SCPI Command Reference

Agilent Technologies
E8663B Analog Signal Generator

For the latest revision of this guide, go to **http://www.agilent.com/find/e8663b** and click Product Library.

Agilent Technologies

Notice

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 to any of the Agilent products to which it pertains, 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 any of the Agilent products to which it pertains. Should Agilent have a written contract with the User and should any of the contract terms conflict with these terms, the contract terms shall control.

Contents

Contents

Contents

Contents

Contents

Contents

Contents

Contents

Contents

Contents

Documentation Overview

Installation Guide
- Safety Information
- Getting Started
- Operation Verification
- Regulatory Information

User's Guide
- Signal Generator Overview
- Basic Operation
- Optimizing Performance
- Analog Modulation
- Troubleshooting

Programming Guide
- Getting Started with Remote Operation
- Using IO Interfaces
- Programming Examples
- Programming the Status Register System
- Creating and Downloading Waveform Files
- Creating and Downloading User-Data Files

SCPI Reference
- Using this Guide
- System Commands
- Basic Function Commands
- Analog Commands
- SCPI Command Compatibility

Service Guide
- Troubleshooting
- Replaceable Parts
- Assembly Replacement
- Post-Repair Procedures
- Safety and Regulatory

Key Reference
- Key function description

1 Using this Guide

In the following sections, this chapter describes how SCPI information is organized and presented in this guide. An overview of the SCPI language is also provided:

- "How the SCPI Information is Organized" on page 1
- "SCPI Basics" on page 2

How the SCPI Information is Organized

SCPI Listings

The table of contents lists the Standard Commands for Programmable Instruments (SCPI) without the parameters. The SCPI subsystem name will generally have the first part of the command in parenthesis that is repeated in all commands within the subsystem. The title(s) beneath the subsystem name is the remaining command syntax. The following example demonstrates this listing:

Communication Subsystem (:SYSTem:COMMunicate)
    :PMETer:CHANnel
    :SERial:ECHO

The following examples show the complete commands from the above Table of Contents listing:

    :SYSTem:COMMunicate:PMETer:CHANnel
    :SYSTem:COMMunicate:SERial:ECHO

Subsystem Groupings by Chapter

A subsystem is a group of commands used to configure and operate a certain function or feature. Like individual commands, subsystems that share a similar scope or role can also be categorized and grouped together. This guide uses chapters to divide subsystems into the following groups:

- System Commands
- Basic Function Commands
- Analog Modulation Commands

Front Panel Operation Cross Reference

The last section in this book provides an index of hardkeys, softkeys, and data fields used in front panel operation, cross-referenced to their corresponding SCPI command. Key and data field names are sorted in two ways:

- individual softkey, hardkey, or data field name
- SCPI subsystem name with associated key and data field names nested underneath

SCPI Basics

This section describes the general use of the SCPI language for the signal generator. It is not intended to teach you everything about the SCPI language; the SCPI Consortium or IEEE can provide that level of detailed information. For a list of the specific commands available for the signal generator, refer to the table of contents.

For additional information, refer to the following publications:

- IEEE Standard 488.1-1987, IEEE Standard Digital Interface for Programmable Instrumentation. New York, NY, 1998.
- IEEE Standard 488.2-1992, IEEE Standard Codes, Formats, Protocols and Command Commands for Use with ANSI/IEEE Standard 488.1-1987. New York, NY, 1998.

Common Terms

The following terms are used throughout the remainder of this section:

Command	A command is an instruction in SCPI consisting of mnemonics (keywords), parameters (arguments), and punctuation. You combine commands to form messages that control instruments.
Controller	A controller is any device used to control the signal generator, for example a computer or another instrument.
Event Command	Some commands are events and cannot be queried. An event has no corresponding setting; it initiates an action at a particular time.
Program Message	A program message is a combination of one or more properly formatted commands. Program messages are sent by the controller to the signal generator.
Query	A query is a special type of command used to instruct the signal generator to make response data available to the controller. A query ends with a question mark. Generally you can query any command value that you set.
Response Message	A response message is a collection of data in specific SCPI formats sent from the signal generator to the controller. Response messages tell the controller about the internal state of the signal generator.

Command Syntax

A typical command is made up of keywords prefixed with colons (:). The keywords are followed by parameters. The following is an example syntax statement:

`[:SOURce]:POWer[:LEVel] MAXimum|MINimum`

In the example above, the `[:LEVel]` portion of the command immediately follows the `:POWer` portion with no separating space. The portion following the `[:LEVel]`, `MINimum|MAXimum`, are the parameters (argument for the command statement). There is a separating space (white space) between the command and its parameter.

Additional conventions in syntax statements are shown in Table 1-1 and Table 1-2.

Table 1-1   Special Characters in Command Syntax

Characters	Meaning	Example

|	A vertical stroke between keywords or parameters indicates alterative choices. For parameters, the effect of the command varies depending on the choice.	`[:SOURce]:AM:`
		`MOD DEEP|NORMal`
		
		DEEP or NORMal are the choices.

[ ]	Square brackets indicate that the enclosed keywords or parameters are optional when composing the command. These implied keywords or parameters will be executed even if they are omitted.	`[:SOURce]:FREQuency[:CW]?` SOURce and CW are optional items.

< >	Angle brackets around a word (or words) indicate they are not to be used literally in the command. They represent the needed item.	`[:SOURce]:FREQuency:`
		`STARt <val><unit>`
		
		In this command, the words <val> and <unit> should be replaced by the actual frequency and unit.
		
		`:FREQuency:STARt 2.5GHZ`

{ }	Braces indicate that parameters can optionally be used in the command once, several times, or not at all.	`[:SOURce]:LIST:`
		`POWer <val>{,<val>}`
		
		a single power listing:
		`LIST:POWer 5`
		a series of power listings:
		`LIST:POWer 5,10,15,20`

Table 1-2   Command Syntax

Characters, Keywords, and Syntax	Example
Upper-case lettering indicates the minimum set of characters required to execute the command.	`[:SOURce]:FREQuency[:CW]?`, FREQ is the minimum requirement.
Lower-case lettering indicates the portion of the command that is optional; it can either be included with the upper-case portion of the command or omitted. This is the flexible format principle called forgiving listening. Refer to "Command Parameters and Responses" on page 5 for more information.	`:FREQuency` Either :FREQ, :FREQuency, or :FREQUENCY is correct.
When a colon is placed between two command mnemonics, it moves the current path down one level in the command tree. Refer to "Command Tree" on page 4 more information on command paths.	`:TRIGger:OUTPut:POLarity?` TRIGger is the root level keyword for this command.
If a command requires more than one parameter, you must separate adjacent parameters using a comma. Parameters are not part of the command path, so commas do not affect the path level.	`[:SOURce]:LIST:` `DWELl <val>{,<val>}`
A semicolon separates two commands in the same program message without changing the current path.	`:FREQ 2.5GHZ;:POW 10DBM`
White space characters, such as <tab> and <space>, are generally ignored as long as they do not occur within or between keywords. However, you must use white space to separate the command from the parameter, but this does not affect the current path.	`:FREQ uency` or `:POWer :LEVel` are not allowed. A <space> between :LEVel and 6.2 is mandatory. `:POWer:LEVel 6.2`

Command Types

Commands can be separated into two groups: common commands and subsystem commands. Figure 1-1, shows the separation of the two command groups. Common commands are used to manage macros, status registers, synchronization, and data storage and are defined by IEEE 488.2. They are easy to recognize because they all begin with an asterisk. For example `*IDN?`, `*OPC`, and `*RST` are common commands. Common commands are not part of any subsystem and the signal generator interprets them in the same way, regardless of the current path setting.

Subsystem commands are distinguished by the colon (`:`). The colon is used at the beginning of a command statement and between keywords, as in `:FREQuency[:CW?]`. Each command subsystem is a set of commands that roughly correspond to a functional block inside the signal generator. For example, the power subsystem (`:POWer`) contains commands for power generation, while the status subsystem (`:STATus`) contains commands for controlling status registers.

Figure 1-1   Command Types
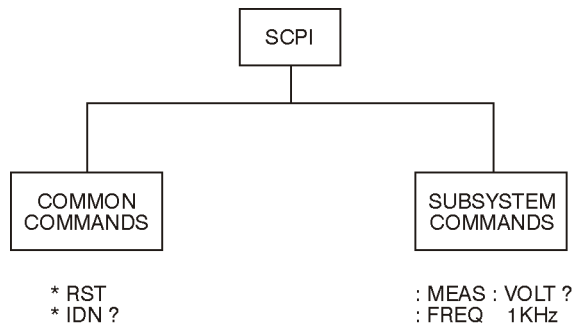
Command Tree

Most programming tasks involve subsystem commands. SCPI uses a structure for subsystem commands similar to the file systems on most computers. In SCPI, this command structure is called a command tree and is shown in Figure 1-2.

Figure 1-2   Simplified Command Tree

ck710a
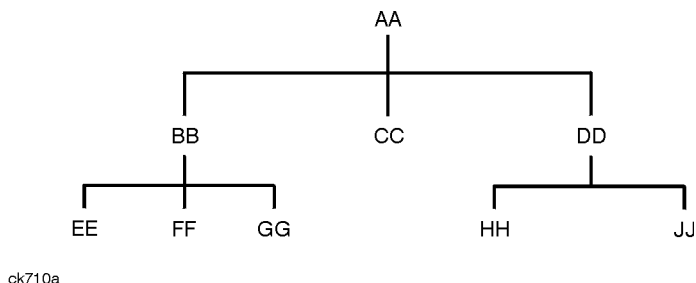
The command closest to the top is the root command, or simply "the root." Notice that you must follow a particular path to reach lower level commands. In the following example, `:POWer` represents AA, `:ALC` represents BB, `:SOURce` represents GG. The complete command path is `:POWer:ALC:SOURce?` (:AA:BB:GG).

Paths Through the Command Tree

To access commands from different paths in the command tree, you must understand how the signal generator interprets commands. The parser, a part of the signal generator firmware, decodes each message sent to the signal generator. The parser breaks up the message into component commands using a set of rules to determine the command tree path used. The parser keeps track of the current path (the level in the command tree) and where it expects to find the next command statement. This is important because the same keyword may appear in different paths. The particular path is determined by the keyword(s) in the command statement.

A message terminator, such as a `<new line>` character, sets the current path to the root. Many programming languages have output statements that automatically send message terminators.

| NOTE | The current path is set to the root after the line-power is cycled or when `*RST` is sent. |

Command Parameters and Responses

SCPI defines different data formats for use in program and response messages. It does this to accommodate the principle of forgiving listening and precise talking. For more information on program data types refer to IEEE 488.2. Forgiving listening means the command and parameter formats are flexible.

For example, with the `:FREQuency:REFerence:STATe ON|OFF|1|0` command, the signal generator accepts `:FREQuency:REFerence:STATe ON`, `:FREQuency:REFerence:STATe 1`, `:FREQ:REF:STAT ON`, `:FREQ:REF:STAT 1` to turn on the frequency reference mode.

Each parameter type has one or more corresponding response data types. A setting that you program using a numeric parameter returns either real or integer response data when queried. Response data (data returned to the controller) is more concise and restricted and is called precise talking.

Precise talking means that the response format for a particular query is always the same.

For example, if you query the power state (`:POWer:ALC:STATe?`) when it is on, the response is always 1, regardless of whether you previously sent `:POWer:ALC:STATe 1` or `:POWer:ALC:STATe ON`.

Table 1-3   Parameter and Response Types

Parameter Types	Response Data Types
Numeric	Real, Integer
Extended Numeric	Real, Integer
Discrete	Discrete
Boolean	Numeric Boolean
String	String

Numeric Parameters

Numeric parameters are used in both common and subsystem commands. They accept all commonly used decimal representations of numbers including optional signs, decimal points, and scientific notation.

If a signal generator setting is programmed with a numeric parameter which can only assume a finite value, it automatically rounds any entered parameter which is greater or less than the finite value. For example, if a signal generator has a programmable output impedance of 50 or 75 ohms, and you specified 76.1 for the output impedance, the value is rounded to 75. The following are examples of numeric parameters:

100	no decimal point required
100.	fractional digits optional
−1.23	leading signs allowed
4.56E<space>3	space allowed after the E in exponential
−7.89E–001	use either E or e in exponential
+256	leading + allowed
.5	digits left of decimal point optional

Extended Numeric Parameters

Most subsystems use extended numeric parameters to specify physical quantities. Extended numeric parameters accept all numeric parameter values and other special values as well.

The following are examples of extended numeric parameters:

Extended Numeric Parameters		Special Parameters	
100	any simple numeric value	DEFault	resets parameter to its default value
1.2GHZ	GHZ can be used for exponential (E009)	UP	increments the parameter
200MHZ	MHZ can be used for exponential (E006)	DOWN	decrements the parameter
−100mV	negative 100 millivolts	MINimum	sets parameter to smallest possible value
10DEG	10 degrees	MAXimum	sets parameter to largest possible value

Discrete Parameters

Discrete parameters use mnemonics to represent each valid setting. They have a long and a short form, just like command mnemonics. You can mix upper and lower case letters for discrete parameters.

The following examples of discrete parameters are used with the command
`:TRIGger[:SEQuence]:SOURce BUS|IMMediate|EXTernal`.

BUS	GPIB, LAN, or RS-232 triggering
IMMediate	immediate trigger (free run)
EXTernal	external triggering

Although discrete parameters look like command keywords, do not confuse the two. In particular, be sure to use colons and spaces correctly. Use a colon to separate command mnemonics from each other and a space to separate parameters from command mnemonics.

The following are examples of discrete parameters in commands:

```
TRIGger:SOURce BUS

TRIGger:SOURce IMMediate

TRIGger:SOURce EXTernal
```

Boolean Parameters

Boolean parameters represent a single binary condition that is either true or false. The two-state boolean parameter has four arguments. The following list shows the arguments for the two-state boolean parameter:

ON	boolean true, upper/lower case allowed
OFF	boolean false, upper/lower case allowed
1	boolean true
0	boolean false

String Parameters

String parameters allow ASCII strings to be sent as parameters. Single or double quotes are used as delimiters.

The following are examples of string parameters:

```
 'This is                      "This is also                     'SO IS THIS'
 valid'                        valid"
```

Real Response Data

Real response data represent decimal numbers in either fixed decimal or scientific notation. Most high-level programming languages that support signal generator input/output (I/O) handle either decimal or scientific notation transparently.

The following are examples of real response data:

+4.000000E+010, −9.990000E+002

−9.990000E+002

+4.0000000000000E+010

+1

0

Integer Response Data

Integer response data are decimal representations of integer values including optional signs. Most status register related queries return integer response data. The following are examples of integer response data:

0	signs are optional	−100	leading − allowed
+100	leading + allowed	256	never any decimal point

Discrete Response Data

Discrete response data are similar to discrete parameters. The main difference is that discrete response data only returns the short form of a particular mnemonic, in all upper case letters. The following are examples of discrete response data:

IMM   EXT   INT   NEG

Numeric Boolean Response Data

Boolean response data returns a binary numeric value of one or zero.

String Response Data

String response data are similar to string parameters. The main difference is that string response data returns double quotes, rather than single quotes. Embedded double quotes may be present in string response data. Embedded quotes appear as two adjacent double quotes with no characters between them. The following are examples of string response data:

```
"This is a string"

"one double quote inside brackets: [""]"

"Hello!"
```

Program Messages

The following commands will be used to demonstrate the creation of program messages:

```
[:SOURce]:FREQuency:STARt              [:SOURce]:FREQuency:STOP

[:SOURce]:FREQuency[:CW]               [:SOURce]:POWer[:LEVel]:OFFSet
```

Example 1

`:FREQuency:STARt 500MHZ;STOP 1000MHZ`

This program message is correct and will not cause errors; STARt and STOP are at the same path level. It is equivalent to sending the following message:

`FREQuency:STARt 500MHZ;FREQuency:STOP 1000MHZ`

Example 2

`:POWer 10DBM;:OFFSet 5DB`

This program message will result in an error. The message makes use of the default POWer[:LEVel] node (root command). When using a default node, there is no change to the current path position. Since there is no command OFFSet at the root level, an error results.

The following example shows the correct syntax for this program message:

`:POWer 10DBM;:POWer:OFFSet 5DB`

Example 3

```
:POWer:OFFSet 5DB;POWer 10DBM
```

This program message results in a command error. The path is dropped one level at each colon. The first half of the message drops the command path to the lower level command `OFFSet`; `POWer` does not exist at this level.

The `POWer 10DBM` command is missing the leading colon and when sent, it causes confusion because the signal generator cannot find `POWer` at the `POWer:OFFSet` level. By adding the leading colon, the current path is reset to the root. The following shows the correct program message:

```
:POWer:OFFSet 5DB;:POWer 10DBM
```

Example 4

```
FREQ 500MHZ;POW 4DBM
```

In this example, the keyword short form is used. The program message is correct because it utilizes the default nodes of `:FREQ[:CW]` and `:POW[:LEVel]`. Since default nodes do not affect the current path, it is not necessary to use a leading colon before `FREQ` or `POW`.

File Name Variables

File name variables, such as `"<file name>"`, represent three formats, `"<file name>"`, `"<file name@file type>"`, and `"</user/file type/file name>"`. The following shows the file name syntax for the three formats, but uses `"FLATCAL"` as the file name in place of the variable `"<file name>"`:

Format 1   `"FLATCAL"`

Format 2   `"FLATCAL@USERFLAT"`

Format 3   `"/USER/USERFLAT/FLATCAL"`

Format 2 uses the file type extension (@USERFLAT) as part of the file name syntax. Format 3 uses the directory path which includes the file name and file type. Use Formats 2 and 3 when the command does not specify the file type. This generally occurs in the Memory (`:MEMory`) or Mass Memory (`:MMEMory`) subsystems. The following examples demonstrate a command where Format 1 applies:

Command Syntax with the file name variable   `:MEMory:STORe:LIST "<file name>"`

Command Syntax with the file name   `:MEMory:STORe:LIST "SWEEP_1"`

This command has `:LIST` in the command syntax. This denotes that `"SWEEP_1"` will be saved in the :List file type location as a list type file.

The following examples demonstrate a command where Format 2 applies:

Command Syntax with the file name variable

    :MMEMory:COPY "<filename>","<filename>"

Command Syntax with the file name

    :MMEMory:COPY "FLATCAL@USERFLAT","FLAT_2CAL@USERFLAT"

This command cannot distinguish which file type "FLATCAL" belongs to without the file type extension (@USERFLAT). If this command were executed without the extension, the command would assume the file type was Binary.

The following examples demonstrate a command where format 3 applies:

Command Syntax with the file name variable

    :MMEMory:DATA "/USER/BBG1/WAVEFORM/<file name>",#ABC

Command Syntax with the file name

    :MMEMory:DATA "/USER/BBG1/WAVEFORM/FLATCAL",#ABC

This command gives the directory path name where the file "FLATCAL" is stored.

A	the number of decimal digits to follow in B.
B	a decimal number specifying the number of data bytes in C.
C	the binary waveform data.

Refer to Table 2-1 on page 37 for a listing of the file systems and types. The entries under file type are used in the directory path.

MSUS (Mass Storage Unit Specifier) Variable

The variable "<msus>" enables a command to be file type specific when working with user files. Some commands use it as the only command parameter, while others can use it in conjunction with a file name when a command is not file type specific. When used with a file name, it is similar to Format 2 in the File Name Variables section on page 10. The difference is the file type specifier (msus) occupies its own variable and is not part of the file name syntax.

The following examples illustrate the usage of the variable "<msus>" when it is the only command parameter:

Command Syntax with the msus variable

`:MMEMory:CATalog? "<msus>"`

Command Syntax with the file system

`:MMEMory:CATalog? "LIST:"`

The variable "<msus>" is replaced with "LIST:". When the command is executed, the output displays only the files from the List file system.

The following examples illustrate the usage of the variable "<file name>" with the variable "<msus>":

*Command Syntax with the file name and msus variable*s

`:MMEMory:DELete[:NAME] "<file name>",["<msus>"]`

Command Syntax with the file name and file system

`:MMEMory:DELete:NAME "LIST_1","LIST:"`

The command from the above example cannot discern which file system LIST_1 belongs to without a file system specifier and will not work without it. When the command is properly executed, LIST_1 is deleted from the List file system.

The following example shows the same command, but using Format 2 from the File Name Variables section on page 10:

`:MMEMory:DELete:NAME "LIST_1@LIST"`

When a file name is a parameter for a command that is not file system specific, either format (<file name>","<msus>" or "<file name@file system>") will work.

Refer to Table 1-1 on page 3 for a listing of special syntax characters.

Quote Usage with SCPI Commands

As a general rule, programming languages require that SCPI commands be enclosed in double quotes as shown in the following example:

`":FM:EXTernal:IMPedance 600"`

However, when a string is the parameter for a SCPI command, additional quotes or other delimiters may be required to identify the string. Your programming language may use two sets of double

quotes, one set of single quotes, or back slashes with quotes to signify the string parameter. The following examples illustrate these different formats:

`"MEMory:LOAD:LIST ""myfile"""` used in BASIC programming languages

`"MEMory:LOAD:LIST \"myfile\""` used in C, C++, Java, and PERL

`"MEMory:LOAD:LIST 'myfile'"` accepted by most programming languages

Consult your programming language reference manual to determine the correct format.

Binary, Decimal, Hexadecimal, and Octal Formats

Command values may be entered using a binary, decimal, hexadecimal, or octal format. When the binary, hexadecimal, or octal format is used, their values must be preceded with the proper identifier. The decimal format (default format) requires no identifier and the signal generator assumes this format when a numeric value is entered without one. The following list shows the identifiers for the formats that require them:

- #B identifies the number as a binary numeric value (base-2).

- #H identifies the number as a hexadecimal alphanumeric value (base-16).

- #Q identifies the number as a octal alphanumeric value (base-8).

The following are examples of SCPI command values and identifiers for the decimal value 45:

#B101101   binary equivalent

#H2D   hexadecimal equivalent

#Q55   octal equivalent

The following example sets the RF output power to 10 dBm (or the equivalent value for the currently selected power unit, such as `DBUV` or `DBUVEMF`) using the hexadecimal value 000A:

    `:POW #H000A`

A unit of measure, such as `DBM` or `mV`, will not work with the values when using a format other than decimal.

The following example sets the bluetooth board address to FFBF7 (hexadecimal):

`:RADio:BLUEtooth:ARB:BDADdr #HFFBF7`

2 System Commands

In the following sections, this chapter provides SCPI descriptions for subsystems dedicated to peripheral signal generator operations that are applicable to this signal generator:

Calibration Subsystem (:CALibration)

:DCFM

`:CALibration:DCFM`

This command initiates a DCFM or DCΦM calibration depending on the currently active modulation. This calibration eliminates any dc or modulation offset of the carrier signal.

Use this calibration for externally applied signals. While the calibration can also be performed for internally generated signals, dc offset is not a normal characteristic for them.

NOTE	If the calibration is performed with a dc signal applied, any deviation provided by the dc signal will be removed and the new zero reference point will be at the applied dc level. The calibration will have to be performed again when the dc signal is removed in order to reset the carrier signal to the correct zero reference.

Key Entry   DCFM/DCΦM Cal

Communication Subsystem (:SYSTem:COMMunicate)

:GPIB:ADDRess

`:SYSTem:COMMunicate:GPIB:ADDRess <number>`
`:SYSTem:COMMunicate:GPIB:ADDRess?`

This command sets the signal generator's general purpose instrument bus (GPIB) address.

The variable <number> is a numeric value between 0 and 30. The signal generator typically uses 19 as the instrument address. The address must be different from other GPIB devices in your system.

The setting enabled by this command is not affected by a signal generator power-on, preset, or *RST command.

Example

`:SYST:COMM:GPIB:ADDR 19`

The preceding example sets the signal generator's GPIB address to 19.

Range   0–30

Key Entry   GPIB Address

:GTLocal

`:SYSTem:COMMunicate:GTLocal`

This command sets the signal generator to local mode, enabling front panel operation.

Range   N/A

Key Entry   Local

:LAN:CONFig

```
:SYSTem:COMMunicate:LAN:CONFig DHCP|MANual
```
```
:SYSTem:COMMunicate:LAN:CONFig?
```

This command selects the signal generator's internet protocol (IP) address. The dynamic host communication protocol (DHCP) selection allows the network to assign an IP address. The manual selection allows the user to enter an IP address.

Example

```
:SYST:COMM:LAN:CONF DHCP
```

The preceding example sets up the signal generator LAN configuration to use a DHCP IP address.

Key Entry   LAN Config

:LAN:GATEway

```
:SYSTem:COMMunicate:LAN:GATEway "<ipstring>"
```
```
:SYSTem:COMMunicate:LAN:GATEway?
```

This command sets the gateway for local area network (LAN) access to the signal generator from outside the current sub-network.

The "<ipstring>" string variable is the LAN gateway address, formatted as xxx.xxx.xxx.xxx. Refer to "Quote Usage with SCPI Commands" on page 12 for information on using quotes for different programming languages.

Using an empty string restricts access to the signal generator to local hosts on the LAN.

Example

```
:SYST:COMM:LAN:GATE "203.149.781.101"
```

The preceding example sets the signal generator's LAN gateway address.

Key Entry   Default Gateway

:LAN:HOSTname

```
:SYSTem:COMMunicate:LAN:HOSTname "<string>"
```
```
:SYSTem:COMMunicate:LAN:HOSTname?
```

This command sets the signal generator's local area network (LAN) connection hostname.

The "<string>" variable is the hostname for the signal generator. Refer to "Quote Usage with SCPI Commands" on page 12 for information on using quotes for different programming languages.

The setting enabled by this command is not affected by a signal generator power-on, preset, or *RST command.

Example

```
:SYST:COMM:LAN:HOSTname "siginst3"
```

The preceding example sets "siginst3" as the signal generator's LAN hostname.

Key Entry   Hostname

:LAN:IP

```
:SYSTem:COMMunicate:LAN:IP "<ipstring>"
```
```
:SYSTem:COMMunicate:LAN:IP?
```

This command sets the signal generator's local area network (LAN) internet protocol (IP) address for your IP network connection.

The "<ipstring>" variable is the signal generator's IP address, formatted as xxx.xxx.xxx.xxx. Refer to "Quote Usage with SCPI Commands" on page 12 for information on using quotes for different programming languages.

The setting enabled by this command is not affected by a signal generator power-on, preset, or *RST command.

Example

```
:SYST:COMM:LAN:IP "202.195.207.193"
```

The preceding example sets the signal generator's LAN IP address.

Key Entry   IP Address

:LAN:SUBNet

```
:SYSTem:COMMunicate:LAN:SUBNet "<ipstring>"
```
```
:SYSTem:COMMunicate:LAN:SUBNet?
```

This command sets the signal generator's local area network (LAN) subnet mask address for your internet protocol (IP) network connection.

The "<ipstring>" variable is the subnet mask for the IP address, formatted as xxx.xxx.xxx.xxx. Refer to "Quote Usage with SCPI Commands" on page 12 for information on using quotes for different programming languages.

The setting enabled by this command is not affected by a signal generator power-on, preset, or *RST command.

Example

```
:SYST:COMM:LAN:SUBN "203.194.101.111"
```

The preceding example sets the signal generator's LAN subnet mask.

Key Entry   **Subnet Mask**

:PMETer:ADDRess

```
:SYSTem:COMMunicate:PMETer:ADDRess <val>
```
```
:SYSTem:COMMunicate:PMETer:ADDRess?
```

This command sets the instrument address for a power meter that is controlled by the signal generator. The power meter is controlled only through a general purpose instrument bus (GPIB) cable.

The variable <number> is an integer numeric value between 0 and 30. The power meter address must be different from the GPIB address of the signal generator and any other GPIB instrument addresses in your system.

The setting enabled by this command is not affected by a signal generator power-on, preset, or *RST command.

Example

```
:SYST:COMM:PMET:ADDR 14
```

The preceding example sets the address to 14 for the power meter that is connected to and controlled by the signal generator.

Range   0–30

Key Entry   **Meter Address**

:PMETer:CHANnel

```
:SYSTem:COMMunicate:PMETer:CHANnel A|B
```
```
:SYSTem:COMMunicate:PMETer:CHANnel?
```

This command sets the measurement channel on a dual channel power meter that is controlled by the signal generator. A single-channel power meter uses channel A and selecting channel B will have no effect.

The setting enabled by this command is not affected by a signal generator power-on, preset, or *RST command. The power meter is controlled only through a general purpose instrument bus (GPIB) cable.

Example

```
:SYST:COMM:PMET:CHAN B
```

The preceding example sets the B measurement channel for the power meter that is connected to and controlled by the signal generator.

Key Entry   Meter Channel A B

:PMETer:IDN

```
:SYSTem:COMMunicate:PMETer:IDN E4418B|E4419B|E4416A|E4417A
```
```
:SYSTem:COMMunicate:PMETer:IDN?
```

This command sets the model number of the power meter that is controlled by the signal generator. The setting enabled by this command is not affected by a signal generator power-on, preset, or *RST command. The power meter is controlled only through a general purpose instrument bus (GPIB) cable.

Example

```
:SYST:COMM:PMET:IDN E4417A
```

The preceding example sets the model number for the power meter that is connected to and controlled by the signal generator.

Key Entry   Power Meter

:PMETer:TIMEout

```
:SYSTem:COMMunicate:PMETer:TIMEout <num>[<time_suffix>]
```
```
:SYSTem:COMMunicate:PMETer:TIMEout?
```

This command sets the period of time that the signal generator will wait for a valid reading from the power meter. The variable <num> has a resolution of 0.001.

The variable <num> is the time expressed as a number. The variable <time_suffix> are the units of time, for example mS (milliseconds) or S (seconds).

The setting enabled by this command is not affected by a signal generator power-on, preset, or *RST command. The power meter is controlled only through a general purpose instrument bus (GPIB) cable. If a timeout occurs, the signal generator reports an error message.

Example

```
:SYST:COMM:PMET:TIME .1SEC
```

The preceding example sets the timeout to 100 milliseconds for the power meter that is connected to and controlled by the signal generator.

Range   1mS–100S

Key Entry   Meter Timeout

:SERial:BAUD

`:SYSTem:COMMunicate:SERial:BAUD <number>`

`:SYSTem:COMMunicate:SERial:BAUD?`

This command sets the baud rate for the rear panel RS-232 interface labeled RS-232. The setting enabled by this command is not affected by a signal generator power-on, preset, or *RST command.

The variable <number> is an integer value corresponding to baud rates: 300, 2400, 4800, 9600, 19200, 38400, and 57600.

Example

`:SYST:COMM:SER:BAUD 9600`

The preceding example sets the baud rate for serial communication to 9600.

Key Entry   RS-232 Baud Rate

:SERial:ECHO

`:SYSTem:COMMunicate:SERial:ECHO ON|OFF`

`:SYSTem:COMMunicate:SERial:ECHO?`

This command enables or disables the RS-232 echo, and is not affected by a power-on, preset, or *RST command. Characters sent to the signal generator are displayed or echoed to the controller display.

Example

`:SYST:COMM:SER:ECHO ON`

The preceding example enables RS-232 echoing.

Key Entry   RS-232 ECHO Off On

:SERial:RESet

`:SYSTem:COMMunicate:SERial:RESet`

This event command resets the RS-232 buffer and discards unprocessed SCPI input received at the RS-232 port.

Key Entry   Reset RS-232

:SERial:TOUT

`:SYSTem:COMMunicate:SERial:TOUT <val>`

`:SYSTem:COMMunicate:SERial:TOUT?`

This command sets the RS-232 serial port timeout value. If further input is not received within the timeout period specified while a SCPI command is processed, the command aborts and clears the input buffer. The variable <val> is entered in seconds. The setting is not affected by a signal generator power-on, preset, or *RST command.

Example

`:SYST:COMM:SER:TOUT 2SEC`

The preceding example sets the RS-232 timeout for 2 seconds.

Range   1–25

Key Entry   RS-232 Timeout

Diagnostic Subsystem (:DIAGnostic[:CPU]:INFOrmation)

:BOARds

`:DIAGnostic[:CPU]:INFOrmation:BOARds?`

This query returns a list of the boards installed in the signal generator. The information is returned in the following format:

`"<board_name,part_number,serial_number,version_number,status>"`

This information format will repeat for each of the signal generator's detected boards.

Key Entry   Installed Board Info

:CCOunt:ATTenuator

`:DIAGnostic[:CPU]:INFOrmation:CCOunt:ATTenuator?`

This query returns the cumulative number of times that the attenuator has switched levels.

Key Entry   Diagnostic Info

:CCOunt:PON

`:DIAGnostic[:CPU]:INFOrmation:CCOunt:PON?`

This query returns the cumulative number of times the signal generator has been powered-on.

Key Entry   Diagnostic Info

:DISPlay:OTIMe

`:DIAGnostic[:CPU]:INFOrmation:DISPlay:OTIMe?`

This query returns the cumulative number of hours the display has been on.

Key Entry   **Diagnostic Info**

:LICENse:AUXiliary

`:DIAGnostic[:CPU]:INFOrmation:LICense:AUXiliary?`

This query returns a listing of current external software application license numbers for an auxiliary instrument.

Key Entry   **Auxiliary Software Options**

:OPTions

`:DIAGnostic[:CPU]:INFOrmation:OPTions?`

This query returns a list of options installed in the signal generator.

Key Entry   **Options Info**

:OPTions:DETail

`:DIAGnostic[:CPU]:INFOrmation:OPTions:DETail?`

This query returns the options installed, option revision, and digital signal processing (DSP) version if applicable.

Key Entry   **Options Info**

:OTIMe

`:DIAGnostic[:CPU]:INFOrmation:OTIMe?`

This query returns the cumulative number of hours that the signal generator has been on.

Key Entry   **Diagnostic Info**

:REVision

`:DIAGnostic[:CPU]:INFOrmation:REVision?`

This query returns the CPU bootstrap read only memory (boot ROM) revision date. In addition, the query returns the revision, creation date, and creation time for the firmware.

Key Entry   **Diagnostic Info**

:SDATe

`:DIAGnostic[:CPU]:INFOrmation:SDATe?`

This query returns the date and time stamp for the signal generator's firmware.

Key Entry   **Diagnostic Info**

Display Subsystem (:DISPlay)

:ANNotation:AMPLitude:UNIT

`:DISPlay:ANNotation:AMPLitude:UNIT DBM|DBUV|DBUVEMF|V|VEMF|DB`
`:DISPlay:ANNotation:AMPLitude:UNIT?`

This command sets the displayed front panel amplitude units.

If the amplitude reference state is set to on, the query returns units expressed in dB. Setting any other unit will cause a setting conflict error stating that the amplitude reference state must be set to off. Refer to for more information.

Example

`:DISP:ANN:AMPL:UNIT DB`

The preceding example sets DB as the amplitude units shown on the signal generator's front panel display.

*RST   **dBm**

:ANNotation:CLOCk:DATE:FORMat

`:DISPlay:ANNotation:CLOCk:DATE:FORMat MDY|DMY`
`:DISPlay:ANNotation:CLOCk:DATE:FORMat?`

This command selects the date format. The choices are month-day-year (MDY) or day-month-year (DMY) format. The date is shown on the signal generator's front panel display.

The setting enabled by this command is not affected by a signal generator power-on, preset, or *RST command.

Example

`:DISP:ANN:CLOC:DATA:FORM DMY`

The preceding example sets the date format shown on the signal generator's front panel display to DMY.

:ANNotation:CLOCk[:STATe]

`:DISPlay:ANNotation:CLOCk[:STATe] ON|OFF|1|0`
`:DISPlay:ANNotation:CLOCk[:STATe]?`

This command enables or disables the digital clock shown at the lower right side of the front panel display.

The setting enabled by this command is not affected by a signal generator power-on, preset, or `*RST` command.

Example

`:DISP:ANN:CLOC OFF`

The preceding example disables the digital clock on the signal generator's front panel display.

:BRIGhtness

`:DISPlay:BRIGhtness <val>`
`:DISPlay:BRIGhtness?`

This command sets the display brightness (intensity). The brightness can be set to the minimum level (0.02), maximum level (1), or in between by using fractional numeric values (0.03–0.99).

The setting enabled by this command is not affected by a signal generator power-on, preset, or `*RST` command.

Example

`:DISP:BRIG .45`

The preceding example sets display intensity to .45.

Range   0.02–1

Key Entry   Brightness

:CAPTure

```
:DISPlay:CAPTure
```

This command allows the user to capture the current display and store it in the signal generator's memory.

The display capture is stored as DISPLAY.BMP in the Binary file system. This file is overwritten with each subsequent display capture. The file can be down-loaded in the following manner:

1. Log on to the signal generator using file transfer protocol (FTP).

2. Change to the BIN directory using the FTP `cd` command.

3. Retrieve the file by using the FTP `get` command.

:CONTrast

```
:DISPlay:CONTrast <val>
```
```
:DISPlay:CONTrast?
```

This command sets the contrast for the signal generator's display. The variable <val> is expressed as a fractional number between 0 and 1. The contrast can be set to the maximum level (1), minimum level (0), or in between by using fractional numeric values (0.001–0.999).

The setting enabled by this command is not affected by a signal generator power-on, preset, or *RST command.

Example

```
:DISP:CONT .45
```

The preceding example sets the display contrast to .45.

Range   0–1

Key Entry   Display contrast hardkeys are located below the display.

:INVerse

```
:DISPlay:INVerse ON|OFF|1|0
```
```
:DISPlay:INVerse?
```

This command sets the display of the source to inverse video mode. The setting enabled by this command is not affected by a signal generator power-on, preset, or *RST command.

Example

```
:DISP:INV OFF
```

The preceding example sets the display video to normal (not inverse).

Key Entry   Inverse Video Off On

:REMote

```
:DISPlay:REMote ON|OFF|1|0
:DISPlay:REMote?
```

This command enables or disables display updating when the signal generator is remotely controlled.

ON (1)   This choice updates the signal generator display so that you can see the settings change as the commands are executed, however, this will decrease the signal generator's response time.

OFF (0)   This choice turns off display updating which will optimizing the signal generator's response time.

The setting enabled by this command is not affected by signal generator preset or *RST command. However, cycling the signal generator power will reset it to zero.

Example

```
:DISP:REM 0
```

The preceding example turns off display updating.

Key Entry   Update in Remote Off On

Display Off On

```
:DISPlay[:WINDow][:STATe] ON|OFF|1|0
:DISPlay[:WINDow][:STATe]?
```

This command is used to either blank out (OFF or 0) the display screen or turn it on (ON or 1).

A signal generator preset, *RST command, or cycling the power will turn the display on.

Example

```
:DISP OFF
```

The preceding example blanks out the signal generator's display.

IEEE 488.2 Common Commands

*CLS

`*CLS`

The Clear Status (CLS) command clears the Status Byte register, the Data Questionable Event register, the Standard Event Status register, and the Standard Operation Status register.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

*ESE

`*ESE <val>`

This command enables bits in the Standard Event Enable register. Bits enabled and set in this register will set the Standard Event Status Summary bit (bit 5) in the Status Byte register. When bit 5 (decimal 32) in the Status Byte register is set, you can read the Standard Event register using the *ESR command and determine the cause.

The Standard Event Enable register state (bits enabled with this command) is not affected by signal generator preset or *RST. The register will be cleared when the signal generator is turned off unless the command *PSC is used before turning it off.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

Example

`*ESE 129`

This command enables bit 0 (decimal 1, Operation Complete) and bit 7 (decimal 128, Power On) in the Standard Event Status Enable register.

Range   0–255

*ESE?

`*ESE?`

This query returns the decimal sum of the enabled bits in the Standard Event Enable register.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

*ESR?

NOTE	This is a destructive read. The data in the register is latched until it is queried. Once queried, the data is cleared. Refer to the *Agilent Signal Generators Programming Guide* for more information.

`*ESR?`

This query returns the decimal sum of the bits set in the Standard Event register.

*IDN?

`*IDN?`

This query requests an identification string from the signal generator. The IDN string consists of the following information:

`<company_name>, <model_number>, <serial_number>, <firmware_revision>`

The identification information can be modified. Refer to ":IDN" on page 54 for more information.

Key Entry   **Diagnostic Info**

*OPC

`*OPC`

The Operation Complete (OPC) command sets bit 0 in the Standard Event register.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

*OPC?

`*OPC?`

The Operation Complete (OPC) query returns the ASCII character 1 in the Standard Event register indicating completion of all pending operations.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

*PSC

`*PSC ON|OFF|1|0`

The power-on Status Clear (PSC) command controls the automatic power-on clearing of the Service Request Enable register, the Standard Event Status Enable register, and the device-specific event enable registers.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

The setting enabled by this command is not affected by a signal generator power-on, preset, or `*RST` command.

| ON (1) | This choice enables the power-on clearing of the listed registers. |
| OFF (0) | This choice disables the clearing of the listed registers and they retain their status when a power-on condition occurs. |

Example

`*PSC ON`

This command clears all listed registers at power-on.

*PSC?

`*PSC?`

The power-on Status Clear (PSC) query returns the flag (1 or 0) setting as enabled by the *PSC command.

*RCL

`*RCL <reg>,<seq>`

The Recall (RCL) command recalls the state from the specified memory register <reg> in the specified sequence <seq>.

| **Range** | *registers*: 0–99 *Sequences*: 0–9 |
| **Key Entry** | **RECALL Reg**   **Select Seq:** |

*RST

`*RST`

The Reset (RST) command resets most signal generator functions to a factory-defined state.

Each command description in this reference shows the *RST value if the signal generator's setting is affected.

*SAV

`*SAV <reg>,<seq>`

The Save (SAV) command saves the state of the signal generator to the specified memory register <reg> of the specified sequence <seq>. Settings such as frequency, attenuation, power, and settings that do not survive a power cycle or an instrument reset can be saved. Data formats, list sweep values, table entries, and so forth are not stored. Only a reference to the data file name is saved. Refer to the *E8663B Analog Signal Generator User's Guide* and *Agilent Signal Generators Programming Guide* for more information on saving and recalling instrument states.

| **Range** | *registers*: 0–99 *Sequences*: 0–9 |
| **Key Entry** | **Save Reg**   **Save Seq[n] Reg[nn]** |

*SRE

`*SRE <val>`

The Service Request Enable (SRE) command enables bits in the Service Request Enable register. Bits enabled and set in this register will set bits in the Status Byte register.

The variable <val> is the decimal sum of the bits that are enabled. Bit 6 (value 64) is not available in this register and therefore cannot be enabled by this command. Because bit 6 is not available, entering values from 64 to 127 is equivalent to entering values from 0 to 63.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

The setting enabled by this command is not affected by signal generator preset or `*RST`. However, cycling the signal generator power will reset this register to zero.

Range   0–63, 128–191

*SRE?

`*SRE?`

The Service Request Enable (SRE) query returns the decimal sum of bits enabled in the Service Request Enable register. Bit 6 (decimal 64) is not available in this register.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

Range   0–63, 128–191

*STB?

`*STB?`

This command reads the decimal sum of the bits set in the Status Byte register.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

Range   0–255

*TRG

`*TRG`

The Trigger (TRG) command triggers the device if BUS is the selected trigger source, otherwise, *TRG is ignored. Refer to ":TRIGger[:SEQuence]:SOURce" on page 63 for more information on triggers.

*TST?

`*TST?`

The Self-Test (TST) query initiates the internal self-test and returns one of the following results:

0   This shows that all tests passed.

1   This shows that one or more tests failed.

Key Entry   **Run Complete Self Test**

*WAI

`*WAI`

The Wait-to-Continue (WAI) command causes the signal generator to wait until all pending commands are completed, before executing any other commands.

Memory Subsystem (:MEMory)

:CATalog:BINary

`:MEMory:CATalog:BINary?`

This command outputs a list of binary files. The return data will be in the following form:

`<mem_used>,<mem_free>{,"<file_listing>"}`

The signal generator will return the two memory usage parameters and as many file listings as there are files in the directory. Each file listing parameter will be in the following form:

`"<file_name,file_type,file_size>"`

Refer to "File Name Variables" on page 10 for information on the file name syntax.

Key Entry   **Binary**

:CATalog:LIST

`:MEMory:CATalog:LIST?`

This command outputs a list of List Sweep files. The return data will be in the following form:
`<mem_used>,<mem_free>{,"<file_listing>"}`

The signal generator will return the two memory usage parameters and as many file listings as there are files in the directory. Each file listing parameter will be in the following form:
`"<file_name,file_type,file_size>"`

Refer to "File Name Variables" on page 10 for information on the file name syntax.

Key Entry   **List**

:CATalog:STATe

`:MEMory:CATalog:STATe?`

This command outputs a list of state files. The return data will be in the following form:

`<mem_used>,<mem_free>{,"<file_listing>"}`

The signal generator will return the two memory usage parameters and as many file listings as there are files in the directory. Each file listing parameter will be in the following form:

`"<file_name,file_type,file_size>"`

Refer to "File Name Variables" on page 10 for information on the file name syntax.

Key Entry   State

:CATalog:UFLT

`:MEMory:CATalog:UFLT?`

This command outputs a list of user-flatness correction files. The return data will be in the following form:

`<mem_used>,<mem_free>{,"<file_listing>"}`

The signal generator will return the two memory usage parameters and as many file listings as there are files in the directory. Each file listing parameter will be in the following form:

`"<file_name,file_type,file_size>"`

Refer to "File Name Variables" on page 10 for information on the file name syntax.

Key Entry   User Flatness

:CATalog[:ALL]

`:MEMory:CATalog[:ALL]?`

This command outputs a list of all files in the memory subsystem. The return data is in the following form:
`<mem_used>,<mem_free>{,"<file_listing>"}`

The signal generator returns the two memory usage parameters and as many file listings as there are files in the memory subsystem. Each file listing parameter is in the following form:
`"<file_name,file_type,file_size>"`

See Table 2-1 on page 37 for file types, and "File Name Variables" on page 10 for syntax.

Key Entry   All

:COPY[:NAME]

`:MEMory:COPY[:NAME] "<src_name>","<dest_name>"`

This command copies the data from one file into another file. The file can use the same name if the specified directory is different. For example, if the file resides in non-volatile waveform memory (NVWFM) it can be copied, using the same name, to the signal generator's volatile memory (WFM1).

"<src_name>"   This variable names a file residing in memory that will be copied. For information on the file name syntax, see "File Name Variables" on page 10.

"<dest_name>"   This variable names the file that is a copy of the "<src_name>" file.

Example

:MEM:COPY "/USER/LIST/LIST1","/USER/LIST/test_LIST"

The preceding example copies the LIST1 file in the signal generator's /USER/LIST directory to a file named test_LIST and saves it in the same directory.

Key Entry   Copy File

:DELete:ALL

CAUTION	Using this command deletes all user files including binary, list, state, and flatness correction files, and any saved setups which use the front panel table editor. You cannot recover the files after executing this command.

:MEMory:DELete:ALL

This command clears the file system of all user files.

Key Entry   Delete All Files

:DELete:BINary

:MEMory:DELete:BINary

This command deletes all binary files.

Key Entry   Delete All Binary Files

:DELete:LIST

:MEMory:DELete:LIST

This command deletes all List files.

Key Entry   Delete All List Files

:DELete:STATe

:MEMory:DELete:STATe

This command deletes all state files.

Key Entry   Delete All State Files

:DELete:UFLT

`:MEMory:DELete:UFLT`

This command deletes all user-flatness correction files.

Key Entry   Delete All UFLT Files

:DELete[:NAME]

`:MEMory:DELete[:NAME] "<file_name>"`

This clears the user file system of "<file_name>".

Refer to "File Name Variables" on page 10 for information on the file name syntax.

Example

`:MEM:DEL "/USER/WAVEFORM/Test_Data"`

The preceding example deletes the file named Test_Data from the signal generator's non–volatile memory.

Key Entry   Delete File

:FREE[:ALL]

`:MEMory:FREE[:ALL]?`

This command returns the number of bytes left in the user file system.

Key Entry   All

:LOAD:LIST

`:MEMory:LOAD:LIST "<file_name>"`

This command loads a List Sweep file.

Example

`:MEM:LOAD:LIST "List_Data"`

The preceding example loads the file "List_Data" into volatile waveform memory.

Key Entry   Load From Selected File

:MOVE

```
:MEMory:MOVE "<src_file>","<dest_file>"
```

This command renames the src_file to dest_file in the signal generator's memory catalog.

Refer to "File Name Variables" on page 10 for information on the file name syntax.

Example

```
:MEM:MOV "NVWFM:Test_Data","NVWFM:New_Data"
```

The preceding example renames the file Test_Data to New_Data in the signal generator's non-volatile memory directory.

Key Entry   **Rename File**

:STATe:COMMent

```
:MEMory:STATe:COMMent <reg_num>,<seq_num>,"<comment>"
:MEMory:STATe:COMMent? <reg_num>,<seq_num>
```

This command lets you to add a descriptive comment to the saved instrument in the state register, <reg_num>,<seq_num>. Comments can be up to 55 characters long.

Example

```
:MEM:STAT:COMM 00,1, "ARB file using external reference"
```

The preceding example writes a descriptive comment to the state file saved in register 00, sequence 1.

Key Entry   **Add Comment To Seq[n] Reg[nn]**

:STORe:LIST

```
:MEMory:STORe:LIST "<file_name>"
```

This command stores the current list sweep data to a file.

Refer to "File Name Variables" on page 10 for information on the file name syntax.

Example

```
:MEM:STOR:LIST "Test_Data"
```

The preceding example writes list sweep data to a file named Test_Data and stores the file in the signal generator's non–volatile memory, List directory.

Key Entry   **Store To File**

Mass Memory Subsystem (:MMEMory)

:CATalog

```
:MMEMory:CATalog? "<msus>"
```

This command outputs a list of the files from the specified file system. The variable "<msus>" (mass storage unit specifier) represents a file system. The file systems and types are shown in Table 2-1.

Table 2-1

File System	File Type
BIN - Binary file	BIN
BIT	BIT
FIR - finite impulse response filter file	FIR
FSK - frequency shift keying modulation file	FSK
LIST - sweep list file	LIST
SHAPE - burst shape file	SHAP
STATE	STAT
USERFLAT - user-flatness file	UFLT
WFM1 - waveform file	WFM1

The return data will be in the following form: <mem_used>,<mem_free>{,"<file_listing>"}

The signal generator will return the two memory usage parameters and as many file listings as there are files in the specified file system. Each file listing will be in the following format:
"<file_name,file_type,file_size>"

Refer to "MSUS (Mass Storage Unit Specifier) Variable" on page 12 for information on the use of the "<msus>" variable.

Key Entry   Binary   List State User Flatness Fir Shape Bit FSK

:COPY

:MMEMory:COPY[:NAME] "<src_name>","<dest_name>"

This command copies the data from one file into another file. The file can use the same name if the specified directory is different. For example, if the file resides in non-volatile waveform memory (NVWFM) it can be copied, using the same name, to the signal generator's volatile memory (WFM1)

"<src_name>"   This variable names a file residing in memory that will be copied. For information on the file name syntax, see "File Name Variables" on page 10

"<dest_name>"   This variable names the file that is a copy of the "<src_name>" file.

Example

:MMEM:COPY "/USER/LIST/LIST1","/USER/LIST/test_LIST"

The preceding example copies the LIST1 file in the signal generator's /USER/LIST directory to a file named test_LIST and saves it in the same directory.

Key Entry   Copy File

:DELete[:NAME]

`:MMEMory:DELete[:NAME] "<file_name>",["<msus>"]`

This command clears the memory file system of "<file_name>" with the option of specifying the file system ["<msus>"] separately.

The variable "<msus>" (mass storage unit specifier) represents the file system. For a list of the file systems refer to Table 2-1 on page 37. Refer to "MSUS (Mass Storage Unit Specifier) Variable" on page 12 for information on the mass storage unit specifier.

If the optional variable "<msus>" is omitted, the file name needs to include the file system extension.

Refer to "File Name Variables" on page 10 for information on the file name syntax.

Example

`:MMEM:DEL "/USER/BIN/Test_Data"`
`:MMEM:DEL "Test_Data",":BIN"`

The preceding examples delete the file named Test_Data from the signal generator's USER/BIN directory. The first example uses the full file name path while the second example uses the "<msus>" specifier.

Key Entry   Delete File

:LOAD:LIST

`:MMEMory:LOAD:LIST "<file_name>"`

This command loads list data from the List file "<file_name>".

Refer to "File Name Variables" on page 10 for information on the file name syntax.

Example

`:MMEM:LOAD:LIST "Sweep_Data"`

The preceding example loads sweep configuration data from the Sweep_Data List file.

Key Entry   Load From Selected File

:MOVE

`:MMEMory:MOVE "<src_file>","<src_file_1>"`

This command renames the src_file to src_file_1 in the memory catalog.

Refer to "File Name Variables" on page 10 for information on the file name syntax and using quotes for different programming languages.

Example

`:MMEM:MOV "NVWFM:Test_Data","NVWFM:New_Data"`

The preceding example renames the file Test_Data to New_Data in the signal generator's non–volatile memory.

Key Entry   Rename File

:STORe:LIST

`:MMEMory:STORe:LIST "<file_name>"`

This command copies the current list sweep data to the "<file_name>" and saves it in the catalog of List files.

Refer to "File Name Variables" on page 10 for information on the file name syntax.

Example

`:MMEM:STOR:LIST "Sweep_Data"`

The preceding example stores the current list sweep data to the file Sweep_Data in the signal generator's catalog of List files.

Key Entry   Store To File

Output Subsystem (:OUTPut)

:BLANking:AUTO

`[:SOURce]:OUTPut:BLANking:AUTO ON|OFF|1|0`
`[:SOURce]:OUTPut:BLANking:AUTO?`

This command sets the state for automatic RF Output blanking. Blanking occurs when the RF output is momentarily turned off as the sweep transitions from one frequency segment (band) to another, allowing the signal to settle. Blanking also occurs during the retrace, so the signal can settle before the next sweep. In CW mode, blanking occurs whenever you change the frequency.

ON (1)   This choice activates the automatic blanking function. The signal generator determines the blanking occurrences for optimum performance.

OFF (0)   This choice turns off the automatic blanking function, which also sets the blanking state to off.

Example

`:OUTP:BLAN:AUTO 0`

The preceding example disables RF output blanking.

*RST   1

Key Entry   Output Blanking Off On Auto

:BLANking:[STATe]

```
[:SOURce]:OUTPut:BLANking[:STATe] ON|OFF|1|0
[:SOURce]:OUTPut:BLANking[:STATe]?
```

This command sets the state for RF Output blanking. Blanking occurs when the RF output is momentarily turned off as the sweep transitions from one frequency segment (band) to another, allowing the signal to settle. Blanking also occurs during the retrace, so the signal can settle before the next sweep. In CW mode, blanking occurs whenever you change the frequency.

ON (1)   This choice activates the blanking function. Blanking occurs on all frequency changes, including segment transitions and retrace

OFF (0)   This choice turns off the blanking function.

Example

```
:OUTP:BLAN:ON
```

The preceding example enables RF output blanking.

Key Entry   Output Blanking Off On Auto

:MODulation[:STATe]

```
:OUTPut:MODulation[:STATe] ON|OFF|1|0
:OUTPut:MODulation[:STATe]?
```

This command enables or disables the modulation of the RF output with the currently active modulation type(s). Most modulation types can be simultaneously enabled except FM with ΦM.

An annunciator on the signal generator always displays to indicate whether modulation is on or off.

Example

```
:OUTP:MOD 0
```

The preceding example disables RF modulation.

*RST   1

Key Entry   Mod On/Off

[:STATe]

```
:OUTPut[:STATe] ON|OFF|1|0
:OUTPut[:STATe]?
```

This command enables or disables the RF output. Although you can configure and engage various modulations, no signal is available at the RF OUTPUT connector until this command is executed.

An annunciator always displays on the signal generator to indicate whether the RF output is on or off.

Example

:OUTP ON

The preceding example turns on the signal generator's RF output.

*RST   0

Key Entry   RF On/Off

Status Subsystem (:STATus)

:OPERation:CONDition

:STATus:OPERation:CONDition?

This query returns the decimal sum of the bits in the Standard Operation Condition register. This register monitors signal generator functions such as sweeping, and measuring. For example, if a sweep is in progress (bit 3), a decimal 8 is returned with this query.

The data in this register is continuously updated and reflects current conditions.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

Range   0–32767

:OPERation:ENABle

:STATus:OPERation:ENABle <val>
:STATus:OPERation:ENABle?

This command enables bits in the Standard Operation Event Enable register. Bits enabled and set in this register will set the Operation Status Summary bit (bit 7) in the Status Byte register. When bit 7 in the Status Byte register is set, you can read the Standard Operation Event register to determine the cause.

The variable <val> is the sum of the decimal values of the bits that you want to enable.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

Example

:STAT:OPER:ENAB 43

This command enables bit 1 (decimal 2, Settling), bit 3 (decimal 8, Sweeping), and bit 5 (decimal 32, Waiting for Trigger) of the Standard Operation Event Enable register.

Range   0–32767

:OPERation:NTRansition

`:STATus:OPERation:NTRansition <val>`

`:STATus:OPERation:NTRansition?`

This command enables bits in the Standard Operation Negative Transition Filter register. A negative transition (1 to 0) of corresponding bits in the Standard Operation Condition register will pass through and be read by the Standard Operation Event register.

The variable <val> is the sum of the decimal values of the bits that you want to enable.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

Example

`:STAT:OPER:NTR 3`

This command enables bit 1 (decimal 2, Settling) in the Standard Operation Negative Transition Filter register.

Range   0–32767

:OPERation:PTRansition

`:STATus:OPERation:PTRansition <val>`
`:STATus:OPERation:PTRansition?`

This command enables bits in the Standard Operation Positive Transition Filter register. A positive transition (0 to 1) of corresponding bits in the Standard Operation Condition register will pass through and be read by the Standard Operation Event register.

The variable <val> is the sum of the decimal values of the bits that you want to enable.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

Example

`:STAT:OPER:PTR 3`

This command enables bit 1 (decimal 2, Settling) in the Standard Operation Positive Transition Filter register.

Range   0–32767

:OPERation[:EVENt]

NOTE	This is a destructive read. The data in the register is latched until it is queried. Once queried, the data is cleared.

`:STATus:OPERation[:EVENt]?`

This query returns the decimal sum of the bits in the Standard Operation Event register.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

Range   0–32767

:PRESet

:STATus:PRESet

This command presets all positive and negative transition filters, enable registers, and error/event queue enable registers.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

:QUEStionable:CALibration:CONDition

:STATus:QUEStionable:CALibration:CONDition?

This query returns the decimal sum of the bits in the Data Questionable Calibration Condition register. For example, if the DCFM or DCΦM zero calibration fails (bit 0), a value of 1 is returned.

The data in this register is continuously updated and reflects the current conditions.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

Range   0–32767

:QUEStionable:CALibration:ENABle

:STATus:QUEStionable:CALibration:ENABle <val>
:STATus:QUEStionable:CALibration:ENABle?

This command enables bits in the Data Questionable Calibration Event Enable register. Bits enabled and set in this register will set the Calibration Summary bit (bit 8) in the Data Questionable Condition register.

The variable <val> is the sum of the decimal values of the bits that you want to enable.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

Example

:STAT:QUES:CAL:ENAB 1

This command enables bit 0 (decimal 1, DCFM/DCΦM Zero Failure) in the Data Questionable Calibration Event Enable register.

Range   0–32767

:QUEStionable:CALibration:NTRansition

```
:STATus:QUEStionable:CALibration:NTRansition <val>
:STATus:QUEStionable:CALibration:NTRansition?
```

This command enables bits in the Data Questionable Calibration Negative Transition Filter register. A negative transition (1 to 0) of corresponding bits in the Data Questionable Calibration Condition register will pass through and be read by the Data Questionable Calibration Event register.

The variable <val> is the sum of the decimal values of the bits that you want to enable.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

Example

```
:STAT:OPER:NTR 3
```

This command enables bit 0 (decimal 1, DCFM/DCΦM Zero Failure) in the Data Questionable Calibration Negative Transition Filter register.

Range   0–32767

:QUEStionable:CALibration:PTRansition

```
:STATus:QUEStionable:CALibration:PTRansition <val>
:STATus:QUEStionable:CALibration:PTRansition?
```

This command enables bits in the Data Questionable Calibration Positive Transition Filter register. A positive transition (0 to 1) of corresponding bits in the Data Questionable Calibration Condition register will pass through and be read by the Data Questionable Calibration Event register.

The variable <val> is the sum of the decimal values of the bits that you want to enable.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

Example

```
:STAT:OPER:PTR 3
```

This command enables bit 0 (decimal 1, DCFM/DCΦM Zero Failure) in the Data Questionable Calibration Positive Transition Filter register.

Range   0–32767

:QUEStionable:CALibration[:EVENt]

NOTE	This is a destructive read. The data in the register is latched until it is queried. Once queried, the data is cleared.

```
:STATus:QUEStionable:CALibration[:EVENt]?
```

This command returns the decimal sum of the bits in the Data Questionable Calibration Event register.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

Range   0–32767

:QUEStionable:CONDition

```
:STATus:QUEStionable:CONDition?
```

This query returns the decimal sum of the bits in the Data Questionable Condition register. For example, if the internal reference oscillator oven is cold (bit 4), a value of 16 is returned.

The data in this register is continuously updated and reflects current conditions.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

Range   0–32767

:QUEStionable:ENABle

```
:STATus:QUEStionable:ENABle <val>
:STATus:QUEStionable:ENABle?
```

This command enables bits in the Data Questionable Event Enable register. Bits enabled and set in this register will set the Data Questionable Summary bit (bit 3) in the Status Byte register. When bit 3 in the Status Byte register is set, you can read the Data Questionable Event register to determine the cause.

The variable <val> is the sum of the decimal values of the bits that you want to enable.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

Example

```
:STAT:QUES:ENAB 8
```

This command enables bit 3 (decimal 8, the Power Summary bit), in the Data Questionable Event Enable register.

Range   0–32767

:QUEStionable:FREQuency:CONDition

```
:STATus:QUEStionable:FREQuency:CONDition?
```

This query returns the decimal sum of the bits in the Data Questionable Frequency Condition register. For example, if the 1 GHz internal reference clock is unlocked (bit 2), a value of 4 is returned.

The data in this register is continuously updated and reflects current conditions.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

Range   0–32767

:QUEStionable:FREQuency:ENABle

```
:STATus:QUEStionable:FREQuency:ENABle <val>
:STATus:QUEStionable:FREQuency:ENABle?
```

This command enables bits in the Data Questionable Frequency Event Enable register. Bits enabled and set in this register will set the Data Questionable Condition register bit 5.

The variable <val> is the sum of the decimal values of the bits that you want to enable.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

Example

```
:STAT:QUES:FREQ:ENAB 7
```

This command enables bit 0 (decimal 1, Synthesizer Unlocked), bit 1 (decimal 2, 10 MHz Reference Unlocked), and bit 2 (decimal 4, 1 GHz reference Unlocked) in the Data Questionable Frequency Event Enable register.

Range   0–32767

:QUEStionable:FREQuency:NTRansition

```
:STATus:QUEStionable:FREQuency:NTRansition <val>
:STATus:QUEStionable:FREQuency:NTRansition?
```

This command enables bits in the Data Questionable Frequency Negative Transition Filter register. A negative transition (1 to 0) of corresponding bits in the Data Questionable Frequency Condition register will pass through and be read by the Data Questionable Frequency Event register.

The variable <val> is the sum of the decimal values of the bits that you want to enable.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

Example

```
:STAT:QUES:FREQ:NTR 96
```

This command enables bit 5 (decimal 32, Sampler Loop Unlocked) and bit 6 (decimal 64, YO Loop Unlocked) in the Data Questionable Frequency Negative Transition Filter register.

Range   0–32767

:QUEStionable:FREQuency:PTRansition

```
:STATus:QUEStionable:FREQuency:PTRansition <val>
:STATus:QUEStionable:FREQuency:PTRansition?
```

This command enables bits in the Data Questionable Frequency Positive Transition Filter register. A positive transition (0 to 1) of corresponding bits in the Data Questionable Frequency Condition register will pass through and be read by the Data Questionable Frequency Event register.

The variable <val> is the sum of the decimal values of the bits that you want to enable.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

Example

```
:STAT:QUES:FREQ:PTR 8
```

This command enables bit 3 (decimal 8, Baseband 1 Unlocked) in the Data Questionable Frequency Positive Transition Filter register.

Range   0–32767

:QUEStionable:FREQuency[:EVENt]

CAUTION	This is a destructive read. The data in the register is latched until it is queried. Once queried, the data is cleared.

```
:STATus:QUEStionable:FREQuency[:EVENt]?
```

This query returns the decimal sum of the bits in the Data Questionable Frequency Event register.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

Range   0–32767

:QUEStionable:MODulation:CONDition

```
:STATus:QUEStionable:MODulation:CONDition?
```

This command returns the decimal sum of the bits in the Data Questionable Modulation Condition register. For example, if the modulation is uncalibrated (bit 4), a value of 16 is returned.

The data in this register is continuously updated and reflects current conditions.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

Range   0–32767

:QUEStionable:MODulation:ENABle

```
:STATus:QUEStionable:MODulation:ENABle <val>
:STATus:QUEStionable:MODulation:ENABle?
```

This command enables bits in the Data Questionable Modulation Event Enable register. Bits enabled and set in this register will set bit 7 in the Data Questionable Condition register.

The variable <val> is the sum of the decimal values of the bits that you want to enable.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

Example

```
:STAT:QUES:MOD:ENAB 20
```

This command enables bit 2 (decimal 4, Modulation 1 Overmod) and bit 4 (decimal 16, Modulation Uncalibrated) in the Data Questionable Modulation Event Enable register.

Range   0–32767

:QUEStionable:MODulation:NTRansition

```
:STATus:QUEStionable:MODulation:NTRansition <val>
:STATus:QUEStionable:MODulation:NTRansition?
```

This command enables bits in the Modulation Questionable Negative Transition Filter register. A negative transition (1 to 0) of corresponding bits in the Modulation Questionable Condition register will pass through and be read by the Modulation Questionable Event register.

The variable <val> is the sum of the decimal values of the bits that you want to enable.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

Example

```
:STAT:QUES:MOD:NTR 3
```

This command enables bit 0 (decimal 1, Modulation 1 Undermod) and bit 1 (decimal 2, Modulation 1 Overmod) in the Data Questionable Modulation Negative Transition Filter register.

Range   0–32767

:QUEStionable:MODulation:PTRansition

```
:STATus:QUEStionable:MODulation:PTRansition <val>
:STATus:QUEStionable:MODulation:PTRansition?
```

This command enables bits in the Data Questionable Modulation Positive Transition Filter register. A positive transition (0 to 1) of corresponding bits in the Data Questionable Modulation Condition register will pass through and be read by the Data Questionable Modulation Event register.

The variable <val> is the sum of the decimal values of the bits that you want to enable.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

Example

```
:STAT:QUES:MOD:PTR 3
```

This command enables bit 0 (decimal 1, Modulation 1 Undermod) and bit 1 (decimal 2, Modulation 1 Overmod) in the Data Questionable Modulation Positive Transition Filter register.

Range   0–32767

:QUEStionable:MODulation[:EVENt]

CAUTION	This is a destructive read. The data in the register is latched until it is queried. Once queried, the data is cleared.

```
:STATus:QUEStionable:MODulation[:EVENt]?
```

This query returns the decimal sum of the bits in the Data Questionable Modulation Event register.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

Range   0–32767

:QUEStionable:NTRansition

```
:STATus:QUEStionable:NTRansition <val>
:STATus:QUEStionable:NTRansition?
```

This command enables bits in the Data Questionable Negative Transition Filter register. A negative transition (1 to 0) of corresponding bits in the Data Questionable Condition register will pass through and be read by the Data Questionable Event register.

The variable <val> is the sum of the decimal values of the bits that you want to enable.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

Example

```
:STAT:QUES:MOD:NTR 3072
```

This command enables bit 10 (decimal 1024, Baseband is busy) and bit 11 (decimal 2048, Sweep Calculating) in the Data Questionable Negative Transition Filter register.

Range   0–32767

:QUEStionable:POWer:CONDition

```
:STATus:QUEStionable:POWer:CONDition?
```

This query returns the decimal sum of the bits in the Data Questionable Power Condition register. For example, if the RF output signal is unleveled (bit 1), a value of 2 is returned.

The data in this register is continuously updated and reflects current conditions.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

Range   0–32767

:QUEStionable:POWer:ENABle

```
:STATus:QUEStionable:POWer:ENABle <val>
:STATus:QUEStionable:POWer:ENABle?
```

This command enables bits in the Data Questionable Power Event Enable register. Bits enabled and set in this register will set bit 3 in the Data Questionable Condition register.

The variable <val> is the sum of the decimal values of the bits that you want to enable.

Refer to the *Agilent Signal Generators Programming Guide* for more information on programming the status registers.

Example

`:STAT:QUES:POW:ENAB 1`

This command enables bit 0 (decimal 1, Reverse Power Protection Tripped) in the Data Questionable Power Event Enable register.

Range   0–32767

:QUEStionable:POWer:NTRansition

```
:STATus:QUEStionable:POWer:NTRansition <val>
:STATus:QUEStionable:POWer:NTRansition?
```

This command enables bits in the Data Questionable Power Negative Transition Filter register. A negative transition (1 to 0) of corresponding bits in the Data Questionable Power Condition register will pass through and be read by the Data Questionable Power Event register.

The variable <val> is the sum of the decimal values of the bits that you want to enable.

See the *Agilent Signal Generators Programming Guide* for more information on programming the status register system.

Example

`:STAT:QUES:POW:NTR 1`

This command enables bit 0 (Reverse Power Protection Tripped) in the Data Questionable Power Negative Transition Filter register.

Range   0–32767

:QUEStionable:POWer:PTRansition

```
:STATus:QUEStionable:POWer:PTRansition <val>
:STATus:QUEStionable:POWer:PTRansition?
```

This command enables bits in the Data Questionable Power Positive Transition Filter register. A positive transition (0 to 1) of corresponding bits in the Data Questionable Power Condition register will pass through and be read by the Data Questionable Power Event register.

The variable <val> is the sum of the decimal values of the bits that you want to enable.

See the *Agilent Signal Generators Programming Guide* for more information on programming the status register system.

Example

`:STAT:QUES:POW:PTR 1`

This command enables bit 0 (decimal 1, Reverse Power Protection Tripped) in the Data Questionable Power Positive Transition Filter register.

Range   0–32767

:QUEStionable:POWer[:EVENt]

CAUTION	This is a destructive read. The data in the register is latched until it is queried. Once queried, the data is cleared.

`:STATus:QUEStionable:POWer[:EVENt]?`

This query returns the decimal sum of the bits in the Data Questionable Power Event register.

See the *Agilent Signal Generators Programming Guide* for more information on programming the status register system.

Range   0–32767

:QUEStionable:PTRansition

`:STATus:QUEStionable:PTRansition <val>`
`:STATus:QUEStionable:PTRansition?`

This command enables bits in the Data Questionable Positive Transition Filter register. A positive transition (0 to 1) of corresponding bits in the Data Questionable Condition register will pass through and be read by the Data Questionable Event register.

See the *Agilent Signal Generators Programming Guide* for more information on programming the status register system.

Example

`:STAT:QUES:PTR 8`

This command enables bit 3 (decimal 8, Power Summary) in the Data Questionable Positive Transition Filter register.

Range   0–32767

:QUEStionable[:EVENt]

CAUTION	This is a destructive read. The data in the register is latched until it is queried. Once queried, the data is cleared.

`:STATus:QUEStionable[:EVENt]?`

This query returns the decimal sum of the bits in the Standard Operation Event register.

See the *Agilent Signal Generators Programming Guide* for more information on programming the status register system.

Range   0–32767

System Subsystem (:SYSTem)

:CAPability

`:SYSTem:CAPability?`

This query returns the signal generator's capabilities and outputs the appropriate specifiers:
`(RFSOURCE WITH((AM|FM|PULM|PM|LFO)&(FSSWEEP|FLIST)&(PSSWEEP|PLIST)`
`&TRIGGER&REFERENCE))`

This is a list of the SCPI-defined basic functionality of the signal generator and the additional capabilities it has in parallel (a&b) and singularly (a|b).

:DATE

`:SYSTem:DATE <year>,<month>,<day>`
`:SYSTem:DATE?`

This command sets the date as shown in the lower right area of the signal generator display.

`<year>`   This variable requires a four digit integer.

The query returns the date in the following format: <+year>, <+month>, <+day>

Example

`:SYST:DATE 2004,12,15`

The preceding example sets the date.

Range   *<month>*: 1–12   *<day>*: 1–31

Key Entry   **Time/Date**

:ERRor[:NEXT]

`:SYSTem:ERRor[:NEXT]?`

This query returns the most recent error message from the signal generator error queue. If there are no error messages, the query returns the following output:

   `+0,"No error"`

When there is more than one error message, the query will need to be sent for each message.

The error messages are erased after being queried.

Key Entry   **Error Info   View Next Error Message**

:ERRor:SCPI[:SYNTax]

`:SYSTem:ERRor:SCPI[:SYNTax] ON|OFF|1|0`
`:SYSTem:ERRor:SCPI[:SYNTax]?`

This command allows you to turn on verbose error messages that point out where the SCPI parser generated an error. Use the ERRor[:NEXT] command to read any reported errors.

Example

```
:SYST:ERR:SCPI ON
```

The preceding example enables the SCPI command error report function.

*RST   0

:FILEsystem:SAFEmode

This command selects the safe mode for file handling. When safe mode is set to `OFF`, volatile waveform files can be edited and saved while the signal generator plays the file without signal interruption. However, it is possible with complex waveforms, for corruption of memory to occur which will be reported as an error on the front-panel display and require a reboot of the signal generator to resolve.

```
:SYSTem:FILEsystem:SAFEmode ON|OFF|1|0
:SYSTem:FILEsystem:SAFEmode?
```

The setting enabled by this command is not affected by a signal generator power-on, preset, or `*RST` command.

Example

```
:SYST:FILE:SAVE ON
```

The preceding example enables the safe mode setting and waveform files cannot be edited without signal disruption while the signal generator plays them.

:HELP:MODE

```
:SYSTem:HELP:MODE SINGle|CONTinuous
:SYSTem:HELP:MODE?
```

This command sets the help function mode of the signal generator.

SINGle   Help is provided only for the next key that you press.

CONTinuous   Help is provided for each key you press. In addition, the function of the key is executed.

The setting enabled by this command is not affected by a signal generator power-on, preset, or `*RST` command.

Example

```
:SYST:HELP:MODE CONT
```

The preceding example enables the Help system continuos mode.

Key Entry   **Help Mode Single Cont**

:IDN

```
:SYSTem:IDN "string"
```

This command modifies the identification string that the *IDN? query returns. Sending an empty string returns the query output of *IDN? to its factory-shipped setting. The maximum string length is 72 characters.

Modification of the *IDN? query output enables the signal generator to identify itself as another signal generator when used as a replacement.

The display diagnostic information, shown by pressing the **Diagnostic Info** softkey, is not affected by this command.

:LANGuage

```
:SYSTem:LANGuage "SCPI"|"8662"|"8663"|
:SYSTem:LANGuage?
```

This command sets the remote language for the signal generator.

SCPI   This choice provides compatibility for SCPI commands.

8662   This choice provides compatibility for the Agilent 8662A Synthesized Signal Generator. The 8662A is controlled only through a GPIB interface.

8663   This choice provides compatibility for the Agilent 8663A Synthesized Signal Generator. The 8663A is controlled only through a GPIB interface.

The setting enabled by this command is not affected by a power-on, preset, or *RST command.

For more information on supported SCPI commands and programming codes, refer to Chapter 5, "SCPI Command Compatibility," on page 131.

Example

```
:SYST:LANG "8662"
```

The preceding example enables the 8662 Synthesized Signal Generator language as the language used to control the signal generator.

Key Entry   **SCPI   8662A,8663A**

:PON:TYPE

```
:SYSTem:PON:TYPE PRESet|LAST
:SYSTem:PON:TYPE?
```

This command sets the defined conditions for the signal generator at power on.

PRESet   This choice sets the conditions to factory- or user-defined as determined by the choice for the preset type. Refer to ":PRESet:TYPE" on page 56 for selecting the type of preset.

LAST   This choice retains the settings at the time the signal generator was last powered down.

The setting enabled by this command is not affected by a signal generator power-on, preset, or *RST command.

NOTE	When LAST is selected, no signal generator interaction can occur for at least 3 seconds prior to cycling the power for the current settings to be saved.

Example

`:SYST:PON:TYPE PRES`

The preceding example sets the preset state for the signal generator to factory settings.

Key Entry   Power On Last Preset

:PRESet

`SYSTem:PRESet`

This command returns the signal generator to a set of defined conditions. It is equivalent to pressing the front panel **Preset** hardkey.

The defined conditions are either factory– or user–defined. Refer to for selecting the type of defined conditions.

Key Entry   Preset

:PRESet:ALL

`:SYSTem:PRESet:ALL`

This command sets all states of the signal generator back to their factory default settings, including states that are not normally affected by a signal generator power-on, preset, or *RST command.

:PRESet:LANGuage

`:SYSTem:PRESet:LANGuage "SCPI"│"8663"│"8663"│`
`:SYSTem:PRESet:LANGuage?`

This command sets the remote language that is available when the signal generator is preset.

SCPI	This choice provides compatibility for SCPI commands.
8662	This choice provides compatibility for 8662A series synthesized waveform generators, which are supported only through a GPIB interface.
8663	This choice provides compatibility for 8663A series synthesized waveform generators, which are supported only through a GPIB interface.

Example

`:SYST:PRES:LANG "8662"`

The preceding example selects 8662 Synthesized Signal Generator as the language used by the signal generator following an instrument preset.

*RST   "SCPI"

Key Entry   SCPI   8662A,8663A

:PRESet:PERSistent

`:SYSTem:PRESet:PERSistent`

This command sets the states that are not affected by a signal generator power-on, preset, or *RST command to their factory default settings.

Key Entry   Restore Sys Defaults

:PRESet:PN9

`:SYSTem:PRESet:PN9 NORMal|QUICk`
`:SYSTem:PRESet:PN9?`

This command sets the preset length of the PN9 sequence for personalities that require software PRBS generation.

NORMal   This choice produces a maximal length PN9 sequence.

QUICk   This choice produces a truncated (216 bits) PN9 sequence.

Example

`:SYST:PRES:PN9 NORMAL`

The preceding example selects a maximum length PN9 sequence.

*RST   NORM

Key Entry   PN9 Mode Preset Normal Quick

:PRESet:TYPE

`:SYSTem:PRESet:TYPE NORMal|USER`
`:SYSTem:PRESet:TYPE?`

This command toggles the preset state between factory- and user-defined conditions. Refer to :PRESet[:USER]:SAVE for saving the USER choice preset settings. The setting enabled by this command is not affected by a signal generator power-on, preset, or *RST command.

Example

`:SYST:PRES:TYPE USER`

The preceding example selects a user defined conditions for the signal generator preset state.

Key Entry   Preset Normal User

:PRESet[:USER]:SAVE

`:SYSTem:PRESet[:USER]:SAVE`

This command saves your user-defined preset conditions to a state file.

Only one user-defined preset file can be saved. Subsequent saved user–defined preset files will overwrite the previously saved file.

Key Entry   Save User Preset

:SECurity:DISPlay

```
:SYSTem:SECurity:DISPlay ON|OFF|1|0
:SYSTem:SECurity:DISPlay?
```

This command enables or disables the secure display mode.

On(1)   This selection turns the signal generator display back on, showing the current settings. Cycling the signal generator power also restores the display, however the current settings may change depending on the power-on configuration choice. See ":PON:TYPE" on page 54 for information on the power-on choices available.

OFF(0)   This selection blanks the signal generator's display, hiding the settings and disabling the front panel keys. While in this mode, the display shows `*** SECURE DISPLAY ACTIVATED ***`.

For more information about security functions, refer to the *E8663B Analog Signal Generator User's Guide*.

Example

```
:SYST:SEC:DISP OFF
```

The preceding example enables the secure display mode.

*RST   1

Range   N/A

Key Entry   Activate Security Display

:SECurity:ERASeall

```
:SYSTem:SECurity:ERASall
```

This command removes all user files, flatness correction files, and baseband generator files. In addition, all table editor files are returned to their original factory values.

This command differs from the :DELete:ALL command, which does not reset table editors to factory values. For more information about security functions, refer to the *E8663B Analog Signal Generator User's Guide*.

Key Entry   Erase All

:SECurity:LEVel

```
:SYSTem:SECurity:LEVel NONE|ERASe|OVERwrite|SANitize
:SYSTem:SECurity:LEVel?
```

This command selects the security level operation for the signal generator.

NONE   This selection causes the signal generator to reset to factory default settings.

ERASe   This selection removes all user files, table editor files, flatness correction files, and baseband generator files.

OVERwrite   This selection removes all user files, table editor files, flatness correction files, and baseband generator files. The memory is then overwritten with random data.

SRAM	All addressable locations will be overwritten with random characters.
Hard Disk	All addressable locations will be overwritten with random characters.
Flash Memory	The flash blocks will be erased.

SANitize   This selection removes all user files, table editor files, flatness correction files, and baseband generator files using the same techniques as the OVERwrite selection for SRAM and flash memory. For the hard disk, the signal generator overwrites all addressable locations with a single character, its complement, and then with a random character.

Once you select the security level, you must execute the command from :SECurity:LEVel:STATe to arm the security level.

NOTE	Once you select a security level and arm it, you cannot change the level.

For other cleaning and security operation descriptions, see ":SECurity:ERASeall" on page 57, ":SECurity:OVERwrite" on page 59, and ":SECurity:SANitize" on page 59. For more information about security functions, refer to the *E8663B Analog Signal Generator User's Guide*.

Example

`:SYST:SEC:LEV NONE`

The preceding example sets the secure mode so it resets the signal generator to factory settings after completing the security operation.

Key Entry   None   Erase   Overwrite   Sanitize

:SECurity:LEVel:STATe

CAUTION	Ensure that you select the security level prior to executing this command with the ON (1) selection. Once you enable the state, you cannot reduce the security level.

`:SYSTem:SECurity:LEVel:STATe ON|OFF|1|0`
`:SYSTem:SECurity:LEVel:STATe?`

This command arms and executes the current security level parameter.

On (1)   This selection arms and prevents any changes to the current security level. Refer to ":SECurity:LEVel" on page 57 for setting the security level.

OFF (0)   This selection performs the actions required for the current security level setting. Cycling the signal generator power also performs the same function.

For more information about security functions, refer to the *E8663B Analog Signal Generator User's Guide*.

Example

`:SYST:SEC:LEV:STAT ON`

The preceding example arms the secure mode selected with the SYSTem:SECurity:LEVel command.

Key Entry   **Enter Secure Mode**

:SECurity:OVERwrite

`:SYSTem:SECurity:OVERwrite`

This command removes all user files, table editor files values, flatness correction files, and baseband generator files. The memory is then overwritten with random data as described below. For more information about security functions, refer to the *E8663B Analog Signal Generator User's Guide*.

SRAM   All addressable locations will be overwritten with random characters.

HARD DISK   All addressable locations will be overwritten with random characters.

FLASH MEMORY   The flash blocks will be erased.

Key Entry   **Erase and Overwrite All**

:SECurity:SANitize

`:SYSTem:SECurity:SANitize`

This command removes all user files, table editor files values, flatness correction files, and baseband generator files. The memory is then overwritten with a sequence of data as described below. For more information about security functions, refer to the *E8663B Analog Signal Generator User's Guide*.

SRAM   All addressable locations will be overwritten with random characters.

HARD DISK   All addressable locations will be overwritten with a single character and then a random character.

FLASH MEMORY   The flash blocks will be erased.

Key Entry   **Erase and Sanitize All**

:SSAVer:DELay

`:SYSTem:SSAVer:DELay <val>`
`:SYSTem:SSAVer:DELay?`

This command sets the amount of time before the display light or display light and text is switched off. The time delay represents the time during which there is no signal generator front panel input. The variable `<val>` is a positive integer number, in hours. The setting enabled by this command is not affected by power-on, preset, or `*RST`. See ":SSAVer:MODE" on page 60 for selecting the screen saver mode.

Example

`:SYST:SSAV:DEL 2`

The preceding example sets two hours delay time for the screen saver mode.

Range   1–12

Key Entry   Screen Saver Delay:

:SSAVer:MODE

`:SYSTem:SSAVer:MODE LIGHt|TEXT`
`:SYSTem:SSAVer:MODE?`

This command toggles the screen saver mode between light only or light and text.

LIGHt   Enables only the light to turn off during the screen saver operation while leaving the text visible on the darkened screen.

TEXT   Enables both the display light and text to turn off during screen saver operation.

The setting is not affected by a signal generator power-on, preset, or `*RST` command.

Example

`:SYST:SSAV:MODE TEXT`

The preceding example sets the screen saver mode.

Key Entry   Screen Saver Mode

:SSAVer:STATe

`:SYSTem:SSAVer:STATe ON|OFF|1|0`
`:SYSTem:SSAVer:STATe?`

This command enables or disables the display screen saver. The setting enabled by this command is not affected by a signal generator power-on, preset, or `*RST` command.

Example

`:SYST:SSAV:STAT 1`

The preceding example enables the screen saver mode.

Key Entry   Screen Saver Off On

:TIME

```
:SYSTem:TIME <hour>,<minute>,<second>
:SYSTem:TIME?
```

This command sets the time displayed in the lower right area of the signal generator's display.

Range   <hour>: 0–23   <minute>: 0-59   <second>: 0-59

Example

```
:SYST:TIME 9,30,45
```

The preceding example sets the signal generator time to 09:30:45.

Key Entry   Time/Date

:VERSion

```
:SYSTem:VERSion?
```

This command returns the SCPI version number with which the signal generator complies.

Trigger Subsystem

:ABORt

```
:ABORt
```

This command causes the List or Step sweep in progress to abort. If INIT:CONT[:ALL] is set to ON, the sweep will immediately re-initiate. The pending operation flag affecting *OPC, *OPC?, and *WAI will undergo a transition once the sweep has been reset.

:INITiate:CONTinuous[:ALL]

```
:INITiate:CONTinuous[:ALL] ON|OFF|1|0
:INITiate:CONTinuous[:ALL]?
```

This command selects either a continuous or single List or Step sweep. Execution of this command does not affect a sweep in progress.

ON (1)   Selects continuous sweep where, after the completion of the previous sweep, the sweep restarts automatically, or waits for a trigger.

OFF (0)   This choice selects a single sweep. Refer to ":INITiate[:IMMediate][:ALL]" on page 62 for single sweep triggering information.

Example

```
:INIT:CONT ON
```

The preceding example enables the continuos mode for the sweep type.

*RST   0

Key Entry   Sweep Repeat Single Cont

:INITiate[:IMMediate][:ALL]

`:INITiate[:IMMediate][:ALL]`

This command either sets or sets and starts a single List or Step sweep, depending on the trigger type. The command performs the following:

- arms a single sweep when BUS, EXTernal, or KEY is the trigger source selection
- arms and starts a single sweep when IMMediate is the trigger source selection

This command is ignored if a sweep is in progress. See ":INITiate:CONTinuous[:ALL]" on page 61 for setting continuous or single sweep. See ":TRIGger[:SEQuence]:SOURce" on page 63 to select the trigger source.

In some atypical cases, the :INIT command could be ignored if it immediately follows an *OPC? command. If the :INIT command is ignored, then use a 10ms sleep function before sending the command.

Key Entry   Single Sweep

:TRIGger:OUTPut:POLarity

`:TRIGger:OUTPut:POLarity POSitive|NEGative`
`:TRIGger:OUTPut:POLarity?`

Sets the TTL signal level present at the TRIGGER OUT connector to either high (5 vdc) or low (0 vdc). The trigger out is asserted after the frequency and/or power is set while the sweep is waiting for its step trigger. In addition, the swept-sine sends a pulse to the TRIGGER OUT at the beginning of each sweep.

Example

`:TRIG:OUTP:POL NEG`

The preceding example enables the continuos mode as the sweep type.

*RST   POS

Key Entry   Trigger Out Polarity Neg Pos

:TRIGger[:SEQuence]:SLOPe

`:TRIGger[:SEQuence]:SLOPe POSitive|NEGative`

`:TRIGger[:SEQuence]:SLOPe?`

This command sets the polarity of the ramp or sawtooth waveform slope present at the TRIGGER IN connector that will trigger a List or Step sweep.

Example

`:TRIG:SLOP POS`

The preceding example sets a positive ramp slope.

*RST   POS

Key Entry   Trigger In Polarity Neg Pos

:TRIGger[:SEQuence]:SOURce

`:TRIGger[:SEQuence]:SOURce BUS|IMMediate|EXTernal|KEY`
`:TRIGger[:SEQuence]:SOURce?`

This command sets the sweep trigger source for a List or Step sweep.

BUS	This choice enables GPIB triggering using the `*TRG` or `GET` command or LAN and RS-232 triggering using the `*TRG` command.
IMMediate	This choice enables immediate triggering of the sweep event.
EXTernal	This choice enables the triggering of a sweep event by an externally applied signal at the TRIGGER IN connector.
KEY	This choice enables front-panel triggering by pressing the Trigger hardkey.

The wait for the BUS, EXTernal, or KEY trigger can be bypassed by sending the `:TRIGger[:SEQuence][:IMMediate]` command.

Example

`:TRIG:SOUR BUS`

The preceding example sets the sweep trigger source to BUS.

*RST   IMM

Key Entry   Bus   Free Run   Ext   Trigger Key

:TRIGger[:SEQuence][:IMMediate]

`:TRIGger[:SEQuence][:IMMediate]`

This event command causes an armed List or Step sweep to immediately start without the selected trigger occurring.

In some atypical cases, the `:TRIG` command could be ignored if it immediately follows an `*OPC?` command. If the `:TRIG` command is ignored, then use a 10ms sleep function before sending the command.

Unit Subsystem (:UNIT)

:POWer

`:UNIT:POWer DBM|DBUV|DBUVEMF|V|VEMF|DB`
`:UNIT:POWer?`

This command terminates an amplitude value in the selected unit of measure.

If the amplitude reference state is set to on, the query returns units expressed in dB. Setting any other unit will cause a setting conflict error stating that the amplitude reference state must be set to off. Refer to, for more information.

All power values in this chapter are shown with DBM as the unit of measure. If a different unit of measure is selected, replace DBM with the newly selected unit whenever it is indicated for the value.

Example

`:UNIT:POW DBM`

The preceding example selects dBm as the unit of amplitude measurement.

*RST	DBM						
Key Entry	dBm	dBuV	dBuVemf	mV	uV	mVemf	uVemf

3 Basic Function Commands

In the following sections, this chapter provides SCPI descriptions for subsystems dedicated to signal generator operations that are applicable to this signal generator:

Correction Subsystem ([:SOURce]:CORRection)

:FLATness:LOAD

[:SOURce]:CORRection:FLATness:LOAD "<file_name>"

This command loads a user-flatness correction file designated by the file name "<file_name>" variable. The file will be loaded from the signal generator's USERFLAT directory. The directory path does not need to be specified in the command. Refer to the *Agilent Signal Generators Programming Guide* for more information on flatness corrections.

For information on file name syntax, refer to "File Name Variables" on page 10.

Example

:CORR:FLAT:LOAD "Flatness_Data"

The preceding example loads a user flatness file named Flatness_Data from the signal generator's user flatness directory.

Key Entry   Load From Selected File

:FLATness:PAIR

[:SOURce]:CORRection:FLATness:PAIR <freq>,<corr>

This command adds or edits a frequency and amplitude correction pair. The maximum number of pairs or points that can be entered is 1601. Refer to the *Agilent Signal Generators Programming Guide* for more information on flatness corrections.

The <corr> variable is the power correction in dB.

Power and frequency ranges for different signal generator models and options are listed on page 92.

Example

:CORR:FLAT:PAIR 10MHZ,.1

The preceding example enters a frequency of 10 megahertz and a power of 0.1dB into the user flatness table.

*RST	*Option 503*: +.3000000000000E+10
	Option 509: +.90000000000000E+10
Range	*Option 503*: 100kHZ–3GHZ
	Option 509: 100kHZ–9GHZ

Key Entry   **Configure Cal Array**

:FLATness:POINts

[:SOURce]:CORRection:FLATness:POINts?

This query returns the number of points in the user-flatness correction file.

:FLATness:PRESet

CAUTION	Once this command is executed, correction data is overwritten; If needed, save the current correction data (see ":FLATness:STORe" on page 67).

[:SOURce]:CORRection:FLATness:PRESet

This command presets the user-flatness correction to a factory-defined setting that consists of one frequency point and one amplitude point with no corrections.

Key Entry   **Preset List**

:FLATness:STORe

[:SOURce]:CORRection:FLATness:STORe "<file_name>"

This command stores the current user-flatness correction data to a file named by the "<file_name>" variable. All user-flatness files are stored in the signal generator's USERFLAT directory. The directory path does not need to be specified in the command.

For information on file name syntax, refer to "File Name Variables" on page 10.

Example

:CORR:FLAT:STOR "New_Flat_data"

The preceding example stores the current user-flatness table entries in a file named "New_Flat_data".

Key Entry   Store To File

[:STATe]

[:SOURce]:CORRection[:STATe] ON|OFF|1|0
[:SOURce]:CORRection[:STATe]?

This command toggles the application of user-flatness corrections to the current signal generator power output.

Example

:CORR OFF

The preceding example turns off correction data.

*RST   0

Key Entry   Flatness Off On

Frequency Subsystem ([:SOURce])

:FREQuency:CHANnels:BAND

[:SOURce]:FREQuency:CHANnels:BAND NBASe|NMOBile|BPGSm|MPGSm|BEGSm|MEGSm|
BRGSm|MRGSm|BDCS|MDCS|BPCS|MPCS|B450|GM450|B480|M480|B850|M850|B8|M8|B15|M15|B
390|B420|B460|B915|M380|M410|M450|M870|PHS|DECT
[:SOURce]:FREQuency:CHANnels:BAND?

This command sets the frequency of the signal generator by specifying a frequency channel band. The frequency channel state must be enabled for this command to work. See
":FREQuency:CHANnels[:STATe]" on page 70.

NBASe	This choice selects Standard Base as the frequency band for NADC.
NMOBile	This choice selects Standard Mobile as the frequency band for NADC.
BPGSm	This choice selects P-Gsm 900 Base as the frequency band for GSM.
MPGSm	This choice selects P-Gsm 900 Mobile as the frequency band for GSM.
BEGSm	This choice selects E-Gsm 900 Base as the frequency band for GSM.
MEGSm	This choice selects E-Gsm 900 Mobile as the frequency band for GSM.
BRGSm	This choice selects R-Gsm 900 Base as the frequency band for GSM.

MRGSm — This choice selects R-Gsm 900 Mobile as the frequency band for GSM.
BDCS — This choice selects DCS 1800 Base as the frequency band for GSM.
MDCS — This choice selects DCS 1800 Mobile as the frequency band for GSM.
BPCS — This choice selects PCS 1900 Base as the frequency band for GSM.
MPCS — This choice selects PCS 1900 Mobile as the frequency band for GSM.
B450 — This choice selects Gsm 450 Base as the frequency band for GSM.
GM450 — This choice selects Gsm 450 Mobile as the frequency band for GSM.
B480 — This choice selects Gsm 480 Base as the frequency band for GSM.
M480 — This choice selects Gsm 480 Mobile as the frequency band for GSM.
B850 — This choice selects Gsm 850 Base as the frequency band for GSM.
M850 — This choice selects Gsm 850 Mobile as the frequency band for GSM.
B8 — This choice selects 800MHz Base as the frequency band for PDC.
M8 — This choice selects 800MHz Mobile as the frequency band for PDC.
B15 — This choice selects 1500MHz Base as the frequency band for PDC.
M15 — This choice selects 1500MHz Mobile as the frequency band for PDC.
B390 — This choice selects Base 390-400 as the frequency band for TETRA.
B420 — This choice selects Base 420-430 as the frequency band for TETRA.
B460 — This choice selects Base 460-470 as the frequency band for TETRA.
B915 — This choice selects Base 915-921 as the frequency band for TETRA.
M380 — This choice selects Mobile 380-390 as the frequency band for TETRA.
M410 — This choice selects Mobile 410-420 as the frequency band for TETRA.
M450 — This choice selects Mobile 450-460 as the frequency band for TETRA.
M870 — This choice selects Mobile 870-876 as the frequency band for TETRA.
PHS — This choice selects Standard PHS as the frequency band.
DECT — This choice selects Standard DECT as the frequency band.

Example

:FREQ:CHAN:BAND DECT

The preceding example sets the frequency band to standard DECT.

*RST — BPGS

Key Entry — P-GSM Base / E-GSM Base / R-GSM Base / DCS Base / PCS Base / GSM 450 Base / GSM 480 Base / GSM 850 Base

NADC Base / 800MHZ Base / 1500MHZ Base

Tetra Base 390/400 / Tetra Base 420/430 / Tetra Base 460/470

Tetra Base 915/921 / PHS Standard / DECT Standard

P-GSM Mobile / E-GSM Mobile / R-GSM Mobile / DCS Mobile

PCS Mobile / GSM 450 Mobile / GSM 480 Mobile / GSM 850 Mobile

NADC Mobile / 800MHZ Mobile / 1500MHZ Mobile

Tetra Mobile 380/390 / Tetra Mobile 410/420 / Tetra Mobile 450/460

Tetra Mobile 870/876

:FREQuency:CHANnels:NUMBer

[:SOURce]:FREQuency:CHANnels:NUMBer <number>
[:SOURce]:FREQuency:CHANnels:NUMBer?

This command sets the frequency of the signal generator by specifying a channel number of a given frequency band.

The channel band and channel state must be enabled for this command to work. Refer to ":FREQuency:CHANnels[:STATe]" on page 70.

Example

:FREQ:CHAN:NUMB 24

The preceding example sets the channel number to 24 for the current band.

*RST	+1	
Range	P-GSM Base/Mobile:	1–24
	E-GSM and R-GSM Base/Mobile:	1–1023
	DCS Base/Mobile:	512–885
	PCS Base/Mobile:	512–900
	GSM-450 Base/Mobile:	259–293
	GSM-480 Base/Mobile:	306–340
	GSM-850 Base/Mobile:	128–251
	NADC Base/Mobile:	1–1023
	800MHZ Base/Mobile:	0–640
	1500MHZ Base/Mobile:	0–960
	TETRA 380/390 Mobile:	3600–4000
	TETRA 390/4000 Base:	3600–4000
	TETRA 410/420 Mobile:	800–1200
	TETRA 420/430 Base:	800–1200
	TETRA 460/470: 2400 through 2800	2400–2800
	TETRA 870/876 Mobile:	600–640
	TETRA 915/921 Base:	600–940
	PHS Standard:	1–255
	DECT Standard:	0–9
Key Entry	**Channel Number**	

:FREQuency:CHANnels[:STATe]

[:SOURce]:FREQuency:CHANnels[:STATe] ON|OFF|1|0
[:SOURce]:FREQuency:CHANnels[:STATe]?

This command enables or disables the frequency channel and band selection. The signal generator frequency will be set to the channel frequency when the state is on. To set frequency channel bands refer to ":FREQuency:CHANnels:BAND" on page 67.

Example

:FREQ:CHAN ON

The preceding example turns on the frequency channel.

*RST   0

Key Entry   Freq Channels Off On

:FREQuency:FIXed

[:SOURce]:FREQuency:FIXed <val><units>
[:SOURce]:FREQuency:FIXed?

This command sets the signal generator output frequency. To set the frequency mode, see ":FREQuency:MODE" on page 71. For a listing of signal generator frequency and power specifications, refer to "[:LEVel][:IMMediate][:AMPLitude]" on page 92.

Example

:FREQ:FIX 1GHZ

The preceding example sets the signal generator frequency to 1 GHz.

*RST	*Option 503*: +.3000000000000E+10
	Option 509: +.90000000000000E+10
Range	*Option 503*: 100kHZ–3GHZ
	Option 509: 100kHZ–9GHZ
Key Entry	**Freq CW**

:FREQuency:MODE

[:SOURce]:FREQuency:MODE FIXed|CW|SWEep|LIST
[:SOURce]:FREQuency:MODE?

This command sets the frequency mode of the signal generator.

FIXed and CW   These choices are synonymous. Any currently running frequency sweeps are turned off, and the current CW frequency settings are used to control the output frequency.

To set the frequency in the CW frequency mode, see ":FREQuency[:CW]" on page 75.

To set the frequency in the fixed frequency mode, see ":FREQuency:FIXed" on page 70.

LIST   This choice selects the swept frequency mode. If sweep triggering is set to immediate along with continuous sweep mode, executing the command starts the LIST or STEP frequency sweep.

NOTE	To perform a frequency and amplitude sweep, you must also select LIST as the power mode (see ":MODE" on page 89).

Example

:FREQ:MODE LIST

The preceding example selects a list frequency sweep.

*RST	CW

Key Entry	Freq CW	Sweep Type	Freq	Off	Freq & Ampl

:FREQuency:MULTiplier

[:SOURce]:FREQuency:MULTiplier <val>
[:SOURce]:FREQuency:MULTiplier?

This command sets the multiplier for the signal generator carrier frequency. For any multiplier other than one, the MULT indicator is shown in the frequency area of the display. The multiplier value is used to multiply the signal generator's displayed frequency. The true frequency remains constant. For example, if the signal generator frequency is 2 GHz and a multiplier of 3 is selected, the displayed frequency will be 6 GHz. This feature is useful when working with mixers and multipliers.

Example

:FREQ:MULT 2

The preceding example sets the carrier multiplier to 2.

*RST   +1.00000000E+000

Key Entry   **Freq Multiplier**

:FREQuency:OFFSet

[:SOURce]:FREQuency:OFFSet <val><units>
[:SOURce]:FREQuency:OFFSet?

This command sets the frequency offset. When an offset has been entered, the OFFS indicator appears in the frequency area of the signal generator's front-panel display and the frequency reading will include the offset value.

When any non-zero value is entered, the frequency offset state turns on; entering zero turns it off. To set the offset state independent of entering offset values see :FREQuency:OFFSet:STATe.

Example

:FREQ:OFFS 1GHZ

The preceding example sets the frequency offset to 1 GHz.

*RST   +0.0000000000000E+00

Range   -200GHZ to 200GHZ

Key Entry   **Freq Offset**

:FREQuency:OFFSet:STATe

[:SOURce]:FREQuency:OFFSet:STATe ON|OFF|1|0
[:SOURce]:FREQuency:OFFSet:STATe?

This command enables or disables the offset frequency.

Entering OFF (0) will set the frequency offset to 0 Hz.

Example

:FREQ:OFFS:STAT 0

The preceding example disables the frequency offset and sets the offset to 0 hertz.

*RST   0

Key Entry   **Freq Offset**

:FREQuency:REFerence

```
[:SOURce]:FREQuency:REFerence <val><units>
[:SOURce]:FREQuency:REFerence?
```

This command sets the output reference frequency for the signal generator. Once the reference frequency is set, any change to the signal generator's CW frequency will be displayed referenced to 0 hertz. For example, if the signal generator's CW frequency is set to 100 megahertz and the frequency reference is set (the frequency reference state will automatically turn on). The frequency display will read 0 Hz. If you change the signal generator's CW frequency to 1 megahertz, the frequency display will read 1 megahertz. However, the true frequency is 101 megahertz. This can be verified by turning the frequency reference state off. The signal generator frequency display will read 101 megahertz. Refer to :FREQuency:REFerence:STATe for more information.

Example

:FREQ:REF 100MHZ

The preceding example sets the output reference frequency to 100 megahertz.

*RST   +0.0000000000000E+00

Key Entry   Freq Ref Set

:FREQuency:REFerence:SET

```
[:SOURce]:FREQuency:REFerence:Set
```

This command sets the current CW output frequency, along with any offset, as a 0 hertz reference value.

*RST   +0.0000000000000E+00

Key Entry   Freq Ref Set

:FREQuency:REFerence:STATe

```
[:SOURce]:FREQuency:REFerence:STATe ON|OFF|1|0
[:SOURce]:FREQuency:REFerence:STATe?
```

This command enables or disables the frequency reference mode. When the frequency reference mode is on, changes in the signal generator's CW frequency are displayed relative to the 0 hertz frequency reference. When the state is off, the front-panel display indicates the true signal generator frequency.

Example

:FREQ:REF:STAT OFF

The preceding example turns off the reference frequency mode.

*RST   0

Key Entry   Freq Ref Off On

:FREQuency:STARt

```
[:SOURce]:FREQuency:STARt <val><units>
[:SOURce]:FREQuency:STARt?
```

This command sets the frequency start point for a step sweep.

Refer to "[:LEVel][:IMMediate][:AMPLitude]" on page 92 for frequency and power specifications for different signal generator options and model numbers.

Example

:FREQ:STAR 1GHZ

The preceding example sets the start frequency for a sweep to 1 GHz.

*RST	*Option 503*: +.3000000000000E+10	
	Option 509: +.90000000000000E+10	
Range	*Option 503*: 100kHZ–3GHZ	
	Option 509: 100kHZ–9GHZ	
Key Entry	**Freq Start**	

:FREQuency:STOP

```
[:SOURce]:FREQuency:STOP <val><units>
[:SOURce]:FREQuency:STOP?
```

This command sets the stop frequency for a step sweep.

Refer to "[:LEVel][:IMMediate][:AMPLitude]" on page 92 for frequency and power specifications for different signal generator options and model numbers.

Example

:FREQ:STOP 1GHZ

The preceding example sets the stop frequency for a sweep to 1 GHz.

*RST	*Option 503*: +.3000000000000E+10	
	Option 509: +.90000000000000E+10	
Range	*Option 503*: 100kHZ–3GHZ	
	Option 509: 100kHZ–9GHZ	
Key Entry	**Freq Stop**	

:FREQuency:SYNThesis

```
[:SOURce]:FREQuency:SYNThesis 1|2
[:SOURce]:FREQuency:SYNThesis?
```

This command sets the phase-lock loop (PLL) bandwidth to optimize phase noise for offsets above and below 10 kHz.

1	This choice will select mode 1 which optimizes phase noise at offsets below 10 kHz.
2	This choice will select mode 2 which optimizes phase noise at offsets above 10 kHz.

Example

:FREQ:SYNT 2

The preceding example sets PLL bandwidth to mode 2.

*RST   +1

Key Entry   Mode 1 Optimize <10kHZ Offset   Mode 2 Optimize >10kHZ Offset

:FREQuency[:CW]

```
[:SOURce]:FREQuency[:CW] <val><unit>
[:SOURce]:FREQuency[:CW]?
```

This command sets the signal generator output frequency for the CW frequency mode.

To set the frequency mode to CW, refer to ":FREQuency:MODE" on page 71.

Example

:FREQ 2GHZ

The preceding example sets signal generator's output frequency to 2 GHz.

*RST   *Option 503*: +.3000000000000E+10

                Option 509: +.90000000000000E+10

Range   *Option 503*: 100kHZ–3GHZ

                Option 509: 100kHZ–9GHZ

Key Entry   **Frequency**

:PHASe:REFerence

```
[:SOURce]:PHASe:REFerence
```

This command sets the output phase reference to zero. Subsequent phase adjustments are set relative to the new reference.

Key Entry   **Phase Ref Set**

:PHASe[:ADJust]

```
[:SOURce]:PHASe[:ADJust] <val><unit>
[:SOURce]:PHASe[:ADJust]?
```

This command adjusts the phase of the modulating signal. The query returns values in radians.

Example

:PHAS 30DEG

The preceding example sets the phase of the modulating signal to 30 degrees relative to the previous phase setting.

*RST	+0.00000000E+000	
Range	*Radians*: –3.14 to 3.14RAD	*Degrees*: –180 to 179DEG

Key Entry   **Adjust Phase**

:ROSCillator:BANDwidth:DEFaults

[:SOURce]:ROSCillator:BANDwidth:DEFaults

This command resets the bandwidth of the reference oscillator to the factory-defined default state. The default value for the internal reference bandwidth is 125 Hz. The default value for the external reference bandwidth is 25 Hz.

Key Entry   **Restore Factory Defaults**

:ROSCillator:BANDwidth:EXTernal

[:SOURce]:ROSCillator:BANDwidth:EXTernal 25HZ|55HZ|125HZ|300HZ|650HZ
[:SOURce]:ROSCillator:BANDwidth:EXTernal?

This command sets the bandwidth of the external reference oscillator.

Example

:ROSC:BAND:EXT 300HZ

The preceding example sets the bandwidth of the external oscillator to 300 hertz.

Key Entry   **External Ref Bandwidth**

:ROSCillator:BANDwidth:INTernal

[:SOURce]:ROSCillator:BANDwidth:INTernal 25HZ|55HZ|125HZ|300HZ|650HZ
[:SOURce]:ROSCillator:BANDwidth:INTernal?

This command sets the bandwidth of the internal reference oscillator.

Example

:ROSC:BAND:INT 125HZ

The preceding example sets the bandwidth of the internal oscillator to 125 hertz.

Key Entry   **Internal Ref Bandwidth**

:ROSCillator:SOURce

[:SOURce]:ROSCillator:SOURce?

This command queries the reference oscillator source: INT (internal) or EXT (external).

:ROSCillator:SOURce:AUTO

```
[:SOURce]:ROSCillator:SOURce:AUTO ON|OFF|1|0
[:SOURce]:ROSCillator:SOURce:AUTO?
```

This command enables or disables the ability of the signal generator to automatically select between the internal and an external reference oscillator.

ON (1)   This choice enables the signal generator to detect when a valid reference signal is present at the 10 MHz IN connector and automatically switches from internal to external frequency reference.

OFF (0)   This choice selects the internal reference oscillator and disables the switching capability between the internal and an external frequency reference.

Example

:ROSC:SOUR:AUTO 0

The preceding example turns off the automatic selection of internal or external reference oscillators.

*RST   1

Key Entry   Ref Oscillator Source Auto Off On

List/Sweep Subsystem ([:SOURce])

A complete sweep setup requires commands from other subsystems. Table 3-1 shows the function and location of these commands.

Table 3-1   Location of Commands from the other Subsystems

Sweep Type	Function	Command Location	Key Entry under Sweep/List key
List and Step	Start/stop frequency sweep	":FREQuency:MODE" (page 71)	Freq Off
	Start/stop amplitude sweep	":MODE" (page 89)	Ampl Off
	Start/stop frequency and amplitude sweep[a]	":MODE" (page 89) ":FREQuency:MODE" (page 71)	Freq & Ampl Off
	Set up & control sweep triggering[b]	"Trigger Sweep Subsystem ([:SOURce])" (page 92)	See the Trigger Sweep Subsystem ([:SOURce])
Step	Start frequency sweep	":FREQuency:STARt" (page 73)	Freq Start
	Stop frequency sweep	":FREQuency:STOP" (page 74)	Freq Stop
	Start amplitude sweep	":STARt" (page 91)	Ampl Start
	Stop amplitude sweep	":STOP" (page 91)	Ampl Stop

a. Execute both commands to start or stop a frequency and amplitude sweep.
b. For point to point triggering, see ":LIST:TRIGger:SOURce" on page 81.

:LIST:DIRection

[:SOURce]:LIST:DIRection UP|DOWN

[:SOURce]:LIST:DIRection?

This command sets the direction of a list or step sweep.

UP   This choice enables a sweep in an ascending order:

- first to last point for a list sweep

- start to stop for a step sweep

DOWN   This choice reverses the direction of the sweep.

Example

:LIST:DIR UP

The preceding example selects an ascending sweep direction.

*RST   UP

Key Entry   Sweep Direction Down Up

:LIST:DWELl

[:SOURce]:LIST:DWELl <val>{,<val>}

[:SOURce]:LIST:DWELl?

This command sets the dwell time for points in the current list sweep.

The variable <val> is expressed in units of seconds with a 0.001 resolution. If only one point is specified, that value is used for all points in the list. Otherwise, there must be a dwell point for each frequency and amplitude point in the list.

NOTE   The dwell time <val> does not begin until the signal generator frequency and/or amplitude change has settled.

Dwell time is used when IMMediate is the trigger source. Refer to ":LIST:TRIGger:SOURce" on page 81 for the trigger setting.

The dwell time is the amount of time the sweep pauses after setting the frequency and/or power for the current point.

The setting enabled by this command is not affected by a signal generator power cycle, preset, or *RST command.

Example

:LIST:DWEL .1,.2,.1,.2,.3

The preceding example sets the dwell time for a list of five points.

Range   0.001–60

:LIST:DWELl:POINts

[:SOURce]:LIST:DWELl:POINts?

This command queries the signal generator for the number of dwell points in the list sweep file.

:LIST:DWELl:TYPE

[:SOURce]:LIST:DWELl:TYPE LIST|STEP
[:SOURce]:LIST:DWELl:TYPE?

This command toggles the dwell time for the list sweep points between the values defined in the list sweep and the value for the step sweep.

LIST   This choice selects the dwell times from the list sweep. Refer to ":LIST:DWELl" on page 78 for setting the list dwell points.

STEP   This choice selects the dwell time from the step sweep. Refer to ":SWEep:DWELl" on page 83 for setting the step dwell.

Example

:LIST:DWEL:TYPE STEP

The preceding example selects the dwell time from step sweep values.

*RST   LIST

Key Entry   **Dwell Type List Step**

:LIST:FREQuency

[:SOURce]:LIST:FREQuency <val>{,<val>}
[:SOURce]:LIST:FREQuency?

This command sets the frequency values for the current list sweep points. The maximum number of points is 1601. The setting enabled by this command is not affected by a signal generator power-on, preset, or *RST command.

The variable <val> is expressed in hertz.

For signal generator frequency and power specifications, refer to "[:LEVel][:IMMediate][:AMPLitude]" on page 92.

Example

:LIST:FREQ 1GHZ,2GHZ,3GHZ,6GHZ

The preceding example sets the frequency value for a four point sweep.

*RST   *Option 503*: +.3000000000000E+10

                     Option 509: +.90000000000000E+10

Range   *Option 503*: 100kHZ–3GHZ

                     Option 509: 100kHZ–9GHZ

:LIST:FREQuency:POINts

[:SOURce]:LIST:FREQuency:POINts?

This command queries the current list sweep file for the number of frequency points.

:LIST:MANual

[:SOURce]:LIST:MANual <val> |UP|DOWN
[:SOURce]:LIST:MANual?

This command selects a list point or step sweep point as the current sweep point controlling the frequency and power output. If list or step mode is controlling frequency or power, or both, the indexed point in the respective list(s) is used.

The MANual mode must be selected and sweep enabled for this command to have an effect.

For information on setting the proper mode, see :LIST:MODE.

If the point selected is beyond the length of the longest enabled list, the point sets to the maximum possible point, and an error is generated.

Example

:LIST:MAN UP

The preceding example selects the next positive–direction, sequential point in the list.

Range   List Sweep: 1– 1601
                   Step Sweep: 1– 65535

Key Entry   Manual Point

:LIST:MODE

[:SOURce]:LIST:MODE AUTO|MANual
[:SOURce]:LIST:MODE?

This command sets the operating mode for the current list or step sweep.

AUTO   This choice enables the selected sweep type to perform a sweep of all points.

MANual   This choice enables you to select an individual sweep point to control the RF output parameters. For more about selecting a sweep point, see ":LIST:MANual" on page 80.

Example

:LIST:MODE AUTO

The preceding example sets the mode to automatic.

*RST   AUTO

Key Entry   Manual Mode Off On

:LIST:POWer

[:SOURce]:LIST:POWer <val>{,<val>}

[:SOURce]:LIST:POWer?

This command sets the amplitude for the current list sweep points.

The setting enabled by this command is not affected by signal generator power-on, preset, or *RST.

During an amplitude sweep operation, the signal generator protects the step attenuator by automatically switching to attenuator hold mode (OFF). The attenuator locks at its current setting and the amplitude sweep range is limited to 40 dB. The maximum number of points is 1601.

Example

:LIST:POW .1,.2,.1,.3,.1,-.1

The preceding example sets the power level for a six point sweep list.

Range   See "[:LEVel][:IMMediate][:AMPLitude]" on page 92.

:LIST:POWer:POINts

[:SOURce]:LIST:POWer:POINts?

This command queries the number of power points in the current list sweep file.

:LIST:RETRace

[:SOURce]:LIST:RETRace ON|OFF|1|0

[:SOURce]:LIST:RETRace?

Upon completion of a single sweep operation, this command either resets the sweep to the first sweep point, or leaves it at the last sweep point. The command is valid for the list or step single-sweep modes.

ON (1)   The sweep resets to the first sweep point.

OFF (0)   The sweep stays at the last sweep point.

Example

:LIST:RETR 1

The preceding example sets the retrace on. The sweep will reset to the first point after completing a sweep.

*RST   1

Key Entry   Sweep Retrace Off On

:LIST:TRIGger:SOURce

[:SOURce]:LIST:TRIGger:SOURce BUS|IMMediate|EXTernal|KEY

[:SOURce]:LIST:TRIGger:SOURce?

This command sets the trigger source for a list or step sweep event.

To set the sweep trigger, see ":TRIGger[:SEQuence]:SOURce" on page 102.

BUS	This choice enables GPIB triggering using the *TRG or GET command, or LAN and RS-232 triggering using the *TRG command.
IMMediate	This choice enables immediate triggering of the sweep event.
EXTernal	This choice enables the triggering of a sweep event by an externally applied signal at the TRIGGER IN connector.
KEY	This choice enables triggering by pressing the front-panel **Trigger** hardkey.

Example

:LIST:TRIG:SOUR BUS

The preceding example sets the trigger source to the instrument BUS.

*RST   IMM

Key Entry   **Bus**   **Free Run**   **Ext**   **Trigger Key**

:LIST:TYPE

[:SOURce]:LIST:TYPE LIST│STEP

[:SOURce]:LIST:TYPE?

This command selects the sweep type.

LIST	This type of sweep has arbitrary frequencies and amplitudes.
STEP	This type of sweep has equally spaced frequencies and amplitudes.

Example

:LIST:TYPE LIST

The preceding example selects list as the sweep type.

*RST   STEP

Key Entry   **Sweep Type List Step**

:LIST:TYPE:LIST:INITialize:FSTep

CAUTION	When you execute this command, the current list sweep data is overwritten. If needed, save the current data. For information on storing list sweep files, see":STORe:LIST" on page 57.

[:SOURce]:LIST:TYPE:LIST:INITialize:FSTep

This command replaces the loaded list sweep data with the settings from the current step sweep data points. You can have only one sweep list at a time.

The maximum number of list sweep points is 1,601. When copying the step sweep settings over to a list sweep, ensure that the number of points in the step sweep do not exceed the maximum list sweep points.

Key Entry   **Load List From Step Sweep**

:LIST:TYPE:LIST:INITialize:PRESet

CAUTION	When you execute this command, the current list sweep data is overwritten. If needed, save the current data. For information on storing list sweep files, see ":STORe:LIST" on page 57.

`[:SOURce]:LIST:TYPE:LIST:INITialize:PRESet`

This command replaces the current list sweep data with a factory-defined file consisting of one point at a frequency, amplitude, and dwell time.

Key Entry   Preset List

:SWEep:DWELl

`[:SOURce]:SWEep:DWELl <val>`

`[:SOURce]:SWEep:DWELl?`

This command enables you to set the dwell time for a step sweep.

The variable <val> is expressed in seconds with a 0.001 resolution.

NOTE	The dwell time <val> does not begin until the signal generator has settled for the current frequency and/or amplitude change.

Dwell time is used when the trigger source is set to IMMediate.
For the trigger setting, refer to ":LIST:TRIGger:SOURce" on page 81.

The dwell time is the amount of time the sweep pauses after setting the frequency or power, or both, for the current point.

Example

:SWE:DWEL .1

The preceding example sets the dwell time for a step sweep to 100 milliseconds.

*RST   +2.00000000E−003

Range   0.001−60S

Key Entry   Step Dwell

:SWEep:POINts

`[:SOURce]:SWEep:POINts <val>`

`[:SOURce]:SWEep:POINts?`

This command enables you to define the number of points in a step sweep.

Example

:SWE:POIN 2001

The preceding example sets the number of step sweep points to 2001.

*RST   2

Range   2–65535

Key Entry   # Points

Power Subsystem ([:SOURce]:POWer)

:ALC:BANDwidth|BWIDth

```
[:SOURce]:POWer:ALC:BANDwidth|BWIDth <num>[<freq_suffix>]
[:SOURce]:POWer:ALC:BANDwidth|BWIDth?
```

This command sets the bandwidth of the automatic leveling control (ALC) loop. You can select bandwidths of 100 Hz, 1 kHz, 10 kHz, or 100kHz. If you do not specify one of these exact bandwidths, your entry rounds to the nearest acceptable value. Refer to the *E8663B Analog Signal Generator User's Guide* for information on ALC and bandwidth considerations.

Example

:POW:ALC:BWID 1KHZ

The preceding example sets the ALC bandwidth to 1 kHz.

*RST   100.0

Key Entry   ALC BW

:ALC:BANDwidth|BWIDth:AUTO

```
[:SOURce]:POWer:ALC:BANDwidth|BWIDth:AUTO ON|OFF|1|0
[:SOURce]:POWer:ALC:BANDwidth|BWIDth:AUTO?
```

This command sets the state of the automatic leveling control (ALC) automatic bandwidth function. When this state is turned on, the signal generator automatically selects the optimum bandwidth for the ALC.

Example

:POW:ALC:BWID:AUTO 0

The preceding example disables the automatic bandwidth optimizing function.

*RST   1

Key Entry   ALC BW

:ALC:LEVel

```
[:SOURce]:POWer:ALC:LEVel <value>DB
[:SOURce]:POWer:ALC:LEVel?
```

This command sets the automatic leveling control (ALC) level when the attenuator hold is active.

Use this command when the automatic attenuation mode is set to OFF (0). Refer to ":ATTenuation:AUTO" on page 88 for choosing the attenuator mode.

Example

:POW:ALC:LEV 10DB

The preceding example sets the ALC to 10 dB.

*RST	+1.00000000E+000
Range	-20 to 25
Key Entry	Set ALC Level

:ALC:SEARch

```
[:SOURce]:POWer:ALC:SEARch ON|OFF|1|0|ONCE
[:SOURce]:POWer:ALC:SEARch?
```

This command enables or disables the internal power search calibration. A power search is recommended for pulse-modulated signals with pulse widths less than one microsecond. Refer to the *E8663B Analog Signal Generator User's Guide* for more information on ALC and the power search function.

ON (1)   This choice executes the power search automatically with each change in RF frequency or power.

OFF (0)   This choice disables the automatic power search routine.

ONCE   This choice executes a single power search of the current RF output signal.

Use this command when the automatic leveling control (ALC) state is set to OFF (0). Refer to ":ALC[:STATe]" on page 88 for setting the ALC state.

If ON was previously selected, executing ONCE will cause OFF to be the current selection after the power search is completed.

Example

:POW:ALC:SEAR ONCE

The preceding example starts a single power search of the RF output signal.

*RST	0	
Key Entry	Power Search Manual Auto	Do Power Search

:ALC:SEARch:REFerence

```
[:SOURce]:POWer:ALC:SEARch:REFerence FIXed|MODulated
[:SOURce]:POWer:ALC:SEARch:REFerence?
```

This command sets either fixed or modulated modes for power search.

FIXed   This choice uses a 0.5 volt reference.

MODulated   This choice uses the RMS value of the current I/Q modulation as measured during
                   the power search.

Example

:POW:ALC:SEAR:REF FIX

The preceding example selects a fixed voltage as the reference for a power search.

*RST   MOD

Key Entry   **Power Search Reference Fixed Mod**

:ALC:SEARch:SPAN:START

[:SOURce]:POWer:ALC:SEARch:SPAN:START <val><units>
[:SOURce]:POWer:ALC:SEARch:SPAN:START?

This command sets the start frequency for a power search over a user-defined range. The start
frequency has no default value. The start frequency value will be set before powering off the
instrument.

Example

:POW:ALC:SEAR:SPAN:START 2GHZ

The preceding example selects 2 GHz as the start frequency for a power search.

Key Entry   **Start Frequency**

:ALC:SEARch:SPAN:STOP

[:SOURce]:POWer:ALC:SEARch:SPAN:STOP <val><units>
[:SOURce]:POWer:ALC:SEARch:SPAN:STOP?

This command sets the stop frequency for a power search over a user-defined range. The stop
frequency has no default value. The stop frequency value will be set before powering off the
instrument

Example

:POW:ALC:SEAR:SPAN:STOP 3GHZ

The preceding example selects 3GHz as the stop frequency for a power search.

Key Entry   **Stop Frequency**

:ALC:SEARch:SPAN:TYPE FULL|USER

[:SOURce]:POWer:ALC:SEARch:SPAN:TYPE FULL|USER
[:SOURce]:POWer:ALC:SEARch:SPAN:TYPE?

This command enables you to select the frequency range for a power search. You can specify the
range (USER) or you can select the full range (FULL) of the signal generator.

Example

:POW:ALC:SEAR:SPAN:TYPE USER

The preceding example selects a user-defined frequency range for the power search.

Key Entry   Span Type User Full

:ALC:SEARch:SPAN[:STATe] ON|OFF|1|0

```
[:SOURce]:POWer:ALC:SEARch:SPAN[:STATe] ON|OFF|1|0
[:SOURce]:POWer:ALC:SEARch:SPAN[:STATe]?
```

This command enables (1) or disables (0) the span mode, allowing you to perform power searches over a selected range of frequencies. The power search corrections are then stored and used whenever the signal generator is tuned within the selected range.

Example

:POW:ALC:SEAR:SPAN ON

The preceding example enables the span mode.

:ALC:SOURce

```
[:SOURce]:POWer:ALC:SOURce INTernal|DIODe
[:SOURce]:POWer:ALC:SOURce?
```

This command enables you to select an automatic level control (ALC) source. You can select the internal ALC source or an external detector source. Refer to the *E8663B Analog Signal Generator User's Guide* for more information on ALC leveling, bandwidth, and the power search function.

Example

:POW:ALC:SOUR DIODe

The preceding example selects the diode detector as the ALC source detector.

*RST   INT

Key Entry   Leveling Mode

:ALC:SOURce:EXTernal:COUPling

```
[:SOURce]:POWer:ALC:SOURce:EXTernal:COUPling <value>DB
[:SOURce]:POWer:ALC:SOURce:EXTernal:COUPling?
```

This command sets the external detector coupling factor. Use this command when DIODe is the selected ALC source (). (0 to 32 coupling value).

Example

:POW:ALC:SOUR:EXT:COUP 20DB

The preceding example sets the external coupling factor to 20 dB.

*RST   +1.60000000E+001

Range   −200DB to 200DB.

Key Entry   Ext Detector Coupling Factor

:ALC[:STATe]

```
[:SOURce]:POWer:ALC[:STATe] ON|OFF|1|0
[:SOURce]:POWer:ALC[:STATe]?
```

This command enables or disables the automatic leveling control (ALC) circuit. The purpose of the ALC circuit is to hold output power at a desired level by adjusting the signal generator power circuits for power drift. Power will drift over time and with changes in temperature. Refer to the *E8663B Analog Signal Generator User's Guide* for more information on the ALC.

Example

:POW:ALC ON

The preceding example sets the ALC on.

*RST	1
Key Entry	**ALC Off On**

:ATTenuation

```
[:SOURce]:POWer:ATTenuation <val><unit>
[:SOURce]:POWer:ATTenuation?
```

This command sets the attenuation level when the attenuator hold is active. The attenuation is set to values of 0, 5, 15, 25, and continues in 5 dB increments.

The output power is the ALC level minus the attenuator setting.

Use this command when the automatic attenuation mode is set to OFF (0). Refer to for choosing the attenuator mode.

Example

:POW:ATT 10DB

The preceding example sets the attenuator to 10 dB.

*RST	+115
Range	0 to 115 dB
Key Entry	**Set Atten**

:ATTenuation:AUTO

```
[:SOURce]:POWer:ATTenuation:AUTO ON|OFF|1|0
[:SOURce]:POWer:ATTenuation:AUTO?
```

This command sets the state of the attenuator hold function.

ON (1)	This choice enables the attenuator to operate normally.
OFF (0)	This choice holds the attenuator at its current setting or at a selected value that will not change during power adjustments.

OFF (0) eliminates the power discontinuity normally associated with the attenuator switching during power adjustments. During an amplitude sweep operation, the signal generator protects the step attenuator by automatically switching to attenuator hold mode (ON). The attenuator is locked at its current setting and the amplitude sweep range is limited to 40 dB.

Example

:POW:ATT:AUTO OFF

The preceding example turns off the attenuator hold function.

*RST	1
Key Entry	**Atten Hold Off On**

:MODE

```
[:SOURce]:POWer:MODE FIXed|SWEep|LIST
[:SOURce]:POWer:MODE?
```

This command starts or stops an amplitude sweep and sets the power mode of the signal generator.

FIXed	This choice stops a power sweep and allows the signal generator to operate at a fixed power level. Refer to "[:LEVel][:IMMediate][:AMPLitude]" on page 92 for more information on running power sweeps and setting CW amplitude settings that control the output power.
LIST	This choice selects the swept power mode. If sweep triggering is set to immediate along with continuous sweep mode, executing the command starts the LIST or STEP frequency sweep.

NOTE	To perform a frequency and amplitude sweep, you must also select LIST or SWEep as the frequency mode (see ":FREQuency:MODE" on page 71).

Example

:POW:MODE LIST

The preceding example sets list as the amplitude sweep mode.

*RST	FIX			
Key Entry	**Sweep Type**	**Ampl**	**Off**	**Freq & Ampl**

:PROTection:STATe

```
[:SOURce]:POWer:PROTection[:STATe] ON|OFF|1|0
[:SOURce]:POWer:PROTection[:STATe]?
```

This command enables or disables the power search protection function. The power search protection function sets the attenuator to its maximum level whenever a power search is initiated. This can be used to protect devices that are sensitive to high average power or high power changes. The trade-off on using the power protection function is decreased attenuator life, as the attenuator will switch to its maximum setting during a power search.

NOTE	Continual or excessive use of the power search protection function can decrease attenuator life.

ON (1)	Causes the attenuator to switch to and hold its maximum level setting during a power search.
OFF (0)	Sets the attenuator normal mode. The attenuator is not used during power search.

Example

:POW:PROT ON

The preceding example enables the power inhibit function.

*RST	0
Key Entry	**RF During Power Search Normal Minimum**

:REFerence

[:SOURce]:POWer:REFerence <val><unit>
[:SOURce]:POWer:REFerence?

This command sets the power level for the signal generator RF output reference. The RF output power is referenced to the value entered in this command.

Example

:POW:REF 50DBM

The preceding example sets the RF output power reference to 50 dBm.

*RST	+0.00000000E+000
Range	-400 to 300 dBm
Key Entry	**Ampl Ref Set**

:REFerence:STATe

[:SOURce]:POWer:REFerence:STATe ON|OFF|1|0
[:SOURce]:POWer:REFerence:STATe?

This command enables or disables the RF output reference.

ON (1)	Sets the power reference state ON. dB is the unit displayed for commands (":ANNotation:AMPLitude:UNIT" on page 24 and ":POWer" on page 64).
OFF (0)	Sets the power reference state OFF.

Once the reference state is ON, all subsequent output power settings are set relative to the reference value. Amplitude offsets can be used with the amplitude reference mode.

Example

:POW:REF:STAT 1

The preceding example sets the reference state on.

*RST	0

Key Entry   **Ampl Ref Off On**

:STARt

[:SOURce]:POWer:STARt <val><unit>
[:SOURce]:POWer:STARt?

This command sets the amplitude of the first point in a step sweep.

During an amplitude sweep operation, the signal generator protects the step attenuator by automatically switching to attenuator hold (ON) mode. The attenuator is locked at its current setting and the amplitude sweep range is limited to 40 dB.

Example

:POW:STAR -30DBM

The preceding example sets the amplitude of the first point in the sweep to –30 dBm.

*RST   Depends on model and option number

Range   Refer to "[:LEVel][:IMMediate][:AMPLitude]" on page 92 for the output power ranges.

Key Entry   **Ampl Start**

:STOP

[:SOURce]:POWer:STOP <val><unit>
[:SOURce]:POWer:STOP?

This command sets the amplitude of the last point in a step sweep.

During an amplitude sweep, the signal generator protects the step attenuator by switching to attenuator hold (ON) mode. The attenuator is locked at its current setting and the amplitude sweep range is limited to 40 dB.

Example

:POW:STOP -10DBM

The preceding example sets the amplitude of the last point in the sweep to –10 dBm.

*RST   Depends on model and option number.

Range   See "[:LEVel][:IMMediate][:AMPLitude]" on page 92 for the available power ranges.

Key Entry   **Ampl Stop**

[:LEVel][:IMMediate]:OFFSet

[:SOURce]:POWer[:LEVel][:IMMediate]:OFFSet <val><unit>
[:SOURce]:POWer[:LEVel][:IMMediate]:OFFSet?

This command sets the power offset value as a dB power offset to the actual RF output. This simulates a power level at a test point beyond the RF OUTPUT connector without changing the actual RF output power. The offset value only affects the displayed amplitude setting.

You can enter an amplitude offset anytime in either normal operation or amplitude reference mode.

Example

:POW:OFFS 10DB

The preceding example sets the amplitude offset to 10 dB.

*RST   +0.00000000E+000

Range   −200dB to 200dB

Key Entry   Ampl Offset

[:LEVel][:IMMediate][:AMPLitude]

[:SOURce]:POWer[:LEVel][:IMMediate][:AMPLitude] <val><unit>
[:SOURce]:POWer[:LEVel][:IMMediate][:AMPLitude]?

This command sets the RF output power.

Example

:POW 0DBM

The preceding example sets the signal generator output power level to 0 dBm.

*RST   Depends on model and option number

Range   The ranges for this command are specified values from the data sheet.

E8663B

Option	Standard	with Option UNW
Option 503 100 kHz–3.2 GHz	−135 to 15DBM	−135 to 12DBM
Option 509 100 kHz–3.2 GHz	−135 to 14DBM	−135 to 11DBM
>3.2 GHz–9 GHz	−135 to 16DBM	−135 to 16DBM

Key Entry   Amplitude

Trigger Sweep Subsystem ([:SOURce])

:TSWeep

[:SOURce]:TSWeep

This command aborts the current sweep, then either arms or arms and starts a single list or step sweep, depending on the trigger type.

The command performs the following:

• arms a single sweep when BUS, EXTernal, or KEY is the trigger source selection

• arms and starts a single sweep when IMMediate is the trigger source selection

Key Entry   Single Sweep

4 Analog Commands

This chapter provides SCPI descriptions for subsystems dedicated to the E8663B Analog signal generator. The following is a list of the subsystems:

Amplitude Subsystem ([:SOURce])

:AM[1]|2...

```
[:SOURce]:AM[1]|2...
```

This prefix enables the selection of the AM path and is part of most SCPI commands associated with this subsystem. The two paths are equivalent to the **AM Path 1 2** softkey.

AM1   **AM Path 1 2** with 1 selected

AM2   **AM Path 1 2** with 2 selected

When just AM is shown in a command, the command defaults to path 1.

Each path is set up separately. When a SCPI command uses AM1, only path one is affected. Consequently, when AM2 is selected, only path two is set up. However, the depth of the signals for the two paths can be coupled.

The two AM paths can be on at the same time provided the following conditions have been met:

- dual–sine or swept–sine is not one of the selections for the waveform type
- each path uses a different source (Internal 1, Internal 2, Ext1, or Ext2)

:AM:INTernal:FREQuency:STEP[:INCRement]

[:SOURce]:AM:INTernal:FREQuency:STEP[:INCRement]<num>|MAXimum|MINimum|DEFault

[:SOURce]:AM:INTernal:FREQuency:STEP[:INCRement]?

This command sets the step value for the AM internal frequency.

The step value set by this command is used with the UP and DOWN choices for the :AM[1]|2:INTernal[1]|2:FREQuency command described on page 95.

The step value set with this command is not affected by a signal generator power-on, preset, or *RST command.

Example

:AM:INT:FREQ:STEP 1E3

The preceding example sets the step size to 1000 hertz.

Range   0.5–1E6

Key Entry   Incr Set

:AM:MODE

[:SOURce]:AM:MODE DEEP|NORMal

[:SOURce]:AM:MODE?

This command sets the mode for amplitude modulation.

DEEP   This choice enables amplitude modulation depth with a greater dynamic range than normal mode which utilizes the ALC. DEEP has no specified parameters and emulates the amplitude modulation NORMal mode with the ALC disabled.

NORMal   This choice maintains the amplitude modulation standard behavior and has specified parameters as outlined in the data sheet.

The ALC is disabled when the carrier amplitude is less than −10 dBm and DEEP is the AM mode.

DEEP is limited to repetitive AM and will not work with a dc modulation signal.

Example

:AM:MODE NORM

The preceding example selects the normal mode for amplitude modulation.

*RST   NORM

Key Entry   AM Mode Normal Deep

:AM[1]|2:EXTernal[1]|2:COUPling

```
[:SOURce]:AM[1]|2:EXTernal[1]|2:COUPling AC|DC
[:SOURce]:AM[1]|2:EXTernal[1]|2:COUPling?
```

This command sets the coupling type for the selected external input. The command does not change the active source or switch the modulation on or off. The modulating signal may be the sum of several signals, with either internal or external sources.

AC   This choice will pass only ac signal components.

DC   This choice will pass both ac and dc signal components.

Example

:AM1:EXT1:COUP AC

The preceding example sets the AM path 1, external 1 source coupling to AC.

*RST   DC

Key Entry   **Ext Coupling DC AC**

:AM[1]|2:EXTernal[1]|2:IMPedance

```
[:SOURce]:AM[1]|2:EXTernal[1]|2:IMPedance <50|600>
[:SOURce]:AM[1]|2:EXTernal[1]|2:IMPedance?
```

This commands sets the impedance for the external input.

Example

:AM1:EXT1:IMP 600

The preceding example sets the AM path 1, external 1 source impedance to 600 ohms.

*RST   +5.00000000E+001

Key Entry   **Ext Impedance 50 Ohm 600 Ohm**

:AM[1]|2:INTernal[1]|2:FREQuency

```
[:SOURce]:AM[1]|2:INTernal[1]|2:FREQuency <val><units>|UP|DOWN
[:SOURce]:AM[1]|2:INTernal[1]|2:FREQuency?
```

This command sets the internal AM rate using the variable <val><units>. The command, used with the UP|DOWN parameters, will change the frequency rate by a user-defined step value. Refer to the :PULM:INTernal[1]:FREQuency:STEP command on for setting the value associated with the UP and DOWN choices.

The command changes:

- the frequency rate of the first tone of a dual–sine waveform

- the start frequency for a swept–sine waveform

- the AM frequency rate for all other waveforms

Refer to ":AM[1]|2:INTernal[1]|2:FUNCtion:SHAPe" on page 98 for the waveform selection.

Example

:AM1:INT2:FREQ UP

The preceding example increases the modulation rate for AM path 1, AM internal source 2 by the step value set with the :AM:INTernal:FREQuency:STEP[:INCRement] command described on page 94.

*RST   +4.00000000E+002

Range   *Dual–Sine & Sine*: 0.5HZ–1MHZ   *Swept–Sine*: 1HZ–1MHZ

                   All Other Waveforms: 0.5HZ–100kHZ

Key Entry   **AM Tone 1 Rate**   **AM Start Rate**   **AM Rate**

:AM[1]|2:INTernal[1]:FREQuency:ALTernate

[:SOURce]:AM[1]|2:INTernal[1]:FREQuency:ALTernate <val><units>
[:SOURce]:AM[1]|2:INTernal[1]:FREQuency:ALTernate?

This command sets the frequency for the alternate signal. The alternate signal frequency is the second tone of a dual–sine or the stop frequency of a swept–sine waveform.

Refer to ":AM[1]|2:INTernal[1]|2:FUNCtion:SHAPe" on page 98 for the waveform selection.

Example

:AM2:INT1:FREQ:ALT 500KHZ

The preceding example sets the alternate frequency (AM path 2, AM internal source 1) for AM tone 2 to 500 kHz.

*RST   +4.00000000E+002

Range   *Dual–Sine*: 0.5HZ–1MHZ   *Swept–Sine*: 1HZ–1MHZ

Key Entry   **AM Tone 2 Rate**   **AM Stop Rate**

:AM[1]|2:INTernal[1]:FREQuency:ALTernate:AMPLitude:PERCent

[:SOURce]:AM[1]|2:INTernal[1]:FREQuency:ALTernate:AMPLitude:
PERCent <val> [:SOURce]:AM[1]|2:INTernal[1]:FREQuency:ALTernate:AMPLitude:PERCent?

This command sets the amplitude of the second tone for a dual–sine waveform as a percentage of the total amplitude. For example, if the second tone makes up 30% of the total amplitude, then the first tone is 70% of the total amplitude.

Refer to ":AM[1]|2:INTernal[1]|2:FUNCtion:SHAPe" on page 98 for the waveform selection.

Example

:AM2:INT1:FREQ:ALT:AMPL:PERC 50

The preceding example sets the amplitude (AM path 2, AM internal source 1) for AM tone 2 to 50% of the total amplitude.

*RST   +5.00000000E+001

Range   0–100PCT

Key Entry   AM Tone 2 Ampl Percent Of Peak

:AM[1]|2:INTernal[1]|2:FUNCtion:NOISe

```
[:SOURce]:AM[1]|2:INTernal[1]|2:FUNCtion:NOISe GAUSsian|UNIForm
[:SOURce]:AM[1]|2:INTernal[1]|2:FUNCtion:NOISe?
```

This command selects a gaussian or uniform noise modulation for the selected waveform.

Refer to ":AM[1]|2:INTernal[1]|2:FUNCtion:SHAPe" on page 98 for the waveform selection.

Example

:AM2:INT1:FUNC:NOIS GAUS

The preceding example selects the gaussian noise waveform for AM modulation on AM path 2, internal source 1.

*RST   UNIF

Key Entry   Gaussian   Uniform

:AM[1]|2:INTernal[1]|2:FUNCtion:RAMP

```
[:SOURce]:AM[1]|2:INTernal[1]|2:FUNCtion:RAMP POSitive|NEGative
[:SOURce]:AM[1]|2:INTernal[1]|2:FUNCtion:RAMP?
```

This command selects a positive or negative slope for the modulating ramp waveform.

Refer to :AM[1]|2:INTernal[1]|2:FUNCtion:SHAPe for the waveform selection.

Example

:AM2:INT1:FUNC:RAMP NEG

The preceding example sets the slope of the ramp modulation for AM path 2, internal source 1, to negative.

*RST   POS

Key Entry   Positive   Negative

:AM[1]|2:INTernal[1]|2:FUNCtion:SHAPe

```
[:SOURce]:AM[1]|2:INTernal[1]|2:FUNCtion:SHAPe SINE|TRIangle|SQUare|
RAMP|NOISe|DUALsine|SWEPtsine
[:SOURce]:AM[1]|2:INTernal[1]|2:FUNCtion:SHAPe?
```

This command sets the AM waveform type. The INTernal2 source selection does not support the dual–sine or Sweep–Sine waveform choices.

Example

:AM1:INT1:FUNC:SHAP DUAL

The preceding example sets the AM waveform type for AM path 1, internal source 1, to dual sine.

*RST	SINE						
Key Entry	Sine	Triangle	Square	Ramp	Noise	Dual-Sine	Swept-Sine

:AM[1]|2:INTernal[1]:SWEep:RATE

```
[:SOURce]:AM[1]|2:INTernal[1]:SWEep:RATE <val><units>
[:SOURce]:AM[1]|2:INTernal[1]:SWEep:RATE?
```

This command sets the sweep rate for the AM swept–sine waveform.

Refer to ":AM[1]|2:INTernal[1]|2:FUNCtion:SHAPe" on page 98 for the waveform selection. The sweep rate function is only available for internal source 1.

Example

:AM2:INT1:SWE:RATE 1KHZ

The preceding example sets the sweep rate for AM path 1, internal source 1to 1 kHz.

*RST	+4.00000000E+002
Range	0.5HZ–100kHZ
Key Entry	AM Sweep Rate

:AM[1]|2:INTernal[1]:SWEep:TRIGger

```
[:SOURce]:AM[1]|2:INTernal[1]:SWEep:TRIGger BUS|IMMediate|EXTernal|KEY
[:SOURce]:AM[1]|2:INTernal[1]:SWEep:TRIGger?
```

This command sets the trigger source for the AM swept–sine waveform.

BUS	This choice enables GPIB triggering using the *TRG or GET command or LAN triggering using the *TRG command.
IMMediate	This choice enables immediate triggering of the sweep event.

EXTernal   This choice enables the triggering of a sweep event by an externally applied signal at the TRIGGER IN connector.

KEY   This choice enables triggering through front panel interaction by pressing the **Trigger** hardkey.

Refer to ":AM[1]|2:INTernal[1]|2:FUNCtion:SHAPe" on page 98 for the waveform selection.

Example

:AM1:INT1:SWE:TRIG EXT

The preceding example sets an external trigger source for the swept–sine waveform on AM path 1.

*RST	IMM			
Key Entry	Bus	Free Run	Ext	Trigger Key

:AM[1]|2:SOURce

[:SOURce]:AM[1]|2:SOURce INT[1]|INT2|EXT[1]|EXT2
[:SOURce]:AM[1]|2:SOURce?

This command selects the source for amplitude modulation.

INT   This choice selects internal source 1 or 2 to provide an ac–coupled signal.

EXT   This choice selects the EXT 1 INPUT or the EXT 2 INPUT connector to provide an externally applied signal that can be ac- or dc-coupled. The externally applied, ac-coupled input signal is tested for a voltage level and an annunciator, on the signal generator's front-panel display, will indicate a high or low condition if that voltage is > ±3% of 1 V_p.

Example

:AM2:SOUR INT1

The preceding example selects internal source 1 as the source for AM path 2.

*RST	INT			
Key Entry	Internal 1	Internal 2	Ext1	Ext2

:AM[1]|2:STATe

[:SOURce]:AM[1]|2:STATe ON|OFF|1|0
[:SOURce]:AM[1]|2:STATe?

This command enables or disables amplitude modulation for the selected path.

The RF carrier is modulated when you have set the signal generator's modulation state to ON, see ":MODulation[:STATe]" on page 40 for more information.

Whenever amplitude modulation is enabled, the AM annunciator appears on the signal generator's front-panel display.

The two paths for amplitude modulation can be simultaneously enabled. Refer to ":AM[1]|2..." on page 93 for more information.

Example

:AM1:STAT ON

The preceding example turns on AM modulation for AM path 1.

*RST   0

Key Entry   **AM Off On**

:AM[1]|2:TYPE

[:SOURce]:AM[1]|2:TYPE LINear|EXPonential
[:SOURce]:AM[1]|2:TYPE?

This command sets the AM type to linear or exponential AM.

LINear   This choice selects linear AM type with depth values in units of percent/volt.

EXPonential   This choice selects exponential AM type with depth values in units of dB/volt.

Example

:AM2:TYPE EXP

The preceding example selects exponential type depth values for AM path 2.

*RST   LIN

Key Entry   **AM Type LIN EXP**

:AM[1]|2[:DEPTh]:EXPonential

[:SOURce]:AM[1]|2[:DEPTh]:EXPonential <val>
[:SOURce]:AM[1]|2[:DEPTh]:EXPonential?

This commands sets the AM depth in dB/volt units. EXPonential must be the current AM type for this command to have any affect. Refer to :AM[1]|2:TYPE for setting the AM type.

Example

:AM2:EXP 20

The preceding example sets the exponential depth to 20 dB for AM path 2.

*RST   +4.00000000E+001

Range   0.00–40.00DB

Key Entry   **AM Depth**

:AM[1]|2[:DEPTh][:LINear]

```
[:SOURce]:AM[1]|2[:DEPTh][:LINear] <val>|UP|DOWN
[:SOURce]:AM[1]|2[:DEPTh][:LINear]?
```

This command sets the AM depth in percent/volt units.The command, used with the UP|DOWN parameters, will change the depth by a user-defined step value. Refer to the :AM[:DEPTh]:STEP[:INCRement] command on page 101 for setting the value associated with the UP and DOWN choices.

LINear must be the current AM type for this command to have any affect. Refer to ":AM[1]|2:TYPE" on page 100 for setting the AM measurement type. When the depth values are coupled, a change made to one path is applied to both. For AM depth value coupling, refer to the command ":AM[1]|2[:DEPTh][:LINear]:TRACk" on page 101.

Example

:AM2 20

The preceding example sets the AM path 2 linear depth to 20%.

*RST	+1.00000000E-001
Range	0.0–100PCT
Key Entry	**AM Depth**

:AM[1]|2[:DEPTh][:LINear]:TRACk

```
[:SOURce]:AM[1]|2[:DEPTh][:LINear]:TRACk ON|OFF|1|0
```

```
[:SOURce]:AM[1]|2[:DEPTh][:LINear]:TRACk?
```

This command enables or disables AM depth value coupling between AM paths 1 and 2. When the depth values are coupled, a change made to one path is applied to both. LINear must be the AM type for this command to have any affect. Refer to ":AM[1]|2:TYPE" on page 100 for setting the AM type.

ON (1)   This choice will link the depth value of AM[1] with AM2; AM2 will assume the AM[1] depth value. For example, if AM[1] depth is set to 15% and AM2 is set to 11%, enabling the depth tracking will cause the AM2 depth value to change to 15%. This applies regardless of the path (AM[1] or AM2) selected in this command

OFF (0)   This choice disables coupling and both paths will have independent depth values.

Example

:AM1:TRAC ON

The preceding example enables AM depth coupling between AM path 1 and AM path 2.

*RST	0
Key Entry	**AM Depth Couple Off On**

:AM[:DEPTh]:STEP[:INCRement]

```
[:SOURce]:AM[:DEPTh]:STEP[:INCRement] <val>|MAXimum|MINimum|DEFault
[:SOURce]:AM[:DEPTh]:STEP[:INCRement]?
```

This command sets the linear depth step value in percent/volt units.

The step value set by this command is used with the UP and DOWN choices for the :AM[1]|2[:DEPTh][:LINear] command on page 101.

The setting enabled by this command is not affected by a signal generator power-on, preset, or *RST command.

Example

:AM:STEP 10

The preceding example sets the step value for AM depth to 10%.

Range   0.1–100

Key Entry   Incr Set

Frequency Modulation Subsystem ([:SOURce])

:FM[1]|2...

[:SOURce]:FM[1]|2...

This prefix enables the selection of the FM path and is associated with all SCPI commands in this subsystem. The two paths are equivalent to the **FM Path 1 2** softkey.

FM1   **FM Path 1 2** with 1 selected

FM2   **FM Path 1 2** with 2 selected

When just FM is shown in a command, this means the command applies to path one only.

Each path is set up separately. When a SCPI command uses FM1, only path one is affected. Consequently, when FM2 is selected, only path two is set up. However, the deviation of the signals for the two paths can be coupled.

Deviation coupling links the deviation value of FM1 to FM2. Changing the deviation value for one path changes it for the other. These two paths can be on at the same time provided the following conditions have been met:

- dual–sine or swept–sine is not the selection for the waveform type

- each path uses a different source (Internal 1, Internal 2, Ext1, or Ext2)

- FM2 must be set to a deviation less than FM1

:FM:INTernal:FREQuency:STEP[:INCRement]

`[:SOURce]:FM:INTernal:FREQuency:STEP[:INCRement]<num>|MAXimum|MINimum|DEFault`

`[:SOURce]:FM:INTernal:FREQuency:STEP[:INCRement]?`

This command sets the step value for the internal frequency modulation.

The step value set by this command is used with the UP and DOWN choices for the command :FM[1]|2:INTernal[1]|2:FREQuency command on page 106.

The setting enabled by this command is not affected by a signal generator power-on, preset, or `*RST` command.

Example

:FM:INT:FREQ:STEP 1E5

The preceding example sets the step value to .1 MHz.

Range   0.5–1E6

:FM[1]|2:EXTernal[1]|2:COUPLing

`[:SOURce]:FM[1]|2:EXTernal[1]|2:COUPling AC|DC`
`[:SOURce]:FM[1]|2:EXTernal[1]|2:COUPling?`

This command sets the coupling type for the selected external input. The command does not change the active source or switch modulation on or off. The modulating signal may be the sum of several signals, from either internal or external sources.

AC   This choice will pass only ac signal components.

DC   This choice will pass both ac and dc signal components.

Example

:FM1:EXT1:COUP AC

The preceding example sets the coupling for FM path 1, external source 1 to AC.

*RST   DC

Key Entry   Ext Coupling DC AC

:FM[1]|2:EXTernal[1]|2:IMPedance

`[:SOURce]:FM[1]|2:EXTernal[1]|2:IMPedance <50|600>`
`[:SOURce]:FM[1]|2:EXTernal[1]|2:IMPedance?`

This command sets the impedance for the external input.

Example

:FM1:EXT2:IMP 600

The preceding example sets the FM path 1, external 1 source impedance to 600 ohms.

*RST   +5.00000000E+001

Key Entry   **Ext Impedance 50 Ohm 600 Ohm**

:FM[1]|2:INTernal[1]:FREQuency:ALTernate

[:SOURce]:FM[1]|2:INTernal[1]:FREQuency:ALTernate <val><units>
[:SOURce]:FM[1]|2:INTernal[1]:FREQuency:ALTernate?

This command sets the internal FM rate of the alternate signal. The alternate signal frequency is the second tone of a dual–sine or the stop frequency of a swept–sine waveform.

Refer to ":FM[1]|2:INTernal[1]|2:FUNCtion:SHAPe" on page 107 for the waveform selection.

Example

:FM1:INT:FREQ:ALT 20KHZ

The preceding example sets the FM tone 2 rate for FM path 1, FM source 1, to 20 kHz.

*RST   +4.00000000E+002

Range   *dual–sine:* 0.5HZ–100kHZ   *swept–sine:* 0.5HZ–100kHZ

Key Entry   **FM Tone 2 Rate   FM Stop Rate**

:FM[1]|2:INTernal[1]:FREQuency:ALTernate:AMPLitude:PERCent

[:SOURce]:FM[1]|2:INTernal[1]:FREQuency:ALTernate:AMPLitude:
PERCent <val><units>
[:SOURce]:FM[1]|2:INTernal[1]:FREQuency:ALTernate:AMPLitude:PERCent?

This command sets the amplitude of the second tone for a dual–sine waveform as a percentage of the total amplitude. For example, if the second tone makes up 30% of the total amplitude, then the first tone is 70% of the total amplitude. Refer to ":FM[1]|2:INTernal[1]|2:FUNCtion:SHAPe" on page 107 for the waveform selection.

Example

:FM1:INT:FREQ:ALT:AMPL:PERC 20

The preceding example sets the amplitude for FM tone 2, FM path 1, FM internal source 1 to 20% of the total amplitude.

*RST   +5.00000000E+001

Range   0–100PCT

Key Entry   **FM Tone 2 Ampl Percent Of Peak**

:FM[1]|2:INTernal[1]:SWEep:RATE

[:SOURce]:FM[1]|2:INTernal[1]:SWEep:RATE <val><units>

[:SOURce]:FM[1]|2:INTernal[1]:SWEep:RATE?

This command sets the sweep rate for the swept–sine waveform. The minimum resolution is 0.5 hertz. Refer to ":FM[1]|2:INTernal[1]|2:FUNCtion:SHAPe" on page 107 for the waveform selection.

Example

:FM1:INT:SWE:RATE 20KHZ

The preceding example sets the sweep rate for the swept–sine waveform to 20 kilohertz.

*RST	+4.00000000E+002
Range	0.5HZ–100kHZ
Key Entry	**FM Sweep Rate**

:FM[1]|2:INTernal[1]:SWEep:TRIGger

[:SOURce]:FM[1]|2:INTernal[1]:SWEep:TRIGger BUS|IMMediate|EXTernal|KEY

[:SOURce]:FM[1]|2:INTernal[1]:SWEep:TRIGger?

This command sets the trigger source for the FM swept–sine waveform. Refer to ":FM[1]|2:INTernal[1]|2:FUNCtion:SHAPe" on page 107 for the waveform selection.

BUS	This choice enables GPIB triggering using the *TRG or GET command or LAN triggering using the *TRG command.
IMMediate	This choice enables immediate triggering of the sweep event. This choice is equivalent to pressing the **Free Run** softkey.
EXTernal	This choice enables the triggering of a sweep event by an externally applied signal at the TRIGGER IN connector.
KEY	Enables triggering through front panel interaction (the **Trigger** hardkey).
*RST	IMM

Example

:FM1:INT:SWE:TRIG BUS

The preceding example selects the bus as the trigger source for FM path 1.

Key Entry	**Bus**	**Free Run**	**Ext**	**Trigger Key**

:FM[1]|2:INTernal[1]|2:FREQuency

[:SOURce]:FM[1]|2:INTernal[1]|2:FREQuency <val><units>|UP|DOWN

[:SOURce]:FM[1]|2:INTernal[1]|2:FREQuency?

This command sets the internal FM rate using the <val><units> variable, or changes the FM rate by a user-defined up/down step value. Refer to the :FM:INTernal:FREQuency:STEP[:INCRement] command on page 103 for setting the value associated with the UP and DOWN choices.

The command changes:

- the FM rate of the first tone of a dual–sine waveform
- the starting FM rate for a swept–sine waveform
- the FM rate for all other waveforms

Refer to ":FM[1]|2:INTernal[1]|2:FUNCtion:SHAPe" on page 107 for the waveform selection.

Example

:FM2:INT:FREQ 40KHZ

The preceding example sets the modulation rate for FM path 2 to 40 kHz.

*RST	+4.00000000E+002		
Range	*Dual-Sine & Sine*: 0.5HZ–1MHZ		*Swept-Sine*: 1HZ–1MHZ
	All Other Waveforms: 0.5HZ–100kHZ		
Key Entry	FM Tone 1 Rate	FM Start Rate	FM Rate

:FM[1]|2:INTernal[1]|2:FUNCtion:NOISe

[:SOURce]:FM[1]|2:INTernal[1]|2:FUNCtion:NOISe GAUSsian|UNIForm

[:SOURce]:FM[1]|2:INTernal[1]|2:FUNCtion:NOISe?

This command selects a gaussian or uniform noise type as the modulation. Refer to ":FM[1]|2:INTernal[1]|2:FUNCtion:SHAPe" on page 107 for the waveform selection.

Example

:FM2:INT2:FUNC:NOIS UNIF

The preceding example selects a uniform noise waveform as the modulation for FM path 2 and FM source 2.

*RST	UNIF	
Key Entry	Gaussian	Uniform

:FM[1]|2:INTernal[1]|2:FUNCtion:RAMP

`[:SOURce]:FM[1]|2:INTernal[1]|2:FUNCtion:RAMP POSitive|NEGative`

`[:SOURce]:FM[1]|2:INTernal[1]|2:FUNCtion:RAMP?`

This command selects a positive or negative ramp as the internal modulating waveform. Refer to :FM[1]|2:INTernal[1]|2:FUNCtion:SHAPe for the waveform selection.

Example

:FM2:INT2:FUNC:RAMP POS

The preceding example selects a positive sloped ramp as the internal modulating waveform.

*RST   POS

Key Entry   **Positive**   **Negative**

:FM[1]|2:INTernal[1]|2:FUNCtion:SHAPe

`[:SOURce]:FM[1]|2:INTernal[1]|2:FUNCtion:SHAPe SINE|TRIangle|SQUare|`
`RAMP|NOISe|DUALsine|SWEPtsine`
`[:SOURce]:FM[1]|2:INTernal[1]|2:FUNCtion:SHAPe?`

This command selects the FM waveform type. The INTernal2 source selection does not support the dual–sine or Sweep–Sine waveform types.

Example

:FM2:INT1:FUNC:SHAP SQU

The preceding example selects a square wave as the internal modulating waveform.

*RST   SINE

Key Entry	**Sine**	**Triangle**	**Square**	**Ramp**	**Noise**	**Dual-Sine**	**Swept-Sine**

:FM[1]|2:SOURce

`[:SOURce]:FM[1]|2:SOURce INT[1]|INT2|EXT1|EXT2`
`[:SOURce]:FM[1]|2:SOURce?`

This command selects the FM source.

INT   This choice selects internal source 1 or 2 to provide an ac‐coupled signal.

EXT   This choice selects the EXT 1 INPUT or the EXT 2 INPUT connector to provide an externally applied signal that can be ac‐ or dc‐coupled. The externally applied, ac‐coupled input signal is tested for a voltage level and an annunciator, on the signal generator's front‐panel display, will indicate a high or low condition if that voltage is > ±3% of 1 V_p.

Example

:FM2:SOUR INT2

The preceding example selects internal source 2 as the FM source for FM path 2.

---

*RST	INT			
Key Entry	**Internal 1**	**Internal 2**	**Ext1**	**Ext2**

:FM[1]|2:STATe

[:SOURce]:FM[1]|2:STATe ON|OFF|1|0

[:SOURce]:FM[1]|2:STATe?

This command enables or disables the selected FM path.

The RF carrier is modulated when you set the signal generator's modulation state to ON, see ":MODulation[:STATe]" on page 40 for more information.

Whenever frequency modulation is enabled, the FM annunciator appears on the signal generator's front-panel display.

The two paths for frequency modulation can be simultaneously enabled. Refer to ":FM[1]|2..." on page 102 for more information.

Example

:FM2:STAT ON

The preceding example enables FM path 2.

*RST	0
Key Entry	**FM Off On**

:FM[1]|2[:DEViation]

[:SOURce]:FM[1]|2[:DEViation] <val><units>

[:SOURce]:FM[1]|2[:DEViation]?

This command sets the FM deviation for the selected FM path.

If deviation tracking is ON, a change to the deviation value on one path will apply to both. Refer to ":FM[1]|2[:DEViation]:TRACk" on page 109 for more information on setting the deviation tracking.

Example

:FM2 1MHZ

The preceding example sets the frequency deviation to 1 megahertz.

*RST	+1.00000000E+003	
Range	*Frequency*	*Deviation*
	250KHZ–250MHZ	0–2MHZ
	> 250–500MHZ	0–1MHZ
	> 0.5–1GHZ	0–2MHZ
	> 1–2GHZ	0–4MHZ
	> 2–3.2GHZ	0–8MHZ
	> 3.2–9GHZ	0–16MHZ

Key Entry   FM DEV

:FM[1]|2[:DEViation]:TRACk

[:SOURce]:FM[1]|2[:DEViation]:TRACk ON|OFF|1|0

[:SOURce]:FM[1]|2[:DEViation]:TRACk?

This command enables or disables deviation coupling between FM paths 1 and 2.

ON (1)	This choice will link the deviation value of FM1 with FM2; FM2 will assume the FM1 deviation value. For example, if FM1 deviation is set to 500 Hz and FM2 is set to 2 kHz, enabling the deviation tracking will cause the FM2 deviation value to change to 500 Hz. This applies regardless of the path (FM1 or FM2) selected.
OFF (0)	This choice disables the coupling and both paths will have independent deviation values.

This command uses exact match tracking, not offset tracking.

Example

:FM2:TRAC 0

The preceding example disables deviation coupling.

*RST   0

Key Entry   FM Dev Couple Off On

Low Frequency Output Subsystem ([:SOURce]:LFOutput)

:LFOutput:AMPLitude

[:SOURce]:LFOutput:AMPLitude <val><units>

[:SOURce]:LFOutput:AMPLitude?

This command sets the amplitude of the signal at the LF OUTPUT connector.

Example

:LFO:AMPL 2.1VP

The preceding example sets the peak amplitude to 2.1 volts.

*RST	0.00
Range	0.000VP–3.5VP
Key Entry	**LF Out Amplitude**

:LFOutput:FUNCtion[1]|2:FREQuency

[:SOURce]:LFOutput:FUNCtion[1]|2:FREQuency <val><units>

[:SOURce]:LFOutput:FUNCtion[1]|2:FREQuency?

This command sets the frequency of function generator 1 or 2. The command sets:

- the frequency of the first tone of a dual–sine waveform
- the start frequency for a swept–sine waveform
- the frequency for all other waveform types

Refer to ":LFOutput:FUNCtion[1]|2:SHAPe" on page 111 for selecting the waveform type.

Example

:LFO:FUNC1:FREQ .1MHZ

The preceding example sets the frequency for function generator 1 to 100 kHz.

*RST	+4.00000000E+002		
Range	*Sine and Dual-Sine*: 0.5HZ–1MHZ		
Range	*Swept-Sine*: 1HZ–1MHZ		
	All Other Waveforms: 0.5HZ–100KHZ		
Key Entry	**LF Out Tone 1 Freq**	**LF Out Start Freq**	LF Out Freq

:LFOutput:FUNCtion[1]:FREQuency:ALTernate

[:SOURce]:LFOutput:FUNCtion[1]:FREQuency:ALTernate <val><units>

[:SOURce]:LFOutput:FUNCtion[1]:FREQuency:ALTernate?

This command sets the frequency for the alternate LF output signal. The alternate frequency is the second tone of a dual–sine or the stop frequency of a swept–sine waveform.

Refer to ":LFOutput:FUNCtion[1]|2:SHAPe" on page 111 for more information on selecting the waveform type.

Example

:LFO:FUNC1:FREQ:ALT 20KHZ

The preceding example sets the alternate frequency to 20 kHz.

*RST	+4.00000000E+002	
Range	*Dual-Sine*: 0.1HZ–100kHZ	*Swept-Sine*: 0.1HZ–100kHZ
Key Entry	**LF Out Tone 2 Freq**	**LF Out Stop Freq**

:LFOutput:FUNCtion[1]:FREQuency:ALTernate:AMPLitude:PERCent

[:SOURce]:LFOutput:FUNCtion[1]:FREQuency:ALTernate:AMPLitude:
PERCent <val><units>

[:SOURce]:LFOutput:FUNCtion[1]:FREQuency:ALTernate:AMPLitude:PERCent?

This command sets the amplitude of the second tone for a dual–sine waveform as a percentage of the total LF output amplitude. For example, if the second tone makes up 30% of the total amplitude, then the first tone is 70% of the total amplitude. Refer to ":LFOutput:FUNCtion[1]|2:SHAPe" on page 111 for selecting the waveform type.

Example

:LFO:FUNC1:FREQ:ALT:AMPL:PERC 50

The preceding example sets the alternate frequency to 50% of the total output amplitude.

*RST	+5.00000000E+001
Range	0–100PCT
Key Entry	**LF Out Tone 2 Ampl % of Peak**

:LFOutput:FUNCtion[1]|2:SHAPe

[:SOURce]:LFOutput:FUNCtion[1]|2:SHAPe SINE|DUALsine|SWEPtsine|TRIangle|
SQUare|RAMP|PULSe|NOISe|DC

[:SOURce]:LFOutput:FUNCtion[1]|2:SHAPe?

This command selects the waveform type. Function Generator 1 must be the source for the dual–sine or the swept–sine waveform. Refer to ":LFOutput:SOURce" on page 114.

Example

:LFO:FUNC2:SHAP TRI

The preceding example selects a triangle wave for the Function Generator 2 LF output.

*RST	SINE						
Key Entry	Sine	Dual-Sine	Swept-Sine	Triangle	Square	Ramp	Pulse
	Noise	DC					

:LFOutput:FUNCtion:[1]|2:SHAPe:NOISe

[:SOURce]:LFOutput:FUNCtion[1]|2:SHAPe:NOISe UNIForm|GAUSsian
[:SOURce]:LFOutput:FUNCtion[1]|2:SHAPe:NOISe?

This command selects a gaussian or uniform noise modulation for the LF output.

Refer to ":LFOutput:FUNCtion[1]|2:SHAPe" on page 111 for selecting the waveform type.

Example

:LFO:FUNC1:SHAP:NOIS GAUS

The preceding example selects a gaussian noise modulation for the Function Generator 1 LF output.

*RST	UNIF	
Key Entry	Uniform	Gaussian

:LFOutput:FUNCtion[1]|2:SHAPe:RAMP

[:SOURce]:LFOutput:FUNCtion[1]|2SHAPe:RAMP POSitive|NEGative
[:SOURce]:LFOutput:FUNCtion[1]|2SHAPe:RAMP?

This command selects a positive or negative slope for the ramp modulation on the LF output.

Refer to ":LFOutput:FUNCtion[1]|2:SHAPe" on page 111 for selecting the waveform type.

Example

:LFO:FUNC1:SHAP:RAMP POS

The preceding example selects a positive ramp slope modulation for the Function Generator 1 LF output.

*RST	POS	
Key Entry	Positive	Negative

:LFOutput:FUNCtion[1]:SWEep:RATE

[:SOURce]:LFOutput:FUNCtion[1]:SWEep:RATE <val><units>
[:SOURce]:LFOutput:FUNCtion[1]:SWEep:RATE?

This command sets the sweep rate for an internally generated swept–sine signal.

Example

:LFO:FUNC1:SWE:RATE 1E5

The preceding example sets the sweep rate for the swept–sine waveform to 100 kHz.

*RST	+4.00000000E+002
Range	0.5HZ–100kHZ
Key Entry	**LF Out Sweep Rate**

:FUNCtion[1]:SWEep:TRIGger

[:SOURce]:LFOutput:FUNCtion[1]:SWEep:TRIGger BUS|IMMediate|EXTernal|KEY

[:SOURce]:LFOutput:FUNCtion[1]:SWEep:TRIGger?

This command sets the trigger source for the internally generated swept–sine signal at the LF output.

BUS	This choice enables GPIB triggering using the *TRG or GET command or LAN and RS-232 triggering using the *TRG command.
IMMediate	This choice enables immediate triggering of the sweep event.
EXTernal	This choice enables the triggering of a sweep event by an externally applied signal at the TRIGGER IN connector.
KEY	This choice enables triggering through front panel interaction by pressing the **Trigger** hardkey.

Refer to ":LFOutput:FUNCtion[1]|2:SHAPe" on page 111 for selecting the waveform type.

Example

:LFO:FUNC1:SWE:TRIG EXT

The preceding example sets an external trigger as the trigger for the swept–sine signal.

*RST	Free Run			
Key Entry	**Bus**	**Free Run**	**Ext**	**Trigger Key**

:LFOutput:SOURce

`[:SOURce]:LFOutput:SOURce INT[1]|INT2|FUNCtion[1]|FUNCtion2`

`[:SOURce]:LFOutput:SOURce?`

This command selects the source for the LF output.

INT   This choice enables you to output a signal where the frequency and shape of the signal is set by internal source 1 or 2. For example, if the internal source is currently assigned to an AM path configuration and AM is turned on, the signal output at the LF OUTPUT connector will have the frequency and shape of the amplitude modulating signal.

FUNCtion   This choice enables the selection of an internal function generator.

Example

:LFO:SOUR FUNC1

The preceding example selects Function Generator 1 as the active LF source.

*RST	INT	
Key Entry	**Internal 1 Monitor**	**Internal 2 Monitor**
	Function Generator 1	**Function Generator 2**

:LFOutput:STATe

`[:SOURce]:LFOutput:STATe ON|OFF|1|0`

`[:SOURce]:LFOutput:STATe?`

This command enables or disables the low frequency output.

Example

:LFO:STAT ON

The preceding example enables the source.

*RST	0
Key Entry	**LF Out Off On**

Phase Modulation Subsystem ([:SOURce])

:PM[1]|2...

[:SOURce]:PM[1]|2...

This prefix enables the selection of the ΦM path and associated with all SCPI commands in this subsystem. The two paths are equivalent to the ΦM Path 1 2 softkey.

PM1   ΦM Path 1 2 with 1 selected

PM2   ΦM Path 1 2 with 2 selected

When just PM is shown in a command, this means the command applies to path 1 only.

Each path is set up separately. When a SCPI command uses PM1, only path one is affected. Consequently, when PM2 is selected, only path two is set up. However, the deviation of the signals for the two paths can be coupled.

Deviation coupling links the deviation value of PM1 to PM2. Changing the deviation value for one path will change it for the other path. These two paths can be on at the same time provided the following conditions have been met:

- dual–sine or Sweep–Sine is not the selection for the waveform type
- each path uses a different source (Internal 1, Internal 2, Ext1, or Ext2)
- PM2 must be set to a deviation less than or equal to PM1

:PM:INTernal:FREQuency:STEP[:INCRement]

[:SOURce]:PM:INTernal:FREQuency:STEP[:INCRement]<num>|MAXimum|MINimum|DEFault

[:SOURce]:PM:INTernal:FREQuency:STEP[:INCRement]?

This command sets the step value of the phase modulation internal frequency.

The step value set by this command is used with the UP and DOWN choices for the :PM[1]|2:INTernal[1]:FREQuency command on page 117.

The setting enabled by this command is not affected by a signal generator power-on, preset, or *RST command.

Example

:PM:INT:FREQ:STEP 1E5

The preceding example sets the step value to 100 kHz.

Range   0.5–1E6

Key Entry   Incr Set

:PM[1]|2:BANDwidth|BWIDth

```
[:SOURce]:PM[1]|2:BANDwidth|BWIDth NORMal|HIGH
[:SOURce]:PM[1]|2:BANDwidth|BWIDth?
```

This command selects normal phase modulation or high bandwidth phase modulation. The command can use either the BANDwidth or BWIDth paths.

Example

:PM1:BAND NORM

The preceding example selects normal phase modulation for ΦM path 1.

*RST   NORM

Key Entry   FM ΦM Normal High BW

:PM[1]|2:EXTernal[1]:COUPling

```
[:SOURce]:PM[1]|2:EXTernal[1]:COUPling AC|DC [:SOURce]:PM[1]|2:EXTernal[1]:COUPling?
```

This command sets the coupling for the phase modulation source at the selected external input connector.

AC   This choice will only pass ac signal components.

DC   This choice will pass both ac and dc signal components.

This command does not change the active source or switch modulation on or off. The modulating signal may be the sum of several signals, from either internal or external sources.

Example

:PM1:EXT:COUP AC

The preceding example selects AC coupling at the external input for ΦM path 1.

*RST   DC

Key Entry   Ext Coupling DC AC

:PM[1]|2:EXTernal[1]|2:IMPedance

```
[:SOURce]:PM[1]|2:EXTernal[1]|2:IMPedance <50|600>
[:SOURce]:PM[1]|2:EXTernal[1]|2:IMPedance?
```

This command selects 50 ohms or 600 ohms as the input impedance for the external input signal.

Example

:PM1:EXT2:IMP 600

The preceding example sets the ΦM path 1, external 2 source impedance to 600 ohms.

*RST   +5.00000000E+001

Key Entry   **Ext Impedance 50 Ohm 600 Ohm**

:PM[1]|2:INTernal[1]:FREQuency

```
[:SOURce]:PM[1]|2:INTernal[1]|2:FREQuency <val><units>
[:SOURce]:PM[1]|2:INTernal[1]|2:FREQuency?
```

This command sets the internal modulation frequency rate. The command sets:

- the frequency of the first tone of a dual–sine waveform
- the start frequency for a swept–sine waveform
- the frequency rate for all other waveforms

Refer to ":LFOutput:FUNCtion[1]|2:SHAPe" on page 111 for selecting the waveform type.

Example

:PM1:INT1:FREQ 20KHZ

The preceding example sets the ΦM path 1, internal source 1 frequency to 20 kHz.

*RST   +4.00000000E+002

Range   *Dual-Sine*: 0.1HZ–100KHZ   *Swept-Sine*: 0.1HZ–100KHZ

                   All Other Waveforms: 0.1HZ–20KHZ

Key Entry   **ΦMTone 1 Rate**   **ΦM Start Rate**   **ΦM Rate**

PM[1]|2:INTernal[1]:FREQuency:ALTernate

```
[:SOURce]:PM[1]|2:INTernal[1]:FREQuency:ALTernate <val><units>
[:SOURce]:PM[1]|2:INTernal[1]:FREQuency:ALTernate?
```

This command sets the frequency rate for the alternate signal. The alternate frequency is the second tone of a dual–sine or the stop frequency of a swept–sine waveform.

Refer to ":PM[1]|2:INTernal[1]:FUNCtion:SHAPe" on page 119 for the waveform selection.

Example

:PM1:INT1:FREQ:ALT 50KHZ

The preceding example sets the alternate frequency rate for the ΦM tone 2, ΦM path 1, source 1 to 50 kHz.

*RST	+4.00000000E+002	
Range	*Dual-Sine*: 0.1HZ–100KHZ	*Swept-Sine*: 0.1HZ–100KHZ
Key Entry	ΦM Stop Rate	ΦM Tone 2 Rate

:PM[1]|2:INTernal[1]|2:FUNCtion:NOISe

[:SOURce]:PM[1]|2:INTernal[1]|2:FUNCtion:NOISe GAUSsian|UNIForm

[:SOURce]:PM[1]|2:INTernal[1]|2:FUNCtion:NOISe?

This commands selects a gaussian or uniform noise modulation for the selected path(s).

Example

:PM1:INT1:FUNC:NOIS GAUS

The preceding example selects a gaussian noise modulation for ΦM path 1, source 1.

*RST	UNIF	
Key Entry	Gaussian	Uniform

:PM[1]|2:INTernal[1]|2:FUNCtion:RAMP

[:SOURce]:PM[1]|2:INTernal[1]|2:FUNCtion:RAMP POSitive|NEGative

[:SOURce]:PM[1]|2:INTernal[1]|2:FUNCtion:RAMP?

This command selects a positive or negative slope for the ramp modulating waveform.

Example

:PM1:INT2:FUNC:RAMP POS

The preceding example selects a positive ramp slope for modulating the signal on ΦM path 1, internal source 2.

*RST	POS	
Key Entry	Positive	Negative

:PM[1]|2:INTernal[1]:FREQuency:ALTernate:AMPLitude:PERCent

```
[:SOURce]:PM[1]|2:INTernal[1]:FREQuency:ALTernate:AMPLitude:
PERCent <val>
[:SOURce]:PM[1]|2:INTernal[1]:FREQuency:ALTernate:AMPLitude:PERCent?
```

This command sets the amplitude of the second tone for the dual–sine waveform as a percentage of the total amplitude. For example, if the second tone makes up 30% of the total amplitude, then the first tone is 70% of the total amplitude. Refer to ":PM[1]|2:INTernal[1]:FUNCtion:SHAPe" on page 119 for the waveform selection.

Example

:PM2:INT:FREQ:ALT:AMPL:PERC 40

The preceding example sets the alternate tone amplitude to 40% of the total amplitude.

*RST   +5.00000000E+001

Range   0–100PCT

Key Entry   ΦM Tone 2 Ampl Percent of Peak

:PM[1]|2:INTernal[1]:FUNCtion:SHAPe

```
[:SOURce]:PM[1]|2:INTernal[1]:FUNCtion:SHAPe SINE|TRIangle|SQUare|RAMP|
NOISe|DUALsine|SWEPtsine
[:SOURce]:PM[1]|2:INTernal[1]:FUNCtion:SHAPe?
```

This command sets the phase modulation waveform type.

Example

:PM1:INT:FUNC:SHAP RAMP

The preceding example selects a ramp modulation for ΦM path 1, source 1.

*RST   SINE

Key Entry	Sine	Triangle	Square	Ramp	Noise	Dual-Sine	Swept-Sine

:PM[1]|2:INTernal[1]:SWEep:RATE

```
[:SOURce]:PM[1]|2:INTernal[1]:SWEep:RATE <val><units>
[:SOURce]:PM[1]|2:INTernal[1]:SWEep:RATE?
```

This command sets the sweep rate for a phase-modulated, swept–sine waveform. Refer to ":PM[1]|2:INTernal[1]:FUNCtion:SHAPe" on page 119 for the waveform selection.

Example

:PM1:INT:SWE:RATE 30KHZ

The preceding example sets the sweep rate to 30 kHz.

*RST	+4.00000000E+002
Range	0.5HZ–100kHZ
Key Entry	ΦM Sweep Rate

:PM[1]|2:INTernal[1]:SWEep:TRIGger

```
[:SOURce]:PM[1]|2:INTernal[1]:SWEep:TRIGger BUS|IMMediate|EXTernal|KEY
[:SOURce]:PM[1]|2:INTernal[1]:SWEep:TRIGger?
```

This command sets the trigger source for the phase-modulated, swept–sine waveform.

BUS	This choice enables GPIB triggering using the *TRG or GET command or LAN and RS-232 triggering using the *TRG command.
IMMediate	This choice enables immediate triggering of the sweep event. This choice is equivalent to pressing the **Free Run** softkey.
EXTernal	This choice enables the triggering of a sweep event by an externally applied signal at the TRIGGER IN connector.
KEY	This choice enables triggering through front panel interaction by pressing the **Trigger** hardkey.

Refer to ":PM[1]|2:INTernal[1]:FUNCtion:SHAPe" on page 119 for the waveform selection.

Example

:PM2:INT:SWE:TRIG BUS

The preceding example selects a BUS trigger as the triggering for the internal source 1 swept–sine waveform on ΦM path 2.

*RST	IMM			
Key Entry	Bus	Free Run	Ext	Trigger Key

:PM[1]|2:SOURce

```
[:SOURce]:PM[1]|2:SOURce INT[1]|INT2|EXT[1]|EXT2
[:SOURce]:PM[1]|2:SOURce?
```

This command selects the source used to generate the phase modulation.

INT   This choice selects internal source 1 or internal source 2 to provide an ac-coupled signal.

EXT   This choice selects the EXT 1 INPUT or the EXT 2 INPUT connector to provide an externally applied signal that can be ac- or dc- coupled.

The externally applied, ac-coupled input signal is tested for a voltage level and an annunciator, on the signal generator's front–panel display, will indicate a high or low condition if that voltage is > $\pm$3% of 1 V_p.

Example

:PM2:SOUR EXT1

The preceding example selects an external signal on the EXT 1 INPUT connector as the source for ΦM path 2 modulation.

*RST	INT			
Key Entry	**Internal 1**	**Internal 2**	**Ext1**	**Ext2**

:PM[1]|2:STATe

```
[:SOURce]:PM[1]|2:STATe ON|OFF|1|0
[:SOURce]:PM[1]|2:STATe?
```

This command enables or disables the phase modulation for the selected path. The RF carrier is modulated when you set the signal generator's modulation state to ON, see ":MODulation[:STATe]" on page 40 for more information.

The ΦM annunciator appears on the signal generator's front-panel display whenever phase modulation is enabled. The two paths for phase modulation can be simultaneously enabled. Refer to ":PM[1]|2..." on page 115 for more information.

Example

:PM2:STAT 1

The preceding example turns on ΦM path 2 phase modulation.

*RST	0
Key Entry	**ΦM Off On**

:PM[1]|2[:DEViation]

```
[:SOURce]:PM[1]|2[:DEViation] <val><units>|UP|DOWN
[:SOURce]:PM[1]|2[:DEViation]?
```

This command sets the deviation of the phase modulation. The variable <units> will accept RAD (radians), PIRAD (pi-radians), and DEG (degrees); however, the query will only return values in radians. If deviation tracking is active, a change to the deviation value on one path will apply to both.

The command, used with the UP|DOWN parameters, will change the deviation by a user-defined step value. Refer to the :PM[:DEViation]:STEP[:INCRement] command on page 123 for setting the value associated with the UP and DOWN choices.

Example

:PM1 135DEG

The preceding example sets the phase modulation to 135 degrees.

*RST	+0.00000000E+000		
Range	*Frequency*	*Normal Bandwidth*	*High Bandwidth*
	250KHZ–250MHZ	0–20rad	0–2rad
	> 250–500MHZ	0–10rad	0–1rad
	> 0.5–1GHZ	0–20rad	0–2rad
	> 1–2GHZ	0–40rad	0–4rad
	> 2–3.2GHZ	0–80rad	0–8rad
	> 3.2–9GHZ	0–160rad	0–16rad
Key Entry	ΦM Dev		

:PM[1]|2[:DEViation]:TRACk

```
[:SOURce]:PM[1]|2[:DEViation]:TRACk ON|OFF|1|0
[:SOURce]:PM[1]|2[:DEViation]:TRACk?
```

This command enables or disables the deviation coupling between the PM paths 1 and 2.

ON (1)   This choice will link the deviation value of PM1 with PM2; PM2 will assume the PM[1] deviation value. For example, if PM1 deviation is set to 500 Hz and PM2 is set to 2 kHZ, enabling the deviation tracking will cause the PM2 deviation value to change to 500 Hz. This applies regardless of the path (PM1 or PM2) selected in this command.

OFF (0)   This choice disables the coupling and both paths will have independent deviation values.

This command uses exact match tracking, not offset tracking.

Example

:PM1:TRAC OFF

The preceding example disables deviation coupling.

*RST   0

Key Entry   ΦM Dev Couple Off On

:PM[:DEViation]:STEP[:INCRement]

```
[:SOURce]:PM[:DEViation]:STEP[:INCRement]<val><units>|MAXimum|MINimum|DEFault
[:SOURce]:PM[:DEViation]:STEP[:INCRement]?
```

This command sets the phase modulation deviation step value.

The value set by this command is used with the UP and DOWN choices for the FM deviation command. Refer to ":PM[1]|2[:DEViation]" on page 122 for more information.

The setting is not affected by a signal generator power-on, preset, or *RST command.

Example

:PM:STEP 20RAD

The preceding example sets the step value to 20 radians.

Range   0.001–1E3RAD

Pulse Modulation Subsystem ([:SOURce])

:PULM:EXTernal:POLarity NORMal:INVerted

[:SOURce]:PULM:EXTernal:POLarity NORMal|INVerted

[:SOURce]:PULM:EXTernal:POLarity?

This command selects the polarity of the TTL input signal at the GATE/PULSE/TRIGGER INPUT front panel connector. The signal generator can respond to either a normal (a TTL high) or an inverted (TTL low) signal.

Example

:PULM:EXT:POL NORM

The preceding example selects normal (TTL high) polarity.

*RST　　　　　　Normal

Key Entry　　　　**Ext Polarity Normal Inverted**

:PULM:INTernal[1]:DELay

[:SOURce]:PULM:INTernal[1]:DELay <num><time_suffix>|UP|DOWN

[:SOURce]:PULM:INTernal[1]:DELay?

This command sets the pulse delay for the internally-generated pulse modulation using the variable <num>[<time_suffix>]. The command, used with the UP|DOWN parameters, will change the delay by a user-defined step value. Refer to the :PULM:INTernal[1]:DELay:STEP command on page 125 for setting the value associated with the UP and DOWN choices.

The optional variable <time_suffix> accepts nS (nanoseconds) to S (seconds).

The range value is dependent on the pulse period. Refer to ":PULM:INTernal[1]:PERiod" on page 126 for pulse period settings.

Example

:PULM:INT:DEL 200E-9

The preceding example sets the internal pulse delay to 200 nanoseconds.

*RST　　　　　　+0.00000000E+000

Range　　　　　　　*Internal Free Run*: depends on pulse period and pulse width settings

　　　　　　　　　　Internal Triggered & Doublet: 70nS to (42 S - 20 nS - pulse width)

Key Entry　　　　**Pulse Delay**

:PULM:INTernal[1]:DELay:STEP

`[:SOURce]:PULM:INTernal[1]:DELay:STEP <num><time_suffix>`

`[:SOURce]:PULM:INTernal[1]:DELay:STEP?`

This command sets the step increment for the pulse delay.

The step value, set by this command, is used with the UP and DOWN choices in the ":PULM:INTernal[1]:DELay" on page 124 command.

The step value set with this command is not affected by a signal generator power-on, preset, or `*RST` command.

Example

:PULM:INT:DEL:STEP 10NS

The preceding example sets the pulse delay step value to 10 nanoseconds.

Range   10nS to (pulse period – 20 nS)

:PULM:INTernal[1]:FREQuency

`[:SOURce]:PULM:INTernal[1]:FREQuency <val><units>|UP|DOWN`

`[:SOURce]:PULM:INTernal[1]:FREQuency?`

This command sets the pulse rate for the internally-generated square wave using the variable <val><units>. The command, used with the UP|DOWN parameters, will change the frequency by a user-defined step value. Refer to the :PULM:INTernal[1]:FREQuency:STEP command for setting the value associated with the UP and DOWN choices.

This command is used when SQUare is the pulse modulation type. Refer to ":PULM:SOURce" on page 128 for the pulse modulation type selection.

Example

:PULM:INT:FREQ 1MHZ

The preceding example sets the square wave pulse rate to 1 megahertz.

*RST   +4.00000000E+002

Range   0.1HZ–10MHZ

Key Entry   Pulse Rate

:PULM:INTernal[1]:FREQuency:STEP

[:SOURce]:PULM:INTernal[1]:FREQuency:STEP[:INCRement] <frequency>

[:SOURce]:PULM:INTernal[1]:FREQuency:STEP[INCRement]?

This command sets the step value for the internally-generated square wave pulse rate.

This command is used when SQUare is the pulse modulation type. Refer to ":PULM:SOURce" on page 128 for the pulse modulation type selection. The step value, set with this command, is used with the UP and DOWN choices in the :PULM:INTernal[1]:FREQuency command.

The step value set with this command is not affected by a power-on, preset, or *RST command.

Example

:PULM:INT:FREQ:STEP MIN

The preceding example sets the step value for the square wave pulse rate to 0.1 Hz, the minimum rate.

Range   0.1HZ–10MHZ

:PULM:INTernal[1]:PERiod

[:SOURce]:PULM:INTernal[1]:PERiod <val><units>|UP|DOWN

[:SOURce]:PULM:INTernal[1]:PERiod?

This command sets the pulse period for the internally–generated pulse modulation using the variables <val><units>. The command, used with the UP|DOWN parameters, will change the pulse period by a user–defined step value. Refer to the :PULM:INTernal[1]:PERiod:STEP[:INCRement] command for setting the value associated with the UP and DOWN choices.

If the entered value for the pulse period is equal to or less than the value for the pulse width, the pulse width changes to a value that is less than the pulse period. Refer to ":PULM:INTernal[1]:PWIDth" on page 127 for setting the pulse width.

Example

:PULM:INT:PER .5S

The preceding example sets the period of the internally–generated pulse to 500 milliseconds.

*RST   +2.00000000E–006

Range   70nS–42S

Key Entry   Pulse Period

:PULM:INTernal[1]:PERiod:STEP[:INCRement]

[:SOURce]:PULM:INTernal[1]:PERiod:STEP[:INCRement]<val><units>|MAXimum|
MINimum|DEFault
[:SOURce]:PULM:INTernal[1]:PERiod:STEP[:INCRement]?

This command sets the step value for the internal pulse period using the variable <val><units>.

The step value, set with this command, is used with the UP and DOWN choices available in the :PULM:INTernal[1]:PERiod command.

The step value set with this command is not affected by a power-on, preset, or *RST command.

Example

:PULM:INT:PER:STEP .1S

The preceding example sets the square wave pulse rate to 100 milliseconds.

*RST   +1.00000000E-006

Range   10nS–42S

:PULM:INTernal[1]:PWIDth

[:SOURce]:PULM:INTernal[1]:PWIDth <num><time_suffix>|UP|DOWN
[:SOURce]:PULM:INTernal[1]:PWIDth?

This command sets the pulse width for the internally generated pulse signal.

This command sets the pulse width for the internally-generated pulse modulation using the variables <num><time_suffix>. The command, used with the UP|DOWN parameters, will change the pulse width by a user-defined step value. Refer to the :PULM:INTernal[1]:PWIDth:STEP command for setting the value associated with the UP and DOWN choices.

If the entered value for the pulse width is equal to or greater than the value for the pulse period, the pulse width changes to a value that is less than the pulse period. For more information, refer to the command ":PULM:INTernal[1]:PWIDth" on page 127.

NOTE	A power search is recommended for signals with pulse widths less than one microsecond. Refer to ":ALC:SEARch" on page 85.

Example

:PULM:INT:PWIDth 100MS

The preceding example sets the pulse width to 100 milliseconds.

*RST   +1.00000000E−006

Range   10nS to (pulse period - 20 nS)

Key Entry   Pulse Width

:PULM:INTernal[1]:PWIDth:STEP

[:SOURce]:PULM:INTernal[1]:PWIDth:STEP<num><time_suffix>|MAXimum|MINimum|DEFault

[:SOURce]:PULM:INTernal[1]:PWIDth:STEP?

This command sets the step increment for the pulse width using the variable <num><time_suffix>.

The step value, set by this command, is used with the UP and DOWN choices available in the :PULM:INTernal[1]:PWIDth command.

The step value, set with this command, is not affected by a power-on, preset, or *RST command.

Example

:PULM:INT:PWID:STEP 100NS

The preceding example sets the pulse width step to 100 nanoseconds.

*RST   +1.00000000E−006

Range   10nS to (pulse period - 20 nS)

:PULM:SOURce:INTernal

[:SOURce]:PULM:SOURce:INTernal SQUare|FRUN|TRIGgered|DOUBlet|GATEd

[:SOURce]:PULM:SOURce:INTernal?

This command selects one of the five internally generated modulation inputs. There are two external sources: Scalar and Ext Pulse which are selected using the :PULM:SOURce command.

Example

:PULM:SOUR:INT SQU

The preceding example selects the internally–generated square wave pulse modulation format.

*RST   FRUN (Int Free–Run)

Key Entry	Internal Square	Int Free-Run	Int Triggered	Int Doublet	Int Gated

:PULM:SOURce

[:SOURce]:PULM:SOURce INTernal|EXTernal|SCALar

[:SOURce]:PULM:SOURce?

This command sets the source for pulse modulation. The INTernal selection accesses one of the five internally generated modulation inputs while EXTernal selects an external pulse (Ext Pulse) and SCALar selects input from a scalar network analyzer.

Example

:PULM:SOUR INT

The preceding example selects the internal free-run, pulse modulation source.

*RST   FRUN (Int Free-Run)

Key Entry	Internal Square	Int Free-Run	Int Triggered	Int Doublet	Int Gated
	Ext Pulse	Scalar			

:PULM:STATe

```
[:SOURce]:PULM:STATe ON|OFF|1|0
```

```
[:SOURce]:PULM:STATe?
```

This command enables or disables pulse modulation for the selected path.

When pulse modulation is enabled, the PULSE annunciator appears on the signal generator's front-panel display.

Example

:PULM:STAT ON

The preceding example enables the pulse modulation.

*RST   0

Key Entry   Pulse Off On

5 SCPI Command Compatibility

This chapter provides a compatibility listing of SCPI commands. Many commands unique to other Agilent signal generator models are also supported by the E8663B Analog signal generator:

:SYSTem:IDN

Supported   All

```
:SYSTem:IDN "<string>"
```

This command modifies the identification string that the `*IDN?` query returns. Sending an empty string returns the query output to its factory shipped setting. The maximum string length is 72 characters.

Modification of the `*IDN?` query output enables the signal generator to identify itself as another signal generator when it is used as a backward compatible replacement.

The display diagnostic information, shown by pressing the **Diagnostic Info** softkey, is not affected by this command.

Example

```
:SYST:IDN "Agilent Technologies, Exxxx, US4000000, c.00.00.1234"
```

The preceding example changes and sets the identification string for the signal generator.

8662A/63A Compatible Commands

The tables in this section provide the following:

Table 5-1 on page 132: a comprehensive list of 8662A/63A programming commands, listed in alphabetical order. The equivalent SCPI command sequence for each supported code is provided. Codes that have no equivalent SCPI command sequence are indicated in the command column, as are codes that are *not* supported by the signal generator family.

Table 5-2 on page 139: a list of the implemented 8662A/63A programming commands that set the active function. This table also indicates which codes are compatible with the increment (up), and the decrement (down) SCPI commands.

To use the commands, select *866xA* as the remote language. See ":LANGuage" on page 54 for selecting the language type.

When using the programming codes in this section, you can:

- set the signal generator system language to 866xA for the current session:

  Utility > GPIB/RS-232 LAN > Remote Language > 866xA

  or send the command:

  :SYST:LANG "8662" or "8663"

- set the signal generator system language to 866xA so that it does not reset on a preset, an instrument power cycle or a *RST command:

  Utility > Power On/Preset > Preset Language > 866xA

  or send the command:

  :SYST:PRESET:LANG "8662" or "8663"

- set the *IDN? response to any 866xA-like response you prefer. Refer to the :SYSTem:IDNcommand on page 131.

NOTE   Compatibility is provided for GPIB only; RS-232 and LAN are *not* supported.

Device Clear does not preset the instrument.

To reproduce the sweep functionality, use the signal generator List Sweep features.

Table 5-1   8662A/63A Commands & Equivalent SCPI Sequences

Command	Description	8662	8663	Equivalent SCPI Command Sequence
@1	Write require service mask	N	N	*not supported*
@2	Deferred execution mode	N	N	*not supported*
@3	Immediate execution mode	N	N	*not supported*
+D	+dBm	Y	Y	DBM

Table 5-1   8662A/63A Commands & Equivalent SCPI Sequences (Continued)

Command	Description	8662	8663	Equivalent SCPI Command Sequence

AM	AM modulation
	
	See also: Table 5-2 on page 139
		Y		`AM:DEPTh <val> <units>`
				`AM:TRAC ON`
				`FM:STAT OFF`
				`AM:STAT ON`

			Y	`AM:DEPTh <val> <units>`
				`AM:TRAC ON`
				`AM:STAT ON`

AO	Amplitude off	Y	Y	`OUTPut:STATe OFF`

AP	Amplitude	Y	Y	`POW:REF:STATe OFF`
				`POWer:AMPL <val> <units>`
				`OUTPut:STATe ON`
				See also: Table 5-2 on page 139

AS|BLSQ	Auto sequence	N	N	*not supported*
BP	BPSK modulation		N	*not supported*
CT	Configure trigger	Y	Y	*no equivalent SCPI command sequence*
−D	−dBm Negates the power value.	Y	Y	`DBM`
DB	dB	Y	Y	`DB`
DG	Degree	Y		`DEG`
DM	dBm	Y	Y	`DBM`
DN	Decrement Passes `DOWN` as parameter of active function command.	Y	Y	*See* Table 5-2 on page 139
FA	Start frequency	Y	Y	*See W2, W3, W4,* *and* Table 5-2 on page 139
FB	Stop frequency	Y	Y	*See W2, W3, W4,* *and* Table 5-2 on page 139

FM	FM modulation
	
	See also: Table 5-2 on page 139
		Y		`FM:DEV <val> <units>`
				`AM:STAT OFF`
				`FM:STAT ON`

			Y	`FM:DEV <val> <units>` `FM:STAT ON`
FR	Center frequency	Y	Y	`FREQuency:CW <val> <units>` *See also: W2, W3, and W4,* *and* Table 5-2 on page 139
FS	Span frequency	Y	Y	*See W2, W3, W4,* *and* Table 5-2 on page 139
GZ	GHz	Y	Y	`GHZ`
HZ	Hz	Y	Y	`HZ`
IS	Set increment Adds `STEP:INCR` to active function command.	Y	Y	*no equivalent SCPI command sequence*
KZ	kHz	Y	Y	`KHZ`

Table 5-1   8662A/63A Commands & Equivalent SCPI Sequences (Continued)

Command	Description	8662	8663	Equivalent SCPI Command Sequence
L1	Learn front panel	N	N	*not supported*
L2	Fast learn	N	N	*not supported*
MO|M0	Modulation off	Y	Y	AM:STATe OFF FM:STATe OFF PULM:STATe OFF PM:STATe OFF
M1	**For 8662A:** <mod> = FM or AM, depending on which is on.	Y		<mod>:SOURce INT1 <mod>:INT1:FREQ 400Hz
	Modulation source internal 400 Hz **For 8663A:** Executes MF with <freq> = 400 Hz		Y	AM:INT1:FREQ 400 MHz FM:INT2:FREQ 400 MHz PM:INT2:FREQ 400 MHz PULM:INT:FREQ 400 MHz
M2	**For 8662A:** <mod> = FM or AM, depending on which is on.	Y		<mod>:SOURce INT1 <mod>:INT1:FREQ 1kHz
	Modulation source internal 1 kHz **For 8663A:** Executes MF with <freq> = 1 kHz		Y	AM:INT1:FREQ 1 kHzFM:INT2:FREQ 1 kHzPM:INT2:FREQ 1 kHzPULM:INT:FREQ 1 kHz
M3	**For 8662A:** <mod> = FM or AM, depending on which is on. Modulation source external AC	Y		<mod>:SOURce EXT <mod>:EXT:COUPling AC <mod>:EXT:IMP 600
	For 8663A: <mod> = AM, FM, or PM, depending on which is on. <n> = 1 for AM, 2 for FM or PM **NOTE**: For PM, the impedance value is set using the SP71/SP70 commands		Y	<mod>:SOURce EXT<n> <mod>:EXT<n>:COUPling AC <mod>:EXT<n>:IMP 600
M4	**For 8662A:** <mod> = FM or AM, depending on which is on. Modulation source external DC	Y		<mod>:SOURce EXT <mod>:EXT:COUPling DC <mod>:EXT:IMP 600
	For 8663A: <mod> = AM, FM, or PM, depending on which is on. <n> = 1 for AM, 2 for FM or PM **NOTE**: For PM, the impedance value is set using the SP71/SP70 commands		Y	<mod>:SOURce EXT<n> <mod>:EXT<n>:COUPling DC <mod>:EXT<n>:IMP 600
MF	Modulation frequency <mod> = FM, or PM, depending on which is on. *Also see*: *M1, M2, and* Table 5-2 on page 139		Y	**AM:** AM:SOUR: INT1 AM:SOUR:INT1:FREQ <freq> **FM or PM:** <mod>:SOUR: INT2 <mod>:SOUR:INT2:FREQ <freq> **Pulse:** PULM:SOUR: INT PULM:INT:FREQ <freq> PULM:SOUR:INT SQUARE
MS	Read status key message Returns status string.	Y	Y	*no equivalent SCPI command sequence*
MV	mV	Y	Y	MV
MZ	MHz	Y	Y	MHZ
N1	Linear 100 steps	Y	Y	*See W2, W3, and W4*

Table 5-1   8662A/63A Commands & Equivalent SCPI Sequences (Continued)

Command	Description	8662	8663	Equivalent SCPI Command Sequence
N2	Linear 1000 steps	Y	Y	*See W2, W3, and W4*
N3	Step size	Y	Y	*See W2, W3,W4,* *and* Table 5-2 on page 139
N4	Log 10% steps	Y	Y	*See W2, W3, and W4*
N5	Log 1% steps	Y	Y	*See W2, W3, and W4*
PC	%	Y	Y	PCT
PL	Pulse modulation Must have an instrument with pulse capability.		Y	PULM:STAT ON
PM	Phase modulation Not compatible with any FM modulation.		Y	PM:STAT ON *See also:* Table 5-2 on page 139
R1	Knob resolution x10	N	N	*not supported*
R2	Knob resolution /10	N	N	*not supported*
R3	Knob off	N	N	*not supported*
R4|BLR1	Knob hold	N	N	*not supported*
R5|BLR2	Knob increment	N	N	*not supported*
RC	Recall	Y	Y	*RCL
RD	Knob down Only for manual sweep	Y	Y	LIST:MANual DOWN
RM	Read require service mask	N	N	*not supported*
RU	Knob up Only for manual sweep	Y	Y	LIST:MANual UP
SP00	System preset Presets the instrument, including the compatibility language.	Y	Y	SYSTem:PRESet
SP10	Frequency offset off	Y	Y	FREQ:OFFS:STAT OFF
SP11	Positive frequency offset The 8662 modifies the output, but does not change the displayed frequency; the signal generator modifies the displayed frequency, but does *not* change the output. Because of this, you must first set the offset, then reapply the frequency to change the output.	Y	Y	FREQ:OFFS -<value> FREQ:OFFS:STAT ON FREQ:CW <displayed value>
SP12	Negative frequency offset The 8662 modifies the output, but does not change the displayed frequency; the signal generator modifies the displayed frequency, but does *not* change the output. Because of this, you must first set the offset, then reapply the frequency to change the output.	Y	Y	FREQ:OFFS <value> FREQ:OFFS:STAT ON FREQ:CW <displayed value>
SP20	ALC bandwidth normal		Y	POWer:ALC:BANDwidth:AUTO ON
SP21	ALC bandwidth < 1 kHz		Y	POWer:ALC:BANDwidth:AUTO OFFPOWer:ALC:BANDwidth 1KHZ
SP30	Amplitude reference off	Y	Y	POW:REF:STATe OFF

Table 5-1 8662A/63A Commands & Equivalent SCPI Sequences (Continued)

Command	Description	8662	8663	Equivalent SCPI Command Sequence
SP31	Amplitude reference	Y	Y	POW:REF <val> <val> = current amplitude setting POW:REF:STATe ON
SP32	Amplitude reference relative to 1 μV		Y	POW:REF 106.99DBM POW:REF:STATe ON POW 1UV
SP40	External AM off	Y		AM:STAT OFF
	Modulation frequency sweep mode off		N	*not supported*
SP41	Internal FM + external AM (AC)	Y		FM:SOUR INT1 FM:INT1:FREQ 400 HZ FM:STAT ON AM:SOUR EXT1 AM:EXT1:IMP 600 AM:DEPTH 95 PCT AM:EXT1:COUP AC AM:STAT ON
	Modulation frequency sweep mode on		N	*not supported*
SP42	Internal FM + external AM (DC)	Y		FM:SOUR INT1 FM:INT1:FREQ 400 HZ FM:STAT ON AM:SOUR EXT1 AM:EXT1:IMP 600 AM:DEPTH 95 PCT AM:EXT1:COUP DC AM:STAT ON
SP50	AUX FM off	Y	Y	FM2:STAT OFF
SP51	AUX FM on **RF (MHz) FM Deviation (kHz)** 0.01–120 25 120–160 6.25 160–320 12.5 320–640 25 640–1280 50 1280–2560 100 <dev> is dependant on output frequency, and mimics the 8662 hardware settings. **NOTE**: The deviation for this command cannot be greater than the deviation of the FM1 path.	Y	Y	FM2:SOUR EXT2 FM2:EXT2:COUP DC FM2:EXT2:IMP 600 FM2:DEV <dev> kHz FM2:STAT ON
SP60	Parameter shift keying off	N	N	*not supported*
SP61	Parameter shift keying up/down (two-key)	N	N	*not supported*
SP62	Parameter shift keying up/down (one-key)	N	N	*not supported*
SP70	External PM input impedance 50Ω Effects the behavior of M3 and M4.		Y	*no equivalent SCPI command sequence*
SP71	External PM input impedance 600Ω Effects the behavior of M3 and M4.		Y	*no equivalent SCPI command sequence*
SP80	Special functions 10-62 off	Y	Y	FM2:STAT OFF AM:STAT OFF FREQ:OFFS:STAT OFF

Table 5-1   8662A/63A Commands & Equivalent SCPI Sequences (Continued)

Command	Description	8662	8663	Equivalent SCPI Command Sequence
SP81	Amplitude conversion (V-dBm)	N	N	*not supported*
SP82	Display GPIB address	N	N	*not supported*
SP83	ROM test	N	N	*not supported*
SP84	RAM test	N	N	*not supported*
SP85	Amplitude correction off	Y	Y	POWer:ALC:STATe OFF
SP86	Amplitude correction on signal generator ALC ON always works with sweep.	Y	Y	POWer:ALC:STATe ON
SP87	Amplitude correction on (includes Sweep)		Y	POWer:ALC:STATe ON
SP87	GPIB operator request response	N		*not supported*
SP88	Auto sequence	N	N	*not supported*
SP89	GPIB operator request response		N	*not supported*
SP90	Set auto sequence step delay		N	*not supported*
SP91	Enable frequency hopping mode		N	*not supported*
SP92	Knob (restore normal operation)		N	*not supported*
SP93	Manual amplitude level control		N	*not supported*
SP94	Knob, 120 increments per revolution		N	*not supported*
SP95	Knob, 120 increments per revolution, reconfigure AUX con.		N	*not supported*
SP96	Modulation oscillator off when modulation is off		N	*not supported*
SP97	Modulation oscillator on		N	*not supported*
SP98	Turn display on		Y	DISP ON
SP99	Turn display off		Y	DISP OFF
SP2.0	Power up preset off		N	*not supported*
SP2.1	Power up preset on		N	*not supported*
SQ	Sequence	N	N	*not supported*
SS|BLST	Set sequence	N	N	*not supported*
ST	Store Saves/recalls register to sequence 0.	Y	Y	*SAV
T1	0.5 ms per step	Y	Y	SWEEP:DWELL 0.5ms *Beyond signal generator range limit; is set to 1ms.*
T2	1 ms per step	Y	Y	SWEEP:DWELL 1ms
T3	2 ms per step	Y	Y	SWEEP:DWELL 2ms
T4	10 ms per step	Y	Y	SWEEP:DWELL 10ms

Table 5-1 8662A/63A Commands & Equivalent SCPI Sequences (Continued)

Command	Description	8662	8663	Equivalent SCPI Command Sequence
T5	100 ms per step	Y	Y	`SWEEP:DWELL 100ms`
TR	Trigger Performs command code setup with `CT` command.	Y	Y	*no equivalent SCPI command sequence*
UP	Increment Passes `UP` as a parameter of the active function command.	Y	Y	*See Table 5-2 on page 139*
UV	mV	Y	Y	`UV`
W1	Sweep off	Y	Y	`FREQ:MODE CW` `LIST:TRIG:SOUR IMM`

W2	Auto sweep mode on
	
	Generates a sweep list based on stored parameters from FA, FB, FR, FS, N1, N2, N3, N4, and N5
	Default values:
	FR = 100 MHz,
	FS = 10 MHz,
	N1, T2 FA = 1 MHz,
	FB = 1279 MHz
		Y	Y	`INIT:CONT ON`
				`SWEEP:MODE AUTO`
				`LIST:TRIG:SOUR IMM`
				`LIST:DWEL1:TYPE STEP`
				`LIST:TYPE LIST`
				`FREQ:MODE LIST`

W3	Manual sweep mode on
	
	Generates a sweep list based on stored parameters from FA, FB, FR, FS, N1, N2, N3, N4, and N5
	Default values:
	FR = 100 MHz,
	FS = 10 MHz,
	N1, T2 FA = 1 MHz,
	FB = 1279 MHz
		Y	Y	`INIT:CONT ON`
				`SWEEP:MODE MANual`
				`LIST:TRIG:SOUR IMM`
				`LIST:DWEL1:TYPE STEP`
				`LIST:TYPE LIST`
				`FREQ:MODE LIST`

W4	Single sweep mode on
	
	Generates a sweep list based on stored parameters from FA, FB, FR, FS, N1, N2, N3, N4, and N5
	Default values:
	FR = 100 MHz,
	FS = 10 MHz,
	N1, T2 FA = 1 MHz,
	FB = 1279 MHz
		Y	Y	`INIT:CONT OFF`
				`SWEEP:MODE AUTO`
				`LIST:TRIG:SOUR IMM`
				`LIST:DWEL1:TYPE STEP`
				`LIST:TYPE LIST`
				`FREQ:MODE LIST`
				`INIT`

X1	Marker 1	N	N	*not supported*
X2	Marker 2	N	N	*not supported*
X3	Marker 3	N	N	*not supported*
X4	Marker 4	N	N	*not supported*
X5	Marker 5	N	N	*not supported*
X6	Marker off	N	N	*not supported*
X7|BLX6	All markers off	N	N	*not supported*
Y0	Remote stepped sweep off	Y	Y	`FREQ:MODE CW` `LIST:TRIG:SOUR IMM`

Table 5-1   8662A/63A Commands & Equivalent SCPI Sequences (Continued)

Command	Description	8662	8663	Equivalent SCPI Command Sequence
Y1|Y2	Remote stepped sweep on	Y	Y	INIT:CONT ON SWEEP:MODE AUTO LIST:DWEL1:TYPE STEP LIST:TYPE LIST FREQ:MODE LIST LIST:TRIG:SOUR BUS
Y3	Execute remote stepped sweep	Y	Y	*TRG

Table 5-2   8662/63B Command Compatibility

Command	Description	Sets Active Function	Compatible with UP/DN	8662	8663	Equivalent SCPI Commands for UP/DN and Increment
AM	AM modulation	Y	Y	Y	Y	AM:DEPTH UP AM:DEPTH DOWN AM:DEPTH:STEP:INCR
AP	Amplitude	Y	Y	Y	Y	POW:AMPL UP POW:AMPL DOWN POW:AMPL:STEP:INCR
FA	Start frequency	Y	Y	Y	Y	FREQ:CW:STEP:INCR
FB	Stop frequency	Y	Y	Y	Y	FREQ:CW:STEP:INCR
FM	FM modulation	Y	Y	Y	Y	FM:DEV UP FM:DEV DOWN FM:DEV:STEP:INCR
FR	Center frequency	Y	Y	Y	Y	FREQ:CW UP FREQ:CW DOWN FREQ:CW:STEP:INCR
FS	Span frequency	Y	Y	Y	Y	FREQ:CW:STEP:INCR
MF	Modulation frequency	Y	Y		Y	<mod>:INT:FREQ UP <mod>:INT:FREQ DOWN <mod>:INT:FREQ:STEP:INCR <mod> = AM|FM|PM|PULM
N3	Step size	Y	Y	Y	Y	*no equivalent SCPI commands*
PM	Phase modulation Not compatible with any FM modulation.	Y	Y		Y	PM:DEV UP PM:DEV DOWN PM:DEV:STEP:INCR

Index

Index

Agilent E8663B Analog Signal Generator SCPI Command Reference

Index

Index

Index

Index

Index