ASSet:DCONverter:USAGe NONE|MANual|SYSTem

Example

SYSTem:ASSet:DCONverter:USAGe SYSTem

Query Syntax

SYSTem:ASSet:DCONverter:USAGe?

Return Format

String

Attribute Summary

Synchronization Required: none

Preset (*RST) State: NONE

SCPI Compliance: Instrument Specific.

Additional Information

If NONE is selected, the down converter is not used in the measurement. If MANual is selected, the downconverter is used in the measurement with its current settings; no commands to set the input frequency, IF gain, etc. are sent. If SYSTem is selected, the down converter is set up via the settings on the Downconverter parameters page and used in the measurement.

SYSTem:ASSet:FFT

Specify the FFT analyzer.

Command Syntax

```
SYSTem:ASSet:FFT <"None" | "asset_name">
```

Asset_name = quoted string as defined in the Asset Manager

Example

```
SYSTem:ASSet:FFT AUTO,"HP XXXXX"
```

```
SYSTem:ASSet:FFT NONE
```

Query Syntax

```
SYSTem:ASSet:FFT?
```

Return Format

String

Attribute Summary

Synchronization Required: none

Preset (*RST) State: NULL

SCPI Compliance: Instrument Specific.

Additional Information

Invalid asset names will be accepted but will cause a server error message when a measurement is initialized.

SYSTem:ASSet:FFT:CUSTom[:SEGTaBle]

Specify the custom segment table for the FFT analyzer.

Command Syntax

SYSTem:ASSet:FFT:CUSTom[:SEGTaBle] <"path\file_name.fst">
<path\file_name.fst> ::= quoted string (full or relative path and filename)<file>

Example

SYSTem:ASSet:FFT:CUSTom "Segtbl_1.fst"

Attribute Summary

Synchronization Required: none

Preset (*RST) State: NULL

SCPI Compliance: Instrument Specific.

Additional Information

[[:SENSE]:RANGE:FFT:SEGTaBle[:MEASUREMENT][:QUALITY] CUSTom must be selected to use the currently loaded custom segment table.

Custom segment tables are edited and saved using the Measurement Client GUI Define menu. The default file extension is (.fst) for Fft Segment Table.

SYSTem:ASSet:PSHifter

Specify the phase shifter.

Command Syntax

SYSTem:ASSet:PSHifter <"None" | "asset_name">

Asset_name = quoted string as defined in the Asset Manager

Query Syntax

SYSTem:ASSet:PSHifter?

Example

SYSTem:ASSet:PSHifter "HPXXXXX"

SYSTem:ASSet:PSHifter "NONE"

Return Format

String

Attribute Summary

Synchronization Required: none

Preset (*RST) State: NULL

SCPI Compliance: Instrument Specific.

Additional Information

Invalid asset names will be accepted but will cause a server error message when a measurement is initialized.

SYSTem:ASSet:REFeRence

Specify the reference source.

Command Syntax

SYSTem:ASSet:REFeRence <"None" | "asset_name">

Asset_name = quoted string as defined in the Asset Manager

Example

SYSTem:ASSet:REFeRence AUTO,"HPXXXXX"

SYSTem:ASSet:REFeRence "NONE"

Query Syntax

SYSTem:ASSet:REFeRence?

Return Format

String

Attribute Summary

Synchronization Required: none

Preset (*RST) State: NULL

SCPI Compliance: Instrument Specific.

Additional Information

Invalid asset names will be accepted but will cause a server error message when a measurement is initialized.

SYSTem:ASSet:RESidual

Specify the residual source.

Command Syntax

SYSTem:ASSet:RESidual <“None”|“asset_name”>

Asset_name = quoted string as defined in the Asset Manager

Query Syntax

SYSTem:ASSet: RESidual?

Example

SYSTem:ASSet: RESidual”HP83650A “

SYSTem:ASSet: RESidual “NONE”

Return Format

Attribute Summary

Synchronization Required: none

Preset (*RST) State: NULL

SCPI Compliance: Instrument Specific.

Additional Information

Invalid asset names will be accepted, but will cause a server error message when a measurement is initialized.

SYSTem:ASSet:SLAVe:SOURce

Specify the slave source associated with the 50kHz-1600MHz RF Output of the carrier noise test set (70427A H16/H17). This slave source is connected to the Auxiliary Input or LO input of the carrier noise test set (see SYSTem:ASSet:SLAVe:SOURce:PORT).

Command Syntax

SYSTem:ASSet:SLAVe:SOURce <"None" | "asset_name">
 asset_name = quoted string as defined in the Asset Manager

Example

SYSTem:ASSet:SLAVe:SOURce "HPXXXXX"
 SYSTem:ASSet:SLAVe:SOURce "NONE"

Query Syntax

SYSTem:ASSet:SLAVe:SOURce?

Return Format

String

Attribute Summary

Synchronization Required: none
 Preset (*RST) State: NULL
 SCPI Compliance: Instrument Specific.

Additional Information

Invalid asset names will be accepted but will cause a server error message when a measurement is initialized.

The carrier noise test set can be used as a source, with the source output port being the 50kHz-1600MHz RF Output. However, another source has to provide a signal to the carrier noise test set's divider chain.

SYSTem:ASSet:SLAVe:SOURce:PORT

Specify the input port on the carrier noise test set to which the slave source is connected.

Command Syntax

`SYSTem:ASSet:SLAVe:SOURce:PORT LOINput | AUXinput`

Example

`SYSTem:ASSet:SLAVe:SOURce:PORT AUXinput`

Query Syntax

`SYSTem:ASSet:SLAVe:SOURce:PORT?`

Return Format

String

Attribute Summary

Synchronization Required: none

Preset (*RST) State: LOINput

SCPI Compliance: Instrument Specific.

Additional Information

- LOINput specifies the LO IN 3-6.6 GHz port
- AUXinput specifies the AUX INPUT 640-1280 MHz port.

SYSTem:ASSet:SWANalyzer

Specify the swept analyzer.

Command Syntax

SYSTem:ASSet:SWANalyzer <"None" | "asset_name">

asset_name = quoted string as defined in the Asset Manager

Example

SYSTem:ASSet:SWANalyzer AUTO,"HPXXXXX"

SYSTem:ASSet:SWANalyzer NONE

Query Syntax

SYSTem:ASSet:SWANalyzer?

Return Format

String

Attribute Summary

Synchronization Required: none

Preset (*RST) State: NULL

SCPI Compliance: Instrument Specific.

Additional Information

Invalid asset names will be accepted but will cause a server error message when a measurement is initialized.

SYSTEM:ASSET:SWANalyzer:CUSTOM [:SEGTable]

Specify the custom segment table to be used by the swept analyzer.

Command Syntax

SYSTEM:ASSET:SWANalyzer:CUSTOM[:SEGTable] <“path\filename.sst”>

<“path\filename.sst”> ::= string (full or relative path and filename)

Example

SYSTEM:ASSET:SWANalyzer:CUSTOM “Segtbl_2.sst”

Attribute Summary

Synchronization Required: none

Preset (*RST) State: NULL

SCPI Compliance: Instrument Specific.

Additional Information

[:SENSe]:RANGe:SWEPt:SEGTable[:MEASurement][:QUALity]

CUSTOM must be selected to use the currently loaded custom segment table.

Custom segment tables are edited and saved using the Measurement Client GUI Define menu. The default file extension is (.sst) for Swept Segment Table.

SYSTem:ASSet:TBASe

Specify the time base.

Command Syntax

SYSTem:ASSet:TBASe <"None" | "asset_name">

asset_name = quoted string as defined in the Asset Manager

Example

SYSTem:ASSet:TBASe AUTO "HPXXXXX"

SYSTem:ASSet:TBASe NONE

Query Syntax

SYSTem:ASSet:TBASe?

Return Format

String

Attribute Summary

Synchronization Required: none

Preset (*RST) State: NULL

SCPI Compliance: Instrument Specific.

Additional Information

Invalid asset names will be accepted but will cause a server error message when a measurement is initialized.

SYSTem:ASSet:TSET

Specify the testset.

Command Syntax

SYSTem:ASSet:TSET <“None”|”asset_name”>

Asset_name = quoted string as defined in the Asset Manager

Example

SYSTem:ASSet:TSET “HP70420A “

SYSTem:ASSet:TSET “NONE”

Query Syntax

SYSTem:ASSet:TSET?

Return Format

String

Attribute Summary

Synchronization Required: none

Preset (*RST) State: NULL

SCPI Compliance: Instrument Specific.

Additional Information

Invalid asset names will be accepted, but will cause a server error message when a measurement is initialized.

SYSTem:ERRor?

Queries an error from the error queue.

Command Syntax

Query Only

Query Syntax

SYSTem:ERRor?

Example

Return Format

<value>, <string>

<value> is an integer from -32768 to +32767.

<string> the text of the error message.

The following is an example of one response:

```
-113,"Undefined header"
```

Attribute Summary

Synchronization Required: none

Preset (*RST) State: NA

SCPI Compliance: Standard

Additional Information

The response for :ERRor is an ASCII string with two arguments separated by a comma. The first is an integer, ranging from -32768 to +32767, and the second is quoted string which string which describes the error in clear text.

For error information, refer to Table 19-1 on page 19-20.

Table 19-1 SCPI Error Messages

Error Number Range	Description - E5500 Specific Errors
100 - 199	Abort - Measurement aborted. Body of message indicates the reason.
200 - 299	Pause Point - Measurement is paused waiting for user interaction. Body of message indicates reason.
300 - 499	Status - Incidental measurement progress. STATus: ADVise:ENABle must be on to receive these messages.
500 - 599	Warning - Measurement will proceed. Body of message indicates reason.
Error Number	Description - General SCPI Errors
-100	Command error (unknown command)
-101	Invalid character
-102	Syntax error
-103	invalid separator
-104	Data type error
-105	GET not allowed
-108	Parameter not allowed
-109	Missing parameter
-112	Program mnemonic too long
-113	Undefined header
-121	Invalid character in number
-123	Numeric overflow
-124	Too many digits
-128	Numeric data not allowed
-130	Suffix error
-131	invalid suffix
-138	Suffix not allowed
-140	Character data error
-141	Invalid character data
-144	Character data too long
-148	Character data not allowed
-150	String data error

Table 19-1 *SCPI Error Messages*

Error Number Range	Description - E5500 Specific Errors
-151	Invalid string data
-158	String data not allowed
-160	Block data error
-161	invalid block data
-168	Block data not allowed
-170	Expression error
-171	Invalid Expression
-178	Expression data not allowed
-200	Execution error
-211	Trigger ignored
-221	Settings conflict
-222	Data out of range
-223	Too much data
-310	System error
-350	Too many errors
-400	Query error
-410	Query INTERRUPTED
-420	Query UNTERMINATED
-430	Query DEADLOCKED
-440	Query UNTERMINATED after indefinite response

SYSTem:GUI:LLOCkout[:STATe]

Disable Measurement Development Client GUI Local menu function.

Command Syntax

SYSTem:GUI:LLOCkout

Example

SYSTem:GUI:LLOCkout

Attribute Summary

Synchronization Required: none

Preset (*RST) State: unaffected

SCPI Compliance: Instrument Specific.

Additional Information

Invalid asset names will be accepted but will cause a server error message when a measurement is initialized.

SYSTem:GUI:REMOte[:STATe]

If ON put Measurement Development Client GUI in Remote mode. If OFF MDC GUI is in Local mode.

Command Syntax

```
SYSTem:GUI:REMOte[:STATe] <boolean>  
<boolean> ::= OFF|0|ON|1
```

Example

```
SYSTem:GUI:REMOte ON
```

Query Syntax

```
SYSTem:GUI:REMOte?
```

Return Format

Integer 0=OFF 1=ON

Attribute Summary

Synchronization Required: none

Preset (*RST) State: unaffected

SCPI Compliance: Instrument Specific.

Additional Information

The GUI must already be in remote or this command has no effect. Returning the GUI to Local (SYSTem:GUI:REMOte OFF) cancels Local Lockout.

SYSTem:HELP:HEADers?

Queries a listing of all the remote programming commands.

Command Syntax

Query Only

Query Syntax

SYSTem:HELP:HEADers?

Return Format

ASCII block

Attribute Summary

Synchronization Required: none

Preset (*RST) State: NA

SCPI Compliance: Standard

Additional Information

The response for :HEADers is the entire SCPI tree, one ASCII line at a time. Each line contains a complete SCPI statement and it's query ability.

SYSTem:PATH:CARRier

Specify the signal input port that the carrier signal is connected to.

Command Syntax

`SYSTem:PATH:CARRier TSET|DCONverter`

Example

`SYSTem:PATH:CARRier TSET`

Query Syntax

`SYSTem:PATH:CARRier?`

Return Format

String

Attribute Summary

Synchronization Required: none

Preset (*RST) State: TSET

SCPI Compliance: Instrument Specific.

SYSTEM:PATH:DCBBanalyzer

Specify the analyzer output port that the downconverter baseband analyzer input port is connected to.

Command Syntax

`SYSTEM:PATH:DCBBanalyzer TSLF | TSHF | TSRF`

Example

`SYSTEM:PATH:DCBBanalyzer TSLF`

Query Syntax

`SYSTEM:PATH:DCBBanalyzer?`

Return Format

String

Attribute Summary

Synchronization Required: none

Preset (*RST) State: TSRF

SCPI Compliance: Instrument Specific.

Additional Information

- TSLF - Test Set Low Frequency analyzer output port (Analyzer <100kHz)
- TSHF - Test Set High Frequency analyzer output port (Analyzer <100MHz)
- TSRF - Test Set RF Analyzer output port

SYSTem:PATH:FFTAnalyzer

Specify the analyzer output port that the FFT analyzer is connected to.

Command Syntax

```
SYSTem:PATH:FFTAnalyzer TSLF|TSHF|TSRF|DCRF
```

Example

```
SYSTem:PATH:FFTAnalyzer TSLF
```

Query Syntax

```
SYSTem:PATH:FFTAnalyzer?
```

Return Format

String

Attribute Summary

Synchronization Required: none

Preset (*RST) State: TSHF

SCPI Compliance: Instrument Specific.

Additional Information

- TSLF - Test Set Low Frequency analyzer output port (Analyzer <100kHz)
- TSHF - Test Set High Frequency analyzer output port (Analyzer <100MHz)
- TSRF - Test Set RF Analyzer output port
- DCRF - Downconverter RF Analyzer output port

SYSTEM:PATH:SWANalyzer

Specify the analyzer output port that the Swept (RF) Analyzer is connected to.

Command Syntax

SYSTEM:PATH:FFTAnalyzer TSLF|TSHF|TSRF|DCRF

Example

SYSTEM:PATH:FFTAnalyzer TSLF

Query Syntax

SYSTEM:PATH:FFTAnalyzer?

Return Format

String

Attribute Summary

Synchronization Required: none

Preset (*RST) State: TSRF

SCPI Compliance: Instrument Specific.

Additional Information

- TSLF - Test Set Low Frequency analyzer output port (Analyzer <100kHz)
- TSHF - Test Set High Frequency analyzer output port (Analyzer <100MHz)
- TSRF - Test Set RF Analyzer output port
- DCRF - Downconverter RF Analyzer output port

SYSTem:PATH:TVCO

Specifies which port is selected to supply the tune voltage output.

Command Syntax

`SYSTem:PATH:TVCO FPANel | RPANel | INTernal`

Example

`SYSTem:PATH:TVCO FPANel`

Query Syntax

`SYSTem:PATH:TVCO?`

Return Format

String

Attribute Summary

Synchronization Required: none

Preset (*RST) State: FPANel

SCPI Compliance: Instrument Specific.

SYSTem:TStart?

Query the start time of the measurement.

Command Syntax

Query Only

Query Syntax

SYSTem:TStart?

Return Format

String

Attribute Summary

Synchronization Required: none

Preset (*RST) State: NA

SCPI Compliance: Instrument Specific.

SYSTem:VERSion?

Queries the version of SCPI the E5500 complies with.

Command Syntax

Query Only

Query Syntax

SYSTem:VERSion?

Return Format

String

Attribute Summary

Synchronization Required: none

Preset (*RST) State: "1995.0"

SCPI Compliance: Standard

Additional Information

The response for :VERSion is an ASCII string numeric value in <N2> format. The value returned represents the SCPI compatibility and is in the form YYYY.V. Where YYYY is the year and the V is the revision with that year.

Index

NOTE: For a full list of SCPI commands please see the [Table of Contents](#).

A

Abort Commands, 8-1
application environment setup, 5-2

C

CALCulate Commands, 9-1
CALibrate Commands, 10-1
commands
 Abort, 8-1
 CALCulate, 9-1
 CALibrate, 10-1
 DISPlay, 11-1
 FORMat, 12-1
 INITiate, 13-1
 MMEMory, 14-1
 PAUSE, 15-1
 required, 7-1
 SENSe, 16-1
 SOURce, 17-1
 SYSTEM, 19-1
configuration editing internal instrument, 2-7
configuring LAN server configuring (optional), 2-7
Configuring the Internal Instrument Interface, 2-2, 2-3, 2-4
Configuring the SCPI (LAN) Client, 2-1

D

DISPlay Commands, 11-1
DOS command prompt example, 3-6

E

editing internal instrument configuration, 2-7

F

FORMat Commands, 12-1

H

HP BASIC example, 5-2
HP Basic program example, 5-6
HP E5500A introduction, 1-2
HP Vee example, 5-3

I

INITiate Commands, 13-1
interface internal instrument, configuring, 2-2, 2-3, 2-4
internal instrument editing configuration, 2-7
internal instrument interface configuring, 2-2, 2-3, 2-4

M

measurement definition SCPI, setup, 5-4
MMEMory Commands, 14-1

P

PAUSE Commands, 15-1
Program Examples, 5-1
programmatically starting SCPI LAN client, 3-6

R

Required Commands, 7-1

S

SCPI setup measurement definition, 5-4
SCPI Remote Interface configuring, 2-1
SCPI Command Reference, 6-1
SCPI Remote Interface starting, 3-2
SCPI logging function, 3-5
SENSe Commands, 16-1
server LAN, configuring (optional), 2-7
setup application environment, 5-2
starting SCPI Remote Interface, 3-2
starting SCPI Remote Interface programmatically, 3-6
Syntax, 6-1
SYSTEM Commands, 19-1

U

using HP E5500 SCPI Assistant, 4-1

W

[windows NT shortcut example, 3-6](#)