Agilent E5505A Phase Noise Measurement System

for E5500 Software Rev A.04.00

SCPI Command Reference

Agilent Technologies

Notices

Manual Part Number

E5505-90002

Edition

December 2012

Printed in USA

Agilent Technologies, Inc.
1400 Fountaingrove Pkwy
Santa Rosa, CA 95403

Warranty

The material contained in this document is provided "as is," and is subject to being changed, without notice, in future editions. Further, to the maximum extent permitted by applicable law, Agilent disclaims all warranties, either express or implied, with regard to this manual and any information contained herein, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. Agilent shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein. Should Agilent and the user have a separate written agreement with warranty terms covering the material in this document that conflict with these terms, the warranty terms in the separate agreement shall control.

Technology Licenses

The hardware and/or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license.

Restricted Rights Legend

If software is for use in the performance of a U.S. Government prime contract or subcontract, Software is delivered and licensed as "Commercial computer software" as defined in DFAR 252.227-7014 (June 1995), or as a "commercial item" as defined in FAR 2.101(a) or as "Restricted computer software" as defined in FAR 52.227-19 (June 1987) or any equivalent agency regulation or contract clause. Use, duplication or disclosure of Software is subject to Agilent Technologies' standard commercial license terms, and non-DOD Departments and Agencies of the U.S. Government will receive no greater than Restricted Rights as defined in FAR 52.227-19(c)(1-2) (June 1987). U.S. Government users will receive no greater than Limited Rights as defined in FAR 52.227-14 (June 1987) or DFAR 252.227-7015 (b)(2) (November 1995), as applicable in any technical data.

Safety Notices

CAUTION

A **CAUTION** notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in damage to the product or loss of important data. Do not proceed beyond a **CAUTION** notice until the indicated conditions are fully understood and met.

WARNING

A WARNING notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in personal injury or death. Do not proceed beyond a WARNING notice until the indicated conditions are fully understood and met.

Contents

8 CALibrate Commands

16 STATus Commands

17 SYSTem Commands

1
SCPI Interface

Introduction

Standard Commands for Programmable Instruments (SCPI) defines how you communicate with an instrument from a computer. It is a programming language designed specifically for electronic test and measurement instruments. SCPI standards are built on the IEEE-488.2 Standard Codes and Formats. For more information on SCPI, visit http://www.scpiconsortium.org.

The SCPI programming commands presented in this manual are used with the E5500 Phase Noise Measurement System software, which operates in all of the E5500 series of phase noise measurement systems (including the E5505A). The E5500 programming commands comply with the SCPI standards.

Configuring the SCPI Interface

The SCPI interface provides connectivity between the E5500 phase noise measurement (software) subsystem and your application programming environment. Multiple types of interfaces and protocols are supported. Before using an interface with the E5500 Phase Noise System Measurement software, some system configuration may be required. You need to configure only the interfaces you plan to use.

Once connected and configured, use the SCPI interface to open other interfaces and send SCPI commands to the phase noise subsystem.

Configuring Telnet and sockets interfaces

These interfaces use TCP/IP protocol through a LAN connection. In order to use these interfaces, you must have Windows® networking installed and configured. If you want to control the E5500 phase noise measurement subsystem from a remote computer, an appropriate LAN card must also be installed and configured.

Configuring the RS-232 interface

This interface provides direct serial port connection using a null-modem cable. COM port settings are set by the E5500 SCPI Remote Interface.

Configuring the VXI-11.2 interface

This interface uses VXI-11.2 protocol which provides full GPIB emulation through a LAN connection. It requires a software simulated GPIB card called an internal instrument installed and configured in Agilent I/O Libraries. The SCPI interface is then used to open this simulated GPIB card and connect it to the Phase Noise Subsystem. The Agilent I/O Libraries software which adds the internal instrument interface is part of the E5500 Installer.

NOTE

The Agilent I/O libraries version 15.1 and the I/O upgrade must be installed before beginning this procedure. Refer to the *E5505A Phase Noise Measurement System Installation Guide* for more information.

Figure 1 shows the relationship between the E5500 phase noise measurement subsystem, the SCPI Interface, the I/O Libraries Internal Instrument, and the user's application program running on either the same or a remote computer.

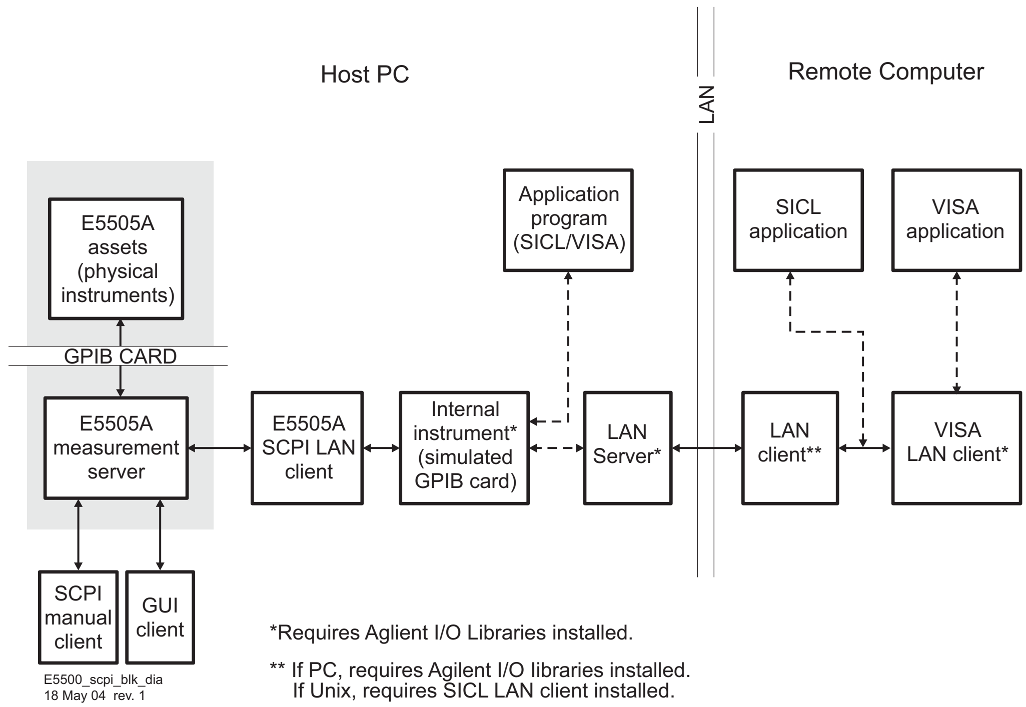

Figure 1 E5500 block diagram

Activating the LAN server (optional)

Activating the LAN Server is only required if you wish to control the E5500 phase noise measurement system using the SCPI VXI-11.2 interface from a remote computer.

Use the Windows start menu as in Figure 2 to navigate to the LAN Server.

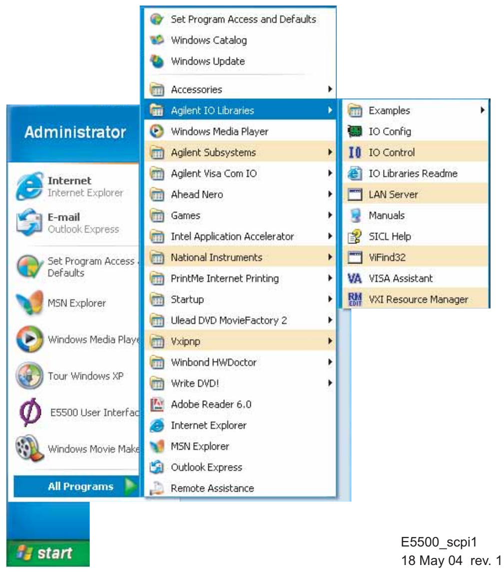

Figure 2 Navigate to LAN server

1 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.

Starting the SCPI Interface

1 Use the Windows start menu as in Figure 3 to navigate to SCPI Interface.

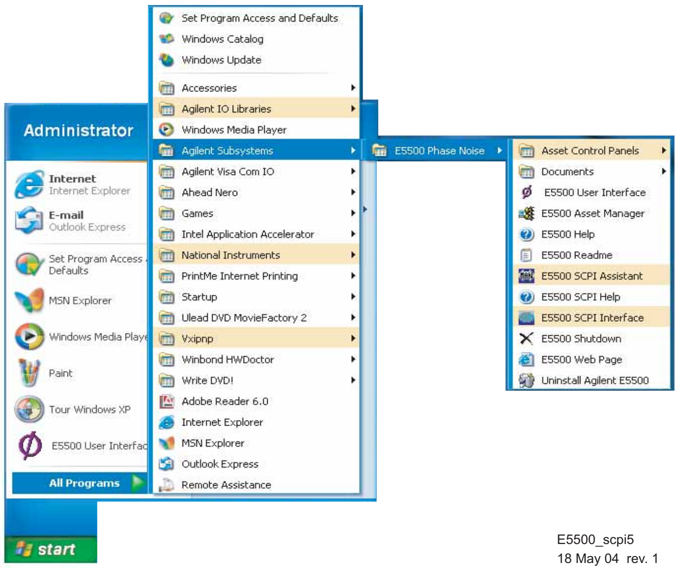

Figure 3 Navigate to SCPI interface

2 The **SCPI Interface Selections** dialog box appears.

NOTE

This dialog box is used to open one or more interfaces to the SCPI 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 through 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-11.2 - VXI-11.2 protocol, requires Agilent I/O Libraries software with a "Internal Instrument" type interface configured. For more information, see **"Configuring the VXI-11.2 interface"** on page 39.
- GPIB Slave card - A second 82350 GPIB card must be installed and configured as a GPIB slave Port. For more information, Refer to the E5505A Phase Noise Measurement System Installation Guide.

3 **Choose which protocol you wish to use by checking either Telnet Enable, Sockets Enable, RS-232 Enable or VXI.11.2 Enable or GPIB Enable (Figure 4 on page 43).**

4 If you selected **VXI-11.2 Enable**, in the **Select SICL Interface** pull-down list, select **gpib (x)** (**gpib7**, for example). Only the Internal Instrument type interfaces defined in **"Introduction"** on page 38, will be listed. You may also change the bus address of the subsystem if desired (default is 0).

5 Click the **OK** button.

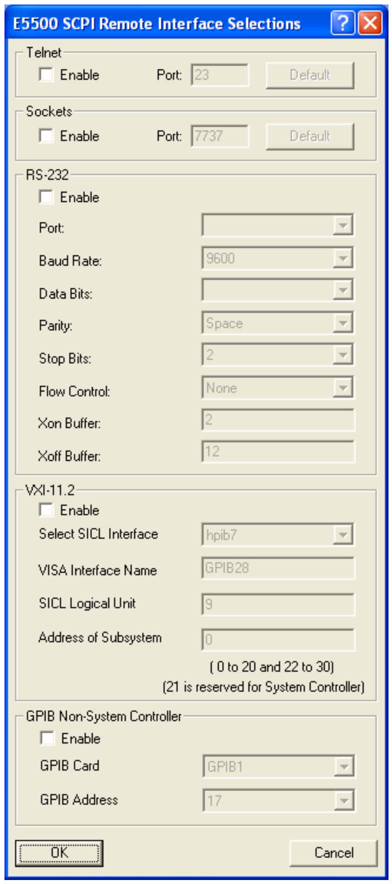

Figure 4 Choosing the interface

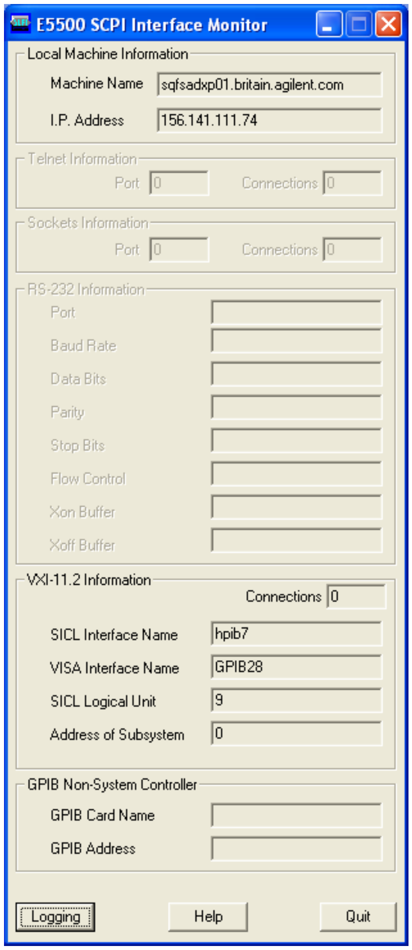

Figure 5 SCPI interface information

The selected interface is now open and connected to the E5505A 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 SCPI Interface **before** running any application software.

SCPI logging function

1 Click the Logging button shown in Figure 5. The SCPI Logging dialog box appears as shown in Figure 6.

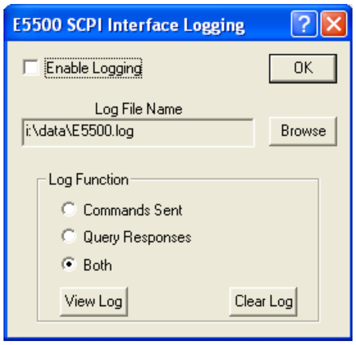

Figure 6 E5500 LAN client logging

2 Click the **OK** button.

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.

Starting the SCPI 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 with the command line. The \driver option takes *one* of the following arguments:

"name" Opens a VXI-11.2 interface where **"name"** is a quoted string SICL 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`

VXI Opens a VXI-11.2 Interface. The port setting last open is used.

gpib Opens a slave GPIB interface. The port setting last opened is used.

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-11.2

```
“C:\Program Files\Agilent\Measurement and Stimulus Subsystems\
E5500 Phase Noise\lanSCPIclient.exe”\driver "hpib7”
```

GPIB slave Port

```
“C:\Program Files\Agilent\Measurement and Stimulus Subsystems\E5500
Phase Noise\lanSCPIclient.exe”\driver gpib
```

2
SCPI Assistant

Starting the SCPI Assistant

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

Use the Windows start menu as in Figure 7 to navigate to SCPI assistant.

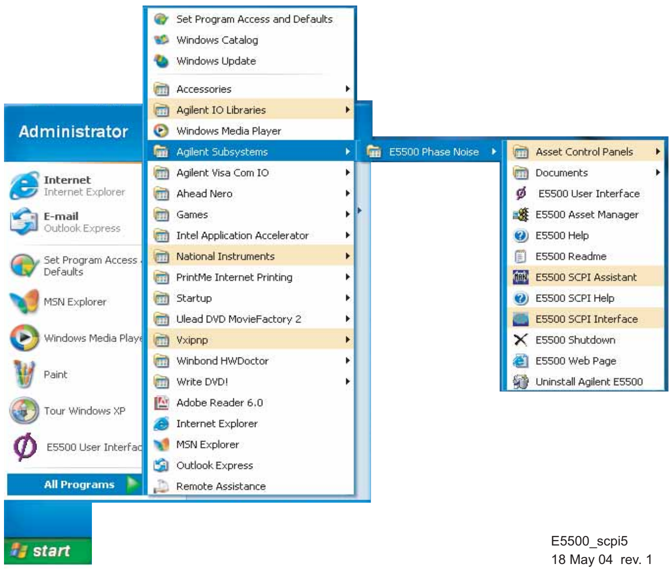

Figure 7 Navigate to SCPI assistant

The SCPI Assistant Interface (Figure 8) 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), click the **View QRG** button. The QRG can be printed using the standard Window printing capability.

Query responses appear in the Query Response Window.

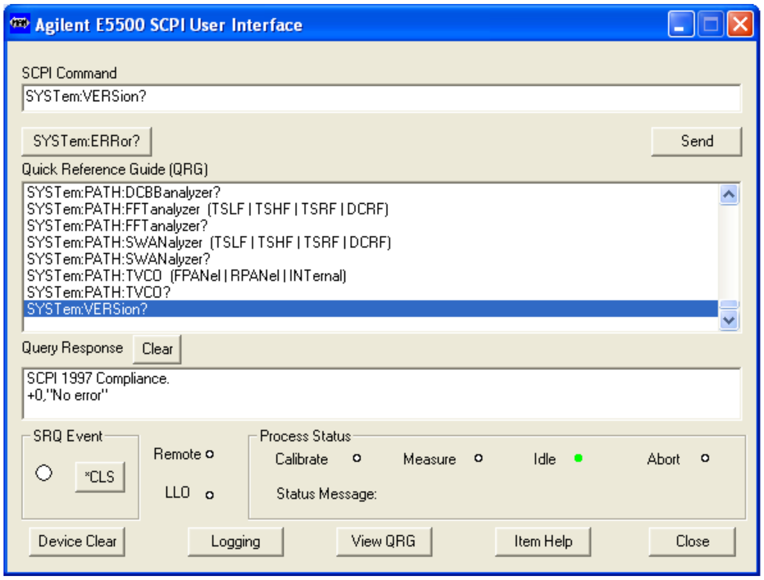

Figure 8 SCPI assistant interface

SCPI logging function

Click the **Logging** button (Figure 8). The **LAN Client Logging** dialog box appears (Figure 9). When enabled, this 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.

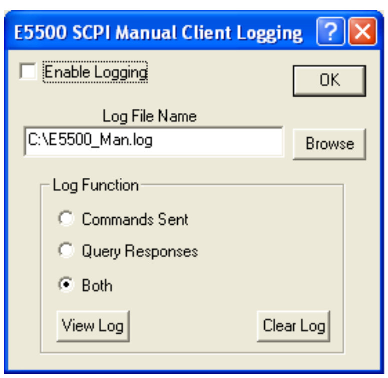

Figure 9 E5500 LAN client logging

- Click the **OK** button.

3
SCPI Program Examples

Application Environment Setup

Application software communicates with the E5505A system measurement server thorough 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 E5505A system or a remote computer.

The following are language specific examples of application setup. In all cases the SCPI interface has opened a VXI-11.2 interface with the SICL Interface Name of “hpib7” and the address of subsystem “0”.

HTBasic for Windows on same PC or remote PC

The autostart (autost) file must use the HPIBS driver to open the interface as follows:

Same PC as E5505A: `LOAD BIN "HPIBS; DEV hpib7 ISC 7"!`

Remote PC name or IP: `LOAD BIN "HPIBS; DEV lan[pc52]:hpib7 ISC 7"!`

- “pc52” is the name of the PC running the E5505A. You may also use an IP address here. “hpib7” is the SICL interface name opened with the SCPI LAN Client.

NOTE Using a remote PC requires installing Agilent I/O Libraries LAN CLIENT on the remote PC.

Your program would then look like:

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

RMBUX on remote workstation

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

```
INTERFACE 9="lan[pc52]:hpib7"; normal
```

- “pc52” is the name of the pc running the E5505A system. You may also use an IP address here. “hpib7” 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 look like:

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

Agilent VEE on same PC 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 E5505A system in the Gateway: pane.

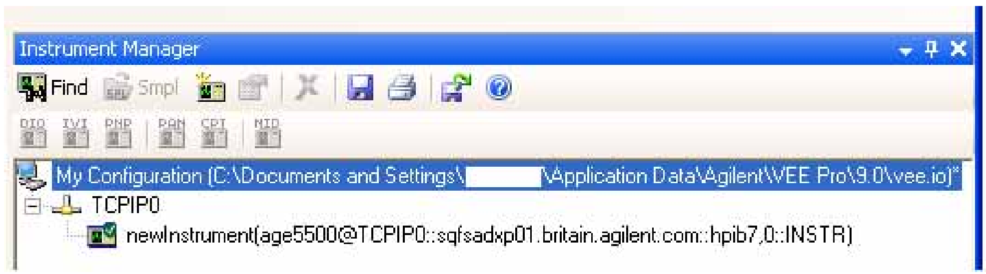

Figure 10 Add IP address

SCPI Programming

Generally, any E5500 program must perform these tasks:

- Set up the measurement definition
- Set up the device under test (DUT)[1]
- Initiate measurement and determine when complete
- Read the data from the measurement
- Process, display and store the data*

As with most programming tasks there is more than one method to accomplish these tasks. Several methods are discussed in this document.

Set up 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 (for example, carrier frequency) 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 path names.

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.

* These tasks are user specific and are not discussed in this document.

Initiate measurement and determine when complete

The SCPI command INITiate[:IMMediate][:ALL] begins a calibration and measurement sequence. A wait of approximately 15 seconds is recommended after issuing this command, as the E5505A system requires nearly 100% of the 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 queue 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.

HTBasic for Windows Program Example

The following program example demonstrates use of the above techniques in a HTBASIC program. It runs with your system in Demo Mode. The program recalls a measurement setup file and then makes measurements at several carrier frequencies. For information about HTBASIC for Windows, refer to the URL, http:\\www.htbasic.com.

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 2004 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 the 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$(J)
660 !
670 ! Code to setup Device Under Test goes here.
680 !
690 Restart: Abort_flg=0
700 !
710 OUTPUT @Pn;"INIT:ALL"   ! Start new calibration & measurement.
720 !
730 PRINT "      Measuring carrier at: ";Freq$(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 !
1500 END
```

4
E5500 SCPI Commands

Syntax Conventions

Table 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
  ```

- The 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
```

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)
```

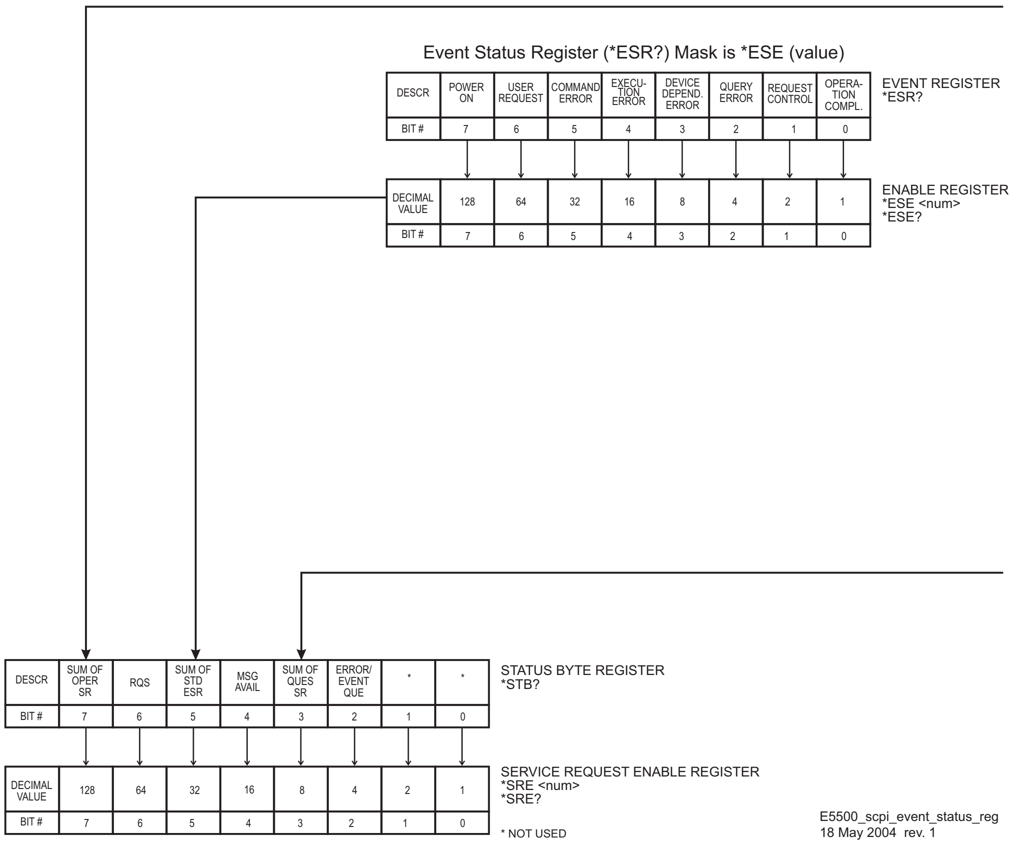

Figure 11 Event status register 1

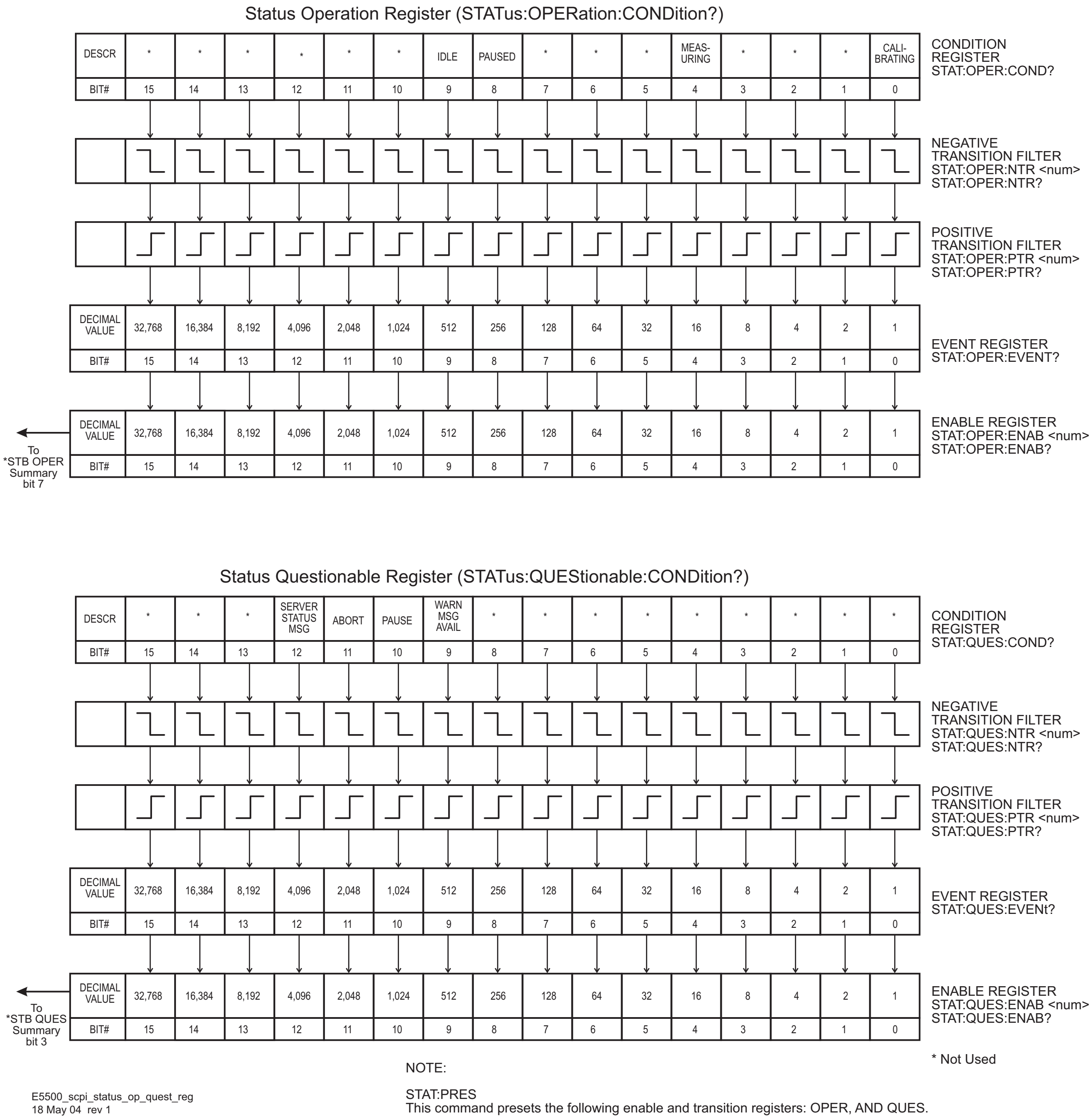

Figure 12 Event status register 2

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?
*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 | SDFF | 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)
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?
[: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]?
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?
```

5
Required Commands

Required Commands Introduction

The following IEEE 488.2 Required Commands are supported:

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

*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

```
E5500, Phase Noise Measurement Sub system, Version A.04.00
```

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? SDFF 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
- 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 and 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 queue 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.

6
ABORt Commands

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

7
CALCulate Commands

CALCulate Commands Introduction

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: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 | SDFF | 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? queue.

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:SPURtable returns “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 CALCule: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.

The 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.
"InFileAC.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|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 (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 by CALCulate:DATA? are affected by the following:

- CALCulate:TRANsform
- CALCulate:YSHift

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

8
CALibrate Commands

CALlibrate Commands Introduction

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|ABEatnote|DCPeak|SSPur|DSPur|FMDeviation|AMINternal|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|ABEatnote|DCPeak|SSPur|DSPur|FMDeviation|AMINternal|PSH
ift|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

- BEATnoteDerive from measured beatnote
- ABEatnote ... Derived from measured beatnote / auto cal. This parameter is valid only for the residual phase noise and the reference source being an E8257D, E8267D or E8663D. Otherwise, this parameter is accepted, but an error “ABORT: The detector constant method is invalid for the specified measurement type.” is popped-up and aborts when the detector constant calibration starts (this is the same behavior as other parameters).
- DCPeakDerive from measured +/- DC peak voltage
- SSPurDerive from single-sided spur
- DSPurDerive from double-sided spur
- FMDeviation Derive from FM rate and deviation
- AMINternal Use AM internal automatic self-calibration.
- NONEUse 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)
    - NONEUse the current detector constant
    - BEATnoteDerive from measured beatnote
- RESidual(Phase Noise without using a Phase Locked Loop)
    - NONEUse the current detector constant
    - BEATnoteDerive from measured beatnote
    - ABEatnoteDerive from measured beatnote / auto cal
    - DCPeakDerive from measured +/- DC peak voltage
    - SSPurDerive from single-sided spur
    - DSPurDerive from double-sided spur
- FMDiscriminator (Phase Noise using an FM Discriminator)
    - NONEUse the current detector constant
    - DSPurDerive from double-sided spur
    - FMDeviationDerive from FM rate and deviation
- AM (AM Noise Measurement)
    - NONEUse the current detector constant
    - AMINternal Use AM internal automatic self-calibration.
    - SSPurDerive from single-sided spur
    - DSPurDerive from double-sided spur
- BBANdBase 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.

VCO Tuning Constant = Expected Tuning Constant * (Rin / (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 by means of 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 2 Voltage Tune Range

Center Voltage	Maximum Voltage Tuning Range
<= 2V	+/-10V
> 2V	+/-12V - 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

9
DISPlay Commands

DISPlay Commands Introduction

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|SDFF|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)
- SDFF = 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.

10
FORMat Commands

FORMat Command Introduction

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.

11
INITiate Commands

INITiate Commands Introduction

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.

12
MMEMory Commands

MMEMory Commands Introduction

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")

MMEMory:LOAD[:ALL]

Load a measurement definition and data file.

Command syntax

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

Example

MMEMory:LOAD “Measfiles\Phase1.pnm

Attribute summary

Synchronization Required: none

*RST State: NA

SCPI Compliance: Standard

Additional information

Use the `MMEMory:STORe` command to store a file to load with `MMEMory: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 with `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).

13
PAUSe Commands

PAUSe Commands Introduction

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)
Accempts 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 an 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.

14
SENSe Commands

SENSe Commands Introduction

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]: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)
Acceps 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 N5507A downconverter is used. The N5507A 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 N5500A OPT 001 and N5507A 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 with in 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 - N5507A 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 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
- NOTSetBase 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 N5500A OPT 001 test set is used. The N5500A 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 N5500A OPT 001 test set is used. The N5500A 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 N5500A OPT 001 and an N5507A 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 N5500A 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 Nose 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 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

15
SOURce Commands

SOURce Commands Introduction

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:

```
Carrier = Residual * (Multiplier / Divisor)
```

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

```
Residual = Carrier / (Multiplier / 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:

```
Carrier = Residual * (Multiplier / Divisor)
```

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

```
Residual = Carrier / (Multiplier / 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:

```
Carrier = Residual * (Multiplier / Divisor)
```

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

```
Residual = Carrier / (Multiplier / 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:

```
Carrier = Residual * (Multiplier / Divisor)
```

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

```
Residual = Carrier / (Multiplier / 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.

16
STATus Commands

STATus Commands Introduction

- 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.

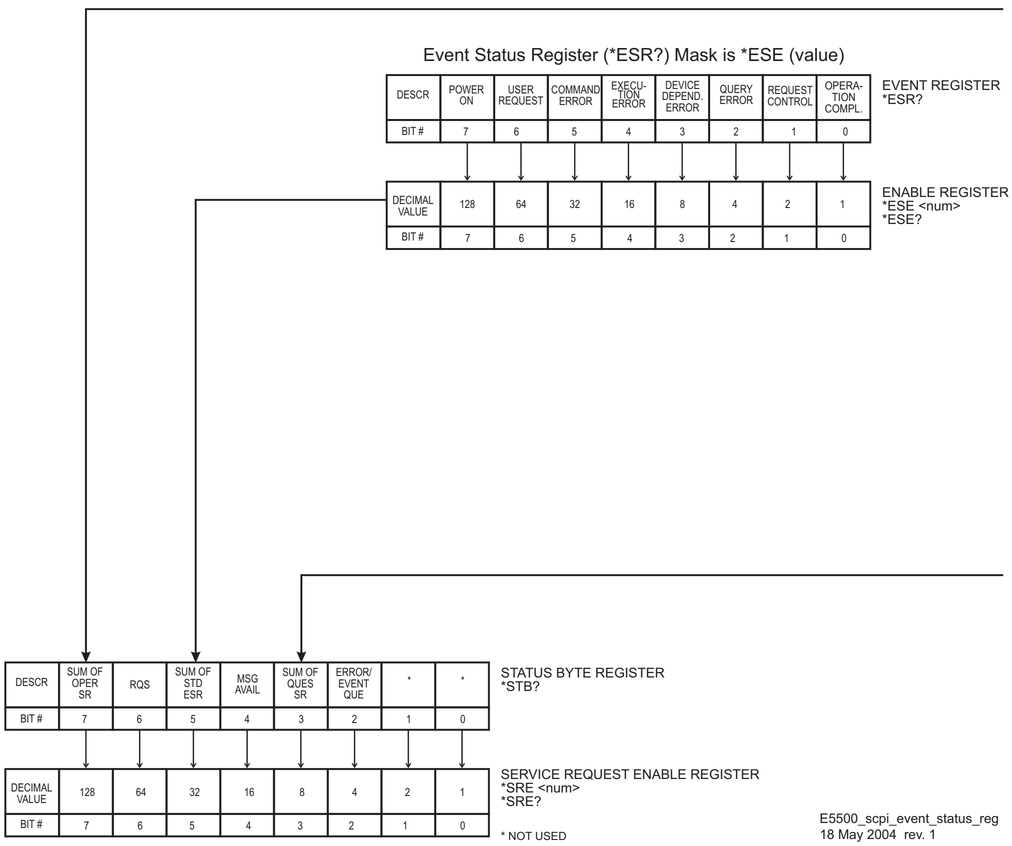

Figure 13 Event status register 1

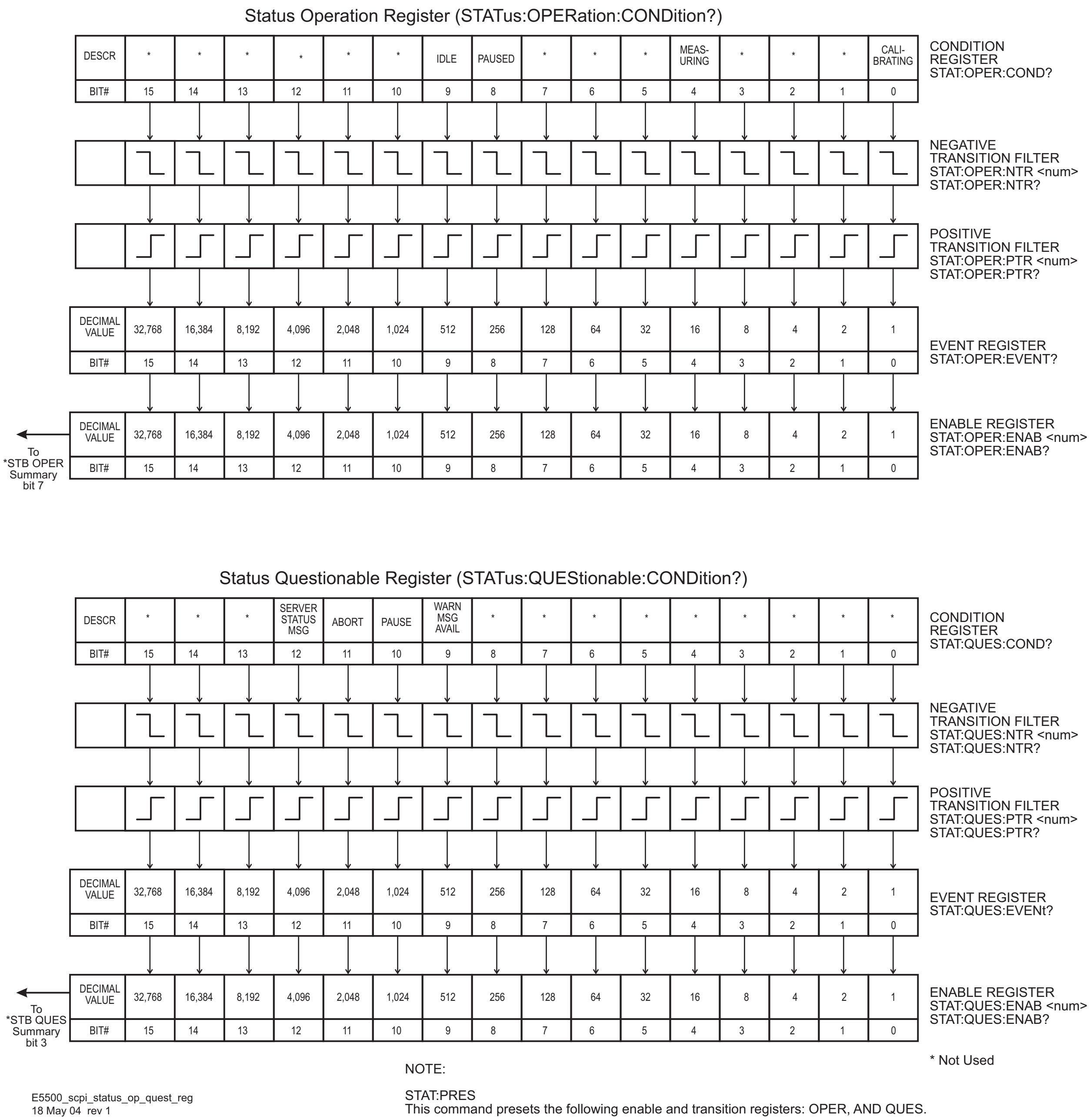

Figure 14 Event status register 2

STATus:ADVise:ENABle

When set to ON, server status messages (non-error) are also placed in the SCPI error queue 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

17
SYSTem Commands

SYSTem Commands Introduction

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”XXXXX “
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 "XXXX"
SYSTem:ASSet:CARRier "NONE"
```

Query syntax

```
SYSTem:ASSet:CARRier?
```

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: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,"XXXXXA-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:DCONverter <"None" | "asset_name">
```

Asset_name = quoted string as defined in the Asset Manager

Example

```
SYSTem:ASSet:DCONverter
AUTO,”XXXXXA-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 downconverter is set up through 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,”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 “XXXXX”
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,"XXXXX"
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"83650A"
SYSTem:ASSet: RESidual "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:SLAVe:SOURce

Specify the slave source associated with the 50kHz-1600MHz RF Output of the carrier noise test set (N5507A 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 ”XXXXX”
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,”XXXXX”
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 "XXXXX"
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 "N5500A"
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

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 3 on page 314.

Table 3 SCPI Error Messages

Error Number Range	E5500 Specific Errors Description
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 3 SCPI Error Messages (continued)

Error Number Range	E5500 Specific Errors Description
-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 numberic 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.