SCPI Command Reference

Agilent Technologies
PSG E8257D/67D Signal Generators

This guide applies to the following signal generator models:

E8257D PSG Analog Signal Generator

E8267D PSG Vector Signal Generator

Due to our continuing efforts to improve our products through firmware and hardware revisions, signal generator design and operation may vary from descriptions in this guide. We recommend that you use the latest revision of this guide to ensure you have up-to-date product information. Compare the print date of this guide (see bottom of page) with the latest revision, which can be downloaded from the following website:

http://www.agilent.com/find/psg

Agilent Technologies

Manufacturing Part Number: E8251-90356

Printed in USA

August 2004

Notice

Questions or Comments about our Documentation?

We welcome any questions or comments you may have about our documentation. Please send us an E-mail at **sources_manuals@am.exch.agilent.com**.

Contents

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 2
- "SCPI Basics" on page 3

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
- Digital 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

Supported Models and Options per Command

Within each command section, the Supported heading describes the signal generator configurations supported by the SCPI command. “All” means that all models and options are supported. When “All with Option xxx” is shown next to this heading, only the stated option(s) is supported.

SCPI Basics

This section describes the general use of the SCPI language for the PSG. 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

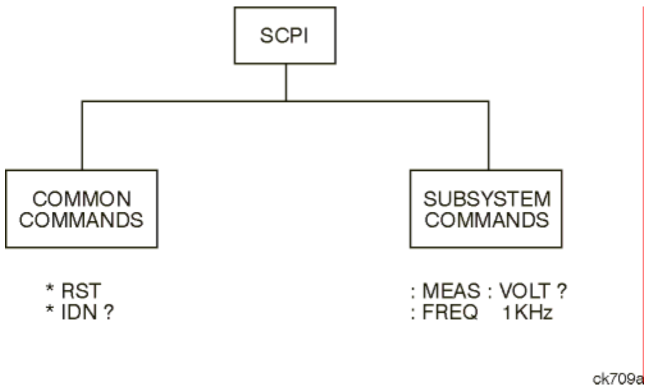
Characters, Keywords, and Syntax	Example
Upper-case lettering indicates the minimum set of characters required to execute the command.	[:SOURce]:FREQuency[:CW]?, FREQ is the minimum requirement.
Lower-case lettering indicates the portion of the command that is optional; it can either be included with 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 7 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 6 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.

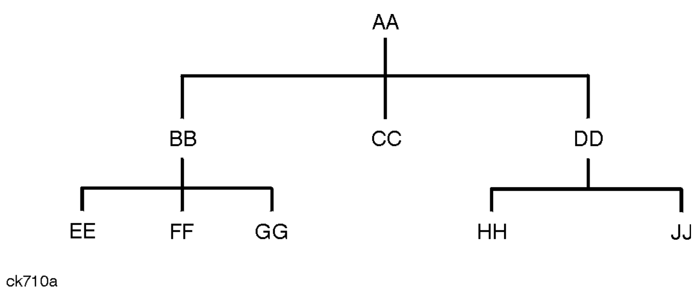
Figure 1-1 **Command Types**



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**



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

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 "<file name>","<file name>"
```

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 55 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 12. 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 variables

```
: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 12:

```
: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 4 for a listing of the file systems and types.

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 common to all PSG models:

Calibration Subsystem (:CALibration)

:DCFM

Supported All with Option UNT

`: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 disconnected to reset the carrier signal to the correct zero reference.

Key Entry **DCFM / DCΦM Cal**

:IQ

Supported E8267D

`:CALibration:IQ`

This command initiates an I/Q calibration for a range of frequencies, which is equivalent to the User selection from the front panel **Calibration Type DC User Full** softkey. For setting range frequencies, refer to “:IQ:STARt” , and “:IQ:STOP” command descriptions.

Key Entry **Execute Cal** **Calibration Type DC User Full**

:IQ:DC

Supported E8267D

`:CALibration:IQ:DC`

This command starts and performs a one-to two-second adjustment that is not traceable to a standard. However, it will minimize errors associated with signal generator internal voltage offsets. This adjustment minimizes errors for the current signal generator setting and at a single frequency. The DC adjustment is volatile and must be repeated with each signal generator setting change. This command can be sent while the RF On/Off is set to Off and the adjustment will still be valid when the RF is enabled.

The I/Q DC adjustment is dependent upon a number of instrument settings. If any of the instrument settings change, the adjustment will become invalid. The dependent instrument settings are:

- RF frequency
- I/Q attenuation level
- Baseband generator settings
- I/Q polarity settings
- Baseband filter settings
- Path settings (Internal I/Q Mux Path 1 or Path 2)
- I/Q calibration (the I/Q DC calibration will be invalidated if any other I/Q calibration is execute)
- Temperature (±5 degrees)

The following instrument states will not invalidate the I/Q DC calibration:

- Power level changes
- I/Q Impairments

***RST** N/A

Key Entry **Execute Cal** **Calibration Type DC User Full**

:IQ:DEFault

Supported E8267D

```
:CALibration:IQ:DEFault
```

This command will restore the original factory calibration data for the internal I/Q modulator.

Key Entry **Revert to Default Cal Settings**

:IQ:FULL

Supported E8267D

```
:CALibration:IQ:FULL
```

This command sets and performs a full-frequency range (regardless of the start and stop frequency settings) I/Q calibration and stores the results in the signal generator’s firmware.

Start and stop frequencies will default to the full frequency range of the signal generator.

Key Entry **Execute Cal** (**Calibration Type DC User Full** set to **Full**)

:IQ:STARt

Supported E8267D

```
:CALibration:IQ:STARt <val><units>
:CALibration:IQ:STARt?
```

This command sets the start frequency and automatically sets the calibration type to User for an I/Q calibration.

The variable `<val>` is the frequency number. The variable `<units>` is the frequency suffix.

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

Range Depends on the signal generator’s frequency option
Refer to “:FREQuency:CENTer” on page 98

Key Entry **Start Frequency**

:IQ:STOP

Supported E8267D

```
:CALibration:IQ:STOP <val><units>
:CALibration:IQ:STOP?
```

This command sets the stop frequency and automatically sets the calibration type to User for an I/Q calibration.

The variable `<val>` is the frequency number. The variable `<units>` is the frequency suffix.

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

Range Depends on the signal generator’s frequency option
Refer to “:FREQuency:CENTer” on page 98

Key Entry **Stop Frequency**

:WBIQ

Supported E8267D with Option 015

```
:CALibration:WBIQ
```

This command initiates a wideband I/Q calibration. The type of calibration can be DC, USER, or FULL, selected using the `:CALibration:WBIQ:<Calibration Type>` command.

Key Entry **Execute Cal**

:WBIQ:DC

Supported E8267D with Option 015

`:CALibration:WBIQ:DC`

This command performs a one to two second adjustment that is not traceable to a standard. However, it will minimize errors associated with offset voltages. This adjustment minimizes errors for the current signal generator setting and at a single frequency. The DC adjustment is volatile and must be repeated with each signal generator setting change. This command can be sent while the RF On/Off is set to Off and the adjustment will be valid when RF is enabled.

The WBI/Q DC adjustment is dependent upon a number of instrument settings. If any of the instrument settings change, the adjustment will become invalid. The dependent instrument settings are:

- RF frequency
- I/Q attenuation level
- Baseband generator settings
- I/Q polarity settings
- Baseband filter settings
- Path settings (Internal I/Q Mux Path 1 or Path 2)
- I/Q calibration (the I/Q DC calibration will be invalidated if any other I/Q calibration is executed)
- Temperature (±5 degrees)

The following instrument states will not invalidate the I/Q DC calibration:

- Power level changes
- I/Q Impairments

***RST** N/A

Key Entry **Execute Cal** **Calibration Type DC User Full**

:WBIQ:DEFault

Supported E8267D with Option 015

`:CALibration:WBIQ:DEFault`

This command will restore the original factory calibration data for the internal I/Q modulator.

Key Entry **Revert to Default Cal Settings**

:WBIQ:FULL

Supported E8267D with Option 015

```
:CALibration:WBIQ:FULL
```

This command sets and performs a full-frequency range (regardless of the start and stop frequency settings) wideband I/Q calibration and stores the results in the signal generator’s firmware.

Start and stop frequencies will default to the full frequency range of the signal generator.

Range Depends on the signal generator’s frequency option
Refer to “:FREQuency:CENTer” on page 98

Key Entry **Execute Cal** **Calibration Type DC User Full**

:WBIQ:STARt

Supported E8267D with Option 015

```
:CALibration:WBIQ:STARt <val><units>
:CALibration:WBIQ:STARt?
```

This command sets the start frequency and automatically sets the calibration type to User for an I/Q calibration.

The variable `<val>` is the frequency number. The variable `<units>` is the frequency suffix.

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

Range Depends on the signal generator’s frequency option
Refer to “:FREQuency:CENTer” on page 98

Key Entry **Start Frequency**

:WBIQ:STOP

Supported E8267D with Option 015

```
:CALibration:WBIQ:STOP <val><units>
:CALibration:WBIQ:STOP?
```

This command sets the stop frequency and automatically sets the calibration type to User for an I/Q calibration.

The variable `<val>` is the frequency number. The variable `<units>` is the frequency suffix.

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

Range Depends on the option available.

Key Entry **Stop Frequency**

Communication Subsystem (:SYSTem:COMMunicate)

:GPIB:ADDRess

Supported All Models

```
: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 signal generator power-on, preset, or `*RST`.

Range 0–30

Key Entry **GPIB Address**

:GTLocal

Supported All

```
:SYSTem:COMMunicate:GTLocal
```

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

Range N/A

Key Entry **Local**

:LAN:CONFig

Supported All Models

```
: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.

Key Entry **LAN Config**

:LAN:GATEway

Supported All Models

```
: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>"` variable is the IP (internet protocol) address, formatted as xxx.xxx.xxx.xxx for the signal generator.

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

Key Entry **Default Gateway**

:LAN:HOSTname

Supported All Models

```
: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.

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

Key Entry **Hostname**

:LAN:IP

Supported All Models

```
: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 IP address, formatted as xxx.xxx.xxx.xxx, for the signal generator.

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

Key Entry **IP Address**

:LAN:SUBNet

Supported All Models

```
: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, for the signal generator.

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

Key Entry **Subnet Mask**

:PMETer:ADDRess

Supported All Models

```
: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 a numeric value between 0 and 30. The power meter address must be different from the address of the signal generator and any other GPIB instrument addresses in your system.

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

Range 0–30

Key Entry **Meter Address**

:PMETer:CHANnel

Supported All Models

```
: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 signal generator power-on, preset, or `*RST`.

The power meter is controlled only through a general purpose instrument bus (GPIB) cable.

Key Entry **Meter Channel A B**

:PMETer:IDN

Supported All Models

```
: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 signal generator power-on, preset, or `*RST`. The power meter is controlled only through a general purpose instrument bus (GPIB) cable.

Key Entry **Power Meter**

:PMETer:TIMEout

Supported All Models

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

This command sets the period of time which 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 units of time, for example msec (microseconds).

The setting enabled by this command is not affected by signal generator power-on, preset, or `*RST`. The power meter is controlled only through a GPIB cable. If a time-out occurs, the signal generator reports an error message.

Range 1mS–100S

Key Entry **Meter Timeout**

:SERial:BAUD

Supported All Models

```
: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 signal generator power-on, preset, or `*RST`.

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

Key Entry **RS-232 Baud Rate**

:SERial:ECHO

Supported All Models

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

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

Key Entry **RS-232 ECHO Off On**

:SERial:RESet

Supported All Models

```
:SYSTem:COMMunicate:SERial:RESet
```

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

Key Entry **Reset RS-232**

:SERial:TOUT

Supported All Models

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

This sets the RS-232 serial port time-out value. If further input is not received within the time-out 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 power-on, preset, or `*RST`.

Range 1–25

Key Entry **RS-232 Timeout**

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

:BOARds

Supported All Models

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

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

`"<board name,part number,serial number,version number,status>"`

This information format will repeat for each detected board in the signal generator.

Key Entry **Installed Board Info**

:CCOunt:ATTenuator

Supported E8267D and E8257D with Option 1E1

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

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

Key Entry **Diagnostic Info**

:CCOunt:PON

Supported All Models

`: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

Supported All Models

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

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

Key Entry **Diagnostic Info**

:LICENse:AUXiliary

Supported All Models

```
: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

Supported All Models

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

This query returns a list of internally installed signal generator options.

Key Entry **Options Info**

:OPTions:DETail

Supported All Models

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

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

Key Entry **Options Info**

:OTIMe

Supported All Models

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

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

Key Entry **Diagnostic Info**

:REVision

Supported All Models

`: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 of the main firmware.

Key Entry **Diagnostic Info**

:SDATe

Supported All Models

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

This query returns the date and time of the main firmware.

Key Entry **Diagnostic Info**

Display Subsystem (:DISPlay)

:ANNotation:AMPLitude:UNIT

Supported All Models

```
: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, ":REFerence:STATe" on page 131 for more information.

***RST** **dBm**

:ANNotation:CLOCk:DATE:FORMat

Supported All Models

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

This command enables the selection of the date format. The choices are month-day-year (MDY) or day-month-year (DMY) format.

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

:ANNotation:CLOCk[:STATe]

Supported All Models

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

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

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

:BRIGhtness

Supported All Models

```
: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 signal generator power-on, preset, or `*RST`.

Range 0.02–1

Key Entry **Brightness**

:CAPTure

Supported All Models

```
:DISPlay:CAPTure
```

This event command enables 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

Supported All Models

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

This command sets the contrast of the LCD 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 signal generator power-on, preset, or `*RST`.

Range 0–1

Key Entry Display contrast hardkeys are located below the display.

:INVerse

Supported All Models

```
: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 signal generator power-on, preset, or *RST.

Key Entry **Inverse Video Off On**

:REMote

Supported All Models

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

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

ON (1) This choice updates the signal generator display so you can see the settings as the commands are executed, however, this will decrease the signal generator speed.

OFF (0) This choice turns off the display updating while further optimizing the signal generator for speed.

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

Key Entry **Update in Remote Off On**

[:WINDow][:STATe]

Supported All Models

```
: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).

The setting enabled by this command is not affected by *RST. However, presetting the signal generator or cycling the power will turn the display on.

IEEE 488.2 Common Commands

*CLS

Supported All Models

`*CLS`

The Clear Status (CLS) command clears the Status Byte Register, the Data Questionable Event Register, the Standard Event Status Register, the Standard Operation Status Register and any other registers that are summarized in the status byte.

*ESE

Supported All Models

`*ESE <data>`

The Standard Event Status Enable (ESE) command sets the Standard Event Status Enable Register. The variable `<data>` represents the sum of the bits that will be enabled. The setting enabled by this command is not affected by signal generator preset or `*RST`, but cycling the signal generator power resets this register to zero. Refer to the *Programming Guide* for more information.

Range 0–255

*ESE?

Supported All Models

`*ESE?`

The Standard Event Status Enable (ESE) query returns the value of the Standard Event Status Enable Register. Refer to the *Programming Guide* for more information.

*ESR?

Supported All Models

CAUTION 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 *Programming Guide* for more information.

`*ESR?`

This query returns the value of the Standard Event Status Register (ESR).

*IDN?

Supported All Models

`*IDN?`

The Identification (IDN) query outputs an identifying string. The response will show the following information:

`<company name>, <model number>, <serial number>, <firmware revision>`

The identification information can be modified. Refer to ":IDN" on page 81 for more information.

Key Entry **Diagnostic Info**

*OPC

Supported All Models

`*OPC`

The Operation Complete (OPC) command sets bit 0 in the Standard Event Status Register when all pending operations have finished.

*OPC?

Supported All Models

`*OPC?`

The Operation Complete (OPC) query returns the ASCII character 1 in the Standard Event Status Register when all pending operations have finished.

*PSC

Supported All Models

`*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.

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.

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

*PSC?

Supported All Models

```
*PSC?
```

The Power-On Status Clear (PSC) query returns the flag setting as enabled by the *PSC command.

*RCL

Supported All Models

```
*RCL <reg>,<seq>
```

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

Range *Registers*: 0–99 *Sequences*: 0–9

Key Entry **RECALL Reg** **Select Seq:**

*RST

Supported All Models

```
*RST
```

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

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

*SAV

Supported All Models

```
*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>.

Range *Registers*: 0–99 *Sequences*: 0–9

Key Entry **Save Reg** **Save Seq[n] Reg[nn]**

*SRE

Supported All Models

`*SRE <data>`

The Service Request Enable (SRE) command sets the value of the Service Request Enable Register.

The variable <data> is the decimal sum of the bits that will be enabled. Bit 6 (value 64) is ignored and cannot be set by this command.

Refer to the *Programming Guide* for more information.

Entering values from 64 to 127 is equivalent to entering values from 0 to 63.

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

Range 0–255

*SRE?

Supported All Models

`*SRE?`

The Service Request Enable (SRE) query returns the value of the Service Request Enable Register.

Refer to the *Programming Guide* for more information.

Range 0–63 or 128–191

*STB?

Supported All Models

`*STB?`

The Read Status Bye (STB) query returns the value of the status byte including the master summary status (MSS) bit. Refer to the *Programming Guide* for more information.

Range 0–255

*TRG

Supported All Models

`*TRG`

The Trigger (TRG) command triggers the device if BUS is the selected trigger source, otherwise, *TRG is ignored.

*TST?

Supported All Models

`*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

Supported All Models

`*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

Supported All Models

```
:MEMory:CATalog:BINary?
```

This command outputs a list of the 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 12 for information on the file name syntax.

Key Entry **Binary**

:CATalog:BIT

Supported E8267D with Option 601 or 602

```
:MEMory:CATalog:BIT?
```

This command outputs a list of the bit 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 12 for information on the file name syntax.

Key Entry **Bit**

:CATalog:DMOD

Supported E8267D with Option 601 or 602

```
:MEMory:CATalog:DMOD?
```

This command outputs a list of the arbitrary waveform digital modulation 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 12 for information on the file name syntax.

Key Entry DMOD

:CATalog:FIR

Supported E8267D with Option 601 or 602

`:MEMory:CATalog:FIR?`

This command outputs a list of the finite impulse response (FIR) filter 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 12 for information on the file name syntax.

Key Entry FIR

:CATalog:FSK

Supported E8267D with Option 601 or 602

`:MEMory:CATalog:FSK?`

This command outputs a list of the frequency shift keying (FSK) 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 12 for information on the file name syntax.

Key Entry FSK

:CATalog:IQ

Supported E8267D with Option 601 or 602

`:MEMory:CATalog:IQ?`

This command outputs a list of the IQ 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 12 for information on the file name syntax.

Key Entry **I/Q**

:CATalog:LIST

Supported All Models

`:MEMory:CATalog:LIST?`

This command outputs a list of the 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 12 for information on the file name syntax.

Key Entry **List**

:CATalog:MDMod

Supported E8267D with Option 601 or 602

`:MEMory:CATalog:MDMod?`

This command outputs a list of the arbitrary waveform multicarrier digital modulation 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 12 for information on the file name syntax.

Key Entry **MDMOD**

:CATalog:MTONe

Supported E8267D with Option 601 or 602

```
:MEMory:CATalog:MTONe?
```

This command outputs a list of the arbitrary waveform multitone 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 12 for information on the file name syntax.

Key Entry **MTONE**

:CATalog:SEQ

Supported E8267D with Option 601 or 602

```
:MEMory:CATalog:SEQ?
```

This command outputs a list of the arbitrary waveform sequence 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 12 for information on the file name syntax.

Key Entry **Seq**

:CATalog:SHAPe

Supported E8267D with Option 601 or 602

```
:MEMory:CATalog:SHAPe?
```

This command outputs a list of the burst shape 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 12 for information on the file name syntax.

Key Entry **Shape**

:CATalog:STATe

Supported All Models

```
:MEMory:CATalog:STATe?
```

This command outputs a list of the 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 12 for information on the file name syntax.

Key Entry **State**

:CATalog:UFLT

Supported All Models

```
:MEMory:CATalog:UFLT?
```

This command outputs a list of the 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 12 for information on the file name syntax.

Key Entry **User Flatness**

:CATalog[:ALL]

Supported All Models

```
:MEMory:CATalog[:ALL]?
```

This command outputs a list of all files in the memory subsystem, but does not include files stored in the Option 601 or 602 baseband generator memory. 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 55 for file types, and "File Name Variables" on page 12 for syntax.

Key Entry **All**

:COPY[:NAME]

Supported All Models

```
:MEMory:COPY[:NAME] "<file name>","<file name>"
```

This command makes a duplicate of the requested file. Refer to "File Name Variables" on page 12 for information on the file name syntax.

Key Entry **Copy File**

:DATA

Supported All Models

```
:MEMory:DATA "<file name>",<datablock>
:MEMory:DATA? "<file name>"
```

This command loads <datablock> into the memory location designated by "<file name>". The query returns the <datablock> associated with the "<file name>"

"<file name>" Represents the user file stored in the signal generator non-volatile memory. The filename must be enclosed with quotation marks.

<datablock> This variable represents the data block.

Example:

```
:MEMory:DATA "userfile", #1912S4078ë7
```

userfile This is the file as it appears in the signal generator's memory.

1	This variable defines the number of decimal digits to follow.
9	This variable defines how many bytes of data are to follow.
12S4078ë7	This is the ASCII representation of the data that is downloaded to the signal generator.

Refer to “File Name Variables” on page 12 for information on the file name syntax.

:DATA:APPend

Supported All

`:MEMory:DATA:APPend ",<file name>",<datablock>`

This commands appends new data to an existing binary user file stored in non-volatile signal generator memory.

`"<file name>"`	This variable represents the name of the existing user file.
`<datablock>`	This variable represents the block-formatted data.

Example:

`:MEMory:DATA:APPend "userfile", #141ë49`

userfile	This name represents the user file name as it appears in the signal generator.
1	This variable defines the number of decimal digits to follow.
4	This variable defines how many bytes of data are to follow.
1ë49	This example is an ASCII representation of the data that is downloaded to the signal generator. Refer to the *PSG Programming Guide* for more information.

***RST** N/A

Range N/A

Key Entry N/A

Refer to “File Name Variables” on page 12 for information on the file name syntax.

:DATA:BIT

Supported E8267D with Option 601 or 602

```
:MEMory:DATA:BIT "<file name>",<bit_count>,<datablock>
:MEMory:DATA:BIT? "<file name>"
```

This command creates a bit file and stores it in the signal generator non-volatile memory.

`"<file name>"`	This variable represents the user file name as it will appear in the signal generator memory. The filename must be enclosed with quotation marks.
`<bit_count>`	This variable represents the number of significant bits in the data block.
`<datablock>`	This variable represents the data block.

Example:

```
:MEMory:DATA:BIT "userfile1", 16, #12Qz
```

`"userfile1"`	This is the name of the user file as it appears in the signal generator.
16	This variable defines the actual number of data bits contained in the datablock.
1	This variable defines the number of decimal digits to follow.
2	This variable defines how many bytes of data are to follow.
Qz	This variable defines the ASCII representation of the 16 bits of data that are downloaded to the signal generator.

Refer to “File Name Variables” on page 12 for information on the file name syntax.

:DATA:FIR

Supported E8267D with Option 601 or 602

```
:MEMory:DATA:FIR "<file name>",osr,coefficient{,coefficient}
:MEMory:DATA:FIR? "<file name>"
```

This command creates a user-defined finite impulse response (FIR) file and stores it in the signal generator non-volatile memory (NVWFM).

osr	The oversample ratio (osr) is the number of filter taps per symbol.
coefficient	This variable is the FIR coefficient. The maximum total number of coefficients is 1024.
{,coefficient}	This optional variable is used when you enter additional coefficients.

Refer to “File Name Variables” on page 12 for information on the file name syntax.

Range osr: 1–32

coefficient: –1000 to 1000

Key Entry **Oversample Ratio**

:DATA:FSK

Supported E8267D with Option 601 or 602

```
:MEMory:DATA:FSK "<file name>",<num_states>,<f0>,<f1>,...<f(n)>
[,<diff_state>,<num_diff_states>,<diff1>,...<diff(n)>]
:MEMory:DATA:FSK? "<file name>"
```

This command creates a custom FSK file and stores it in the signal generator non-volatile memory.

The query returns data in the following form:

```
<num_states>,<f0>,<f1>,...<f(n)>,<diff_state>,<num_diff_states>,<diff1>,
...<diff(n)>
```

`"<file name>"`	This variable string identifies the name of the FSK file. The filename must be enclosed with quotation marks.
`<num_states>`	This variable identifies the number of frequency states.
`<f0>`	This variable identifies the value of the first frequency state.
`<f1>,...<f(n)>`	This variable identifies the value of the second and subsequent frequency states with a frequency resolution of 0.1Hz.
`<diff_state>`	This variable enables or disables differential encoding.
`<num_diff_states>`	This variable identifies the number of differential states.
`<diff0>`	This variable identifies the value of the first differential state.
`<diff1>,...<diff(n)>`	This variable identifies the value of the second and subsequent differential states.

The following example creates and stores a four-level FSK file named 4FSK that has four states (frequencies): –2kHZ, –1kHZ, 2kHZ, 1kHZ; differential encoding is toggled ON, and there are two differential states 1 and 0.

```
:MEM:DATA:FSK "4FSK",4,–2kHZ,–1kHZ,2kHZ,1kHZ,ON,2,1,0
```

Range num_diff_states: 0–256

num_states: 2–16

f0–f(n): –20MHZ to 20MHZ

diff0–diff(n): –128 to 127

:DATA:IQ

Supported E8267D with Option 601 or 602

```
:MEMory:DATA:IQ "<file name>",<offsetQ>,<num_states>,<i0>,<q0>,<i1>,
<q1>,...<i(n)>,<q(n)>
[,<diff_state>,<num_diff_states>,<diff0>,<diff1>,...<diff(n)>]
:MEMory:DATA:IQ? "<file name>"
```

This command creates a custom I/Q file and stores it in the signal generator non-volatile memory (NVWFM).

The query returns data in the following form:

```
<offsetQ>,<num_states>,<i0>,<q0>,<i1>,<q1>,...<i(n)>,<q(n)>,<diff_state>
,<num_diff_states>,<diff0>,<diff1>,...<diff(n)>
```

<file name> This variable string identifies the name of the I/Q file. The filename must be enclosed with quotation marks.

<offsetQ> This variable enables or disables the Q output delay by 1/2 symbol from the I output.

<num_states> This variable identifies the number of symbols.

<i0>...<i(n)> This variable identifies the I value of the first and subsequent I symbols.

<q0>...<q(n)> This variable identifies the Q value of the first and subsequent Q symbols.

<diff_state> This variable enables and disables differential encoding.

<num_diff_states> This variable identifies the number of differential states.

<diff0> This variable identifies the value of the first differential state.

<diff1,...diff(n)> This variable identifies the value of the second and subsequent differential states.

The following example creates and stores a two-symbol I/Q file named `testBPSK` that has the Q offset.

```
:MEM:DATA:IQ "testBPSK",1,2,1,0,0,0
```

Range num_states: 2–256

i0–i(n): –1 to 1

q0–q(n): –1 to 1

num_diff_states: 0–256

diff0–diff(n): -128 to 127

:DATA:PRAM:BLOCk

Supported E8267D with Option 601 or 602

```
:MEMory:DATA:PRAM:BLOCk <"<filename>"<datablock>
```

This command downloads block-formatted data directly into pattern RAM volatile memory. PRAM files are specified as an array of bytes.

:DATA:PRAM:LIST

Supported E8267D with Option 601 or 602

```
:MEMory:DATA:PRAM:LIST "<filename>",<uint8>[,<uint8>,<...>]
```

This command downloads the list-formatted data directly into pattern RAM volatile memory. This command should be preceded by a *WAI (Wait-to-Continue) command to ensure that all pending commands are completed, before loading the list.

`"<filename>"` This variable is the file name of the file downloaded to pattern RAM memory.

`<uint8>` This variable is any of the valid 8-bit, unsigned integer values between 0 and 255.

`[,<uint8>,<...>]` This variable identifies the value of the second and subsequent 8-bit unsigned integer variables.

Range 0–255

:DATA:SHAPe

Supported E8267D with Option 601 or 602

```
:MEMory:DATA:SHAPe <"file name">,<num_rise_points>,<rp0>,
<rp1>,...<num_fall_points>,<fp0>,<fp1>,...<fp(n)>
:MEMory:DATA:SHAPe? <"file name">
```

This command creates a new burst shape file and stores it in the signal generator non-volatile memory.

`"<filename>"` This variable string identifies the name of the burst shape file.

`num_rise_points` This variable specifies how many rise points used in the command.

`rp0,...rp(n)` This variable defines each successive rise point, where 0 is no power and 1 is full power.

`num_fall_points` specifies how many fall points used in the command.

`fp0,...fp(n)` defines each successive fall point, where 0 is no power and 1 is full power.

Range	num_rise_points: 2–256
	num_fall_points: 2–256
	rp0–rp(n): 0.0–1.0
	fp0–fp(n): 0.0–1.0

:DATA:UNPRotected

Supported E8267D with Option 601 or 602

```
:MEMory:DATA:UNPRotected <"<filename>"<datablock>
```

This command downloads waveform data to the signal generator's /USER/SECUREWAVE directory. The `UNPRotected` command parameter allows waveform data to be retrieved or uploaded from the signal generator.

Waveform files created with Agilent's Signal Studio are encrypted. These files can be used in other signal generators (provided the other signal generator has the same options and licenses required by the file) only if the SECUREWAVE directory is used for both downloading and uploading. Refer to the *Programming Guide* for more information.

:DELete:ALL

Supported All Models

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. However, this does not include files stored in the Option 601 or 602 baseband generator memory. 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

Supported All Models

```
:MEMory:DELete:BINary
```

This command deletes all binary files.

Key Entry **Delete All Binary Files**

:DELete:BIT

Supported E8267D with Option 601 or 602

`:MEMory:DELete:BIT`

This command deletes all bit files.

Key Entry **Delete All Bit Files**

:DELete:DMOD

Supported E8267D with Option 601 or 602

`:MEMory:DELete:DMOD`

This command deletes all arbitrary waveform digital modulation (DMOD) files.

Key Entry **Delete All ARB DMOD Files**

:DELete:FIR

Supported E8267D with Option 601 or 602

`:MEMory:DELete:FIR`

This command deletes all finite impulse response (FIR) filter files.

Key Entry **Delete All FIR Files**

:DELete:FSK

Supported E8267D with Option 601 or 602

`:MEMory:DELete:FSK`

This command deletes all frequency shift keying (FSK) files.

Key Entry **Delete All FSK Files**

:DELete:IQ

Supported E8267D with Option 601 or 602

`:MEMory:DELete:IQ`

This command deletes all I/Q files.

Key Entry **Delete All I/Q Files**

:DELete:LIST

Supported All Models

`:MEMory:DELete:LIST`

This command deletes all List files.

Key Entry **Delete All List Files**

:DELete:MDMod

Supported E8267D with Option 601 or 602

`:MEMory:DELete:MDMod`

This command deletes all arbitrary waveform multicarrier digital modulation (MDMod) files.

Key Entry **Delete All ARB MDMOD Files**

:DELete:MTONe

Supported E8267D with Option 601 or 602

`:MEMory:DELete:MTONe`

This command deletes all arbitrary waveform multitone files.

Key Entry **Delete All ARB MTONE Files**

:DELete:SEQ

Supported E8267D with Option 601 or 602

`:MEMory:DELete:SEQ`

This command deletes all sequence files.

Key Entry **Delete All Sequence Files**

:DELete:SHAPe

Supported E8267D with Option 601 or 602

`:MEMory:DELete:SHAPe`

This command deletes all burst shape files.

Key Entry **Delete All Shape Files**

:DELete:STATe

Supported All Models

`:MEMory:DELete:STATe`

This command deletes all state files.

Key Entry **Delete All State Files**

:DELete:UFLT

Supported All Models

`:MEMory:DELete:UFLT`

This command deletes all user-flatness correction files.

Key Entry **Delete All UFLT Files**

:DELete[:NAME]

Supported All Models

`:MEMory:DELete[:NAME] "<file name>"`

This clears the user file system of `"<file name>"`. See "File Name Variables" on page 12 for syntax. When deleting an ARB waveform file, the associated marker and header files are also deleted.

Key Entry **Delete File**

:FREE[:ALL]

Supported All Models

`:MEMory:FREE[:ALL]?`

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

Key Entry **All**

:LOAD:LIST

Supported All Models

`:MEMory:LOAD:LIST "<file name>"`

This command loads a list sweep file.

Key Entry **Load From Selected File**

:MOVE

Supported All Models

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

This command renames the `src_file` to the `dest_file` in the memory catalog. Refer to “File Name Variables” on page 12 for information on the file name syntax.

Key Entry **Rename File**

:STATe:COMMent

Supported All Models

```
: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 state <reg_num>,<seq_num>. Comments can be up to 55 characters long.

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

:STORe:LIST

Supported All Models

```
:MEMory:STORe:LIST "<file name>"
```

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

Key Entry **Store To File**

Mass Memory Subsystem (:MMEMory)

:CATalog

Supported All Models

`: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
BINARY	BIN
BIT	BIT
DMOD - ARB digital modulation file	DMOD
FIR - finite impulse response filter file	FIR
FSK - frequency shift keying modulation file	FSK
I/Q - modulation file	IQ
LIST - sweep list file	LIST
MDMOD - ARB multicarrier digital modulation file	MDM
MTONE - ARB multitone file	MTON
NVMKR - non-volatile arbitrary waveform marker file	NVMKR
NVWFM - non-volatile arbitrary waveform file	NVWFM
SEQ - ARB sequence file	SEQ
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 14 for information on the use of the "<msus>" variable.

Key Entry **Binary** **List** **State** **User Flatness** **FIR** **Shape** **Bit** **FSK** **I/Q** **Seq** **DMOD** **MTONE** **MDMOD** **WFM1** **NVMKR** **NVWFM**

:COPY

Supported All Models

```
:MMEMory:COPY "<file name>","<file name>"
```

This command makes a duplicate of the requested file. Refer to “File Name Variables” on page 12 for information on the file name syntax.

Key Entry **Copy File**

:DATA

Supported All Models

```
:MMEMory:DATA "<file name>",<datablock>
:MMEMory:DATA? "<file name>"
```

This command loads <datablock> into the memory location designated by "<file name>". The query returns the <datablock> associated with the "<file name>".

"<file name>" Represents the user file stored in the signal generator non-volatile memory. The filename must be enclosed with quotation marks.

<datablock> This variable represents the block-formatted data.

Example:

```
:MEMory:DATA "userfile", #1912S4078ë7
```

userfile This is the user file as it appears in the signal generator.

1 This variable defines the number of decimal digits to follow.

9 This variable defines how many bytes of data are to follow.

12S4078ë7 This is the ASCII representation of the data that is downloaded to the signal generator.

Refer to “File Name Variables” on page 12 for information on the file name syntax.

:DELete:NVWFm

Supported E8267D with Option 601 or 602

`:MMEMory:DELete:NVWFm`

This command clears the user file system of all non-volatile arbitrary waveform (NVWFM) files.

Key Entry **Delete All NVWFM Files**

:DELete:WFM

Supported E8267D with Option 601 or 602

`:MMEMory:DELete:WFM`

This command clears the user file system of all volatile arbitrary waveform (WFM1) files. It performs the same function as DELete:WFM1.

Key Entry **Delete All WFM1 Files**

:DELete[:NAME]

Supported All Models

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

This command clears the user file system of `"<file name>"` with the option of specifying the file system 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 55.

If the optional variable `"<msus>"` is omitted, the file name needs to include the file system extension. Refer to “File Name Variables” on page 12 and “MSUS (Mass Storage Unit Specifier) Variable” on page 14 for information on the use of the file variables.

Key Entry **Delete File**

:HEADer:CLEar

Supported All Models

`:MMEMory:HEADer:CLEar "<file name>"`

This command deletes the header file for the waveform file `"<file name>"`. This command does not require a personality modulation to be on. The header file contains signal generator settings and marker routings associated with the waveform file.

***RST** N/A

Key Entry **Clear Header**

:HEADer:DESCription

Supported All Models

```
:MMEMory:HEADer:DESCription "<file name>","<description>"
:MMEMory:HEADer:DESCription? "<file name>"
```

This command inserts a description for the header file named. The header description is limited to 32 characters. Refer to the *PSG User's Guide* for more information on header files.

***RST** N/A

Key Entry **Edit Description**

:LOAD:LIST

Supported All Models

```
:MMEMory:LOAD:LIST "<file name>"
```

This command loads a List sweep file.

Key Entry **Load From Selected File**

:MOVE

Supported All Models

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

This command renames the file `"<src_file>"` in the memory catalog. The `"<src_file>"` file name will be renamed to the `"<dest_file>"` file name.

Refer to "File Name Variables" on page 12 for information on the file name syntax.

Key Entry **Rename File**

:STORe:LIST

Supported All Models

```
:MMEMory:STORe:LIST "<file name>"
```

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

Key Entry **Store To File**

Output Subsystem (:OUTPut)

:BLANking:AUTO

Supported All Models

```
[: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.

***RST** 1

Key Entry **Output Blanking Off On Auto**

:BLANking:[STATe]

Supported All Models

```
[: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.

Key Entry **Output Blanking Off On Auto**

:MODulation[:STATe]

Supported E8267D and E8257D with Option UNT

```
: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 is always displayed to indicate whether modulation is switched on or off.

***RST** 1

Key Entry **Mod On/Off**

[:STATe]

Supported All Models

```
: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 is always displayed on the signal generator to indicate whether the RF output is switched on or off.

***RST** 0

Key Entry **RF On/Off**

Route Subsystem (:ROUTe:HARDware:DGENerator)

:INPut:BPOLarity

Supported E8267D with Option 601 or 602

```
:ROUTe:HARDware:DGENerator:INPut:BPOLarity POSitive|NEGative
:ROUTe:HARDware:DGENerator:INPut:BPOLarity?
```

This command sets the signal generator up to respond to either a high (+5 vdc) or low (0 vdc) level TTL input signal at the BURST GATE IN connector. This command performs the same function as ":IPOLarity:BGATe" on page 62.

***RST** POS

Key Entry **Burst Gate In Polarity Neg Pos**

:INPut:CPOLarity

Supported E8267D with Option 601 or 602

```
:ROUTe:HARDware:DGENerator:INPut:CPOLarity POSitive|NEGative
:ROUTe:HARDware:DGENerator:INPut:CPOLarity?
```

This command sets the signal generator up to respond to either a high (+5 vdc) or low (0 vdc) level TTL input signal at the DATA CLOCK connector. This command performs the same function as ":IPOLarity:CLOCk" on page 62.

***RST** POS

Key Entry **Data Clock Polarity Neg Pos**

:INPut:DPOLarity

Supported E8267D with Option 601 or 602

```
:ROUTe:HARDware:DGENerator:INPut:DPOLarity POSitive|NEGative
:ROUTe:HARDware:DGENerator:INPut:DPOLarity?
```

This command sets the signal generator up to respond to either a high (+5 vdc) or low (0 vdc) level TTL input signal at the DATA connector. This command performs the same function as ":IPOLarity:DATA" on page 63.

***RST** POS

Key Entry **Data Polarity Neg Pos**

:INPut:SPOLarity

Supported E8267D with Option 601 or 602

`:ROUTe:HARDware:DGENerator:INPut:SPOLarity POSitive|NEGative`
`:ROUTe:HARDware:DGENerator:INPut:SPOLarity?`

This command sets the signal generator up to respond to either a high (+5 vdc) or low (0 vdc) level TTL input signal at the SYMBOL SYNC connector.

This command performs the same function as ":IPOLarity:SSYNc" on page 63.

***RST** POS

Key Entry **Symbol Sync Polarity Neg Pos**

:IPOLarity:BGATe

Supported E8267D with Option 601 or 602

`:ROUTe:HARDware:DGENerator:IPOLarity:BGATe POSitive|NEGative`
`:ROUTe:HARDware:DGENerator:IPOLarity:BGATe?`

This command sets the signal generator up to respond to either a high (+5 vdc) or low (0 vdc) level TTL signal at the BURST GATE IN connector. This command performs the same function as ":INPut:BPOLarity" on page 61

***RST** POS

Key Entry **Burst Gate In Polarity Neg Pos**

:IPOLarity:CLOCk

Supported E8267D with Option 601 or 602

`:ROUTe:HARDware:DGENerator:IPOLarity:CLOCk POSitive|NEGative`
`:ROUTe:HARDware:DGENerator:IPOLarity:CLOCk?`

This command sets the signal generator up to respond to either a high (+5 vdc) or low (0 vdc) level TTL input signal at the DATA CLOCK connector.

This command performs the same function as ":INPut:CPOLarity" on page 61.

***RST** POS

Key Entry **Data Clock Polarity Neg Pos**

:IPOLarity:DATA

Supported E8267D with Option 601 or 602

```
:ROUTe:HARDware:DGENerator:IPOLarity:DATA POSitive|NEGative
:ROUTe:HARDware:DGENerator:IPOLarity:DATA?
```

This command sets the signal generator up to respond to either a high (+5 vdc) or low (0 vdc) level TTL input signal at the DATA connector. This command performs the same function as ":INPut:DPOLarity" on page 61

***RST** POS

Key Entry **Data Polarity Neg Pos**

:IPOLarity:SSYNc

Supported E8267D with Option 601 or 602

```
:ROUTe:HARDware:DGENerator:IPOLarity:SSYNc POSitive|NEGative
:ROUTe:HARDware:DGENerator:IPOLarity:SSYNc?
```

This command sets the signal generator up to respond to either a high (+5 vdc) or low (0 vdc) level TTL input signal at the SYMBOL SYNC connector.

This command performs the same function as ":INPut:SPOLarity" on page 62.

***RST** POS

Key Entry **Symbol Sync Polarity Neg Pos**

:OPOLarity:CLOCk

Supported E8267D with Option 601 or 602

```
:ROUTe:HARDware:DGENerator:OPOLarity:CLOCk POSitive|NEGative
:ROUTe:HARDware:DGENerator:OPOLarity:CLOCk?
```

This command sets the signal generator up to output either a high (+5 vdc) or low (0 vdc) level TTL Data Clock Out signal at the DATA CLK OUT pin on the rear panel AUXILIARY I/O connector.

This command performs the same function as ":OUTPut:CPOLarity" on page 65.

***RST** POS

Key Entry **Data Clock Out Neg Pos**

:OPOLarity:DATA

Supported E8267D with Option 601 or 602

```
:ROUTe:HARDware:DGENerator:OPOLarity:DATA POSitive|NEGative
:ROUTe:HARDware:DGENerator:OPOLarity:DATA?
```

This command sets the signal generator up to output either a high (+5 vdc) or low (0 vdc) level TTL DATA OUT signal at the DATA OUT pin on the rear panel AUXILIARY I/O connector.

This command performs the same function as ":OUTPut:DPOLarity" on page 65.

***RST** POS

Key Entry **Data Out Polarity Neg Pos**

:OPOLarity:EVENt[1]|2|3|4

Supported E8267D with Option 601 or 602

```
:ROUTe:HARDware:DGENerator:OPOLarity:EVENt[1]|2|3|4 POSitive|NEGative
:ROUTe:HARDware:DGENerator:OPOLarity:EVENt[1]|2|3|4?
```

This command sets the signal generator up to output either a high (+5 vdc) or low (0 vdc) level TTL signal at the EVENT 1 or EVENT 2 connector.

This command performs the same function as ":OUTPut:EPOL[1]|2|3|4" on page 66.

***RST** POS

Key Entry **Event 1 Polarity Neg Pos** **Event 2 Polarity Neg Pos** **Event 3 Polarity Neg Pos** **Event 4 Polarity Neg Pos**

:OPOLarity:SSYNc

Supported E8267D with Option 601 or 602

```
:ROUTe:HARDware:DGENerator:OPOLarity:SSYNc POSitive|NEGative
:ROUTe:HARDware:DGENerator:OPOLarity:SSYNc?
```

This command sets the signal generator up to output either a high (+5 vdc) or low (0 vdc) level TTL SYMBOL SYNC signal at the SYM SYNC OUT pin on the rear panel AUXILIARY I/O connector.

This command performs the same function as ":OUTPut:SPOLarity" on page 66.

***RST** POS

Key Entry **Symbol Sync Out Polarity Neg Pos**

:OUTPut:CPOLarity

Supported E8267D with Option 601 or 602

```
:ROUTe:HARDware:DGENerator:OUTPut:CPOLarity POSitive|NEGative
:ROUTe:HARDware:DGENerator:OUTPut:CPOLarity?
```

This command sets the signal generator up to output either a high (+5 vdc) or low (0 vdc) level TTL DATA CLOCK OUT signal at the DATA CLK OUT pin on the rear panel AUXILIARY I/O connector.

This command performs the same function as ":OPOLarity:CLOCk" on page 63.

***RST** POS

Key Entry **Data Clock Polarity Neg Pos**

:OUTPut:DCS[:STATe]

Supported E8267D with Option 601 or 602

```
:ROUTe:HARDware:DGENerator:OUTPut:DCS[:STATe] ON|OFF|1|0
:ROUTe:HARDware:DGENerator:OUTPut:DCS[:STATe]?
```

This command is used to enable or disable the DATA OUT, DATA CLK OUT, and SYM SYNC OUT signals from the rear panel AUXILIARY I/O connector. Normally, these output signals should be enabled (On). However, disabling these outputs will decrease the spurs that are sometimes present when operating at high symbol rates.

***RST** 1

Key Entry **DATA / CLK / SYNC Rear Outputs Off On**

:OUTPut:DPOLarity

Supported E8267D with Option 601 or 602

```
:ROUTe:HARDware:DGENerator:OUTPut:DPOLarity POSitive|NEGative
:ROUTe:HARDware:DGENerator:OUTPut:DPOLarity?
```

This command sets the signal generator up to output either a high (+5 vdc) or low (0 vdc) level TTL signal at the DATA OUT connector.

This command performs the same function as ":OPOLarity:DATA" on page 64.

***RST** POS

Key Entry **Data Out Polarity Neg Pos**

:OUTPut:EPOL[1]|2|3|4

Supported E8267D with Option 601 or 602

```
:ROUTe:HARDware:DGENerator:OUTPut:EPOLarity[1]|2|3|4 POSitive|NEGative
:ROUTe:HARDware:DGENerator:OUTPut:EPOLarity[1]|2|3|4?
```

This command sets the signal generator up to output either a high (+5 vdc) or low (0 vdc) level TTL signal at the EVENT1 or EVENT 2 connector.

This command performs the same function as ":OPOLarity:EVENt[1]|2|3|4" on page 64.

***RST** POS

Key Entry **Event 1 Polarity Neg Pos** **Event 2 Polarity Neg Pos** **Event 3 Polarity Neg Pos** **Event 4 Polarity Neg Pos**

:OUTPut:SPOLarity

Supported E8267D with Option 601 or 602

```
:ROUTe:HARDware:DGENerator:OUTPut:SPOLarity POSitive|NEGative
:ROUTe:HARDware:DGENerator:OUTPut:SPOLarity?
```

This command sets the signal generator up to output either a high (+5 vdc) or low (0 vdc) level TTL signal at the SYMBOL SYNC connector.

***RST** POS

Key Entry **Symbol Sync Out Polarity Neg Pos**

Status Subsystem (:STATus)

:OPERation:BASeband:CONDition

Supported	E8267D with Option 601 or 602

```
:STATus:OPERation:BASeband:CONDition?
```

This query returns the decimal sum of the bits in the Baseband Operation Condition Register. For example, if the baseband is busy (bit 0), the value 1 is returned.

The data in this register is continuously updated and reflects the current conditions. Refer to the *Programming Guide* for more information.

Range	0–32767

:OPERation:BASeband:ENABle

Supported	E8267D with Option 601 or 602

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

This command determines which bits in the Baseband Operation Event Register will set the Baseband is Busy bit (bit 10) in the Standard Operation Condition Register.

The variable `<val>` is the sum of the decimal values of the bits you want to enable. Refer to the *Programming Guide* for more information.

Range	0–32767

:OPERation:BASeband:NTRansition

Supported	E8267D with Option 601 or 602

```
:STATus:OPERation:BASeband:NTRansition <val>
:STATus:OPERation:BASeband:NTRansition?
```

This command determines which bits in the Baseband Operation Condition Register will set the corresponding bit in the Baseband Operation Event Register when that bit has a negative transition (1 to 0). The variable `<val>` is the sum of the decimal values of the bits that you want to enable. Refer to the *Programming Guide* for more information.

Range	0–32767

:OPERation:BASeband:PTRansition

Supported E8267D with Option 601 or 602

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

This command determines which bits in the Baseband Operation Condition Register will set the corresponding bit in the Baseband Operation Event Register when that bit has a positive transition (0 to 1).

The variable `<val>` is the sum of the decimal values of the bits that you want to enable. Refer to the *Programming Guide* for more information.

Range 0–32767

:OPERation:BASeband[:EVENt]

Supported E8267D with Option 601 or 602

```
:STATus:OPERation:BASeband[:EVENt]?
```

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

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

The equivalent PTR and NTR filters must be set before the condition register can set the corresponding bit in the event register. Refer to the *Programming Guide* for more information.

Range 0–32767

:OPERation:CONDition

Supported All Models

```
:STATus:OPERation:CONDition?
```

This query returns the decimal sum of the bits for the registers that are set to one and are part of the Standard Operation Status Group. For example, if a sweep is in progress (bit 3), the value 8 is returned.

The data in this register is continuously updated and reflects current conditions. Refer to the *Programming Guide* for more information.

Range 0–32767

:OPERation:ENABle

Supported All Models

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

This command determines which bits in the Standard Operation Event Register will set the Standard Operation Status Summary bit (bit 7) in the Status Byte Register.

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

Refer to the *Programming Guide* for more information.

Range 0–32767

:OPERation:NTRansition

Supported All Models

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

This command determines which bits in the Standard Operation Condition Register will set the corresponding bit in the Standard Operation Event Register when that bit has a negative transition (1 to 0).

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

Refer to the *Programming Guide* for more information.

Range 0–32767

:OPERation:PTRansition

Supported All Models

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

This command determines which bits in the Standard Operation Condition Register will set the corresponding bit in the Standard Operation Event Register when that bit has a positive transition (0 to 1).

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

Refer to the *Programming Guide* for more information.

Range 0–32767

:OPERation[:EVENt]

Supported All Models

CAUTION 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.

The equivalent PTR or NTR filters must be set before the condition register can set the corresponding bit in the event register.

Refer to the *Programming Guide* for more information.

Range 0–32767

:PRESet

Supported All Models

```
:STATus:PRESet
```

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

Refer to the *Programming Guide* for more information.

:QUEStionable:CALibration:CONDition

Supported All Models

```
: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 *Programming Guide* for more information.

Range 0–32767

:QUEStionable:CALibration:ENABle

Supported All Models

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

This command determines which bits in the Data Questionable Calibration Event 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 *Programming Guide* for more information.

Range 0–32767

:QUEStionable:CALibration:NTRansition

Supported All Models

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

This command determines which bits in the Data Questionable Calibration Condition Register will set the corresponding bit in the Data Questionable Calibration Event Register when that bit has a negative transition (1 to 0).

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

Refer to the *Programming Guide* for more information.

Range 0–32767

:QUEStionable:CALibration:PTRansition

Supported All Models

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

This command determines which bits in the Data Questionable Calibration Condition Register will set the corresponding bit in the Data Questionable Calibration Event Register when that bit has a positive transition (0 to 1).

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

Refer to the *Programming Guide* for more information.

Range 0–32767

:QUEStionable:CALibration[:EVENt]

Supported All Models

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:CALibration[:EVENt]?
```

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

The equivalent PTR or NTR filters must be set before the condition register can set the corresponding bit in the event register.

Refer to the *Programming Guide* for more information.

Range 0–32767

:QUEStionable:CONDition

Supported All Models

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

This query returns the decimal sum of the bits in the Data Questionable Condition Register. For example, if the 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 *Programming Guide* for more information.

Range 0–32767

:QUEStionable:ENABle

Supported All Models

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

This command determines which bits in the Data Questionable Event Register will set the Data Questionable Status Group Summary bit (bit 3) in the Status Byte Register.

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

Refer to the *Programming Guide* for more information.

Range 0–32767

:QUEStionable:FREQuency:CONDition

Supported All Models

`: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 *Programming Guide* for more information.

Range 0–32767

:QUEStionable:FREQuency:ENABle

Supported All Models

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

This command determines which bits in the Data Questionable Frequency Event Register will set the frequency summary bit (bit 5) 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 *Programming Guide* for more information.

Range 0–32767

:QUEStionable:FREQuency:NTRansition

Supported All Models

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

This command determines which bits in the Data Questionable Frequency Condition Register will set the corresponding bit in the Data Questionable Frequency Event Register when that bit has a negative transition (1 to 0).

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

Refer to the *Programming Guide* for more information.

Range 0–32767

:QUEStionable:FREQuency:PTRansition

Supported All Models

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

This command determines which bits in the Data Questionable Frequency Condition Register will set the corresponding bit in the Data Questionable Frequency Event Register when that bit has a positive transition (0 to 1).

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

Refer to the *Programming Guide* for more information.

Range 0–32767

:QUEStionable:FREQuency[:EVENt]

Supported All Models

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.

The equivalent PTR or NTR filters must be set before the condition register can set the corresponding bit in the event register.

Refer to the *Programming Guide* for more information.

Range 0–32767

:QUEStionable:MODulation:CONDition

Supported All Models

```
: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 *Programming Guide* for more information.

Range 0–32767

:QUEStionable:MODulation:ENABle

Supported All Models

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

This command determines which bits in the Data Questionable Modulation Event Register will set the modulation summary bit (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 *Programming Guide* for more information.

Range 0–32767

:QUEStionable:MODulation:NTRansition

Supported All Models

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

This command determines which bits in the Data Questionable Modulation Condition Register will set the corresponding bit in the Data Questionable Modulation Event Register when that bit has a negative transition (1 to 0).

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

Refer to the *Programming Guide* for more information.

Range 0–32767

:QUEStionable:MODulation:PTRansition

Supported All Models

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

This command determines which bits in the Data Questionable Modulation Condition Register will set the corresponding bit in the Data Questionable Modulation Event Register when that bit has a positive transition (0 to 1).

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

Refer to the *Programming Guide* for more information.

Range 0–32767

:QUEStionable:MODulation[:EVENt]

Supported All Models

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.

The equivalent PTR or NTR filters must be set before the condition register can set the corresponding bit in the event register.

Refer to the *Programming Guide* for more information.

Range 0–32767

:QUEStionable:NTRansition

Supported All Models

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

This command determines which bits in the Data Questionable Condition Register will set the corresponding bit in the Data Questionable Event Register when that bit has a negative transition (1 to 0).

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

Refer to the *Programming Guide* for more information.

Range 0–32767

:QUEStionable:POWer:CONDition

Supported All Models

```
: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 *Programming Guide* for more information.

Range 0–32767

:QUEStionable:POWer:ENABle

Supported All Models

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

This command determines which bits in the Data Questionable Power Event Register will set the power summary bit (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 *Programming Guide* for more information.

Range 0–32767

:QUEStionable:POWer:NTRansition

Supported All Models

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

This command determines which bits in the Data Questionable Power Condition Register will set the corresponding bit in the Data Questionable Power Event Register when that bit has a negative transition (1 to 0).

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

Refer to the *Programming Guide* for more information.

Range 0–32767

:QUEStionable:POWer:PTRansition

Supported All Models

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

This command determines which bits in the Data Questionable Power Condition Register will set the corresponding bit in the Data Questionable Power Event Register when that bit has a positive transition (0 to 1).

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

Refer to the *Programming Guide* for more information.

Range 0–32767

:QUEStionable:POWer[:EVENt]

Supported All Models

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.

The equivalent PTR or NTR filters must be set before the condition register can set the corresponding bit in the event register. Refer to the *Programming Guide* for more information.

Range 0–32767

:QUEStionable:PTRansition

Supported All Models

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

This command determines which bits in the Data Questionable Condition Register will set the corresponding bit in the Data Questionable Event Register when that bit has a positive transition (0 to 1). The variable `<val>` is the sum of the decimal values of the bits that you want to enable.

Refer to the *Programming Guide* for more information.

Range 0–32767

:QUEStionable[:EVENt]

Supported All Models

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 Data Questionable Event Register. The equivalent PTR or NTR filters must be set before the condition register can set the corresponding bit in the event register. Refer to the *Programming Guide* for more information.

Range 0–32767

System Subsystem (:SYSTem)

:ALTernate

Supported All Models with Option 007

```
:SYSTem:ALTernate <reg num>
:SYSTem:ALTernate? [MAXimum|MINimum]
```

This command sets up the signal generator to use a sweep state stored in a state register to alternate with the current sweep. The alternate sweep state must be stored in state registers 1 through 9 in sequence 0. Alternate sweep must be selected and both sweeps must be ramp sweeps.

Key Entry **Alternate Sweep Seq 0, Register 1–9**

:ALTernate:STAte

Supported All Models with Option 007

```
:SYSTem:ALTernate:STAte ON|OFF|1|0
:SYSTem:STAte?
```

This command enables or disables the alternate sweep state for the signal generator. With alternate state on, the signal generator uses the current sweep setup and alternates with a sweep saved in on of the state registers. Both sweeps must be ramp sweeps.

Key Entry **Alternate Sweep Off On**

:CAPability

Supported All Models

```
: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

Supported All Models

```
: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>

Range <month>: 1–12 <day>: 1–31

Key Entry **Time / Date**

:ERRor[:NEXT]

Supported All Models

```
: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]

Supported All

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

This command allows you to check for standard commands for programmable instruments (SCPI) errors by reading the signal generator's error queue. The capability to read the SCPI errors must be enabled before you can read the error queue. Use the `:ERRor[:NEXT]` command to read any reported errors.

***RST** 1

Key Entry N/A

:HELP:MODE

Supported All Models

`: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 signal generator power-on, preset, or `*RST`.

Key Entry **Help Mode Single Cont**

:IDN

Supported All Models

`: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

Supported All Models

`:SYSTem:LANGuage "SCPI"|"8340"|"8360"|"83712"|"83732"|"83752"|"8757"`
`:SYSTem:LANGuage?`

This command sets the remote language for the signal generator.

SCPI	This choice provides compatibility for SCPI commands.
8340	This choice provides compatibility for 8340B and 8341B microwave sources, which are supported by using the GPIB interface.
8360	This choice provides compatibility for 8360 series swept signal generators, which are supported only through a GPIB interface.

83712	This choice provides compatibility for 83711B and 83712B synthesized CW generators, which are supported only through a GPIB interface.
83732	This choice provides compatibility for 83731B and 83732B synthesized signal generators, which are supported only through a GPIB interface.
83752	This choice provides compatibility for 83751B and 83752B synthesized sweepers, which are supported only through a GPIB interface.
8757	This choice provides compatibility for a system, comprising a PSG signal generator and a 8757D scalar network analyzer. It is supported only through a GPIB interface.

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

For more information on supported SCPI commands and programming codes, refer to Chapter 6, "SCPI Command Compatibility," on page 281.

Key Entry	**SCPI 83731B,83732B**	**8360 Series 8340B,8341B**	**83711B,83712B 83751B,83752B**	**8757D System**

:PON:TYPE

Supported All Models

```
: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 84 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 signal generator power-on, preset, or `*RST`.

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.

Key Entry **Power On Last Preset**

:PRESet

Supported All Models

`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 ":PRESet:TYPE" on page 84 for selecting the type of defined conditions.

Key Entry **Preset**

:PRESet:ALL

Supported All Models

`: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 signal generator power-on, preset, or `*RST`.

:PRESet:LANGuage

Supported All Models

```
:SYSTem:PRESet:LANGuage "SCPI"|"8340"|"8360"|"83712"|"83732"|"83752"|
"8757"
: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.
8340	This choice provides compatibility for 8340B and 8341B microwave sources, which are supported by using the GPIB interface.
8360	This choice provides compatibility for 8360 series swept signal generators, which are supported only through a GPIB interface.
83712	This choice provides compatibility for 83711B and 83712B synthesized CW generators, which are supported only through a GPIB interface.
83732	This choice provides compatibility for 83731B and 83732B synthesized signal generators, which are supported only through a GPIB interface.
83752	This choice provides compatibility for 83751B and 83752B synthesized sweepers, which are supported only through a GPIB interface.

8757	This choice provides compatibility for a system, comprising a PSG signal generator and a 8757D scalar network analyzer. It is supported only through a GPIB interface.
***RST**	"SCPI"

Key Entry	**SCPI**	**8360 Series**	**83711B,83712B**	**8757D System**
	83731B,83732B	**8340B,8341B**	**83751B,83752B**	

:PRESet:PERSistent

Supported All Models

```
:SYSTem:PRESet:PERSistent
```

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

Key Entry **Restore Sys Defaults**

:PRESet:PN9

Supported All Models

```
: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.
***RST**	NORM
Key Entry	**PN9 Mode Preset Normal Quick**

:PRESet:TYPE

Supported All Models

```
: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 signal generator power-on, preset, or *RST.

Key Entry **Preset Normal User**

:PRESet[:USER]:SAVE

Supported All Models

`: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 ON|OFF|1|0

Supported All Models

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

This command enables (1) or disables (0) the secure display mode. Enabling this mode blanks the display, except for the message `*** SECURE DISPLAY ACTIVATED ***`, and disables the front panel keys. Once this function is activated, the power must be cycled to re-enable the display and front panel keys. For more information about security functions, refer to the *User's Guide*.

***RST** 1

Range N/A

Key Entry N/A

:SECurity:ERASeall

Supported All Models

`: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 *User's Guide*.

Key Entry **Erase All**

:SECurity:LEVel NONE|ERASe|OVERwrite|SANitize

Supported All Models

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

This command selects the secure mode for the signal generator. Once you select the security level operation you must set the command :SECurity:LEVel:STATe to ON to activate the security level. For other cleaning and security operation descriptions, see SECurity:ERASall, SECurity:OVERwrite, and SECurity:SANitize. You can exit the secure mode by entering SYST:SECUrity:LEVel NONE, resetting the signal generator with the *RST command, or recycling power to the signal generator. For more information about security functions, refer to the *User's Guide*.

Key Entry **Security Level**

:SECurity:LEVel:STATe ON|OFF|1|0

Supported All Models

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

When this command is enabled (1) it activates the selected security level. When disabled (0) it executes the selected security level. For more information about security functions, refer to the *User's Guide*.

CAUTION Once the security level is selected and the security state set to on, the operation will complete when the command :SECurity:LEVel:STATe OFF is executed. Ensure that the security level is set to the level you want before executing this command.

Key Entry **Enter Secure Mode**

Remarks The security level is set with the SECurity:LEVel:STATe ON command but the operation is executed only when the SECurity:LEVel:STATe OFF command is executed. You can disable the security level state by sending a *RST command, recycling power to the signal generator, or sending the :SECurity:LEVel NONE command.

:SECurity:OVERwrite

Supported All Models

`: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 *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

Supported All Models

`: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 *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

Supported All Models

```
: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 whole number, in hours. The setting enabled by this command is not affected by power-on, preset, or *RST. See ":SSAVer:MODE" on page 88 for selecting the screen saver mode.

Range 1–12

Key Entry **Screen Saver Delay:**

:SSAVer:MODE

Supported All Models

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

Key Entry **Screen Saver Mode**

:SSAVer:STATe

Supported All Models

```
: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 signal generator power-on, preset, or *RST.

Key Entry **Screen Saver Off On**

:TIME

Supported All Models

```
: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

Key Entry **Time / Date**

:VERSion

Supported All Models

```
:SYSTem:VERSion?
```

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

Trigger Subsystem

:ABORt

Supported All Models

`: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]

Supported All Models

```
: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 91 for single sweep triggering information.

***RST** 0

Key Entry **Sweep Repeat Single Cont**

:INITiate[:IMMediate][:ALL]

Supported All Models

`: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 90 for setting continuous or single sweep. See ":TRIGger[:SEQuence]:SOURce" on page 92 to select the trigger source.

Key Entry **Single Sweep**

:TRIGger:OUTPut:POLarity

Supported All Models

`: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.

***RST** POS

Key Entry **Trigger Out Polarity Neg Pos**

:TRIGger[:SEQuence]:SLOPe

Supported All Models

`: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.

***RST** POS

Key Entry **Trigger In Polarity Neg Pos**

:TRIGger[:SEQuence]:SOURce

Supported All Models

```
: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 triggering through front panel interaction by pressing the **Trigger** hardkey.

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

***RST** IMM

Key Entry **Bus** **Free Run** **Ext** **Trigger Key**

:TRIGger[:SEQuence][:IMMediate]

Supported All Models

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

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

Unit Subsystem (:UNIT)

:POWer

Supported All Models

```
: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, ":REFerence:STATe" on page 131 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.

***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 common to all PSG models:

Correction Subsystem ([:SOURce]:CORRection)

:FLATness:LOAD

Supported All Models

`[:SOURce]:CORRection:FLATness:LOAD "<file name>"`

This command loads a user-flatness correction file designated by the file name `"<file name>"` variable.

Key Entry **Load From Selected File**

:FLATness:PAIR

Supported All Models

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

This command sets a frequency and amplitude correction pair. The maximum number of pairs or points that can be entered is 1601.

The `<corr.>` variable is the power correction number.

Refer to "[:LEVel][:IMMediate][:AMPLitude]" on page 133 for power specifications for different signal generator options, model numbers, and frequency ranges.

*RST	*Option 520*: +2.0000000000000E+10
	Option 532: +3.20000000000000E+10
	Option 540: +4.0000000000000E+10
	Option 544: +4.4000000000000E+10
	Option 550: +5.0000000000000E+10
	Option 567: +6.7000000000000E+10
Range	*Option 520*: 250kHZ–20GHZ
	Option 532: 250kHZ–32GHZ
	Option 540: 250kHZ–40GHZ
	Option 544: 250kHZ–44GHZ
	Option 550: 250kH–50GHZ
	Option 567: 250kHZ–70GHZ[a]

a. 67-70 GHz performance not specified

Key Entry **Configure Cal Array**

:FLATness:POINts

Supported All Models

```
[:SOURce]:CORRection:FLATness:POINts?
```

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

:FLATness:PRESet

Supported All Models

CAUTION Once this command is executed, correction data is overwritten; If needed, save the current correction data (See ":FLATness:STORe" on page 97).

```
[: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.

Key Entry **Preset List**

:FLATness:STORe

Supported All Models

```
[:SOURce]:CORRection:FLATness:STORe "<file name>"
```

This command stores the current user-flatness correction data to a file named by the `"<file name>"` variable.

For information on file name syntax, refer to "File Name Variables" on page 12.

Key Entry **Store To File**

[:STATe]

Supported All Models

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

This command enables or disables user-flatness corrections.

***RST** 0

Key Entry **Flatness Off On**

Frequency Subsystem ([:SOURce])

:FREQuency:CENTer

Supported All Models with Option 007

```
[:SOURce]:FREQuency:CENTer <num>[<freq suffix>]|UP|DOWN
[:SOURce]:FREQuency:CENTer? [MAXimum|MINimum]
```

This command sets the center frequency for a ramp sweep. The center frequency symmetrically divides the selected frequency span and is coupled to the start and stop frequency settings. The frequency range and reset values are dependent on the signal generator model and option number.

*RST *Option 520*: +2.0000000000000E+10

Option 532: +3.20000000000000E+10

Option 540: +4.0000000000000E+10

Option 544: +4.4000000000000E+10

Option 550: +5.0000000000000E+10

Option 567: +7.0000000000000E+10

Range *Option 520*: 250kHZ–20GHZ

Option 532: 250kHZ–32GHZ

Option 540: 250kHZ–40GHZ

Option 544: 250kHZ–44GHZ

Option 550: 250kH–50GHZ

Option 567: 250kHZ–70GHZ[a]

a. 67-70 GHz performance not specified

Key Entry **Freq Center**

:FREQuency:CHANnels:BAND

Supported All Models

```
[: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|B390|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 101.

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.

***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

Supported All Models

```
[: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 frequency channel state must be enabled for this command to work. Refer to ":FREQuency:CHANNels[:STATe]" on page 101.

***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]

Supported All Models

```
[: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 channels band refer to “:FREQuency:CHANnels:BAND” on page 98.

***RST** 0

Key Entry **Freq Channels Off On**

:FREQuency:FIXed

Supported All Models

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

This command sets the signal generator output frequency. A frequency change may affect the current output power. To set the frequency mode, see “:FREQuency:MODE” on page 103.

*RST	*Option 520*: +2.0000000000000E+10
	Option 532: +3.20000000000000E+10
	Option 540: +4.0000000000000E+10
	Option 544: +4.4000000000000E+10
	Option 550: +5.0000000000000E+10
	Option 567: +6.7000000000000E+10
Range	*Option 520*: 250kHZ–20GHZ
	Option 532: 250kHZ–32GHZ
	Option 540: 250kHZ–40GHZ
	Option 544: 250kHZ–44GHZ
	Option 550: 250kH–50GHZ
	Option 567: 250kHZ–70GHZ[a]

a. 67-70 GHz performance not specified

:FREQuency:MANual

Supported All Models with Option 007

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

This command sets the RF output frequency when performing a ramp sweep in manual mode. The frequency value selected must fall within the range of the current start and stop frequency settings.

Entering a value with this command has no effect unless manual sweep mode is activated. Refer to ":SWEep:MODE" on page 119 for setting the proper mode.

The variable <val> is a numeric value. The <units> variable can be HZ, KHZ, MHZ, or GHZ.

*RST	*Option 520*: +2.0000000000000E+10
	Option 532: +3.20000000000000E+10
	Option 540: +4.0000000000000E+10
	Option 544: +4.4000000000000E+10
	Option 550: +5.0000000000000E+10
	Option 567: +6.7000000000000E+10
Range	*Option 520*: 250kHZ–20GHZ
	Option 532: 250kHZ–32GHZ
	Option 540: 250kHZ–40GHZ

Option 544: 250kHZ–44GHZ

Option 550: 250kH–50GHZ

Option 567: 250kHZ–70GHZ[a]

a. 67-70 GHz performance not specified

Key Entry **Manual Freq**

:FREQuency:MODE

Supported All Models

```
[: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 control the output frequency.

To set the frequency in the CW frequency mode, see “:FREQuency[:CW]” on page 108.

To set the frequency in the fixed frequency mode, see “:FREQuency:FIXed” on page 101.

SWEep The effects of this choice are determined by the sweep generation type selected (refer to “:SWEep:GENeration” on page 119). In analog sweep generation, the ramp sweep frequency settings (start, stop, center, and span) control the output frequency. In step sweep generation, the current step sweep frequency settings control the output frequency. In both cases, this selection also activates the sweep. This choice is available with Option 007 only.

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 or SWEep as the power mode (see “:MODE” on page 130).

***RST** CW

Key Entry **Freq CW** **Sweep Type** **Freq** **Off** **Freq & Ampl**

:FREQuency:MULTiplier

Supported All Models

```
[: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.

***RST** +1.00000000E+000

Key Entry **Freq Multiplier**

:FREQuency:OFFSet

Supported All Models

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

This command sets the frequency offset. The query of this command returns a value equal to the original output frequency times the multiplier value, plus the frequency offset value.

When an offset has been entered, the `OFFS` indicator is turned on in the frequency area of the display.

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" .

***RST** +0.0000000000000E+00

Range –200GHZ to 200GHZ

Key Entry **Freq Offset**

:FREQuency:OFFSet:STATe

Supported All Models

```
[: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.

***RST** 0

Key Entry **Freq Offset**

:FREQuency:REFerence

Supported All Models

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

This command sets the output reference frequency for the signal generator.

Refer to "[:LEVel][:IMMediate][:AMPLitude]" on page 133 for frequency and power specifications for different signal generator options and model numbers.

***RST** +0.0000000000000E+00

Key Entry **Freq Ref Set**

:FREQuency:REFerence:SET

Supported All Models

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

This command sets the output reference frequency.

Refer to "[:LEVel][:IMMediate][:AMPLitude]" on page 133 for frequency and power specifications for different signal generator options and model numbers.

***RST** +0.0000000000000E+00

Key Entry **Freq Ref Set**

:FREQuency:REFerence:STATe

Supported All Models

```
[: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, subsequent frequency parameters are set relative to the reference value.

***RST** 0

Key Entry **Freq Ref Off On**

:FREQuency:SPAN

Supported All Models with Option 007

```
[:SOURce]:FREQuency:SPAN <num>[<freq suffix>]|UP|DOWN
[:SOURce]:FREQuency:SPAN? [MAXimum|MINimum]
```

This command sets the length of the frequency range for a ramp sweep. Span setting is symmetrically divided by the selected center frequency and is coupled to the start and stop frequency settings. The span range is dependent on the signal generator model and option number.

***RST** +0.0000000000000E+00

Key Entry **Freq Span**

:FREQuency:STARt

Supported All Models

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

This command sets the frequency start point for a step sweep or ramp sweep (Option 007). In a ramp sweep setup, the selected value must be less than or equal to the value selected for the frequency stop point. In ramp sweep, this setting is coupled with the span and center frequency settings.

Refer to “[:LEVel][:IMMediate][:AMPLitude]” on page 133 for frequency and power specifications for different signal generator options and model numbers.

*RST	*Option 520*: +2.0000000000000E+10
	Option 532: +3.20000000000000E+10
	Option 540: +4.0000000000000E+10
	Option 544: +4.4000000000000E+10
	Option 550: +5.0000000000000E+10
	Option 567: +6.7000000000000E+10
Range	*Option 520*: 250kHZ–20GHZ
	Option 532: 250kHZ–32GHZ
	Option 540: 250kHZ–40GHZ
	Option 544: 250kHZ–44GHZ
	Option 550: 250kH–50GHZ
	Option 567: 250kHZ–70GHZ[a]

a. 67-70 GHz performance not specified

Key Entry **Freq Start**

:FREQuency:STOP

Supported All Models

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

This command sets the frequency stop point for a step sweep or ramp sweep (Option 007). In a ramp sweep setup, the selected value must be greater than or equal to the value selected for the frequency start point. In ramp sweep, this setting is coupled with the span and center frequency settings.

Refer to "[:LEVel][:IMMediate][:AMPLitude]" on page 133 for frequency and power specifications for different signal generator options and model numbers.

*RST	*Option 520*: +2.0000000000000E+10
	Option 532: +3.20000000000000E+10
	Option 540: +4.0000000000000E+10
	Option 544: +4.4000000000000E+10
	Option 550: +5.0000000000000E+10
	Option 567: +6.7000000000000E+10
Range	*Option 520*: 250kHZ–20GHZ
	Option 532: 250kHZ–32GHZ
	Option 540: 250kHZ–40GHZ
	Option 544: 250kHZ–44GHZ
	Option 550: 250kH–50GHZ
	Option 567: 250kHZ–70GHZ[a]

a. 67-70 GHz performance not specified

Key Entry **Freq Stop**

:FREQuency:SYNThesis

Supported All Models except Option UNR

```
[: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 optimize phase noise at offsets below 10 kHz.
2	This choice will select mode 2 which optimizes phase noise at offsets above 10 kHz.
***RST**	+1
Key Entry	**Mode 1 Optimize <10kHZ Offset** **Mode 2 Optimize >10kHZ Offset**

:FREQuency[:CW]

Supported All Models

```
[: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 103.

*RST	*Option 520*: +2.0000000000000E+10
	Option 532: +3.20000000000000E+10
	Option 540: +4.0000000000000E+10
	Option 544: +4.4000000000000E+10
	Option 550: +5.0000000000000E+10
	Option 567: +6.7000000000000E+10
Range	*Option 520*: 250kHZ–20GHZ
	Option 532: 250kHZ–32GHZ
	Option 540: 250kHZ–40GHZ
	Option 544: 250kHZ–44GHZ
	Option 550: 250kH–50GHZ
	Option 567: 250kHZ–70GHZ[a]

a. 67-70 GHz performance not specified

Key Entry **Frequency**

:PHASe:REFerence

Supported All Models

`[: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]

Supported All Models

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

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

***RST** +0.00000000E+000

Range *Radians*: –3.14 to 3.14RAD *Degrees*: –180 to 179DEG

Key Entry **Adjust Phase**

:ROSCillator:BANDwidth:DEFaults

Supported All Models with Option UNR

`[: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

Supported All Models with Option UNR

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

This command sets the bandwidth of the external reference oscillator.

Key Entry **External Ref Bandwidth**

:ROSCillator:BANDwidth:INTernal

Supported All Models with Option UNR

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

This command sets the bandwidth of the internal reference oscillator.

Key Entry **Internal Ref Bandwidth**

:ROSCillator:SOURce

Supported All Models

```
[:SOURce]:ROSCillator:SOURce?
```

This command queries the source of the reference oscillator. It returns either INT (internal) or EXT (external).

:ROSCillator:SOURce:AUTO

Supported All Models without Option UNR

```
[: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.

***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 103)	**Freq Off**
	Start/stop amplitude sweep	":MODE" (page 130)	**Ampl Off**
	Start/stop frequency and amplitude sweep[a]	":MODE" (page 130) ":FREQuency:MODE" (page 103)	**Freq & Ampl Off**
	Set up & control sweep triggering[b]	"Trigger Subsystem" (page 90)	See the "Trigger Subsystem"
Step	Start frequency sweep	":FREQuency:STARt" (page 106)	**Freq Start**
	Stop frequency sweep	":FREQuency:STOP" (page 107)	**Freq Stop**
	Start amplitude sweep	":STARt" (page 131)	**Ampl Start**
	Stop amplitude sweep	":STOP" (page 132)	**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 116.

:LIST:DIRection

Supported All Models

```
[: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.

***RST** UP

Key Entry **Sweep Direction Down Up**

:LIST:DWELl

Supported All Models

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

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

The variable <val> is expressed in units of seconds with a 0.001 resolution.

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 116 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 signal generator power-on, preset, or *RST.

Range 0.001–60

:LIST:DWELl:POINts

Supported All Models

```
[:SOURce]:LIST:DWELl:POINts?
```

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

:LIST:DWELl:TYPE

Supported All Models

```
[: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 112 for setting the list dwell points.
STEP	This choice selects the dwell time from the step sweep. Refer to “:SWEep:DWELl” on page 118 for setting the step dwell.
***RST**	LIST
Key Entry	**Dwell Type List Step**

:LIST:FREQuency

Supported All Models

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

This command sets the frequency values for the current list sweep points.

The variable <val> is expressed in Hertz.

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

The maximum number of list sweep points is 1,601.

For signal generator frequency and power specifications, refer to “[:LEVel][:IMMediate][:AMPLitude]” on page 133.

***RST**	*Option 520*: +2.0000000000000E+10
	Option 532: +3.2000000000000E+10
	Option 540: +4.0000000000000E+10
	Option 544: +4.4000000000000E+10
	Option 550: +5.0000000000000E+10
	Option 567: +6.7000000000000E+10
Range	*Option 520*: 250kHZ–20GHZ
	Option 532: 250kHZ–32GHZ
	Option 540: 250kHZ–40GHZ
	Option 544: 250kHZ–44GHZ
	Option 550: 250kH–50GHZ
	Option 567: 250kHZ–70GHZ[a]

a. 67–70 GHz performance not specified

:LIST:FREQuency:POINts

Supported All Models

`[:SOURce]:LIST:FREQuency:POINts?`

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

:LIST:MANual

Supported All Models

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

This command sets a list 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.

MANual must be the selected mode for this command to have an effect.

For information on setting the proper mode, see ":LIST:MODE" on page 114.

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.

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

Key Entry **Manual Point**

:LIST:MODE

Supported All Models

```
[: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 114.

***RST** AUTO

Key Entry **Manual Mode Off On**

:LIST:POWer

Supported All Models

```
[: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, signal generators with Option 1E1 protect 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.

Range See “[:LEVel][:IMMediate][:AMPLitude]” on page 133.

:LIST:POWer:POINts

Supported All Models

```
[:SOURce]:LIST:POWer:POINts?
```

This command queries the number of power points in the current list sweep file.

:LIST:RETRace

Supported All Models

```
[: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, step, or ramp (Option 007) single-sweep modes.

ON (1) The sweep resets to the first sweep point.

OFF (0) The sweep stays at the last sweep point.

***RST** 1

Key Entry **Sweep Retrace Off On**

:LIST:TRIGger:SOURce

Supported All Models

```
[: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 92.

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.
***RST**	IMM
Key Entry	**Bus** **Free Run** **Ext** **Trigger Key**

:LIST:TYPE

Supported All Models

```
[: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.
***RST**	STEP
Key Entry	**Sweep Type List Step**

:LIST:TYPE:LIST:INITialize:FSTep

Supported All Models

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 54.

```
[: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

Supported All Models

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 54.

```
[: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:CONTrol:STATe

Supported All Models with Option 007

```
[:SOURce]:SWEep:CONTrol:STATe ON|OFF|1|0
[:SOURce]:SWEep:CONTrol:STATe?
```

This command sets the sweep control state for a PSG in a dual-PSG ramp sweep setup. When the sweep control is on, you can designate whether the PSG is operating as the master or the slave. For information on setting master and slave designations, see “:SWEep:CONTrol:TYPE” on page 118.

The dual-PSG ramp sweep setup uses a serial cable to connect the two signal generators. This connection enables one PSG to function as the master so that sweep, bandcross, and retrace times are synchronized between the two. Each PSG can have a different sweep range, but they must have identical sweep time settings.

***RST** 0

Key Entry **Sweep Control**

:SWEep:CONTrol:TYPE

Supported All Models with Option 007

```
[:SOURce]:SWEep:CONTrol:TYPE MASTer|SLAVe
[:SOURce]:SWEep:CONTrol:TYPE?
```

In a dual-PSG ramp sweep setup, this command designates whether the PSG is performing as the master or the slave.

MASTer This choice enables the PSG to provide the triggering.

SLAVe This choice causes the PSG to submit to the triggering parameters provided by the master PSG.

You must set the slave PSG triggering to continuous (see :INITiate:CONTinuous[:ALL] on page 89). XREF

***RST** 0

Key Entry **Master** or **Slave**

:SWEep:DWELl

Supported All Models

```
[: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 116.

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

***RST**	+2.00000000E–003
Range	0.001–60
Key Entry	**Step Dwell**

:SWEep:GENeration

Supported	All Models with Option 007

```
[:SOURce]:SWEep:GENeration ANALog|STEPped
[:SOURce]:SWEep:GENeration?
```

This command sets the sweep type to analog or stepped.

ANALog	This choice selects a ramp sweep.
STEPped	This choice selects a step sweep.
***RST**	ANAL
Key Entry	**Sweep Type**

:SWEep:MODE

Supported	All Models with Option 007

```
[:SOURce]:SWEep:MODE AUTO|MANual
[:SOURce]:SWEep:MODE?
```

This command sets the current ramp sweep operating mode.

AUTO	This choice enables the signal generator to automatically sweep through the selected frequency range.
MANual	This choice enables you to select a single frequency value within the current sweep range to control the RF output. For information on selecting the frequency value, see ":FREQuency:MANual" on page 102.
***RST**	AUTO
Key Entry	**Manual Mode Off On**

:SWEep:POINts

Supported All Models

```
[:SOURce]:SWEep:POINts <val>
[:SOURce]:SWEep:POINts?
```

This command enables you to define the number of points in a step sweep.

***RST** 2

Range 2–65535

Key Entry **# Points**

:SWEep:TIME

Supported All Models with Option 007

```
[:SOURce]:SWEep:TIME <val><units>
[:SOURce]:SWEep:TIME?
```

This command enables you to set the sweep time for a ramp sweep. If this command is executed while the signal generator is in automatic sweep time mode, the manual sweep time mode is activated and the new sweep time value is applied. The sweep time cannot be set to a value faster than what the automatic mode provides.

The sweep time is the duration of the sweep from the start frequency to the stop frequency. It does not include the bandcross time that occurs during a sweep or the retrace time that occurs between sweep repetitions.

***RST** 1.00000000E–002

Range 10mS–99S

Key Entry **Sweep Time**

:SWEep:TIME:AUTO

Supported All Models with Option 007

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

This command enables you to set the sweep time mode for a ramp sweep.

The sweep time is the duration of the sweep from the start frequency to the stop frequency. It does not include the bandcross time that occurs during a sweep or the retrace time that occurs between sweep repetitions.

ON This choice enables the signal generator to automatically calculate and set the fastest allowable sweep time.

OFF This choice enables you to select the sweep time. The sweep time cannot be set to a value faster than what the automatic mode provides. To set the sweep time refer to":SWEep:TIME" on page 120.

***RST** 1

Key Entry **Sweep Time Manual Auto**

Marker Subsystem–Option 007 ([:SOURce])

:MARKer[n]:AMPLitude[:STATe]

Supported All Models with Option 007

```
[:SOURce]:MARKer[n]:AMPLitude[:STATe] ON|OFF|1|0
[:SOURce]:MARKer[n]:AMPLitude[:STATe]?
```

This command sets the amplitude marker state for the currently activated markers. When the state is switched on, the RF output signal exhibits a spike with a magnitude relative to the power level at each marker's set frequency. (To set the magnitude of the spike, refer to ":MARKer[n]:AMPLitude:VALue" on page 122.) The width of the amplitude spike is a nominal eight buckets, based on 1601 buckets per sweep.

While an individual marker number (0 through 9) may be specified in the command syntax where [n] is located, it has no effect. The command continues to act as a global switch for all markers. The marker designator [n] is allowed as a programming convenience only.

***RST** 0

Key Entry **Amplitude Markers Off On**

:MARKer[n]:AMPLitude:VALue

Supported All Models with Option 007

```
[:SOURce]:MARKer[n]:AMPLitude:VALue <num>[DB]
[:SOURce]:MARKer[n]:AMPLitude:VALue?
```

This command sets the relative power for the amplitude spikes at each marker's set frequency when the amplitude marker mode is activated. (To activate the amplitude markers, refer to ":MARKer[n]:AMPLitude[:STATe]" on page 122.)

While an individual marker number (0 through 9) may be specified in the command syntax where [n] is located, it has no effect. The command continues to set the power value for all markers. The marker designator [n] is allowed as a programming convenience only.

***RST** 2DB

Range –10DB to +10DB

Key Entry **Marker Value**

:MARKer[n]:AOFF

Supported All Models with Option 007

```
[:SOURce]:MARKer[n]:AOFF
```

This command turns off all active markers.

While an individual marker number (0 through 9) may be specified in the command syntax where [n] is located, it has no effect. The command continues to turn off all markers. The marker designator [n] is allowed as a programming convenience only.

Key Entry **Turn Off Markers**

:MARKer[n]:DELTa?

Supported All Models with Option 007

```
[:SOURce]:MARKer[n]:DELTa? <num>,<num>
```

This query returns the frequency difference between two markers. The variables <num> are used to designate the marker numbers.

Range 0–9

:MARKer[n]:FREQuency

Supported All Models with Option 007

```
[:SOURce]:MARKer[n]:FREQuency <val><unit>
[:SOURce]:MARKer[n]:FREQuency? MAXimum|MINimum
```

This command sets the frequency for a specific marker. If the marker designator [n] is not specified, marker 0 is the default. The frequency value must be within the current sweep range. Using MAXimum or MINimum in the query syntax returns the boundary values for allowable marker frequencies.

If the marker frequency mode is set to delta when the query is sent, the returned value is not absolute, but is relative to the reference marker. (See ":MARKer[n]:MODe" on page 124 for more information.)

***RST** +5.25000000E+008

Range equivalent to current sweep range

Key Entry **Marker Freq**

:MARKer[n]:MODe

Supported All Models with Option 007

```
[:SOURce]:MARKer[n]:MODe FREQuency|DELTa
[:SOURce]:MARKer[n]:MODe?
```

This command sets the frequency mode for all markers. While an individual marker number (0 through 9) may be specified in the command syntax ([n]), it has no effect; the command continues to set the mode for all markers. The marker designator [n] is allowed as a programming convenience.

FREQuency The frequency values for the markers are absolute.

DELTa The frequency values for the markers are relative to the designated reference marker. The reference marker must be designated before this mode is selected. (See :MARKer[n]:REFerence” to select reference marker.)

***RST** FREQuency
Key Entry **Marker Delta Off On**

:MARKer[n]:REFerence

Supported All Models with Option 007

```
[:SOURce]:MARKer[n]:REFerence <n>
[:SOURce]:MARKer[n]:REFerence?
```

This command designates the reference marker when using markers in delta mode. The variable <n> designates the marker number. While an individual marker number (0 through 9) may be specified in the command syntax ([n]), it has no effect; the marker designator [n] is a programming convenience.

***RST** 0
Range 0–9
Key Entry **Delta Ref Set**

:MARKer[n][:STATe]

Supported All Models with Option 007

```
[:SOURce]:MARKer[n][:STATe] ON|OFF|1|0
[:SOURce]:MARKer[n][:STATe]?
```

This command sets the state for a specific marker. If the marker designator [n] is not specified, marker 0 is the default.

***RST** 0
Key Entry **Marker On Off**

Power Subsystem ([:SOURce]:POWer)

:ALC:BANDwidth|BWIDth

Supported All Models

```
[: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. The bandwidth choices for this command are not effective if an internal I/Q source is being used.

***RST** 100.0

Key Entry **ALC BW**

:ALC:BANDwidth|BWIDth:AUTO

Supported All Models

```
[: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.

***RST** 1

Key Entry **ALC BW**

:ALC:LEVel

Supported E8257D with Option 1E1 and E8267D

```
[: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 129 for choosing the attenuator mode.

***RST** +1.00000000E+000

Range -20 to 25

Key Entry **Set ALC Level**

:ALC:SEARch

Supported All Models

```
[: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.

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 128 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.

***RST** 0

Key Entry **Power Search Manual Auto** **Do Power Search**

:ALC:SEARch:REFerence

Supported All Models

```
[:SOURce]:POWer:ALC:SEARch:REFerence FIXed|MODulated
[:SOURce]:POWer:ALC:SEARch:REFerence?
```

This command sets either fixed or modulated modes of power search.

FIXed This choice uses a 0.5 volt reference.

MODulated This choice uses the RMS value of the current I/Q modulation.

***RST** MOD

Key Entry **Power Search Reference Fixed Mod**

:ALC:SEARch:SPAN:START

Supported All Models

```
[:SOURce]:POWer:ALC:SEARch:SPAN:START <val><units>
[:SOURce]:POWer:ALC:SEARch:SPAN:START?
```

This command sets the start frequency for a span power search over a user specified range. The start frequency has no default value. The start frequency value will be the last value set before powering off the instrument.

Key Entry **Start Frequency**

:ALC:SEARch:SPAN:STOP

Supported All Models

```
[:SOURce]:POWer:ALC:SEARch:SPAN:STOP <val><units>
[:SOURce]:POWer:ALC:SEARch:SPAN:STOP?
```

This command sets the stop frequency for a span power search over a user specified range. The stop frequency has no default value. The stop frequency value will be the last value set before powering off the instrument

Key Entry **Stop Frequency**

:ALC:SEARch:SPAN:TYPE FULL|USER

Supported All Models

```
[: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 span power search. You can specify the range (USER) or you can select the full range (FULL) of the signal generator.

Key Entry **Span Type User Full**

:ALC:SEARch:SPAN[:STATe] ON|OFF|1|0

Supported All Models

```
[: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.

:ALC:SOURce

Supported All Models

```
[:SOURce]:POWer:ALC:SOURce INTernal|DIODe|MMHead
[:SOURce]:POWer:ALC:SOURce?
```

This command enables you to select an automatic level control (ALC) source.

***RST** INT

Key Entry **Leveling Mode**

:ALC:SOURce:EXTernal:COUPling

Supported All Models

```
[: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 (“:ALC:SOURce” on page 128). (0 to 32 coupling value).

***RST** +1.60000000E+001

Range –200 to 200.

Key Entry **Ext Detector Coupling Factor**

:ALC[:STATe]

Supported All Models

```
[: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 the desired level in spite of drift due to temperature and time.

***RST** 1

Key Entry **ALC Off On**

:ATTenuation

Supported E8257D with Option 1E1 and E8267D

```
[:SOURce]:POWer:ATTenuation <val><unit>
[:SOURce]:POWer:ATTenuation?
```

This command sets the attenuation level when the attenuator hold is active. For the E8267D, the attenuation is set in increments of 5 dB. For the E8257D with Option 1E1, the progression is 0, 5, 15, 25 and continues in 10 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 ":ATTenuation:AUTO" on page 129 for choosing the attenuator mode.

***RST** +115

Key Entry **Set Atten**

:ATTenuation:AUTO

Supported E8257D with Option 1E1 and E8267D

```
[: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, signal generators with Option 1E1 protect 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.

***RST** 1

Key Entry **Atten Hold Off On**

:MODE

Supported All Models

```
[:SOURce]:POWer:MODE FIXed|SWEep|LIST
[:SOURce]:POWer:MODE?
```

This command sets the power mode of the signal generator.

FIXed — This choice turns off any currently running power sweeps and the current CW amplitude settings control the output power.

SWEep — The effects of this choice are determined by the sweep generation type selected (refer to ":SWEep:GENeration" on page 119). If you are using analog sweep generation, the current ramp sweep amplitude settings (start and stop) control the output power. If you are using step sweep generation, the current step sweep amplitude settings control the output power. In both cases, this selection also activates the sweep. This choice is available with Option 007 only.

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 or SWEep as the frequency mode (see ":FREQuency:MODE" on page 103).

***RST** FIX

Key Entry **Sweep Type** **Ampl** **Off** **Freq & Ampl**

:PROTection:STATe

Supported E8257D with Option 1E1 and E8267D

```
[:SOURce]:POWer:PROTection:[STATe] ON|OFF|1|0
[:SOURce]:POWer:PROTection:[STATe]?
```

This command enables or disables the power inhibit function, which sets the attenuation to maximum when doing a power search. This can be used to protect devices sensitive to high average power.

ON (1) — Causes the attenuator to hold its maximum setting during a power search

OFF (0) — Enables the attenuator to operate normally

***RST** 0

:REFerence

Supported All Models

```
[: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.

***RST**	+0.00000000E+000
Range	-400 to 300 dBm
Key Entry	**Ampl Ref Set**

:REFerence:STATe

Supported All Models

```
[: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 31 and ":POWer" on page 93).
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.

***RST**	0
Key Entry	**Ampl Ref Off On**

:STARt

Supported All Models

```
[:SOURce]:POWer:STARt <val><unit>
[:SOURce]:POWer:STARt?
```

This command sets the amplitude of the first point in a step or ramp sweep (Option 007).

During an amplitude sweep operation, signal generators with Option 1E1 protect 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.

***RST**	−1.35000000E+002

Range Refer to “[:LEVel][:IMMediate][:AMPLitude]” on page 133 for the output power ranges.

Key Entry **Ampl Start**

:STOP

Supported All Models

```
[:SOURce]:POWer:STOP <val><unit>
[:SOURce]:POWer:STOP?
```

This command sets the amplitude of the last point in a step or ramp sweep (Option 007).

During an amplitude sweep operation, signal generators with Option 1E1 protect 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.

***RST** –1.35000000E+002

Range Refer to “[:LEVel][:IMMediate][:AMPLitude]” on page 133 for the output power ranges.

Key Entry **Ampl Stop**

[:LEVel][:IMMediate]:OFFSet

Supported All Models

```
[: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 any time in either normal operation or amplitude reference mode.

***RST** +0.00000000E+000

Range –200dB to 200dB

Key Entry **Ampl Offset**

[:LEVel][:IMMediate][:AMPLitude]

Supported All Models

```
[:SOURce]:POWer[:LEVel][:IMMediate][:AMPLitude] <val><unit>
[:SOURce]:POWer[:LEVel][:IMMediate][:AMPLitude]?
```

This command sets the RF output power.

The ranges for this command are specified values from the data sheet.

***RST** −1.35000000E+002

Range

E8257D

Option	Standard	*with Option UNW*	*with Option 1EH*	*with Option UNW & 1EH*
Option 520				
250 kHz–3.2 GHz	−20 to 13DBM	−20 to 13DBM	−20 to 13DBM	−20 to 10DBM
>3.2 GHz–20 GHz	−20 to 13DBM	−20 to 13DBM	−20 to 13DBM	−20 to 13DBM
Option 540				
250 kHz–40 GHz	−20 to 9DBM	−20 to 9DBM	−20 to 9DBM	−20 to 9DBM
Option 550 and 567				
250 kHz–70 GHz	−20 to 5DBM	−20 to 5DBM	−20 to 5DBM	−20 to 5DBM

E8257D with Option 1EA

Option	*Option 1EA only*	*with Option UNW*	*with Option 1EH*	*with Option UNW & 1EH*
Option 520				
250 kHz–3.2 GHz	−20 to 16DBM	−20 to 13DBM	−20 to 13DBM	−20 to 10DBM
>3.2 GHz–20 GHz	−20 to 13DBM	−20 to 13DBM	−20 to 13DBM	−20 to 20DBM
Option 540				
250 kHz–3.2 GHz	−20 to 15DBM	−20 to 12DBM	−20 to 12DBM	−20 to 9DBM
>3.2 GHz–20 GHz	−20 to 18DBM	−20 to 18DBM	−20 to 18DBM	−20 to18DBM
>20 GHz–40 GHz	−20 to 14DBM	−20 to 14DBM	−20 to 14DBM	−20 to 14DBM
Option 550 and 567				
250 kHz–3.2 GHz	−20 to 14DBM	−20 to 11DBM	−20 to 11DBM	−20 to 8DBM
>3.2 GHz–20 GHz	−20 to 16DBM	−20 to 16DBM	−20 to 16DBM	−20 to16DBM
>20 GHz–65 GHz	−20 to 13DBM	−20 to 13DBM	−20 to 13DBM	−20 to13DBM
>65 GHz–67 GHz	−20 to 12DBM	−20 to 12DBM	−20 to 12DBM	−20 to12DBM
>67 GHz–70 GHz[a]	−20 to 11DBM	−20 to 11DBM	−20 to 11DBM	−20 to 11DBM

a. 67-70 GHz performance not specified

E8257D with Option 1E1

Option	*Option 1E1 only*	*with Option UNW*	*with Option 1EH*	*with Option UNW & 1EH*
Option 520				
250 kHz–3.2 GHz	−135 to 11DBM	−135 to 11DBM	−135 to 11DBM	−135 to 9DBM

Option	Option 1E1 only	with Option UNW	with Option 1EH	with Option UNW & 1EH
>3.2GHz–20 GHz	–135 to 11DBM	–135 to 11DBM	–135 to 11DBM	–135 to 11DBM
Option 540				
250 kHz–40 GHz	–135 to 7DBM	–135 to 7DBM	–135 to 7DBM	–135 to 7DBM
Option 550 and 567				
250 kHz–67 GHz	–110 to 3DBM	–110 to 3DBM	–110 to 3DBM	–110 to 3DBM
>67 GHz–70 GHz[a]	–110 to 3DBM	–110 to 3DBM	–110 to 3DBM	–110 to 3DBM

a. 67-70 GHz performance not specified

E8257D with Option 1EA and 1E1

Option	Option 1EA &1E1 only	with Option UNW	with Option 1EH	with Option UNW & 1EH
Option 520				
250 kHz–3.2 GHz	–135 to 15DBM	–135 to 12DBM	–135 to 12DBM	–135 to 9DBM
>3.2 GHz–20 GHz	–135 to 18DBM	–135 to 18DBM	–135 to 18DBM	–135 to 18DBM
Option 540				
250 kHz–3.2 GHz	–135 to 14DBM	–135 to 11DBM	–135 to 11DBM	–135 to 8DBM
>3.2 GHz–20 GHz	–135 to 16DBM	–135 to 16DBM	–135 to 16DBM	–135 to 16DBM
>20 GHz–40 GHz	–135 to 12DBM	–135 to 12DBM	–135 to 12DBM	–135 to 12DBM
Option 550 and 567				
250 kHz–3.2 GHz	–110 to 13DBM	–110 to 10DBM	–110 to 10DBM	–110 to 7DBM
>3.2 GHz–20 GHz	–110 to 14DBM	–110 to 14DBM	–110 to 14DBM	–110 to 14DBM
>20 GHz–65 GHz	–110 to 9DBM	–110 to 9DBM	–110 to 9DBM	–110 to 9DBM
>65 GHz–67 GHz	–110 to 8DBM	–110 to 8DBM	–110 to 8DBM	–110 to 8DBM
>67 GHz–70 GHz[a]	–110 to 6DBM	–110 to 6DBM	–110 to 6DBM	–110 to 6DBM

a. 67-70 GHz performance not specified

E8267D

Option	Standard	with Option UNW	with Option 1EH	with Option UNW & 1EH
Option 520				
250 kHz–3.2 GHz	–130 to 13DBM	–130 to 10DBM	–130 to 10DBM	–130 to 7DBM
>3.2GHz–20 GHz	–130 to 18DBM	–130 to 18DBM	–135 to 18DBM	–135 to 18DBM
Option 532 and 544				
250 kHz–3.2 GHz	–130 to 12DBM	–130 to 9DBM	–135 to 9DBM	–135 to 6DBM
>3.2 GHz–32 GHz	–130 to 14DBM	–130 to 14DBM	–135 to 14DBM	–135 to 14DBM
>32 GHz–40 GHz	–130 to 12DBM	–130 to 12DBM	–130 to 12DBM	–135 to 12DBM
>40 GHz–44 GHz	–130 to 10DBM	–130 to 10DBM	–130 to 10DBM	–135 to 10DBM

Key Entry **Amplitude**

Tsweep Subsystem ([:SOURce])

:TSWeep

Supported All Models

`[:SOURce]:TSWeep`

This command aborts the current sweep, then either arms or arms and starts a single list, step, or ramp sweep (Option 007), 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 Modulation Commands

In the following sections, this chapter provides SCPI descriptions for subsystems dedicated to E8257D PSG Analog and E8267D PSG Vector signal generators:

Amplitude Modulation Subsystem ([:SOURce])

:AM[1]|2...

Supported E8257D and E8267D

`[: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.

AM[1]	**AM Path 1 2** with 1 selected
AM2	**AM Path 1 2** with 2 selected

When just AM is shown in a command, this means the command applies globally to both paths.

Each path is set up separately. When a SCPI command uses AM[1], 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.

Depth coupling links the depth value of AM[1] to AM2. 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:

- DUALsine or SWEPtsine is not the selection for the waveform type
- each path uses a different source (Internal 1, Internal 2, Ext1, or Ext2)

:AM:INTernal:FREQuency:STEP[:INCRement]

Supported E8257D and E8267D

```
[:SOURce]:AM:INTernal:FREQuency:STEP[:INCRement] <num>
[:SOURce]:AM:INTernal:FREQuency:STEP[:INCRement]?
```

This command sets the step increment for the amplitude modulation internal frequency.

The variable <num> sets the entered value in units of hertz.

The value set by this command is used with the UP and DOWN choices for the AM frequency setting. Refer to ":AM[1]|2:INTernal[1]|2:FREQuency" on page 141 for more information.

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

Range 0.5–1E6

Key Entry **Incr Set**

:AM:MODE

Supported All with Option UNT

```
[:SOURce]:AM:MODE DEEP|NORMal
[:SOURce]:AM:MODE?
```

This command sets the mode for the amplitude modulation.

DEEP This choice enables the amplitude modulation depth greater dynamic range with the ALC enabled. The minimum carrier amplitude with this choice is -10 dBm. 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 will passively disable 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.

***RST** NORM

Key Entry **AM Mode Normal Deep**

:AM:WIDeband:SENSitivity

Supported E8267D and Option UNT

```
[:SOURce]:AM:WIDeband:SENSitivity <val>
[:SOURce]:AM:WIDeband:SENSitivity?
```

This command sets the sensitivity level of the wideband AM signal in units of dB/volt. Sensitivity is .5V = 100% and is linear so that .25V = 50%. Wideband AM uses the front panel I INPUT.

***RST** +2.00000000E+001

Range 0–40DB

Key Entry **AM Depth**

:AM:WIDeband:STATe

Supported E8267D with Option UNT

```
[:SOURce]:AM:WIDeband:STATe ON|OFF|1|0
[:SOURce]:AM:WIDeband:STATe?
```

This command enables or disables wideband amplitude modulation. The RF carrier is modulated when you have set the signal generator's modulation state to ON, see ":MODulation[:STATe]" on page 60 for more information. The signal generator's I input is used to drive wideband AM modulation.

Whenever wideband amplitude modulation is enabled, the AM annunciator is turned on in the display. Wideband amplitude modulation can be simultaneously enabled with AM paths 1 and 2. Refer to ":AM[1]|2..." on page 138 for more information.

***RST** 0

Key Entry **AM Off On**

:AM[1]|2:EXTernal[1]|2:COUPling

Supported All

```
[:SOURce]:AM[1]|2:EXTernal[1]|2:COUPling AC|DC
[:SOURce]:AM[1]|2:EXTernal[1]|2:COUPling?
```

This command sets the coupling for the amplitude modulation source through the selected external input connector. The command does not change the currently active source or switch the modulation on or off. The modulating signal may be the sum of several signals, 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.

***RST** DC

Key Entry **Ext Coupling DC AC**

:AM[1]|2:EXTernal[1]|2:IMPedance

Supported All

```
[: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 selected external input.

***RST** +5.00000000E+001

Key Entry **Ext Impedance 50 Ohm 600 Ohm**

:AM[1]|2:INTernal[1]|2:FREQuency

Supported All with Option UNT

```
[: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 amplitude modulation rate for the following applications:

- the first tone of a dual-sine waveform
- the start frequency for a swept-sine waveform
- the frequency rate for all other waveforms

The variable <val> is expressed as a number. The variable `<units>` is expressed in units of Hertz.

Refer to ":AM:INTernal:FREQuency:STEP[:INCRement]" on page 138 for setting the value associated with the UP and DOWN choices.

Refer to ":AM[1]|2:INTernal[1]|2:FUNCtion:SHAPe" on page 143 for the waveform selection.

***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

Supported All with Option UNT

```
[: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.

The variable <val> is expressed as a number. The variable `<units>` is expressed in units of Hertz.

Refer to ":AM[1]|2:INTernal[1]|2:FUNCtion:SHAPe" on page 143 for the waveform selection.

***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

Supported All with Option UNT

```
[:SOURce]:AM[1]|2:INTernal[1]:FREQuency:ALTernate:AMPLitude:
PERCent <val><units>
[: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.

The variable <val> is expressed as a percentage. The variable `<units>` is expressed in units of decibels (dB).

Refer to ":AM[1]|2:INTernal[1]|2:FUNCtion:SHAPe" on page 143 for the waveform selection.

***RST** +5.00000000E+001

Range 0–100PCT

Key Entry **AM Tone 2 Ampl Percent Of Peak**

:AM[1]|2:INTernal[1]|2:FUNCtion:NOISe

Supported All with Option UNT

```
[:SOURce]:AM[1]|2:INTernal[1]|2:FUNCtion:NOISe GAUSsian|UNIForm
[:SOURce]:AM[1]|2:INTernal[1]|2:FUNCtion:NOISe?
```

This commands sets the noise type when NOISe is the waveform choice.

Refer to ":AM[1]|2:INTernal[1]|2:FUNCtion:SHAPe" on page 143 for the waveform selection.

***RST** UNIF

Key Entry **Gaussian** **Uniform**

:AM[1]|2:INTernal[1]|2:FUNCtion:RAMP

Supported All with Option UNT and Option 007

```
[:SOURce]:AM[1]|2:INTernal[1]|2:FUNCtion:RAMP POSitive|NEGative
[:SOURce]:AM[1]|2:INTernal[1]|2:FUNCtion:RAMP?
```

This command sets the slope type for the ramp modulated waveform.

Refer to ":AM[1]|2:INTernal[1]|2:FUNCtion:SHAPe" for the waveform selection.

***RST**	POS
Key Entry	**Positive** **Negative**

:AM[1]|2:INTernal[1]|2:FUNCtion:SHAPe

Supported All with Option UNT

```
[: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 DUALsine and SWEPtsine waveform choices.

***RST**	SINE
Key Entry	**Sine** **Triangle** **Square** **Ramp** **Noise** **Dual-Sine** **Swept-Sine**

:AM[1]|2:INTernal[1]:SWEep:RATE

Supported All with Option UNT

```
[:SOURce]:AM[1]|2:INTernal[1]:SWEep:RATE <val><unit>
[:SOURce]:AM[1]|2:INTernal[1]:SWEep:RATE?
```

This command sets the sweep rate for the amplitude-modulated, swept-sine waveform.

The variable <val> is a numeric expression. The <units> variable is expressed in Hertz with a minimum resolution of 0.5 hertz.

Refer to ":AM[1]|2:INTernal[1]|2:FUNCtion:SHAPe" on page 143 for the waveform selection.

***RST**	+4.00000000E+002
Range	0.5HZ–100kHZ
Key Entry	**AM Sweep Rate**

:AM[1]|2:INTernal[1]:SWEep:TRIGger

Supported All with Option UNT

```
[: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 amplitude modulated 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 143 for the waveform selection.

***RST** IMM

Key Entry **Bus** **Free Run** **Ext** **Trigger Key**

:AM[1]|2:SOURce

Supported All with Option UNT

```
[:SOURce]:AM[1]|2:SOURce INT[1]|INT2|EXT[1]|EXT2
[:SOURce]:AM[1]|2:SOURce?
```

This command sets the source to generate the 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 a display annunciator will report a high or low condition if that voltage is > ±3% of 1 V_p.

***RST** INT

Key Entry **Internal 1** **Internal 2** **Ext1** **Ext2**

:AM[1]|2:STATe

Supported All with Option UNT

```
[:SOURce]:AM[1]|2:STATe ON|OFF|1|0
[:SOURce]:AM[1]|2:STATe?
```

This command enables or disables the 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 60 for more information.

Whenever amplitude modulation is enabled, the `AM` annunciator is turned on in the display.

The two paths for amplitude modulation can be simultaneously enabled. Refer to “:AM[1]|2...” on page 138 for more information.

***RST**	0
Key Entry	**AM Off On**

:AM[1]|2:TYPE

Supported	All with Option UNT

```
[:SOURce]:AM[1]|2:TYPE LINear|EXPonential
[:SOURce]:AM[1]|2:TYPE?
```

This command sets the measurement type and unit for the depth of the AM signal.

LINear	This choice enables linear depth values in units of percent/volt.
EXPonential	This choice enables exponential depth values in units of dB/volt.
***RST**	LIN
Key Entry	**AM Type LIN EXP**

:AM[1]|2[:DEPTh]:EXPonential

Supported	All with Option UNT

```
[:SOURce]:AM[1]|2[:DEPTh]:EXPonential <val>
[:SOURce]:AM[1]|2[:DEPTh]:EXPonential?
```

This commands sets the depth of the AM signal. The <val> variable is expressed in units of dB/volt. EXPonential must be the current measurement choice for this command to have any affect. Refer to “:AM[1]|2:TYPE” for setting the AM measurement mode.

***RST**	+4.00000000E+001
Range	0.00–40.00DB
Key Entry	**AM Depth**

:AM[1]|2[:DEPTh][:LINear]

Supported All with Option UNT

```
[:SOURce]:AM[1]|2[:DEPTh][:LINear] <val><unit>|UP|DOWN
[:SOURce]:AM[1]|2[:DEPTh][:LINear]?
```

This commands sets the depth of the AM signal.

LINear must be the current measurement choice for this command to have any affect. Refer to ":AM[1]|2:TYPE" on page 145 for setting the AM measurement mode. When the depth values are coupled, a change made to one path is applied to both. For AM depth value coupling, see ":AM[1]|2[:DEPTh][:LINear]:TRACk" on page 146.

Refer to ":AM[:DEPTh]:STEP[:INCRement]" on page 147 for setting the value associated with the UP and DOWN choices.

***RST** +1.00000000E-001

Range 0.0–100PCT

Key Entry **AM Depth**

:AM[1]|2[:DEPTh][:LINear]:TRACk

Supported All with Option UNT

```
[:SOURce]:AM[1]|2[:DEPTh][:LINear]:TRACk ON|OFF|1|0
[:SOURce]:AM[1]|2[:DEPTh][:LINear]:TRACk?
```

This command enables or disables the coupling of the AM depth values between the paths (AM[1] and AM2). When the depth values are coupled, a change made to one path is applied to both. LINear must be the current unit of measure choice for this command to have any affect. Refer to ":AM[1]|2:TYPE" on page 145 for setting the AM measurement unit.

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 the coupling and both paths will have independent depth values.

***RST** 0

Key Entry **AM Depth Couple Off On**

:AM[:DEPTh]:STEP[:INCRement]

Supported All with Option UNT

```
[:SOURce]:AM[:DEPTh]:STEP[:INCRement] <num><units>
[:SOURce]:AM[:DEPTh]:STEP[:INCRement]?
```

This command sets the depth increment value for the LINear measurement choice. The variable <val> is a percentage. The <units> variable sets the increment value in units of percent.

Refer to “:AM[1]|2:TYPE” on page 145 for setting the AM measurement choice. The value set by this command is used with the UP and DOWN choices for the AM linear depth command. Refer to “:AM[1]|2[:DEPTh][:LINear]” on page 146 for more information. The setting enabled by this command is not affected by signal generator power-on, preset, or `*RST`.

Range 0.1–100

Key Entry **Incr Set**

Frequency Modulation Subsystem ([:SOURce])

:FM[1]|2...

Supported E8257D and E8267D

`[:SOURce]:FM[1]|2...`

This prefix enables the selection of the FM path and is part of most SCPI commands associated with this subsystem. The two paths are equivalent to the **FM Path 1 2** softkey.

FM[1]	**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 globally to both paths.

Each path is set up separately. When a SCPI command uses FM[1], 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 FM[1] to FM2. Changing the deviation value for one path changes it for the other. If the following conditions are met, these two paths can be on at the same time:

- DUALsine or SWEPtsine 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 FM[1]

:FM:INTernal:FREQuency:STEP[:INCRement]

Supported All with Option UNT

```
[:SOURce]:FM:INTernal:FREQuency:STEP[:INCRement] <num>
[:SOURce]:FM:INTernal:FREQuency:STEP[:INCRement]?
```

This command sets the step increment for the internal frequency modulation.

The variable <num> sets the entered value in units of hertz.

The value set by this command is used with the UP and DOWN choices for the FM frequency setting. Refer to “:FM[1]|2:INTernal[1]|2:FREQuency” on page 151 for more information.

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

Range 0.5–1E6

:FM[1]|2:EXTernal[1]|2:COUPLing

Supported All with Option UNT

```
[:SOURce]:FM[1]|2:EXTernal[1]|2:COUPling AC|DC
[:SOURce]:FM[1]|2:EXTernal[1]|2:COUPling?
```

This command sets the coupling for the frequency modulation source through the selected external input connector. The command does not change the currently active source or switch the current modulation on or off. The modulating signal may be the sum of several signals, 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.

***RST** DC

Key Entry **Ext Coupling DC AC**

:FM[1]|2:EXTernal[1]|2:IMPedance

Supported All with Option UNT

```
[:SOURce]:FM[1]|2:EXTernal[1]|2:IMPedance <50|600>
[:SOURce]:FM[1]|2:EXTernal[1]|2:IMPedance?
```

This command sets the input impedance for the selected external input.

***RST** +5.00000000E+001

Key Entry **Ext Impedance 50 Ohm 600 Ohm**

:FM[1]|2:INTernal[1]:FREQuency:ALTernate

Supported All with Option UNT

```
[:SOURce]:FM[1]|2:INTernal[1]:FREQuency:ALTernate <val><unit>
[:SOURce]:FM[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 ":FM[1]|2:INTernal[1]|2:FUNCtion:SHAPe" on page 152 for the waveform selection.

***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

Supported All with Option UNT

```
[:SOURce]:FM[1]|2:INTernal[1]:FREQuency:ALTernate:AMPLitude:
PERCent <val><unit>
[: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 152 for the waveform selection.

***RST** +5.00000000E+001

Range 0–100PCT

Key Entry **FM Tone 2 Ampl Percent Of Peak**

:FM[1]|2:INTernal[1]:SWEep:RATE

Supported All with Option UNT

```
[:SOURce]:FM[1]|2:INTernal[1]:SWEep:RATE <val><unit>
[: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 152 for the waveform selection.

***RST**	+4.00000000E+002
Range	0.5HZ–100kHZ
Key Entry	**FM Sweep Rate**

:FM[1]|2:INTernal[1]:SWEep:TRIGger

Supported All with Option UNT

```
[: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 frequency modulated swept-sine waveform. Refer to ":FM[1]|2:INTernal[1]|2:FUNCtion:SHAPe" on page 152 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
Key Entry	**Bus** **Free Run** **Ext** **Trigger Key**

:FM[1]|2:INTernal[1]|2:FREQuency

Supported All with Option UNT

```
[:SOURce]:FM[1]|2:INTernal[1]|2:FREQuency <val><unit>
[:SOURce]:FM[1]|2:INTernal[1]|2:FREQuency?
```

This command sets the internal frequency modulation rate for the following applications:

- 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 ":FM:INTernal:FREQuency:STEP[:INCRement]" on page 148 for setting the value associated with the UP and DOWN choices. Refer to ":FM[1]|2:INTernal[1]|2:FUNCtion:SHAPe" on page 152 for the waveform selection.

***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

Supported All with Option UNT

```
[:SOURce]:FM[1]|2:INTernal[1]|2:FUNCtion:NOISe GAUSsian|UNIForm
[:SOURce]:FM[1]|2:INTernal[1]|2:FUNCtion:NOISe?
```

This command sets the noise type when NOISe is the waveform choice. Refer to ":FM[1]|2:INTernal[1]|2:FUNCtion:SHAPe" on page 152 for the waveform selection.

***RST** UNIF

Key Entry **Gaussian** **Uniform**

:FM[1]|2:INTernal[1]|2:FUNCtion:RAMP

Supported All with Option UNT and Option 007

```
[:SOURce]:FM[1]|2:INTernal[1]|2:FUNCtion:RAMP POSitive|NEGative
[:SOURce]:FM[1]|2:INTernal[1]|2:FUNCtion:RAMP?
```

This command sets either a positive or negative ramp as the internally modulated waveform. Refer to ":FM[1]|2:INTernal[1]|2:FUNCtion:SHAPe" for the waveform selection.

***RST** POS

Key Entry **Positive** **Negative**

:FM[1]|2:INTernal[1]|2:FUNCtion:SHAPe

Supported All with Option UNT

```
[: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 sets the FM waveform type. The INTernal2 source selection does not support the DUALsine and SWEPtsine waveform choices.

***RST** SINE

Key Entry **Sine** **Triangle** **Square** **Ramp** **Noise** **Dual-Sine** **Swept-Sine**

:FM[1]|2:SOURce

Supported All with Option UNT

```
[:SOURce]:FM[1]|2:SOURce INT[1]|INT2|EXT1|EXT2
[:SOURce]:FM[1]|2:SOURce?
```

This command sets the source to generate the frequency 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 a display annunciator will report a high or low condition if that voltage is > ±3% of 1 V_p.

***RST** INT

Key Entry **Internal 1** **Internal 2** **Ext1** **Ext2**

:FM[1]|2:STATe

Supported All with Option UNT

```
[:SOURce]:FM[1]|2:STATe ON|OFF|1|0
[:SOURce]:FM[1]|2:STATe?
```

This command enables or disables the frequency 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 60 for more information.

Whenever frequency modulation is enabled, the `FM` annunciator is turned on in the display.

The two paths for frequency modulation can be simultaneously enabled. Refer to ":FM[1]|2..." on page 148 for more information.

***RST** 0

Key Entry **FM Off On**

:FM[1]|2[:DEViation]

Supported All with Option UNT

```
[:SOURce]:FM[1]|2[:DEViation] <val><units>
[:SOURce]:FM[1]|2[:DEViation]?
```

This command sets the frequency modulation deviation.

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 155 for more information on setting the deviation tracking.

***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
> 3.2–10GHZ	0–16MHZ
> 10–20GHZ	0–32MHZ
> 20–28.5GHZ[a]	0–48MHZ
> 20–40GHZ	0–64MHZ
> 28.5–44GHZ[a]	0–80MHZ
> 40–67GHZ	0–128MHZ

a. E8267D Only

Key Entry **FM DEV**

:FM[1]|2[:DEViation]:TRACk

Supported All with Option UNT

```
[:SOURce]:FM[1]|2[:DEViation]:TRACk ON|OFF|1|0
[:SOURce]:FM[1]|2[:DEViation]:TRACk?
```

This command enables or disables the deviation coupling between the paths (FM[1] and Fm2).

ON (1) This choice will link the deviation value of FM[1] with FM2; FM2 will assume the FM[1] deviation value. For example, if FM[1] 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 (FM[1] or FM2) 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.

***RST** 0

Key Entry **FM Dev Couple Off On**

Low Frequency Output Subsystem ([:SOURce]:LFOutput)

:AMPLitude

Supported All with Option UNT

```
[:SOURce]:LFOutput:AMPLitude <val><unit>
[:SOURce]:LFOutput:AMPLitude?
```

This command sets the amplitude for the signal at the LF OUTPUT connector.

***RST** 0.00

Range 0.000VP–3.5VP

Key Entry **LF Out Amplitude**

:FUNCtion[1]|2:FREQuency

Supported All with Option UNT

```
[:SOURce]:LFOutput:FUNCtion[1]|2:FREQuency <val><unit>
[:SOURce]:LFOutput:FUNCtion[1]|2:FREQuency?
```

This command sets the internal modulation frequency for the following applications:

- 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 “:FUNCtion[1]|2:SHAPe” on page 157 for selecting the waveform type.

***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**

:FUNCtion[1]:FREQuency:ALTernate

Supported All with Option UNT

```
[:SOURce]:LFOutput:FUNCtion[1]:FREQuency:ALTernate <val><unit>
[: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 ":FUNCtion[1]|2:SHAPe" on page 157 for selecting the waveform type.

***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**

:FUNCtion[1]:FREQuency:ALTernate:AMPLitude:PERCent

Supported All with Option UNT

```
[: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 ":FUNCtion[1]|2:SHAPe" on page 157 for selecting the waveform type.

***RST** +5.00000000E+001

Range 0–100PCT

Key Entry **LF Out Tone 2 Ampl % of Peak**

:FUNCtion[1]|2:SHAPe

Supported All with Option UNT

```
[:SOURce]:LFOutput:FUNCtion[1]|2:SHAPe SINE|DUALsine|SWEPtsine|TRIangle|
SQUare|RAMP|PULSe|NOISe|DC
[:SOURce]:LFOutput:FUNCtion[1]|2:SHAPe?
```

This command sets the waveform type for the generated signal at the LF output. Function Generator must be the source selection to support DUALsine or the SWEPtsine waveform. Refer to ":SOURce" on page 159.

***RST** SINE

Key Entry **Sine** **Dual-Sine** **Swept-Sine** **Triangle** **Square** **Ramp** **Pulse** **Noise** **DC**

:FUNCtion:[1]|2:SHAPe:NOISe

Supported All with Option UNT

```
[:SOURce]:LFOutput:FUNCtion[1]|2:SHAPe:NOISe UNIForm|GAUSsian
[:SOURce]:LFOutput:FUNCtion[1]|2:SHAPe:NOISe?
```

This command sets the noise type at the LF output when NOISe is the selected waveform.

Refer to ":FUNCtion[1]|2:SHAPe" on page 157 for selecting the waveform type.

***RST** UNIF

Key Entry **Uniform** **Gaussian**

:FUNCtion[1]|2:SHAPe:RAMP

Supported All with Option UNT and Option 007

```
[:SOURce]:LFOutput:FUNCtion[1]|2SHAPe:RAMP POSitive|NEGative
[:SOURce]:LFOutput:FUNCtion[1]|2SHAPe:RAMP?
```

This command sets the slope type for the ramp waveform at the LF output.

Refer to ":FUNCtion[1]|2:SHAPe" on page 157 for selecting the waveform type.

***RST** POS

Key Entry **Positive** **Negative**

:FUNCtion[1]:SWEep:RATE

Supported All with Option UNT

```
[:SOURce]:LFOutput:FUNCtion[1]:SWEep:RATE <val><unit>
[:SOURce]:LFOutput:FUNCtion[1]:SWEep:RATE?
```

This command sets the sweep rate for an internally generated swept-sine signal at the LF output.

***RST** +4.00000000E+002

Range 0.5HZ–100kHZ

Key Entry **LF Out Sweep Rate**

:FUNCtion[1]:SWEep:TRIGger

Supported All with Option UNT

```
[: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 waveform 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 ":FUNCtion[1]|2:SHAPe" on page 157 for selecting the waveform type.

***RST** Free Run

Key Entry **Bus** **Free Run** **Ext** **Trigger Key**

:SOURce

Supported All with Option UNT

```
[:SOURce]:LFOutput:SOURce INT[1]|INT2|FUNCtion[1]|FUNCtion2
[:SOURce]:LFOutput:SOURce?
```

This command sets the low frequency 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.

***RST** INT

Key Entry **Internal 1 Monitor** **Internal 2 Monitor** **Function Generator 1** **Function Generator 2**

:STATe

Supported All with Option UNT

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

This command enables or disables the low frequency output.

***RST** 0

Key Entry **LF Out Off On**

Phase Modulation Subsystem ([:SOURce])

:PM[1]|2...

Supported E8257D and E8267D

`[:SOURce]:PM[1]|2...`

This prefix enables the selection of the ΦM path and is part of most SCPI commands associated with this subsystem. The two paths are equivalent to the **ΦM Path 1 2** softkey.

PM[1]	**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 globally to both paths.

Each path is set up separately. When a SCPI command uses PM[1], 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 PM[1] 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:

- DUALsine or SWEPtsine is not the selection for the waveform type
- each path uses a different source (Internal 1, Ext1, or Ext2)
- PM2 must be set to a deviation less than or equal to PM[1]

:PM:INTernal:FREQuency:STEP[:INCRement]

Supported All with Option UNT

```
[:SOURce]:PM:INTernal:FREQuency:STEP[:INCRement] <num>
[:SOURce]:PM:INTernal:FREQuency:STEP[:INCRement]?
```

This command sets the step increment of the phase modulation internal frequency. The variable <num> sets the entered value in units of Hertz.

The value set by this command is used with the UP and DOWN choices for the FM frequency command. Refer to ":PM[1]|2:INTernal[1]:FREQuency" on page 163for more information.

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

Range 0.5–1E6

Key Entry **Incr Set**

:PM[1]|2:BANDwidth|BWIDth

Supported All with Option UNT

```
[:SOURce]:PM[1]|2:BANDwidth|BWIDth NORMal|HIGH
[:SOURce]:PM[1]|2:BANDwidth|BWIDth?
```

This command toggles between normal phase modulation and high bandwidth phase modulation mode.

***RST** NORM

Key Entry **FM ΦM Normal High BW**

:PM[1]|2:EXTernal[1]:COUPling

Supported All with Option UNT

```
[: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 through 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 currently active source or switch the current modulation on or off. The modulating signal may be the sum of several signals, either internal or external sources.

***RST** DC

Key Entry **Ext Coupling DC AC**

:PM[1]|2:EXTernal[1]|2:IMPedance

Supported All with Option UNT

```
[:SOURce]:PM[1]|2:EXTernal[1]|2:IMPedance <50|600>
[:SOURce]:PM[1]|2:EXTernal[1]|2:IMPedance?
```

This command sets the input impedance for the selected external input.

***RST** +5.00000000E+001

Key Entry **Ext Impedance 50 Ohm 600 Ohm**

:PM[1]|2:INTernal[1]:FREQuency

Supported All with Option UNT

```
[:SOURce]:PM[1]|2:INTernal[1]|2:FREQuency <val><unit>
[:SOURce]:PM[1]|2:INTernal[1]|2:FREQuency?
```

This command sets the internal modulation frequency rate for the following formats:

- 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 ":FUNCtion[1]|2:SHAPe" on page 157 for selecting the waveform type.

***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

Supported All with Option UNT

```
[:SOURce]:PM[1]|2:INTernal[1]:FREQuency:ALTernate <val><unit>
[:SOURce]:PM[1]|2:INTernal[1]:FREQuency:ALTernate?
```

This command sets the frequency 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 165 for the waveform selection.

***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

Supported All with Option UNT

```
[:SOURce]:PM[1]|2:INTernal[1]|2:FUNCtion:NOISe GAUSsian|UNIForm
[:SOURce]:PM[1]|2:INTernal[1]|2:FUNCtion:NOISe?
```

This commands sets the noise type when NOISe is the waveform choice.

***RST** UNIF

Key Entry **Gaussian** **Uniform**

:PM[1]|2:INTernal[1]|2:FUNCtion:RAMP

Supported All with Option UNT and Option 007

```
[:SOURce]:PM[1]|2:INTernal[1]|2:FUNCtion:RAMP POSitive|NEGative
[:SOURce]:PM[1]|2:INTernal[1]|2:FUNCtion:RAMP?
```

This command sets the slope type for the ramp modulated waveform.

***RST** POS

Key Entry **Positive** **Negative**

:PM[1]|2:INTernal[1]:FREQuency:ALTernate:AMPLitude:PERCent

Supported All with Option UNT

```
[:SOURce]:PM[1]|2:INTernal[1]:FREQuency:ALTernate:AMPLitude:
PERCent <val><unit>
[: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 165 for the waveform selection.

***RST** +5.00000000E+001

Range 0–100PCT

Key Entry **ΦM Tone 2 Ampl Percent of Peak**

:PM[1]|2:INTernal[1]:FUNCtion:SHAPe

Supported All with Option UNT

```
[: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. The INTernal1 source selection does not support the DUALsine and SWEPtsine waveform choices.

***RST** SINE

Key Entry **Sine** **Triangle** **Square** **Ramp** **Noise** **Dual-Sine** **Swept-Sine**

:PM[1]|2:INTernal[1]:SWEep:RATE

Supported All with Option UNT

```
[:SOURce]:PM[1]|2:INTernal[1]:SWEep:RATE <val><unit>
[: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" for the waveform selection.

***RST** +4.00000000E+002

Range 0.5HZ–100kHZ

Key Entry **ΦM Sweep Rate**

:PM[1]|2:INTernal[1]:SWEep:TRIGger

Supported All with Option UNT

```
[: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 165 for the waveform selection.

***RST** IMM

Key Entry **Bus** **Free Run** **Ext** **Trigger Key**

:PM[1]|2:SOURce

Supported All with Option UNT

```
[:SOURce]:PM[1]|2:SOURce INT[1]|INT2|EXT[1]|EXT2
[:SOURce]:PM[1]|2:SOURce?
```

This command sets the source 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 a display annunciator will report a high or low condition if that voltage is > ±3% of 1 V_p.

***RST** INT

Key Entry **Internal 1** **Internal 2** **Ext1** **Ext2**

:PM[1]|2:STATe

Supported All with Option UNT

```
[: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 60 for more information.

Whenever phase modulation is enabled, the ΦM annunciator is turned on in the display. The two paths for phase modulation can be simultaneously enabled. Refer to ":PM[1]|2..." on page 161 for more information.

***RST** 0

Key Entry **ΦM Off On**

:PM[1]|2[:DEViation]

Supported All with Option UNT

```
[:SOURce]:PM[1]|2[:DEViation] <val><unit>
[:SOURce]:PM[1]|2[:DEViation]?
```

This command sets the deviation of the phase modulation. The variable <unit> 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.

Refer to ":PM[:DEViation]:STEP[:INCRement]" on page 168 for setting the value associated with the UP and DOWN choices.

***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
> 3.2–10GHZ	0–160rad	0–16rad
> 10.0–20GHZ	0–320rad	0–32rad
> 20.0–28.5GHZ[a]	0–480rad	0–48rad
> 20.0–40.0GHZ	0–640rad	0–64rad
> 28.5–44.0GHZ[a]	0–800rad	0–80rad
>40–67.0GHZ[b]	0–1280rad	0–128rad

Key Entry **ΦM Dev**

a. E8267D Only
b. Performance is not specified above 50 GHz

:PM[1]|2[:DEViation]:TRACk

Supported All with Option UNT

```
[: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 paths (PM[1] and PM2).

ON (1) This choice will link the deviation value of PM[1] with PM2; PM2 will assume the PM[1] deviation value. For example, if PM[1] 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 (PM[1] 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.

***RST** 0

Key Entry **ΦM Dev Couple Off On**

:PM[:DEViation]:STEP[:INCRement]

Supported All with Option UNT

```
[:SOURce]:PM[:DEViation]:STEP[:INCRement] <val><unit>
[:SOURce]:PM[:DEViation]:STEP[:INCRement]?
```

This command sets the phase modulation deviation step increment.

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 167 for more information.

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

Range 0.001–1E3RAD

Pulse Subsystem ([:SOURce]:PULSe)

:FREQuency:STEP

Supported All with Option UNT

```
[:SOURce]:PULSe:FREQuency:STEP freq
[:SOURce]:PULSe:FREQuency:STEP?
```

This command sets the step increment for the pulse frequency.

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

Range 0.1 Hz–10MHZ

Pulse Modulation Subsystem ([:SOURce])

:PULM:EXTernal:POLarity NORMal:INVerted

Supported All with Option UNT

```
[:SOURce]:PULM:EXTernal:POLarity NORMal|INVerted
[:SOURce]:PULM:EXTernal:POLarity?
```

This command configures the polarity of the TTL input signal at the GATE/PULSE/TRIGGER INPUT front panel connector. The signal generator can respond to either normal (a TTL high) or inverted (TTL low) signal.

***RST** Normal

Key Entry **Ext Polarity Normal Inverted**

:PULM:INTernal[1]:DELay

Supported All with Option UNT

```
[:SOURce]:PULM:INTernal[1]:DELay <num>[<time suffix>]|UP|DOWN
[:SOURce]:PULM:INTernal[1]:DELay?
```

This command sets the pulse delay of the internally generated pulse modulation source.

The optional variable `[<time suffix>]` accepts nS (nanoseconds) to S (seconds).

The range value is dependent on the value set for the pulse period. Refer to ":PULM:INTernal[1]:PERiod" on page 171 for pulse period settings.

Refer to ":PULM:INTernal[1]:DELay:STEP" on page 171 for setting the value associated with the UP and DOWN choices.

***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

Supported All with Option UNT

```
[: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 optional variable [<time suffix>] accepts nS (nano-seconds) to S (seconds).

The value set by this command is used with the UP and DOWN choices for the pulse modulation delay command. Refer to ":PULM:INTernal[1]:DELay" on page 170 for more information.

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

Range 10nS to (pulse period - 20 nS)

:PULM:INTernal[1]:FREQuency

Supported All with Option UNT

```
[:SOURce]:PULM:INTernal[1]:FREQuency <val><unit>UP|DOWN
[:SOURce]:PULM:INTernal[1]:FREQuency?
```

This command sets the rate of the internal square wave pulse modulation source.

This command is used when SQUare is the current pulse modulation type. Refer to ":PULM:SOURce" on page 173 for the pulse modulation type selection.

***RST** +4.00000000E+002

Range 0.1HZ–10MHZ

Key Entry **Pulse Rate**

:PULM:INTernal[1]:PERiod

Supported All with Option UNT

```
[:SOURce]:PULM:INTernal[1]:PERiod <val><unit>|UP|DOWN
[:SOURce]:PULM:INTernal[1]:PERiod?
```

This command sets the period for the internally generated pulse modulation source.

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]:PERiod:STEP[:INCRement]" for setting the value associated with the UP and DOWN choices.

***RST** +2.00000000E–006

Range 70nS–42S

Key Entry **Pulse Period**

:PULM:INTernal[1]:PERiod:STEP[:INCRement]

Supported All with Option UNT

```
[:SOURce]:PULM:INTernal[1]:PERiod:STEP[:INCRement] <val><unit>UP|DOWN
[:SOURce]:PULM:INTernal[1]:PERiod:STEP[:INCRement]?
```

This command sets the step increment for the internal pulse period.

The value set by this command is used with the UP and DOWN choices for the pulse period command. Refer to ":PULM:INTernal[1]:PERiod" for more information.

***RST** +1.00000000E-006

Range 10nS–42S

:PULM:INTernal[1]:PWIDth

Supported All with Option UNT

```
[:SOURce]:PULM:INTernal[1]:PWIDth <num>[<time suffix>]
[:SOURce]:PULM:INTernal[1]:PWIDth?
```

This command sets the pulse width for the internally generated pulse modulation source.

NOTE A power search is recommended for signals with pulse widths less than one microsecond. Refer to ":ALC:SEARch" on page 126.

The optional variable [<time suffix>] accepts nS (nano-seconds) to S (seconds).

If the entered value for the pulse width is equal to or greater than the value for the pulse period, the pulse width will change to a value that is less than the pulse period.

Refer to ":PULM:INTernal[1]:PWIDth:STEP" for setting the value associated with the UP and DOWN choices.

***RST** +1.00000000E–006

Range 10nS to (pulse period - 20 nS)

Key Entry **Pulse Width**

:PULM:INTernal[1]:PWIDth:STEP

Supported All with Option UNT

`[:SOURce]:PULM:INTernal[1]:PWIDth:STEP <num>[<time suffix>]`
`[:SOURce]:PULM:INTernal[1]:PWIDth:STEP?`

This command sets the step increment for the pulse width.

The optional variable [<time suffix>] accepts nS (nano-seconds) to S (seconds).

The value set by this command is used by the UP and DOWN choices for the pulse width command. Refer to “:PULM:INTernal[1]:PWIDth” for more information.

***RST** +1.00000000E–006

Range 10nS to (pulse period - 20 nS)

:PULM:INTernal

Supported All with Option UNT

`[: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.

***RST** FRUN (Int Free-Run)

Key Entry **Internal Square** **Int Free-Run** **Int Triggered** **Int Doublet** **Int Gated**

:PULM:SOURce

Supported All with Option UNT

`[:SOURce]:PULM:SOURce INTernal|EXTernal|SCALar`
`[:SOURce]:PULM:SOURce?`

This command sets the source for the 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.

***RST** FRUN (Int Free-Run)

Key Entry **Internal Square** **Int Free-Run** **Int Triggered** **Int Doublet** **Int Gated**
Ext Pulse **Scalar**

:PULM:STATe

Supported All with Option UNT

```
[: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 is shown in the display

***RST** 0

Key Entry **Pulse Off On**

5 Digital Modulation Commands

In the following sections, this chapter provides SCPI descriptions for subsystems dedicated to the E8267D PSG Vector signal generator:

All Subsystem–Option 601 and 602 ([:SOURce])

:RADio:ALL:OFF

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:ALL:OFF
```

This command disables all digital modulation personalities on a particular baseband. This command does not affect analog modulation.

Custom Subsystem–Option 601 and 602 ([:SOURce]:RADio:CUSTom)

:ALPha

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:CUSTom:ALPHa <val>
[:SOURce]:RADio:CUSTom:ALPHa?
```

This command changes the Nyquist or root Nyquist filter's alpha value. The filter alpha value can be set to a minimum level (0), a maximum level (1), or in between by using fractional numeric values (0.001–0.999). To change the current filter type, refer to ":FILTer" on page 186.

***RST**	+3.50000000E–001
Range	0.000–1.000
Key Entry	Filter Alpha

:BBCLock

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:CUSTom:BBCLock INT[1]|EXT[1]
[:SOURce]:RADio:CUSTom:BBCLock?
```

This command toggles the data (bit) clock input to the baseband generator board to either internal or external. This command is independent in each mode and works for both non-burst (continuous) and burst modes. This allows for a matrix of selections between burst/non-burst, internal/external data generation, internal/external data clock, and external bit/symbol data clock.

INT[1] This choice selects the signal generator internal data clock.

EXT[1] This choice selects an external data clock input.

A data clock or continuous symbol sync input must be supplied when external mode is used. This is ignored if the external reference is set to EXTernal (see ":EREFerence" on page 185).

***RST** INT

Key Entry **BBG Data Clock Ext Int**

:BBT

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:CUSTom:BBT <val>
[:SOURce]:RADio:CUSTom:BBT?
```

This command changes the bandwidth-multiplied-by-bit-time (BbT) filter parameter. The filter BbT value can be set to the maximum level (1) or in between the minimum level (0.100) and maximum level by using fractional numeric values (0.101–0.999). This command is effective only after choosing a Gaussian filter. It does not effect other types of filters (see ":FILTer" on page 186).

***RST** +5.00000000E–001

Range 0.100–1.000

Key Entry Filter BbT

:BRATe

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:CUSTom:BRATe <val>
[:SOURce]:RADio:CUSTom:BRATe?
```

This command sets the bit rate. The variable `<val>` is expressed in bits per second (bps–Mbps) and the maximum range value depends on the data source (internal or external), the modulation type, and filter. When user-defined filters are selected (see ":FILTer" on page 186), the upper bit rate is restricted using the following criteria:

- FIR filter length > 32 symbols: upper limit is 12.5 Msps
- FIR filter length > 16 symbols: upper limit is 25 Msps

When internal FIR filters are used, these limit restrictions always apply. For higher symbol rates, the FIR filter length will be truncated and will impact the relative timing of the modulated data, as well as the actual filter response (see ":SRATe" on page 189).

A change in the bit rate value effects the symbol rate value; refer to ":SRATe" on page 189 for a list of the minimum and maximum symbol rate values.

To change the modulation type, refer to ":MODulation[:TYPE]" on page 188.

***RST** +4.86000000E+004

Range

Modulation Type	*Bits per Symbol*	*Internal Data*	*External Serial Data*
BPSK	1	45 bps–50 Mbps	45 bps–50 Mbps
FSK2			
MSK			
C4FM	2	90 bps–100 Mbps	45 bps–50 Mbps
FSK4			
OQPSK			
OQPSK195			
P4QPPSK			
QAM4			
QPSK			
QPSKIS95			
QPSKISAT			
D8PSK	3	135 bps–150 Mbps	45 bps–50 Mbps
EDGE			
FSK8			
PSK8			
FSK16	4	180 bps–200 Mbps	45 bps–50 Mbps
PSK16			
QAM16			
QAM32	5	225 bps–250 Mbps	45 bps–50 Mbps
QAM64	6	270 bps–300 Mbps	45 bps–50 Mbps
QAM128	7	315 bps–350 Mbps	45 bps–50 Mbps
QAM256	8	360 bps–400 Mbps	45 bps–50 Mbps

:BURSt:SHAPe:FALL:DELay

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:CUSTom:BURSt:SHAPe:FALL:DELay <val>
[:SOURce]:RADio:CUSTom:BURSt:SHAPe:FALL:DELay?
```

This command sets the burst shape fall delay.

The variable `<val>` is expressed in bits.

To change the modulation type, refer to ":MODulation[:TYPE]" on page 188. Refer to ":SRATe" on page 189 for a list of the minimum and maximum symbol rate values.

":BURSt:SHAPe:FDELay" on page 179 performs the same function; in compliance with the SCPI standard, both commands are listed.

For concept information on burst shaping, refer to the *PSG User's Guide*.

***RST** +0.00000000E+000

Range –22.3750 to 99

Key Entry **Fall Delay**

:BURSt:SHAPe:FALL:TIME

Supported E8267D with Option 601or 602601 or 602

```
[:SOURce]:RADio:CUSTom:BURSt:SHAPe:FALL:TIME <val>
[:SOURce]:RADio:CUSTom:BURSt:SHAPe:FALL:TIME?
```

This command sets the burst shape fall time. The variable `<val>` is expressed in bits.

To change the modulation type, refer to ":MODulation[:TYPE]" on page 188. Refer to ":SRATe" on page 189 for a list of the minimum and maximum symbol rate values.

":BURSt:SHAPe:FTIMe" on page 180 performs the same function; in compliance with the SCPI standard, both commands are listed.

For concept information on burst shaping, refer to the *PSG User's Guide*.

***RST** +1.00000000E+001

Range 0.1250–255.8750

Key Entry **Fall Time**

:BURSt:SHAPe:FDELay

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:CUSTom:BURSt:SHAPe:FDELay <val>
[:SOURce]:RADio:CUSTom:BURSt:SHAPe:FDELay?
```

This command sets the burst shape fall delay. The variable `<val>` is expressed in bits.

To change the modulation type, refer to ":MODulation[:TYPE]" on page 188. Refer to ":SRATe" on page 189 for a list of the minimum and maximum symbol rate values.

":BURSt:SHAPe:FALL:DELay" on page 178 performs the same function; in compliance with the SCPI standard, both commands are listed.

For concept information on burst shaping, refer to the *PSG User's Guide*.

***RST**. +0.00000000E+000

Range. –22.3750 to 99

Key Entry **Fall Delay**

:BURSt:SHAPe:FTIMe

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:CUSTom:BURSt:SHAPe:FTIMe <val>
[:SOURce]:RADio:CUSTom:BURSt:SHAPe:FTIMe?
```

This command sets the burst shape fall time.

The variable `<val>` is expressed in bits.

To change the modulation type, refer to ":MODulation[:TYPE]" on page 188. Refer to ":SRATe" on page 189 for a list of the minimum and maximum symbol rate values.

":BURSt:SHAPe:FALL:TIME" on page 179 performs the same function; in compliance with the SCPI standard, both commands are listed.

For concept information on burst shaping, refer to the *PSG User's Guide*.

***RST** +0.00000000E+000

Range 0.1250–255.8750

Key Entry **Fall Time**

:BURSt:SHAPe:RDELay

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:CUSTom:BURSt:SHAPe:RDELay <val>
[:SOURce]:RADio:CUSTom:BURSt:SHAPe:RDELay?
```

This command sets the burst shape rise delay.

The variable `<val>` is expressed in bits.

To change the modulation type, refer to ":MODulation[:TYPE]" on page 188. Refer to ":SRATe" on page 189 for a list of the minimum and maximum symbol rate values.

":BURSt:SHAPe:RISE:DELay" on page 181 performs the same function; in compliance with the SCPI standard, both commands are listed.

For concept information on burst shaping, refer to the *PSG User's Guide*.

***RST** +0.00000000E+000

Range –17.3750 to 99

Key Entry **Rise Delay**

:BURSt:SHAPe:RISE:DELay

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:CUSTom:BURSt:SHAPe:RISE:DELay <val>
[:SOURce]:RADio:CUSTom:BURSt:SHAPe:RISE:DELay?
```

This command sets the burst shape rise delay.

The variable `<val>` is expressed in bits.

To change the modulation type, refer to ":MODulation[:TYPE]" on page 188. Refer to ":SRATe" on page 189 for a list of the minimum and maximum symbol rate values.

":BURSt:SHAPe:RDELay" on page 180 performs the same function; in compliance with the SCPI standard, both commands are listed.

For concept information on burst shaping, refer to the *PSG User's Guide*.

***RST** +0.00000000E+000

Range –17.3750 to 99

Key Entry **Rise Delay**

:BURSt:SHAPe:RISE:TIME

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:CUSTom:BURSt:SHAPe:RISE:TIME <val>
[:SOURce]:RADio:CUSTom:BURSt:SHAPe:RISE:TIME?
```

This command sets the burst shape rise time.

The variable `<val>` is expressed in bits.

To change the modulation type, refer to ":MODulation[:TYPE]" on page 188. Refer to ":SRATe" on page 189 for a list of the minimum and maximum symbol rate values.

":BURSt:SHAPe:RTIMe" on page 182 performs the same function; in compliance with the SCPI standard, both commands are listed.

For concept information on burst shaping, refer to the *PSG User's Guide*.

***RST** +1.00000000E+001

Range 0.1250–121.5000

Key Entry **Rise Time**

:BURSt:SHAPe:RTIMe

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:CUSTom:BURSt:SHAPe:RTIMe <val>
[:SOURce]:RADio:CUSTom:BURSt:SHAPe:RTIMe?
```

This command sets the burst shape rise time.

The variable `<val>` is expressed in bits.

To change the modulation type, refer to ":MODulation[:TYPE]" on page 188. Refer to ":SRATe" on page 189 for a list of the minimum and maximum symbol rate values.

":BURSt:SHAPe:RISE:TIME" on page 181 performs the same function; in compliance with the SCPI standard, both commands are listed.

For concept information on burst shaping, refer to the *PSG User's Guide*.

***RST** +1.00000000E+001

Range 0.1250–121.5000

Key Entry **Rise Time**

:BURSt:SHAPe[:TYPE]

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:CUSTom:BURSt:SHAPe[:TYPE] SINE|"<file name>"
[:SOURce]:RADio:CUSTom:BURSt:SHAPe[:TYPE]?
```

This command specifies the burst shape (`"<file name>"`).

SINE This choice selects a state that is defined by the burst rise and fall *RST values, as the default burst shape type.

`"<file name>"` This choice selects a user-designated file from signal generator memory (non-volatile). Filenames must be enclosed by quotation marks.

***RST** SINE

Key Entry **Sine** **User File**

:CHANnel

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:CUSTom:CHANnel EVM|ACP
[:SOURce]:RADio:CUSTom:CHANnel?
```

This command optimizes the Nyquist and root Nyquist filters to minimize error vector magnitude (EVM) or to minimize adjacent channel power (ACP).

EVM This choice provides the most ideal passband.

ACP This choice improves stopband rejection.

To change the current filter type, refer to ":FILTer" on page 186.

***RST** EVM

Key Entry **Optimize FIR for EVM ACP**

:DACS:ALIGn

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:CUSTOM:DACS:ALIGn
```

This command aligns and zeros out the signal generator's I/Q DAC circuitry. This operation is required any time the external VCO clock signal is lost and re-acquired or when an external VCO clock signal is first applied to the BASEBAND GEN CLK IN connector.

***RST** N/A

Range N/A

Key Entry **Align DACs**

:DATA

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:CUSTom:DATA PN9|PN11|PN15|PN20|PN23|FIX4|"<file name>"|
EXT|P4|P8|P16|P32|P64
[:SOURce]:RADio:CUSTom:DATA?
```

This command sets the data pattern for unframed transmission. For information on the file name syntax, see "File Name Variables" on page 12.

***RST** PN23

Key Entry **PN9** **PN11** **PN15** **PN20** **PN23** **FIX4** **User File** **Ext**

4 1's & 4 0's **8 1's & 8 0's** **16 1's & 16 0's** **32 1's & 32 0's**

64 1's & 64 0's

:DATA:FIX4

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:CUSTom:DATA:FIX4 <val>
[:SOURce]:RADio:CUSTom:DATA:FIX4?
```

This command sets the binary, 4-bit repeating sequence data pattern for unframed transmission according to the modulation type, symbol rate, filter, and burst shape selected for the custom modulation format. FIX4 must already be defined as the data type.

***RST** #B0000

Range #B0000–#B1111 or 0–15

Key Entry **FIX4**

:DATA:PRAM

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:CUSTom:DATA:PRAM <"<file name>">
[:SOURce]:RADio:CUSTom:DATA:PRAM?
```

This command selects PRAM as the data pattern type. Selecting PRAM as the data source forces the burst source to internal to allow framing control.

Range #B0000–#B1111 or 0–15

Key Entry **FIX4**

:DENCode

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:CUSTom:DENCode ON|OFF|1|0
[:SOURce]:RADio:CUSTom:DENCode?
```

This command enables or disables the differential data encoding function. Executing this command encodes the data bits prior to modulation; each modulated bit is 1 if the data bit is different from the previous one or 0 if the data bit is the same as the previous one.

***RST** 0

Key Entry **Diff Data Encode Off On**

:EDATa:DELay

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:CUSTom:EDATa:DELay?
```

This query returns the amount of delay (in symbols) from the external data input to the beginning of the symbol on the I OUT and Q OUT rear-panel connectors and the front panel RF OUTPUT connector. When the format is turned off, the delay value is unchanged; the query will return the same delay value if the format is on or off.

:EDCLock

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:CUSTom:EDCLock SYMBol|NORMal
[:SOURce]:RADio:CUSTom:EDCLock?
```

This command sets the external data clock use. In internal clock mode, neither choice has an effect. Refer to ":BBCLock" on page 176 to select EXT as the data clock type.

SYMBol This choice specifies that a continuous symbol clock signal must be provided to the SYMBOL SYNC input connector.

NORMal This choice specifies that the DATA CLOCK input connector requires a bit clock. The SYMBOL SYNC input connector requires a (one-shot or continuous) symbol sync signal.

***RST** NORM

Key Entry **Ext Data Clock Normal Symbol**

:EREFerence

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:CUSTom:EREFerence INTernal|EXTernal
[:SOURce]:RADio:CUSTom:EREFerence?
```

This command selects either an internal or external bit-clock reference for the data generator.

If the EXTernal choice is selected, the external frequency value must be applied to the BASEBAND GEN REF IN rear-panel connector. See ":EREFerence:VALue" on page 186 to enter the external reference frequency.

***RST** INT

Key Entry **BBG Ref Ext Int**

:EREFerence:VALue

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:CUSTom:EREFerence:VALue <val>
[:SOURce]:RADio:CUSTom:EREFerence:VALue?
```

This command retrieves the expected reference frequency value of an externally applied reference to the signal generator. The variable `<val>` is expressed in units of Hertz (Hz–MHz).

The value specified by this command is effective only when you are using an external ARB reference applied to the BASEBAND GEN REF IN rear-panel connector. Refer to ":EREFerence" on page 185 to select EXTernal as the reference for the bit clock reference of the data generator.

***RST** +1.30000000E+007

Range 2.5E5–1E8

Key Entry **Ext BBG Ref Freq**

:FILTer

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:CUSTom:FILTer RNYQuist|NYQuist|GAUSsian|RECTangle|AC4Fm|
UGGaussian|"<user FIR>"
[:SOURce]:RADio:CUSTom:FILTer?
```

This command selects the pre-modulation filter type.

AC4Fm This is a predefined Association of Public Safety Communications Officials (APCO) specified compatible 4-level frequency modulation (C4FM) filter.

UGGaussian This choice selects a GSM Gaussian filter with a fixed Bbt value of 0.300.

"<User FIR>" This variable is any filter file that you have stored into memory. For information on the file name syntax, see "File Name Variables" on page 12.

***RST** RNYQ

Key Entry **Root Nyquist** **Nyquist** **Gaussian** **Rectangle** **APCO 25 C4FM** **UN3/4 GSM Gaussian** **User FIR**

:IQ:SCALe

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:CUSTom:IQ:SCALe <val>
[:SOURce]:RADio:CUSTom:IQ:SCALe?
```

This command sets the amplitude of the I/Q outputs for better adjacent channel power (ACP); lower scaling values equate to better ACP.

The variable `<val>` is expressed in units of percent.

This command has no effect with MSK or FSK modulation.

***RST**	+70
Range	1–200
Key Entry	**I/Q Scaling**

:MODulation:FSK[:DEViation]

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:CUSTom:MODulation:FSK[:DEViation] <val>
[:SOURce]:RADio:CUSTom:MODulation:FSK[:DEViation]?
```

This command sets the symmetric FSK frequency deviation value.

The variable `<val>` is a numeric expression in units of Hertz. The maximum range value equals the current symbol rate value multiplied by four and is limited to 20 MHz.

To change the modulation type, refer to ":MODulation[:TYPE]" on page 188.

Refer to ":SRATe" on page 189 for a list of the minimum and maximum symbol rate values.

To set an asymmetric FSK deviation value, refer to the *PSG User's Guide* for more information.

***RST**	+4.00000000E+002
Range	0–2E7
Key Entry	**Freq Dev**

:MODulation:MSK[:PHASe]

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:CUSTom:MODulation:MSK[:PHASe] <val>
[:SOURce]:RADio:CUSTom:MODulation:MSK[:PHASe]?
```

This command sets the MSK phase deviation value.

The variable `<val>` is expressed in units of degrees.

***RST**	+9.00000000E+001
Range	0–100
Key Entry	**Phase Dev**

:MODulation:UFSK

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:CUSTom:MODulation:UFSK "<file name>"
[:SOURce]:RADio:CUSTom:MODulation:UFSK?
```

This command selects a user-defined FSK file from the signal generator memory.

The user-defined FSK file is held in signal generator memory until the command that selects user FSK as the modulation type is sent. Refer to “:MODulation[:TYPE]” on page 188 to change the current modulation type.

For information on the file name syntax, see “File Name Variables” on page 12.

Key Entry **User FSK**

:MODulation:UIQ

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:CUSTom:MODulation:UIQ "<file name>"
[:SOURce]:RADio:CUSTom:MODulation:UIQ?
```

This command selects a user-defined I/Q file from the signal generator memory.

The user-defined I/Q file is held in signal generator memory until the command that selects user I/Q as the modulation type is sent. Refer to “:MODulation[:TYPE]” on page 188 to change the current modulation type.

For information on the file name syntax, see “File Name Variables” on page 12.

Key Entry **User I/Q**

:MODulation[:TYPE]

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:CUSTom:MODulation[:TYPE] BPSK|QPSK|IS95QPSK|GRAYQPSK|
OQPSK|IS95OQPSK|P4DQPSK|PSK8|PSK16|D8PSK|MSK|FSK2|FSK4|FSK8|FSK16|C4FM|
QAM4|QAM16|QAM32|QAM64|QAM128|QAM256|UIQ|UFSK
[:SOURce]:RADio:CUSTom:MODulation[:TYPE]?
```

This command sets the modulation type for the Custom personality.

***RST** P4DQPSK

Key Entry **BPSK** **QPSK** **IS-95 QPSK** **Gray Coded QPSK** **OQPSK**
IS-95 OQPSK **π/4 DQPSK** **8PSK** **16PSK** **D8PSK** **MSK** **2-Lvl FSK**

4-Lvl FSK 8-Lvl FSK 16-Lvl FSK C4FM 4QAM 16QAM 32QAM 64QAM 128QAM 256QAM User I/Q User FSK

:POLarity[:ALL]

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:CUSTom:POLarity[:ALL] NORMal|INVerted
[:SOURce]:RADio:CUSTom:POLarity[:ALL]?
```

This command sets the rotation direction of the phase modulation vector.

NORMal	This choice selects normal phase polarity.
INVerted	This choice inverts the internal Q signal.
***RST**	NORM
Key Entry	**Phase Polarity Normal Invert**

:SRATe

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:CUSTom:SRATe <val>
[:SOURce]:RADio:CUSTom:SRATe?
```

This command sets the transmission symbol rate.

The variable `<val>` is expressed in units of bits per second (bps–Mbps) and the maximum range value is dependent upon the source of data (internal or external), the modulation type, and filter.

When user-defined filters are selected using the command in section “:FILTer” on page 186, the upper bit rate will be restricted using the following criteria:

- FIR filter length > 32 symbols: upper limit is 12.5 Msps
- FIR filter length > 16 symbols: upper limit is 25 Msps

When internal FIR filters are used, these limit restrictions always apply. For higher symbol rates, the FIR filter length will be truncated as follows:

- Above 12.5 Msps, the FIR length will be truncated to 32 symbols
- Above 25 Msps, the FIR length will be truncated to 16 symbols
  This will impact the relative timing of the modulated data, as well as the actual filter response (see “:BRATe” on page 177).

A change in the symbol rate value effects the bit rate value; refer to “:BRATe” on page 177 for a list of the minimum and maximum symbol rate values.

To change the modulation type, refer to ":MODulation[:TYPE]" on page 188.

***RST** +2.43000000E+004

Range

Modulation Type	*Bits per Symbol*	*Internal Data*	*External Serial Data*
BPSK	1	45 sps–50 Msps	45 sps–50 Msps
FSK2			
MSK			
C4FM	2	45 sps–50 Msps	45 sps–25 Msps
FSK4			
OQPSK			
OQPSK195			
P4QPPSK			
QAM4			
QPSK			
QPSKIS95			
QPSKISAT	2	45 sps–50 Msps	45 sps–25 Msps
D8PSK	3	45 sps–50 Msps	45 sps–16.67 Msps
EDGE			
FSK8			
PSK8			
FSK16	4	45 sps–50 Msps	45 sps–12.5 Msps
PSK16			
QAM16			
QAM32	5	45 sps–50 Msps	45 sps–10 Msps
QAM64	6	45 sps–50 Msps	45 sps–8.33 Msps
QAM128	7	45 sps–50 Msps	45 sps–7.142857142 Msps
QAM256	8	45 sps–50 Msps	45 sps–6.25 Msps

Key Entry **Symbol Rate**

:STANdard:SELect

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:CUSTom:STANdard:SELect NONE|AC4Fm|ACQPsk|BLUEtooth|CDPD
[:SOURce]:RADio:CUSTom:STANdard:SELect?
```

This command selects a predefined setup for Custom (with the appropriate defaults) and/or clears the selection.

NONE This choice clears the current predefined Custom format.

AC4Fm This choice sets up an Association of Public Safety Communications Officials (APCO) compliant, compatible 4-level frequency modulation (C4FM) format.

ACQPsk	This choice sets up an Association of Public Safety Communications Officials (APCO) compliant, compatible quadrature phase shift keying (CQPSK) format.
BLUEtooth	This choice sets up a Bluetooth (2-level frequency shift keying) format.
CDPD	This choice sets up a minimum shift keying Cellular Digital Packet Data (CDPD) format.

***RST** NONE

Key Entry **None** **APCO 25w/C4FM** **APCO 25 w/CQPSK** **Bluetooth** **CDPD**

:TRIGger:TYPE

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:CUSTom:TRIGger:TYPE CONTinuous|SINGle|GATE
[:SOURce]:RADio:CUSTom:TRIGger:TYPE?
```

This commands sets the trigger mode (type) that controls the data transmission.

Triggers control the data transmission by telling the PSG when to transmit the modulating signal. Depending on the trigger settings for the PSG, the data transmission can occur once, continuously, or the PSG may start and stop the transmission repeatedly (GATE mode).

A trigger signal comprises both positive and negative signal transitions (states), which are also called high and low periods. You can configure the PSG to trigger on either state of the trigger signal. It is common to have multiple triggers, also referred to as trigger occurrences or events, occur when the signal generator requires only a single trigger. In this situation, the PSG recognizes the first trigger and ignores the rest.

When you select a trigger mode, you may lose the signal (carrier plus modulating) from the RF output until you trigger the modulating signal. This is because the PSG sets the I and Q signals to zero volts prior to the first trigger event, which suppresses the carrier. After the first trigger event, the signal's final I and Q levels determine whether you see the carrier signal or not (zero = no carrier, other values = visible carrier). At the end of most data patterns, the final I and Q points are set to a value other than zero. If you create your own data file, you can set the initial I and Q voltages to values other than zero, and set the last I and Q values to zero. Create your own file using the front-panel UI (see the *PSG User's Guide*), or download a file you create external to the PSG (see the *PSG Programming Guide*).

There are four parts to configuring the trigger:

- Choosing the trigger type, which controls the data transmission.
- Setting the data pattern's response to triggers:
  - — CONTinuous, see ':TRIGger:TYPE:CONTinuous[:TYPE]' on page 193
  - — SINGle, selecting the mode also sets the response (This differs from using the single mode for the ARB formats.)

— GATE, selecting the mode also sets the response

- Selecting the trigger source (see ‘:TRIGger[:SOURce]’ on page 194), which determines how the PSG receives its trigger signal, internally or externally. The GATE choice requires an external trigger.
- Setting the trigger polarity when using an external source:

— CONTinuous and SINGle, see ‘:TRIGger[:SOURce]:EXTernal:SLOPe’ on page 196
— GATE, see “:TRIGger:TYPE:GATE:ACTive” on page 193

For more information on triggering, see the *PSG User's Guide*.

The following list describes the trigger type command choices:

CONTinuous	Upon triggering, the data pattern repeats continuously.
SINGle	Upon triggering, the data pattern plays once.
GATE	An external trigger signal controls the data transmission. The modulating signal waits for the first active trigger signal state to begin. After the initial trigger, the behavior is dependent on whether the signal incorporates framed or unframed data. Because the PSG provides only unframed data for real-time custom, to transmit a framed data signal you must create an external file that incorporates the framing and download it to the PSG. The following list describes the behavior differences between the two types of data transmissions:

- For unframed data, an external trigger signal repeatedly starts and stops the data transmission. The length of each transmission depends on the duty period of the trigger signal and the gate polarity selection (see “:TRIGger:TYPE:GATE:ACTive” on page 193). Data transmits during the active polarity selection state and stops during the inactive state. The active state can be set high or low.

NOTE The real-time custom gating behavior described above is opposite to the ARB gating behavior.

- For framed data, an external trigger signals the PSG to start transmitting at the beginning of a frame during active states, but only stops at the end of a frame when the end occurs during the inactive states. If the end of the frame extends into the next active trigger state, the signal transmits continuously. For information on downloading files, see the *PSG Programming Guide*.

***RST** CONT

Key Entry **Continuous** **Single** **Gated**

:TRIGger:TYPE:CONTinuous[:TYPE]

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:CUSTom:TRIGger:TYPE:CONTinuous[:TYPE] FREE|TRIGger
[:SOURce]:RADio:CUSTom:TRIGger:TYPE:CONTinuous[:TYPE]?
```

This command selects the data pattern’s response to a trigger signal while using the continuous trigger mode.

For more information on triggering and to select the continuous trigger mode, see ‘:TRIGger:TYPE’ on page 191.

The following list describes the data pattern’s response to each of the command choices:

FREE Turning custom on immediately triggers the modulating signal. The signal repeats the data pattern until you turn the signal off, select another trigger, or choose another data pattern.

TRIGger The modulating signal waits for a trigger before transmission begins. When the signal receives the trigger, it transmits the data continuously until you turn the signal off, select another trigger, or choose another data pattern.

***RST** FREE

Key Entry **Free Run** **Trigger & Run**

:TRIGger:TYPE:GATE:ACTive

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:CUSTom:TRIGger:TYPE:GATE:ACTive LOW|HIGH
[:SOURce]:RADio:CUSTom:TRIGger:TYPE:GATE:ACTive?
```

This command selects the active state (gate polarity) of the gate while using the gating trigger mode.

The LOW and HIGH selections correspond to the low and high states of an external trigger signal. For example, when you select HIGH, the active state occurs during the high of the trigger signal. The PSG uses the active state to transmit the data pattern. When the inactive state occurs, the transmission stops at the last transmitted symbol, then restarts at the next symbol when the active state occurs. For more information on triggering and to select gating as the trigger mode, see ‘:TRIGger:TYPE’ on page 191.

The following list describes the PSG’s gating behavior for the polarity selections:

LOW The PSG transmits the data pattern while the trigger signal is low (active state) and stops when the trigger signal goes high (inactive state).

HIGH The PSG transmits the data pattern while the trigger signal is high (active state) and stops when the trigger signal goes low (inactive state).

***RST** HIGH

Key Entry **Gate Active Low High**

:TRIGger[:SOURce]

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:CUSTom:TRIGger[:SOURce] KEY|EXT|BUS
[:SOURce]:RADio:CUSTom:TRIGger[:SOURce]?
```

This command sets the trigger source.

For more information on triggering, see ‘:TRIGger:TYPE’ on page 191. The following list describes the command choices:

KEY This choice enables manual triggering by pressing the front-panel **Trigger** hardkey.

EXT An externally applied signal triggers the modulating signal. This is the only choice that works with gating. The following settings affect an external trigger:

- The input connector for the trigger signal. You have a choice between the rear-panel PATTERN TRIG IN connector or the PATT TRIG IN 2 pin on the rear-panel AUXILIARY I/O connector. To make the connector selection, see ‘:TRIGger[:SOURce]:EXTernal[:SOURce]’ on page 195.

  For more information on the connectors and on connecting the cables, see the *PSG User's Guide*.

- The trigger signal polarity:
  - gating mode, see ‘:TRIGger:TYPE:GATE:ACTive’ on page 193
  - continuous and single modes, see “:TRIGger[:SOURce]:EXTernal:SLOPe” on page 196

- Any desired delay between when the PSG receives a trigger and when the data pattern responds to the trigger. There are two parts to setting the delay:
  - setting the amount of delay, see ‘:TRIGger[:SOURce]:EXTernal:DELay’ on page 195
  - turning the delay on, see ‘:TRIGger[:SOURce]:EXTernal:DELay:STATe’ on page 196

BUS This choice enables triggering over the GPIB using the `*TRG` or `GET` command, or the LAN and the AUXILIARY INTERFACE (RS-232) using the `*TRG` command.

***RST** EXT

Key Entry **Trigger Key** **Ext** **Bus**

:TRIGger[:SOURce]:EXTernal[:SOURce]

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:CUSTom:TRIGger[:SOURce]:EXTernal[:SOURce] EPT1|EPT2|
EPTRIGGER1|EPTRIGGER2
[:SOURce]:RADio:CUSTom:TRIGger[:SOURce]:EXTernal[:SOURce]?
```

This command selects which rear-panel connector the PSG uses to accept an externally applied trigger signal when external is the trigger source selection.

For more information on configuring an external trigger source and to select external as the trigger source, see ':TRIGger[:SOURce]' on page 194. For more information on the rear-panel connectors, see the *PSG User's Guide*.

The following list describes the command choices:

EPT1 This choice is synonymous with EPTRIGGER1 and selects the PATTERN TRIG IN rear-panel connector.

EPT2 This choice is synonymous with EPTRIGGER2 and selects the PATT TRIG IN 2 pin on the rear-panel AUXILIARY I/O connector.

EPTRIGGER1 This choice is synonymous with EPT1 and selects the PATTERN TRIG IN rear-panel connector.

EPTRIGGER2 This choice is synonymous with EPT2 and selects the PATT TRIG IN 2 pin on the rear-panel AUXILIARY I/O connector.

***RST** EPT1

Key Entry **Patt Trig In 1** **Patt Trig In 2**

:TRIGger[:SOURce]:EXTernal:DELay

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:CUSTom:TRIGger[:SOURce]:EXTernal:DELay <val>
[:SOURce]:RADio:CUSTom:TRIGger[:SOURce]:EXTernal:DELay?
```

This command sets the number of bits to delay the PSG's response to an external trigger.

The bit delay is a delay between when the PSG receives the trigger and when it responds to the trigger. The delay uses the clocks of the bit-clock to time the delay. After the PSG receives the trigger and the set number of delay bits (clocks) occurs, the PSG transmits the data pattern.

The delay does not occur until you turn it on (see ":TRIGger[:SOURce]:EXTernal:DELay:STATe"). You can set the number of bits either before or after turning it on.

For more information on configuring an external trigger source and to select external as the trigger source, see ‘:TRIGger[:SOURce]’ on page 194.

***RST** +0

Range 0–1048575

Key Entry **Ext Delay Bits**

:TRIGger[:SOURce]:EXTernal:DELay:STATe

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:CUSTom:TRIGger[:SOURce]:EXTernal:DELay:STATe ON|OFF|1|0
[:SOURce]:RADio:CUSTom:TRIGger[:SOURce]:EXTernal:DELay:STATe?
```

This command turns the trigger delay on or off when using an external trigger source.

For setting the delay time, see “:TRIGger[:SOURce]:EXTernal:DELay” , and for more information on configuring an external source, see‘:TRIGger[:SOURce]’ on page 194.

***RST** 0

Key Entry **Ext Delay Off On**

:TRIGger[:SOURce]:EXTernal:SLOPe

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:CUSTom:TRIGger[:SOURce]:EXTernal:SLOPe POSitive|NEGative
[:SOURce]:RADio:CUSTom:TRIGger[:SOURce]:EXTernal:SLOPe?
```

This command sets the polarity for an external trigger signal while using the continuous or single triggering modes. To set the polarity for gating, see ‘:TRIGger:TYPE:GATE:ACTive’ on page 193.

The POSitive and NEGative selections correspond to the high (positive) and low (negative) states of the external trigger signal. For example, when you select POSitive, the waveform responds (transmits) during the high state of the trigger signal. When the PSG receives multiple trigger occurrences when only one is required, the signal generator uses the first trigger and ignores the rest.

For more information on configuring an external trigger source and to select external as the trigger source, see ‘:TRIGger[:SOURce]’ on page 194.

***RST** NEG

Key Entry **Ext Polarity Neg Pos**

[:STATe]

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:CUSTom[:STATe] ON|OFF|1|0
[:SOURce]:RADio:CUSTom[:STATe]?
```

This command enables or disables the Custom modulation format.

Although the Custom modulation is enabled with this command, the RF carrier is not modulated unless you also activate the front panel **Mod On/Off** hardkey.

***RST**	0
Key Entry	**Custom Off On**

:VCO:CLOCk

Supported	E8267D with Option 601or 602

```
[:SOURce]:RADio:CUSTOM:VCO:CLOCk INTernal|EXTernal
[:SOURce]:RADio:CUSTOM:VCO:CLOCk?
```

This command enables an internal or external VCO clock. The external VCO clock is connected to the rear-panel BASEBAND GEN CLK IN connector. Use the `:DACS:ALIGn` command after an external VCO clock is first applied to the BASEBAND GEN CLK IN connector or when the VCO signal is lost and then re-acquired.

***RST**	Int
Key Entry	**VCO Clock Ext Int**

Digital Modulation Subsystem ([:SOURce]:DM)

:EXTernal:Filter

Supported E8267D with Option 601or 602

```
[:SOURce]:DM:EXTernal:Filter 40e6|THRough
[:SOURce]:DM:EXTernal:Filter?
```

This command selects the filter or through path for I/Q signals routed to the rear-panel I and Q outputs.

40e6	This choice applies a 40 MHz baseband filter.
THRough	This choice bypasses filtering.
***RST**	THR
Key Entry	**40.000 MHz** **Through**

:EXTernal:Filter:AUTO

Supported E8267D with Option 601or 602

```
[:SOURce]:DM:EXTernal:Filter:AUTO ON|OFF|1|0
[:SOURce]:DM:EXTernal:Filter:AUTO?
```

This command enables or disables the automatic selection of the filters for I/Q signals out the rear-panel of the instrument.

ON	This choice automatically selects a digital modulation filter.
OFF	This choice disables the auto feature, which enables you to select a digital modulation filter or through path (see “:EXTernal:Filter” on page 198).
***RST**	1 (ON)
Key Entry	**I/Q Output Filter Manual Auto**

:EXTernal:HCRest

Supported E8267D with Option 601or 602

```
[:SOURce]:DM:EXTernal:HCRest[STATe] ON|OFF|1|0
[:SOURce]:DM:EXTernal:HCRest [STATe]?
```

This command changes the operating condition to accommodate I/Q inputs with a high crest factor.

`ON` (1)	This choice turns high crest mode on for externally applied signals with high crest factors. High crest mode allows the signal generator to process these signals with less distortion. For crest factors higher than 4 dB, I/Q drive levels should be reduced by 1 dB for each dB above that level. In high crest mode, the maximum output level is reduced and power level accuracy is degraded.
`OFF` (0)	This choice disables the high crest mode.
***RST**	NORM
Key Entry	**High Crest Mode Off On**

:EXTernal:POLarity

Supported E8267D with Option 601or 602

```
[:SOURce]:DM:EXTernal:POLarity NORMal|INVert|INVerted
[:SOURce]:DM:EXTernal:POLarity?
```

This command sets the phase polarity for the I/Q signal.

This command is for backward compatibility with the appropriate ESG E44xxB.

***RST**	NORM
Key Entry	**Int Phase Polarity Normal Invert**

:EXTernal:SOURce

Supported E8267D with Option 601or 602

```
[:SOURce]:DM:EXTernal:SOURce EXTernal|INTernal|BBG1|EXT600|OFF|SUM
[:SOURce]:DM:EXTernal:SOURce?
```

This command selects the I/Q signal source that is routed to the rear-panel I and Q output connectors.

EXTernal	This choice routes a portion of the externally applied signals at the 50 ohm I and Q input connectors to the rear-panel I and Q output connectors.
INTernal	This choice is for backward compatibility and performs the same function as the BBG1 selection.
BBG1	This choice routes a portion of the baseband generator I/Q signals to the rear-panel I and Q connectors and requires Option 602.
EXT600	This choice routes a portion of the externally applied signals at the 600 ohm I and Q input connectors to the rear-panel I and Q output connectors.
OFF	This choice disables the output to the rear-panel I and Q output connectors.

The output is the analog component of the I and Q signals.

For selecting the I/Q source, refer to ":SOURce" on page 210.

***RST** EXT

Key Entry **Ext 50 Ohm** **BBG1** **Ext 600 Ohm** **Off**

:IQADjustment:DELay

Supported E8267D with Option 601or 602

```
[:SOURce]:DM:IQADjustment:DELay <val><units>
[:SOURce]:DM:IQADjustment:DELay?
```

This command sets a delay for both I and Q from the baseband to the I/Q outputs and to the RF output. This will allow you to time shift the I/Q with respect to triggering and markers. The absolute phase of both I and Q will change with respect to triggers and markers. The range limits are dependent on the current modulation format.

The variable `<val>` is a numeric expression. The `<units>` variable is expressed in seconds.

***RST** +0.00000000E+000

Key Entry **I/Q Delay**

:IQADjustment:EXTernal:COFFset

Supported E8267D with Option 601or 602

```
[:SOURce]:DM:IQADjustment:EXTernal:COFFset <val><units>
[:SOURce]:DM:IQADjustment:EXTernal:COFFset?
```

This command sets the common mode offset voltage for both the in-phase (I) and quadrature-phase (Q) signals going to the rear-panel I and Q output connectors.

The variable `<val>` is a numeric expression. The `<units>` variable is expressed in units of volts (mV–V).

This command is effective only if the state of the I/Q adjustment function is set to ON. Refer to ":IQADjustment[:STATe]" on page 206.

***RST** +0.00000000E+000

Range –3 to 3

Key Entry **Common Mode I/Q Offset**

:IQADjustment:EXTernal:DIOFfset

Supported E8267D with Option 601or 602

```
[:SOURce]:DM:IQADjustment:EXTernal:DIOFfset <val><units>
[:SOURce]:DM:IQADjustment:EXTernal:DIOFfset?
```

This command sets the differential offset voltage for an in-phase (I) signal routed to the I output connectors.

The variable `<val>` is a numeric expression. The `<units>` variable is expressed in units of volts (mV–V).

This command is effective only if the state of the I/Q adjustment function is set to ON. Refer to ":IQADjustment[:STATe]" on page 206.

***RST** +0.00000000E+000

Range –3 to 3

Key Entry **Diff. Mode I Offset**

:IQADjustment:EXTernal:DQOFfset

Supported E8267D with Option 601or 602

```
[:SOURce]:DM:IQADjustment:EXTernal:DQOFfset <val><units>
[:SOURce]:DM:IQADjustment:EXTernal:DQOFfset?
```

This command sets the differential offset voltage for a quadrature-phase (Q) signal routed to the Q output connectors.

The variable `<val>` is a numeric expression. The `<units>` variable is expressed in units of volts (mV–V).

This command is effective only if the state of the I/Q adjustment function is set to ON. Refer to ":IQADjustment[:STATe]" on page 206.

***RST** +0.00000000E+000

Range –3 to 3

Key Entry **Diff. Mode Q Offset**

:IQADjustment:EXTernal:GAIN

Supported E8267D with Option 601or 602

```
[:SOURce]:DM:IQADjustment:EXTernal:GAIN[1|2] <val><units>
[:SOURce]:DM:IQADjustment:EXTernal:GAIN?
```

This command sets the I/Q gain ratio (I/Q balance) for signals routed to the rear-panel I and Q output connectors. The I signal (`GAIN 1`) is increased for positive values and the Q signal (`GAIN 2`) level increases with negative values

This command is effective only if the state of the I/Q adjustment function is set to ON. Refer to ":IQADjustment[:STATe]" on page 206.

***RST** +0.00000000E+000

Range –4 to 4

Key Entry **I/Q Out Gain Balance**

:IQADjustment:EXTernal:IOFFset

Supported E8267D with Option 601or 602

```
[:SOURce]:DM:IQADjustment:EXTernal:IOFFset <val><units>
[:SOURce]:DM:IQADjustment:EXTernal:IOFFset?
```

This command sets the offset voltage for a signal applied to the 600 ohm I input connector.

The variable `<val>` is a numeric expression. The `<units>` variable is expressed in units of volts (mV–V).

This command is effective only if the state of the I/Q adjustment function is set to ON. Refer to ":IQADjustment[:STATe]" on page 206.

***RST** +0.00000000E+000

Range –5 to 5

Key Entry **Ext In 600 Ohm I Offset**

:IQADjustment:EXTernal:IQATten

Supported E8267D with Option 601or 602

```
[:SOURce]:DM:IQADjustment:EXTernal:IQATten <val><units>
[:SOURce]:DM:IQADjustment:EXTernal:IQATten?
```

This command sets the I/Q output attenuation level.

The variable `<val>` is a numeric expression. The `<units>` variable is expressed in units of decibels (dB).

The value set by this command is active even if the I/Q adjustment function is off.

***RST** +6.00000000E+000

Range 0–40

Key Entry **I/Q Output Atten**

:IQADjustment:EXTernal:QOFFset

Supported E8267D with Option 601or 602

```
[:SOURce]:DM:IQADjustment:EXTernal:QOFFset <val><units>
[:SOURce]:DM:IQADjustment:EXTernal:QOFFset?
```

This command sets the offset voltage for a signal applied to the 600 ohm Q input connector. The variable `<val>` is a numeric expression. The `<units>` variable is expressed in units of volts (mV–V).

This command is effective only if the state of the I/Q adjustment function is set to ON. Refer to ":IQADjustment[:STATe]" on page 206.

***RST** +0.00000000E+000

Range –5 to 5

Key Entry **Ext In 600 Ohm Q Offset**

:IQADjustment:GAIN

Supported E8267D with Option 601or 602

```
[:SOURce]:DM:IQADjustment:GAIN[1|2] <val>
[:SOURce]:DM:IQADjustment:GAIN?
```

This command sets the gain for the I signal (`GAIN 1`) relative to the Q signal, (`GAIN 2`). The gain ratio is expressed in decibels (dB).

This command is effective only if the state of the I/Q adjustment function is set to ON. Refer to ":IQADjustment[:STATe]" on page 206.

***RST** +0.00000000E+000

Range –4 to 4 dB

Key Entry **I/Q Gain Balance Source 1**

:IQADjustment:IOFFset

Supported E8267D with Option 601or 602

```
[:SOURce]:DM:IQADjustment:IOFFset <val><units>
[:SOURce]:DM:IQADjustment:IOFFset?
```

This command adjusts the I channel offset value.

The variable <val> is a numeric expression. The <units> variable is expressed in units of percent with 100% equivalent to 500 mV DC at the input connector. The minimum resolution is 0.025 percent.

When using this command to minimize the LO feedthrough signal, optimum performance is achieved when the command is sent after all other I/Q path commands are executed, such as those that change the internal phase polarity or adjust the modulator attenuator. If other adjustments are made after minimizing is performed, the LO feedthrough signal may increase.

This command is effective only if the state of the I/Q adjustment function is set to ON. Refer to ":IQADjustment[:STATe]" on page 206.

***RST** +0.00000000E+000

Range –5E1 to +5E1

Key Entry **I Offset**

:IQADjustment:QOFFset

Supported E8267D with Option 601or 602

```
[:SOURce]:DM:IQADjustment:QOFFset <val><units>
[:SOURce]:DM:IQADjustment:QOFFset?
```

This command adjusts the Q channel offset value.

The variable <val> is a numeric expression. The <units> variable is expressed in units of percent with 100% equivalent to 500 mV DC at the input connector. The minimum resolution is 0.025 percent.

When using this command to minimize the LO feedthrough signal, optimum performance is achieved when the command is sent after all other I/Q path commands are executed, such as those that change the internal phase polarity or adjust the modulator attenuator. If other adjustments are made after minimizing is performed, the LO feedthrough signal may increase.

This command is effective only if the state of the I/Q adjustment function is set to ON. Refer to ":IQADjustment[:STATe]" on page 206.

***RST** +0.00000000E+000

Range –5E1 to +5E1

Key Entry **Q Offset**

:IQADjustment:QSKew

Supported E8267D with Option 601or 602

```
[:SOURce]:DM:IQADjustment:QSKew <val><units>
[:SOURce]:DM:IQADjustment:QSKew?
```

This command adjusts the phase angle between the I and Q vectors.

The variable `<val>` is a numeric expression. The `<units>` variable is expressed in units of degrees with a minimum resolution of 0.1.

If the signal generator is operating at frequencies greater than 3.3 GHz, quadrature skew settings greater than ±5 degrees will not be within specifications.

Positive skew increases the angle from 90 degrees while negative skew decreases the angle from 90 degrees. When the quadrature skew is zero, the phase angle is 90 degrees.

This command is effective only if the state of the I/Q adjustment function is set to ON. Refer to ":IQADjustment[:STATe]" on page 206.

***RST** +0.00000000E+000

Range −1E1 to +1E1

Key Entry **Quadrature Skew**

:IQADjustment:SKEW

Supported E8267D with Option 601or 602

```
[:SOURce]:DM:IQADjustment:SKEW <val><units>
[:SOURce]:DM:IQADjustment:SKEW?
```

This command changes the input skew which is a time delay between I and Q. Equal and opposite skew is applied to both paths (RF Output path and I/Q output paths) simultaneously. A positive value delays the I signal relative to the Q signal, and a negative value delays the Q signal relative to the I signal.

If the internal I/Q correction path is set to RF or BB the I/Q signals are already optimized and adjusting I/Q skew would add an impairment to the signals. If the internal I/Q correction path is set to Off, then adjusting the I/Q skew could improve the I/Q signals. The I/Q skew adjustment cannot be performed on the MSK, FSK, and C4FM constant envelope modulations.

I/Q skew adjustments are preserved when the instrument state is saved. I/Q skew adjustments are also preserved when instrument settings are changed. If the signal generator is calibrated, the skew adjustments are added to the calibration value used for the given signal generator state. If the signal generator is uncalibrated, the skew adjustments are re-applied directly.

Using I/Q skew while playing a user FIR file greater than 32 symbols will generate an error.

The variable `<val>` is expressed as a number. The variable `<unit>` is expressed in units of picoseconds or nanoseconds. Limits are determined by the modulation configuration.

***RST**	+0.00000000E+000
Range	–5.0 to 5.0 Seconds
Key Entry	**I/Q Skew**

:IQADjustment[:STATe]

Supported E8267D

```
[:SOURce]:DM:IQADjustment[:STATe] ON|OFF|1|0
[:SOURce]:DM:IQADjustment[:STATe]?
```

This command enables or disables the I/Q adjustments.

***RST**	0 (OFF)
Key Entry	**I/Q Adjustments Off On**

:MODulation:ATTen

Supported E8267D with Option 601or 602

```
[:SOURce]:DM:MODulation:ATTen <val>
[:SOURce]:DM:MODulation:ATTen?
```

This command attenuates the I/Q signals being modulated through the signal generator’s RF path and automatically sets the modulator attenuator into manual mode.

***RST**	+2.00000000E+000
Range	0–40 dB
Key Entry	**Modulator Atten Manual Auto**

:MODulation:ATTen:AUTO

Supported E8267D with Option 601or 602

```
[:SOURce]:DM:MODulation:ATTen:AUTO ON|OFF|1|0
[:SOURce]:DM:MODulation:ATTen:AUTO?
```

This command enables or disables the I/Q attenuation auto mode. The auto mode will be switched to manual if the signal generator receives a `[:SOURce]:DM:MODulation:ATTen <val>` command.

`ON` (`1`) This choice enables the attenuation auto mode which optimizes the modulator attenuation for the current signal generator settings.

`OFF (0)`	This choice holds the attenuator at its current setting or at a selected value. Refer to ‘:MODulation:ATTen’ on page 206 for setting the attenuation value.
***RST**	1
Key Entry	**Modulator Atten Manual Auto**

:MODulation:ATTen:EXTernal

Supported E8267D

```
[:SOURce]:DM:MODulation:ATTen:EXTernal DEFault|MANual|MEASure
[:SOURce]:DM:MODulation:ATTen:EXTernal?
```

This command selects the method for setting the external I/Q input level for automatic attenuation. Modulation attenuation must be in auto mode.

DEFault	Use this choice to set the external I/Q input level to the default value of 500.0 mV.
MANual	Use this choice to manually set the external I/Q input level. Refer to ‘:MODulation:ATTen:EXTernal:LEVel’ on page 207 to set the input level.
MEASurement	Use this choice to measure the voltage level at the external I/Q inputs. The measurement will be used as the input level setting. Refer to ‘:MODulation:ATTen:EXTernal:LEVel:MEASurement’ on page 208 to perform the measurement.
***RST**	DEFault
Key Entry	**Ext Input Level (nnn mV) Default Man Meas**

:MODulation:ATTen:EXTernal:LEVel

Supported E8267D

```
[:SOURce]:DM:MODulation:ATTen:EXTernal:LEVel <val><volt_units>
[:SOURce]:DM:MODulation:ATTen:EXTernal:LEVel?
```

This command manually attenuates the I/Q signals at the rear-panel I/Q outputs and sets the external I/Q input level for automatic attenuation.

The variable <val> is expressed as a number. The variable <volt_units> is expressed in units of volts root-mean-square (RMS).

***RST**	+4.00000000E–001
Range	1E–1 to 1E0

Key Entry **I/Q Output Atten**

:MODulation:ATTen:EXTernal:LEVel:MEASurement

Supported E8267D

```
[:SOURce]:DM:MODulation:ATTen:EXTernal:LEVel:MEASurement
```

Use this command to measure the voltage level at the external I/Q inputs. The measurement will be used as the input level setting for automatic attenuation. See ':MODulation:ATTen:EXTernal' on page 207.

Key Entry **Do External Input Level Measurement**

:MODulation:ATTen:OPTimize:BANDwidth

Supported E8267D

```
[:SOURce]:DM:MODulation:ATTen:OPTimize:BANDwidth <val><bw_rate_units>
[:SOURce]:DM:MODulation:ATTen:OPTimize:BANDwidth?
```

This command sets the expected bandwidth of the external I/Q signal. The automatic external I/Q attenuator will use this setting to determine the proper attenuation level.

The variable <bw_rate_unit> is expressed in units of samples per second (sps).

***RST** +1.00000000E+006

Range 1E3–100E6

Key Entry **Optimize for (nnn sps) Bandwidth**

:MODulation:FILTer

Supported E8267 with Option 601or 602

```
[:SOURce]:DM:MODulation:FILTer 40e6|THRough
[:SOURce]:DM:MODulation:FILTer?
```

This command enables you to select a filter or through path for I/Q signals modulated onto the RF carrier. Selecting a filter with this command automatically sets ":MODulation:FILTer:AUTO" to `OFF` (`0`).

40E6 This choice applies a 40 MHz baseband filter to the I/Q signals.

THRough This choice bypasses filtering.

***RST** THR

Key Entry **40.000 MHz** **Through**

:MODulation:FILTer:AUTO

Supported E8267D with Option 601or 602

```
[:SOURce]:DM:MODulation:FILTer:AUTO ON|OFF|1|0
[:SOURce]:DM:MODulation:FILTer:AUTO?
```

This command enables or disables the automatic selection of the filters for I/Q signals modulated onto the RF carrier.

ON(1)	This choice will automatically select a digital modulation filter.
OFF(0)	This choice disables the auto feature which lets you select a digital modulation filter or through path. Refer to ":IQ:MODulation:FILTer" on page 215 for selecting a filter or through path.
***RST**	1
Key Entry	**I/Q Mod Filter Manual Auto**

:POLarity[:ALL]

Supported E8267D with Option 601or 602

```
[:SOURce]:DM:POLarity[:ALL] NORMal|INVert|INVerted
[:SOURce]:DM:POLarity?
```

This command sets the digital modulation phase polarity.

NORMal	This choice selects normal phase polarity for the I and Q signals.
INVert (INVerted)	This choice flips the I and Q signals by routing the I signal to the Q input of the I/Q modulator and the Q signal to the I input.
***RST**	NORM
Key Entry	**Int Phase Polarity Normal Invert**

:SKEW:PATH

Supported E8267D with Option 601or 602

```
[:SOURce]:DM:SKEW:PATH RF|BB
[:SOURce]:DM:SKEW:PATH?
```

This command selects the skew path.

RF	When RF is selected, the skew is optimized for the I/Q signal applied to the RF Output. The baseband (BB) output will be functional, but the I/Q skew applied will be optimized for the RF path. When using this choice, seven symbols of

latency are added to the Arb based waveform. While in real-time mode, the maximum number of user symbols for the FIR is limited to 32.

BB — When BB is selected, the skew is optimized for the I/Q signal outputs on the rear-panel. The RF Output will be functional, but the I/Q skew applied will be optimized for the BB path. When using this choice, seven symbols of latency are added to the ARB based waveform. While in real-time mode, the maximum number of user symbols for the FIR is limited to 32.

***RST** INT

Key Entry **Int I/Q Skew Corrections RF BB Off**

:SKEW[:STATe]

Supported E8267D with Option 601or 602

```
[:SOURce]:DM:SKEW[:STATe] ON|OFF|1|0
[:SOURce]:DM:SKEW[:STATe]?
```

This command enables or disables the I/Q skew correction function.

***RST** 1

Key Entry **Int I/Q Skew Corrections RF BB Off**

:SOURce

Supported E8267D with Option 601or 602

```
[:SOURce]:DM:SOURce[1]|2 EXTernal|INTernal|BBG1|EXT600|OFF
[:SOURce]:DM:SOURce?
```

This command selects the I/Q modulator source.

EXTernal — This choice selects a 50 ohm impedance for the I and Q input connectors and routes the applied signals to the I/Q modulator.

INTernal — This choice is for backward compatibility with the appropriate ESG E44xxB and performs the same function as the BBG1 selection.

BBG1 — This choice selects the baseband generator as the source for the I/Q modulator.

EXT600 — This choice selects a 600 ohm impedance for the I and Q input connectors and routes the applied signals to the I/Q modulator.

OFF — This choice disables the digital modulation source.

***RST** EXT

Key Entry **Ext 50 Ohm** **BBG1** **Ext 600 Ohm** **Off**

:SRATio

Supported All

```
[:SOURce]:DM:SRATio <val><units>
[:SOURce]:DM:SRATio?
```

This command enables you to set the power level difference (ratio) between the source one and the source two signals when the two signals are summed together. A positive ratio value reduces the amplitude for source two while a negative ratio value reduces the amplitude for source one.

The range for the summing ratio is dependent on the modulator attenuator (mod atten) setting for the signal generator that is summing the signals together. The minimum range is achieved when the modulator attenuator setting is zero and the maximum range is reached when the maximum attenuator value is used. The range can be calculated using the following formula:

± Range = 50 dB + Mod Atten

The variable `<val>` is expressed as a number. The variable `<units>` is expressed in units of decibels (dB).

For setting the modulator attenuator for real-time modulation formats, see ‘:IQ:MODulation:ATTen’ on page 239 and ‘:IQ:MODulation:ATTen:AUTO’ on page 239. For setting the modulator attenuator for Arb modulation formats, refer to the SCPI command subsystem for the Arb format being used and find the commands that contain the command mnemonics IQ:MODulation:ATTen.

***RST** +0.00000000E+000

Range *Min*: ± 50 dB *Max*: ± 90 dB

Key Entry **Summing Ratio (SRC1/SRC2) x.xx dB**

:STATe

Supported E8267D with Option 601or 602

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

This command enables or disables the internal I/Q modulator.

The I/Q modulator is enabled whenever a digital format is turned on.

The `I/Q` annunciator will be shown on the signal generator display whenever the I/Q modulator is on.

***RST** 0

Key Entry **I/Q Off On**

Dual ARB Subsystem–Option 601 or 602 ([:SOURce]:RADio:ARB)

:CLIPping

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:ARB:CLIPping "<file name>",IJQ|IORQ,<val>[,<val>]
```

This command sets the clipping level of the selected waveform segment to a percentage of its highest peak. For more information about clipping, refer to the *PSG User's Guide*.

The variable `<val>` is expressed as a percentage within a 10–100% range.

IJQ This choice clips the composite I/Q waveform.

IORQ This choice clips I and Q separately. When this choice is enabled, percentage values for both I and Q must be specified.

A value of 100 percent equates to no clipping.

For information on the file name syntax, see "File Name Variables" on page 12.

***RST** IJQ <*val*>: +100

Range <*val*>: 10–100 (0.1% resolution)

Key Entry **Clipping** **Clipping Type |I+jQ| |I|,|Q|** **Clip |I+jQ| To**
Clip |I| To **Clip |Q| To**

:DACS:ALIGn

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:ARB:DACS:ALIGn
```

This command aligns and zeros out the signal generator's I/Q DAC circuitry. This operation is required any time the external VCO clock signal is lost and re-acquired or when an external VCO clock signal is first applied to the BASEBAND GEN CLK IN connector.

***RST** N/A

Range N/A

Key Entry **Align DACs**

:GENerate:SINE

Supported E8267D with Option 601or 602

[:SOURce]:RADio:ARB:GENerate:SINE ["<file name>"][,<osr>],[<scale>],[I|Q|IQ]

This command creates a file (using the "<file name>" name) and stores a generated sine wave.

<osr> This variable sets the oversample ratio, which must be a value that is ≥ 4. If the specified over sample ratio is < 60 (the minimum number of samples or I/Q points), multiple periods are generated to create a waveform with at least 60 samples. The number of periods that will be created is 60 ÷ <osr> (quotient will round off to a whole number). A waveform with an oversample ratio ≥ 60 has one period.

The maximum value for the range below is determined by the available baseband memory.

Executing this command without the "<file name>" variable will generate a factory default SINE_TEST_WFM file. When using the variable "<file name>" for this command, the “@” or “:” characters are not allowed.

Range **4–32Msamples**

:HEADer:CLEar

Supported E8267D with Option 601or 602

[:SOURce]:RADio:ARB:HEADer:CLEar

This command clears the header information from the header file used by this modulation format. Header information consists of signal generator settings and marker routings associated with the waveform file. Refer to the *PSG User’s Guide* for information on header files.

The dual ARB must be on for this command to function. To turn on the dual ARB, see ‘[:STATe]’ on page 234

***RST** N/A

Key Entry **Clear Header**

:HEADer:SAVE

Supported E8267D with Option 601or 602

[:SOURce]:RADio:ARB:HEADer:SAVE

This command saves the header information to the header file used by this modulation format. Header information consists of signal generator settings and marker routings associated with the waveform file. Refer to the *PSG User's Guide* for information on header files.

The dual ARB must be on for this command to function. To turn on the dual ARB, see '[:STATe]' on page 234

***RST**	N/A
Key Entry	**Save Setup To Header**

:IQ:EXTernal:FILTer

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:ARB:IQ:EXTernal:FILTer 40e6|THRough
[:SOURce]:RADio:ARB:IQ:EXTernal:FILTer?
```

This command selects the filter or through path for I/Q signals routed to the rear-panel I and Q outputs. The filter has no effect on the modulated RF signal. Selecting a filter using this command will automatically set ':IQ:EXTernal:FILTer:AUTO' on page 214 to OFF(0) mode.

40e6	This choice applies a 40 MHz baseband filter.
THRough	This choice bypasses filtering.
***RST**	THR
Key Entry	**40.000 MHz** **Through**

:IQ:EXTernal:FILTer:AUTO

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:ARB:IQ:EXTernal:FILTer:AUTO ON|OFF|1|0
[:SOURce]:RADio:ARB:IQ:EXTernal:FILTer:AUTO?
```

This command enables or disables the automatic selection of the filters for I/Q signals routed to the rear-panel I/Q outputs.

`ON` (1)	This choice will automatically select a digital modulation filter optimized for the current signal generator settings.
`OFF` (0)	This choice disables the auto feature which lets you select a digital modulation filter or through path. Refer to ":IQ:EXTernal:FILTer" on page 214 for selecting a filter or through path.
***RST**	1
Key Entry	**I/Q Output Filter Manual Auto**

:IQ:MODulation:ATTen

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:ARB:IQ:MODulation:ATTen <val><units>
[:SOURce]:RADio:ARB:IQ:MODulation:ATTen?
```

This command sets the attenuation level of the I/Q signals being modulated through the signal generator RF path.

The variable `<val>` is expressed as a number. The variable `<units>` are in decibels (dB).

***RST** +2.00000000E+000

Range 0–40

Key Entry **Modulator Atten Manual Auto**

:IQ:MODulation:ATTen:AUTO

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:ARB:IQ:MODulation:ATTen:AUTO ON|OFF|1|0
[:SOURce]:RADio:ARB:IQ:MODulation:ATTen:AUTO?
```

This command enables or disables the I/Q attenuation auto mode.

`ON` (`1`) This choice enables the attenuation auto mode which optimizes the modulator attenuation for the current conditions.

`OFF` (`0`) This choice holds the attenuator at its current setting or at a selected value. Refer to ':IQ:MODulation:ATTen' on page 215 for setting the attenuation value.

***RST** 1

Key Entry **Modulator Atten Manual Auto**

:IQ:MODulation:FILTer

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:ARB:IQ:MODulation:FILTer 40e6|THRough
[:SOURce]:RADio:ARB:IQ:MODulation:FILTer?
```

This command enables you to select a filter or through path for I/Q signals modulated onto the RF carrier. This filter has no effect on the I/Q signal out the rear-panel. Selecting a filter using this command will automatically set ':IQ:MODulation:FILTer:AUTO' on page 216 to OFF(0) mode.

40E6 This choice applies a 40 MHz baseband filter to the I/Q signals.

THRough This choice bypasses filtering.

***RST** THR

Key Entry **40.000 MHz** **Through**

:IQ:MODulation:FILTer:AUTO

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:ARB:IQ:MODulation:FILTer:AUTO ON|OFF|1|0
[:SOURce]:RADio:ARB:IQ:MODulation:FILTer:AUTO?
```

This command enables or disables the automatic selection of the filters for I/Q signals modulated onto the RF carrier.

`ON` (1) This choice will automatically select a digital modulation filter.

`OFF` (0) This choice disables the auto feature which lets you select a digital modulation filter or through path. Refer to ":IQ:MODulation:FILTer" on page 215 for selecting a filter or through path.

***RST** 1

Key Entry **I/Q Mod Filter Manual Auto**

:MARKer:CLEar

Supported E8267D with Option 601 or 602

```
[:SOURce]:RADio:ARB:MARKer:CLEar "<file name>",<marker>,<first_Point>,
<last_point>
```

This command clears a single marker point or a range of marker points on a waveform segment for the selected marker (1–4). The Dual ARB player and all of the ARB formats use this command.

`"<file name>"` This variable specifies the name of the waveform segment file. AUTOGEN_WAVEFORM is the file to use when clearing marker points for an active ARB format. The PSG automatically creates and assigns this name to the file when you turn an ARB format on, and it uses the same name for all ARB formats. When all formats are off, this file is still available for use in the Dual ARB player. For information on the file name syntax, see "File Name Variables" on page 12.

`<marker>` This variable sets the marker number; an integer value from one to four.

`<first_point>` This variable defines the first point in a range of points. The number must be greater than or equal to one, and less than or equal to the total number of waveform points.

If you enter a value for either the first marker point or the last marker point that would make the first marker point occur after the last, the last marker point automatically adjusts to match the first marker point.

`<last_point>` This variable defines the last point in a range of points. The number must be greater than or equal to the first point, and less than or equal to the total number of waveform points.

To clear a single marker point, use the same marker point for the first and last point variables. For more information on markers and ARB files, refer to the *PSG User's Guide*.

Range *<marker>*: 1–4

<first_Point>: 1–number of waveform points

<last_point>: <first_Point>–number of waveform points

Key Entry **Set Marker Off Range Of Points** **Marker 1 2 3 4** **First Mkr Point** **Last Mkr Point**

:MARKer:CLEar:ALL

Supported E8267D with Option 601 or 602

```
[:SOURce]:RADio:ARB:MARKer:CLEar:ALL "<file name>",<marker>
```

This command clears all marker points on a waveform segment for the selected marker (1–4). The Dual ARB player and all of the ARB formats use this command. With all marker points cleared, the event output signal level is zero volts.

`"<file name>"` This variable specifies the name of the waveform segment file. AUTOGEN_WAVEFORM is the file to use when clearing marker points for an active ARB format. The PSG automatically creates and assigns this name to the file when you turn an ARB format on, and it uses the same name for all ARB formats. When all formats are off, this file is still available for use in the Dual ARB player. For information on the file name syntax, see "File Name Variables" on page 12.

`<marker>` This variable sets the marker number; an integer value from one to four.

Range *<marker>*: 1–4

Key Entry **Marker 1 2 3 4** **Set Marker Off All Points**

:MARKer:ROTate

Supported E8267D with Option 601 or 602

```
[:SOURce]:RADio:ARB:MARKer:ROTate "<file name>",<rotate_count>
```

This command shifts the marker points for all markers in a waveform segment by the value of the <rotate_count> variable. The Dual ARB player and all of the ARB formats use this command.

You can use a positive or negative value. When a marker point is close to the end of the waveform and the <rotate_count> value is greater than the number of remaining sample points, but less than the total number of waveform points (waveform points and sample points are the same thing), the marker points that would move beyond the end of the waveform wrap to the beginning of the waveform. For example, if a marker point resides at sample point 195 out of 200, and the <rotate_count> value is twenty-five, the marker point wraps to the beginning of the waveform and finally reside at the twentieth waveform point.

To set the marker points in a waveform, refer to “:MARKer:[SET]” on page 218.

"<file name>" This variable specifies the name of the waveform segment file. AUTOGEN_WAVEFORM is the file to use when modifying marker points for an active ARB format. The PSG automatically creates and assigns this name to the file when you turn an ARB format on, and it uses the same name for all ARB formats. When all formats are off, this file is still available for use in the Dual ARB player. For information on the file name syntax, see “File Name Variables” on page 12.

Range – (n – 1) to (n – 1)

n = number of points in the waveform

:MARKer:[SET]

Supported E8267D with Option 601 or 602

```
[:SOURce]:RADio:ARB:MARKer:[SET] "<file name>",<marker>,<first_point>,
<last_point>,<skip_count>
```

This command sets a single marker point or a range of marker points on a waveform segment for the selected marker (1–4). The Dual ARB player and all of the ARB formats use this command.

The PSG provides four independent markers. Each marker routes an output signal to the rear-panel event connector number (BNC—EVENT 1 and EVENT 2 or AUXILIARY I/O—EVENT 3 and EVENT 4) that corresponds to the marker number. A marker consists of marker points placed at defined sample points in a waveform segment. This means that a marker point cannot be less than one or greater than the last sample point in the waveform. Marker points are cumulative, so multiple command executions with different range values, without first clearing the existing points, places additional marker points on the waveform. Because of this cumulative behavior, it is a good practice to clear existing marker points prior to setting new points. This will eliminate unexpected marker pulses. Refer to ‘:MARKer:CLEar’ on page 216 and ‘:MARKer:CLEar:ALL’ on page 217 for information on clearing marker points.

For waveforms generated on the signal generator (baseband generator), the PSG automatically places a marker point at the first waveform sample for markers one and two.

NOTE You can set markers for either positive or negative polarity. The following discussions for this command assume positive marker polarity. When using negative marker polarity, the marker pulses occur during the periods of no marker points.

There are three ways to place marker points using this command:

- consecutive marker points over a range that collectively create a single marker pulse that spans the range
- equally spaced marker points over a range, so that a marker pulse occurs at each sample point that coincides with a marker point (Using this method, you can configure a clock signal by setting the <skip_count> variable to one.)
- a single marker point placed at a specific sample point in the waveform, which outputs a single pulse relative to the marker point location (To configure a single marker point, set the first and last points to the same number.)

For more information on markers, refer to the *PSG User's Guide*.

The following list describes the command variables:

"<file name>" This variable specifies the name of the waveform segment file. Use the file AUTOGEN_WAVEFORM when configuring marker points for an active ARB format. The PSG automatically creates and assigns this name to the file when you turn an ARB format on, and it uses the same name for all ARB formats. When all formats are off, this file is still available for use in the Dual ARB player. For information on the file name syntax, see "File Name Variables" on page 12.

<marker> This variable sets the marker number; an integer value from one to four.

<first_point> This variable defines the first point in the range over which the marker is placed. This number must be greater than or equal to one, and less than or equal to the total number of waveform points.

If you enter a value for either the first marker point or the last marker point that would make the first marker point occur after the last, the last marker point is automatically adjusted to match the first marker point.

<last_point> This variable defines the last point in the range over which the marker will be placed. This value must be greater than or equal to the first point, and less than or equal to the total number of waveform points.

<skip_count> This variable defines the marker point pattern across the range. A zero value means the marker points occur consecutively across the range. A value greater than zero creates a repeating marker point pattern across the range, where the gap between the marker points is equal to the <skip_count> value. The gaps begin

after the first marker point. Each marker point in the pattern, which is only one point wide, produces a marker pulse.

Range <*marker*>: 1–4

<*first_Point*>: 1–number of waveform points

<*last_point*>: <first_Point>–number of waveform points

<*skip_count*>: 0–number of points in the range

Key Entry **Set Marker on Range Of Points** **Marker 1 2 3 4** **First Mkr Point** **Last Mkr Point** **# Skipped Points** **Apply to Waveform**

:MDEStination:ALCHold

Supported E8267D with Option 601 or 602

CAUTION Incorrect ALC sampling can create a sudden unleveled condition that may create a spike in the RF output potentially damaging a DUT or connected instrument. Ensure that you set markers to let the ALC sample over an amplitude that accounts for the high power levels within the signal.

```
[:SOURce]:RADio:ARB:MDEStination:ALCHold NONE|M1|M2|M3|M4
[:SOURce]:RADio:ARB:MDEStination:ALCHold?
```

This command disables the marker ALC hold function, or it enables the marker hold function for the selected marker. For setting markers, see ":MARKer:[SET]" on page 218.

Use the ALC hold function when you have a waveform signal that incorporates idle periods, or when the increased dynamic range encountered with RF blanking is not desired. The ALC leveling circuitry responds to the marker signal during the marker pulse (marker signal high), averaging the modulated signal level during this period.

The ALC hold function operates during the low periods of the marker signal. The marker polarity determines when the marker signal is high. For a positive polarity, this is during the marker points. For a negative polarity, this is when there are no marker points. For setting a marker's polarity, see ':MPOLarity:MARKer1|2|3|4' on page 222.

NOTE Do not use the ALC hold for more than 100 ms, because it can affect the waveform's output amplitude.

The marker signal has a minimum of a two-sample delay in its response relative to the waveform signal response. To compensate for the marker signal delay, offset marker points from the waveform sample point at which you want the ALC sampling to begin.

The ALC hold setting is part of the file header information, so saving the setting to the file header saves the current marker routing for the waveform file.

NOTE A waveform file that has unspecified settings in the file header uses the previous waveform's routing settings.

For more information on the marker ALC hold function, see the *PSG User's Guide*. For setting the marker points, see ':MARKer:[SET]' on page 218.

NONE	This terminates the marker ALC hold function.
M1–M4	These are the marker choices. The ALC hold feature uses only one marker at a time.
***RST**	NONE
Key Entry	**None** **Marker 1** **Marker 2** **Marker 3** **Marker 4**
Remarks	N/A

:MDEStination:PULSe

Supported E8267D with Option 601 or 602

CAUTION The pulse function incorporates ALC hold. Incorrect ALC sampling can create a sudden unleveled condition that may create a spike in the RF output potentially damaging a DUT or connected instrument. Ensure that you set markers to let the ALC sample over an amplitude that accounts for the high power levels within the signal.

```
[:SOURce]:RADio:ARB:MDEStination:PULSe NONE|M1|M2|M3|M4
[:SOURce]:RADio:ARB:MDEStination:PULSe?
```

This command disables the marker RF blanking/pulse function, or it enables the marker RF blanking/pulse function for the selected marker.

This function automatically incorporates the ALC hold function, so there is no need to select both functions for the same marker.

NOTE Do not use ALC hold for more than 100 ms, because it can affect the waveform's output amplitude.

The signal generator blanks the RF output when the marker signal goes low. The marker polarity determines when the marker signal is low. For a positive polarity, this is during the marker points. For a negative polarity, this is when there are no marker points. For setting a marker's polarity, see ':MPOLarity:MARKer1|2|3|4' on page 222.

NOTE Set marker points prior to using this function. Enabling this function without setting marker points may create a continuous low or high marker signal, depending on the marker polarity. This creates the condition where there is either no RF output or a continuous RF output. See ':MARKer:[SET]' on page 218 for setting the marker points.

The marker signal has a minimum of a two-sample delay in its response relative to the waveform signal response. To compensate for the marker signal delay, offset marker points from the waveform sample point at which you want the RF blanking to begin.

The RF blanking setting is part of the file header information, so saving the setting to the file header saves the current marker routing for the waveform file.

NOTE A waveform file that has unspecified settings in the file header uses the previous waveform's routing settings. This could create the situation where there is no RF output signal, because the previous waveform used RF blanking.

For more information on the marker RF blanking function, see the *PSG User's Guide*.

NONE	This terminates the marker RF blanking/pulse function.
M1–M4	These are the marker choices. The RF blanking/pulse feature uses only one marker at a time.
***RST**	NONE
Key Entry	**None** **Marker 1** **Marker 2** **Marker 3** **Marker 4**

:MPOLarity:MARKer1|2|3|4

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:ARB:MPOLarity:MARKer1|2|3|4 NEGative|POSitive
[:SOURce]:RADio:ARB:MPOLarity:MARKer1|2|3|4?
```

This command sets the polarity for the selected marker.

For a positive marker polarity, the marker signal is high during the marker points. For a negative marker polarity, the marker signal is high during the period of no marker points.

***RST** POS

Key Entry **Marker 1 Polarity Neg Pos** **Marker 2 Polarity Neg Pos** **Marker 3 Polarity Neg Pos** **Marker 4 Polarity Neg Pos**

:REFerence:EXTernal:FREQuency

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:ARB:REFerence:EXTernal:FREQuency <val>
[:SOURce]:RADio:ARB:REFerence:EXTernal:FREQuency?
```

This command allows you to enter the frequency of the external reference.

The variable `<val>` is expressed in units of Hertz (Hz–MHz).

The value specified by this command is effective only when you are using an external ARB reference applied to the BASEBAND GEN REF IN rear-panel connector.

To specify external as the ARB reference source type, refer to “:REFerence[:SOURce]” on page 223.

***RST** +1.00000000E+007

Range 2.5E5–1E8

Key Entry **Reference Freq**

:REFerence[:SOURce]

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:ARB:REFerence[:SOURce] INTernal|EXTernal
[:SOURce]:RADio:ARB:REFerence[:SOURce]?
```

This command selects either an internal or external reference for the waveform clock.

If the EXTernal choice is selected, the external frequency value *must* be entered and the signal must be applied to the BASEBAND GEN REF IN rear-panel connector.

Refer to “:REFerence:EXTernal:FREQuency” on page 223 to enter the external reference frequency.

***RST** INT

Key Entry **ARB Reference Ext Int**

:RETRigger

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:ARB:RETRigger ON|OFF|IMMediate
[:SOURce]:RADio:ARB:RETRigger?
```

This commands selects the waveform’s response to a trigger signal while using the single trigger mode.

When the PSG receives multiple trigger occurrences when only one is required, the signal generator uses the first trigger and ignores the rest. For more information on triggering and to select the single trigger mode, see ‘:TRIGger:TYPE’ on page 227.

The following list describes the waveform’s response to each of the command choices:

ON The waveform waits for a trigger before play begins and accepts a subsequent trigger during playback. If there is a subsequent trigger during playback, the waveform completes its current playback and then plays one more time. If there is no subsequent trigger, the waveform plays once and stops until it receives another trigger.

OFF The waveform waits for a trigger before play begins and ignores triggers during playback. To restart the waveform, you must send a trigger after the playback completes.

IMMediate The waveform waits for a trigger before play begins and accepts a subsequent trigger during playback. Upon receipt of the subsequent trigger, the waveform immediately resets and begins playing from the beginning of the file. For a waveform sequence, this means to the beginning of the first segment in the sequence.

***RST** ON

Key Entry **On** **Off** **Immediate**

:RSCAling

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:ARB:RSCaling <val>
[:SOURce]:RADio:ARB:RSCaling?
```

This command adjusts the scaling value that is applied to a waveform while it is playing. The variable `<val>` is expressed as a percentage. Runtime scaling does not alter the waveform data file. For more information about runtime scaling, refer to the *PSG User’s Guide*.

***RST** +7.00000000E+001

Range 1–100

Key Entry **Waveform Runtime Scaling**

:SCALing

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:ARB:SCALing "<file name>",<val>
```

This command sets the scaling value for the designated waveform file.
The variable `<val>` is expressed as a percentage, 1–100%. For information on file name syntax, see “File Name Variables” on page 12. For more information about waveform file scaling, refer to the *PSG User’s Guide*.

Range 1–100

Key Entry **Scaling** **Scale Waveform Data**

:SCLock:RATE

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:ARB:SCLock:RATE <1.0 kHz – 1.0e8 kHz {1.0e8 kHz}>
[:SOURce]:RADio:ARB:SCLock:RATE?
```

This command sets the sample clock rate.

The ARB should be turned on before executing this command. If this command is executed before the ARB is active, the entered value will be overridden by a calculated factory default value. Refer to ‘[:STATe]’ on page 257 to activate the modulation format.

***RST** +1.00000000E+008

Range 1–1E8

Key Entry **ARB Sample Clock**

:SEQuence

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:ARB:SEQuence[:MWAVeform]
"<filename>","<waveform>",<reps>,NONE|M1|M2|M3|M4|M1M2|M1M3|M1M4|M2M3|M2
M4|M3M4|M1M2M3|M1M2M4|M1M3M4|M2M3M4|ALL,{"<waveform>",<reps>,NONE|M1|M2|
M3|M4|M1M2|M1M3|M1M4|M2M3|M2M4|M3M4|M1M2M3|M1M2M4|M1M3M4|M2M3M4|ALL}
[:SOURce]:RADio:ARB:SEQuence[:MWAVeform]? "<file name>"
```

This command creates a waveform sequence file. The query returns the contents and segment settings of the waveform sequence file.

A waveform sequence file consists of waveform segment files, which contain data and marker information. To use a segment's marker settings, you must enable the segment's markers in the sequence. The segments play in the same order as placed into the sequence by the command. Once you create the file, you cannot edit the segment settings or add further waveform segments. Using the same waveform sequence name overwrites the existing file with the new definitions.

When you create a waveform sequence, the PSG also creates a file header for the waveform. This file header takes priority over the file headers for each waveform segment. Refer to the *PSG User's Guide* for more information on file headers. To save the file header, see ':HEADer:SAVE' on page 213.

"<file name>"	This variable names the waveform *sequence* file. For information on the file name syntax, see "File Name Variables" on page 12.
"<waveform>"	This variable specifies the name of an existing waveform *segment* file. To use a waveform segment, it must reside in volatile memory. For information on the file name syntax, see "File Name Variables" on page 12, and for more information on waveform segments, see the *PSG User's Guide*.
<reps>	This variable sets the number of times a segment plays (repeats) before the next segment in the sequence plays.
NONE	This choice disables all four markers for the waveform segment. Disabling markers means that the waveform sequence ignores the segment's marker settings.
M1, M2, M3, M4	These choices, either individually or a combination of them, enable the markers for the current waveform segment.
All	This choice enables all four markers for the waveform segment.

The following example creates a waveform sequence using the factory-supplied waveform segments:

```
rad:arb:seq "TestFile@seq","ramp_test_wfm@wfm1",25,M1M4,
"sine_test_wfm@wmf1",193,all
```

The waveform sequence consists of two segments:

- The first segment, ramp_test_wfm, has 25 repetitions with markers 1 and 4 enabled.
- The second segment, sine_test_wfm, has 193 repetitions with all four markers enabled.

Range <reps>: 1–65535

Key Entry **Build New Waveform Sequence** **Name and Store** **Insert Waveform** **Edit Repetitions** **Toggle Marker 1** **Toggle Marker 2** **Toggle Marker 3** **Toggle Marker 4**

:TRIGger:TYPE

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:ARB:TRIGger:TYPE CONTinuous|SINGle|GATE|SADVance
[:SOURce]:RADio:ARB:TRIGger:TYPE?
```

This command sets the trigger mode (type) that controls how the waveform plays.

Triggers control the playback by telling the PSG when to transmit the modulating signal (waveform). Depending on the trigger settings for the PSG, the waveform transmission can occur once, continuously, or the PSG may start and stop the transmission repeatedly (GATE mode). For waveform sequences, you can even control when each segment plays (SADVance—segment advance mode).

A trigger signal comprises both positive and negative signal transitions (states), which are also called high and low periods. You can configure the PSG to trigger on either state of the trigger signal. It is common to have multiple triggers, also referred to as trigger occurrences or events, occur when the signal generator requires only a single trigger. In this situation, the PSG recognizes the first trigger and ignores the rest.

When you select a trigger mode, you may lose the signal (carrier plus modulating) from the RF output until you trigger the waveform. This is because the PSG sets the I and Q signals to zero volts prior to the first trigger event, which suppresses the carrier. After the first trigger event, the waveform's final I and Q levels determine whether you will see the carrier signal or not (zero = no carrier, other values = carrier visible). At the end of most files, the final I and Q points are set to a value other than zero. If desired, you can create and download an external file (see the *PSG Programming Guide*) with the initial I and Q voltages set to values other than zero. Conversely, you can set the last I and Q points to zero.

There are four parts to configuring the trigger:

- Choosing the trigger type, which controls the waveform's transmission.
- Setting the waveform's response to triggers:
  - CONTinuous, see ':TRIGger:TYPE:CONTinuous[:TYPE]' on page 229
  - SINGle, see ':RETRigger' on page 224
  - SADVance, see ":TRIGger:TYPE:SADVance[:TYPE]" on page 230
  - GATE, selecting the mode also sets the response
- Selecting the trigger source (see ':TRIGger[:SOURce]' on page 231), which determines how the PSG receives its trigger signal, internally or externally. The GATE choice requires an external trigger.
- Setting the trigger polarity when using an external source:
  - CONTinuous, SINGle, and SADVance, see ':TRIGger[:SOURce]:EXTernal:SLOPe' on page 233

— GATE, see ":TRIGger:TYPE:GATE:ACTive" on page 229

For more information on triggering, see the *PSG User's Guide*.

The following list describes the trigger type command choices:

CONTinuous	Upon triggering, the waveform repeats continuously.
SINGle	Upon triggering, the waveform segment or sequence plays once.
GATE	An external trigger signal repeatedly starts and stops the waveform's playback (transmission). The length of each transmission depends on the duty period of the trigger signal and the gate polarity selection (see ":TRIGger:TYPE:GATE:ACTive" on page 229). The waveform plays during the inactive state and stops during the active polarity selection state. The active state can be set high or low. The gate mode works only with an external trigger source.

NOTE The ARB gating behavior described above is opposite to the gating behavior for real-time custom.

SADVance	The trigger controls the segment advance within a waveform sequence. To use this choice, a waveform sequence must be the active waveform. Ensure that all segments in the sequence reside in volatile memory.
***RST**	CONT
Key Entry	**Continuous** **Single** **Gated** **Segment Advance**

:TRIGger:TYPE:CONTinuous[:TYPE]

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:ARB:TRIGger:TYPE:CONTinuous[:TYPE] FREE|TRIGger|RESet
[:SOURce]:RADio:ARB:TRIGger:TYPE:CONTinuous[:TYPE]?
```

This commands selects the waveform's response to a trigger signal while using the continuous trigger mode.

For more information on triggering and to select the continuous trigger mode, see ":TRIGger:TYPE" on page 227.

The following list describes the waveform's response to each of the command choices:

FREE Turning the ARB format on immediately triggers the waveform. The waveform repeats until you turn the format off, select another trigger, or choose another waveform file.

TRIGger The waveform waits for a trigger before play begins. When the waveform receives the trigger, it plays continuously until you turn the format off, select another trigger, or choose another waveform file.

RESet The waveform waits for a trigger before play begins. When the waveform receives the trigger, it plays continuously. Subsequent triggers reset the waveform to the beginning. For a waveform sequence, this means to the beginning of the first segment in the sequence.

***RST** FREE

Key Entry **Free Run** **Trigger & Run** **Reset & Run**

:TRIGger:TYPE:GATE:ACTive

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:ARB:TRIGger:TYPE:GATE:ACTive LOW|HIGH
[:SOURce]:RADio:ARB:TRIGger:TYPE:GATE:ACTive?
```

This command selects the active state (gate polarity) of the gate while using the gating trigger mode.

The LOW and HIGH selections correspond to the low and high states of an external trigger signal. For example, when you select HIGH, the active state occurs during the high of the trigger signal. When the active state occurs, the PSG stops the waveform playback at the last played sample point, then restarts the playback at the next sample point when the inactive state occurs. For more information on triggering and to select gating as the trigger mode, see ":TRIGger:TYPE" on page 227.

The following list describes the PSG's gating behavior for the polarity selections:

LOW The waveform playback stops when the trigger signal goes low (active state) and restarts when the trigger signal goes high (inactive state).

HIGH — The waveform playback stops when the trigger signal goes high (active state) and restarts when the trigger signal goes low (inactive state).

***RST** — HIGH

Key Entry — **Gate Active Low High**

:TRIGger:TYPE:SADVance[:TYPE]

Supported — E8267D with Option 601or 602

```
[:SOURce]:RADio:ARB:TRIGger:TYPE:SADVance[:TYPE] SINGle|CONTinuous
[:SOURce]:RADio:ARB:TRIGger:TYPE:SADVance[:TYPE]?
```

This commands selects the waveform's response to a trigger signal while using the segment advance (SADVance) trigger mode.

When the PSG receives multiple trigger occurrences when only one is required, the signal generator uses the first trigger and ignores the rest. For more information on triggering and to select segment advance as the trigger mode, see ':TRIGger:TYPE' on page 227.

The following list describes the waveform's response to each of the command choices:

SINGle — Each segment in the sequence requires a trigger to play, and a segment plays only once, ignoring a segment's repetition value (see ':SEQuence' on page 225 for repetition information). The following list describes a sequence's playback behavior with this choice:

- After receiving the first trigger, the first segment plays to completion.
- When the waveform receives a trigger after a segment completes, the sequence advances to the next segment and plays that segment to completion.
- When the waveform receives a trigger during play, the current segment plays to completion. Then the sequence advances to the next segment, and it plays to completion.
- When the waveform receives a trigger either during or after the last segment in a sequence plays, the sequence resets and the first segment plays to completion.

CONTinuous — Each segment in the sequence requires a trigger to play. After receiving a trigger, a segment plays continuously until the waveform receives another trigger. The following list describes a sequence's playback behavior with this choice:

- After receiving the first trigger, the first segment plays continuously.
- A trigger during the current segment play causes the segment to play to the

end of the segment file, then the sequence advances to the next segment, which plays continuously.

- When last segment in the sequence receives a trigger, the sequence resets and the first segment plays continuously.

***RST** CONT

Key Entry **Single** **Continuous**

:TRIGger[:SOURce]

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:ARB:TRIGger[:SOURce] KEY|EXT|BUS
[:SOURce]:RADio:ARB:TRIGger[:SOURce]?
```

This command sets the trigger source.

For more information on triggering, see ‘:TRIGger:TYPE’ on page 227. The following list describes the command choices:

KEY This choice enables manual triggering by pressing the front-panel **Trigger** hardkey.

EXT An externally applied signal triggers the waveform. This is the only choice that works with gating. The following settings affect an external trigger:

- The input connector for the trigger signal. You have a choice between the rear-panel PATTERN TRIG IN connector or the PATT TRIG IN 2 pin on the rear-panel AUXILIARY I/O connector. To make the connector selection, see ‘:TRIGger[:SOURce]:EXTernal[:SOURce]’ on page 232.

  For more information on the connectors and on connecting the cables, see the *PSG User’s Guide*.

- The trigger signal polarity:

  — gating mode, see ‘:TRIGger:TYPE:GATE:ACTive’ on page 229
  — continuous, single, and segment advance modes, see ‘:TRIGger[:SOURce]:EXTernal:SLOPe’ on page 233

- The time delay between when the PSG receives a trigger and when the waveform responds to the trigger. There are two parts to setting the delay:

  — setting the amount of delay, see ‘:TRIGger[SOURce]:EXTernal:DELay’ on page 232
  — turning the delay on, see ‘:TRIGger[:SOURce]:EXTernal:DELay:STATe’ on page 233

BUS — This choice enables triggering over the GPIB using the `*TRG` or `GET` commands, or the LAN and the AUXILIARY INTERFACE (RS-232) using the `*TRG` command.

***RST** EXT

Key Entry **Trigger Key** **Ext** **Bus**

:TRIGger[:SOURce]:EXTernal[:SOURce]

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:ARB:TRIGger[:SOURce]:EXTernal[:SOURce] EPT1|EPT2|
EPTRIGGER1|EPTRIGGER2
[:SOURce]:RADio:ARB:TRIGger[:SOURce]:EXTernal[:SOURce]?
```

This command selects which PATTERN TRIG IN connection the PSG uses to accept an externally applied trigger signal when external is the trigger source selection.

For more information on configuring an external trigger source and to select external as the trigger source, see ":TRIGger[:SOURce]" on page 231. For more information on the rear-panel connectors, see the *PSG User's Guide*.

The following list describes the command choices:

EPT1 — This choice is synonymous with EPTRIGGER1 and selects the PATTERN TRIG IN rear-panel connector.

EPT2 — This choice is synonymous with EPTRIGGER2 and selects the PATT TRIG IN 2 pin on the rear-panel AUXILIARY I/O connector.

EPTRIGGER1 — This choice is synonymous with EPT1 and selects the PATTERN TRIG IN rear-panel connector.

EPTRIGGER2 — This choice is synonymous with EPT2 and selects the PATT TRIG IN 2 pin on the rear-panel AUXILIARY I/O connector.

***RST** EPT1

Key Entry **Patt Trig In 1** **Patt Trig In 2**

:TRIGger[SOURce]:EXTernal:DELay

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:ARB:TRIGger[:SOURce]:EXTernal:DELay <val>
[:SOURce]:RADio:ARB:TRIGger[:SOURce]:EXTernal:DELay?
```

This command sets the amount of time to delay the PSG's response to an external trigger.

The delay is a path (time) delay between when the PSG receives the trigger and when it responds to the trigger. For example, configuring a trigger delay of two seconds, causes the PSG to wait two seconds after receipt of the trigger before the PSG transmits the waveform.

The delay does not occur until you turn it on (see “:TRIGger[:SOURce]:EXTernal:DELay:STATe”). You can set the delay value either before or after turning it on.

For more information on configuring an external trigger source and to select external as the trigger source, see “:TRIGger[:SOURce]” on page 231.

The unit of measurement for the variable <val> is in seconds (nsec–sec).

***RST** +1.00000000E–003

Range 1E–8 to 4E1

Key Entry **Ext Delay Time**

:TRIGger[:SOURce]:EXTernal:DELay:STATe

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:ARB:TRIGger[:SOURce]:EXTernal:DELay:STATe ON|OFF|1|0
[:SOURce]:RADio:ARB:TRIGger[:SOURce]:EXTernal:DELay:STATe?
```

This command turns the trigger delay on or off when using an external trigger source.

For setting the delay time, see “:TRIGger[SOURce]:EXTernal:DELay” , and for more information on configuring an external source, see“:TRIGger[:SOURce]” on page 231.

***RST** 0

Key Entry **Ext Delay Off On**

:TRIGger[:SOURce]:EXTernal:SLOPe

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:ARB:TRIGger[:SOURce]:EXTernal:SLOPe POSitive|NEGative
[:SOURce]:RADio:ARB:TRIGger[:SOURce]:EXTernal:SLOPe?
```

This command sets the polarity for an external trigger signal while using the continuous, single, or segment advance triggering modes. To set the polarity for gating, see ‘:TRIGger:TYPE:GATE:ACTive’ on page 229.

The POSitive and NEGative selections correspond to the high (positive) and low (negative) states of the external trigger signal. For example, when you select POSitive, the waveform responds (plays) during the high state of the trigger signal. When the PSG receives multiple trigger occurrences when only one is required, the signal generator uses the first trigger and ignores the rest.

For more information on configuring an external trigger source and to select external as the trigger source, see “:TRIGger[:SOURce]” on page 231.

***RST** NEG

Key Entry **Ext Polarity Neg Pos**

:VCO:CLOCk

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:ARB:VCO:CLOCk INTernal|EXTernal
[:SOURce]:RADio:ARB:VCO:CLOCk?
```

This command enables an internal or external VCO clock. the external VCO clock is connected to the rear-panel `BASEBAND GEN CLK IN` connector. Use the `:DACS:ALIGn` command after an external VCO clock is first applied to the `BASEBAND GEN CLK IN` connector or when the VCO signal is lost and then re-acquired.

***RST** Int

Key Entry **VCO Clock Ext Int**

:WAVeform

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:ARB:WAVeform "<file_name>"
[:SOURce]:RADio:ARB:WAVeform?
```

This command for the Dual ARB selects a waveform sequence or segment, for the dual arbitrary waveform generator to generate. The "<file_name>" can be indicated as follows:

WFM1:<file_name> This choice selects a single waveform segment.

SEQ:<file_name> This choice selects a sequence of segments. This choice selects a sequence waveform file with the <file_name> name.

Key Entry **Select Waveform**

For information on the file name syntax, see “File Name Variables” on page 12.

[:STATe]

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:ARB[:STATe] ON|OFF|1|0
[:SOURce]:RADio:ARB[:STATe]?
```

This command enables or disables the operating state of the signal generator’s dual arbitrary waveform (ARB) generator.

***RST**	0
Key Entry	**ARB Off On**

Dmodulation Subsystem–Option 601 or 602 ([:SOURce]:RADio:DMODulation:ARB)

:IQ:EXTernal:FILTer

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:DMODulation:ARB:IQ:EXTernal:FILTer 40e6|THRough
[:SOURce]:RADio:DMODulation:ARB:IQ:EXTernal:FILTer?
```

This command selects the filter or through path for I/Q signals routed to the rear-panel I and Q outputs. Selecting a filter using this command will automatically set ‘:IQ:EXTernal:FILTer:AUTO’ on page 236 to OFF(0) mode.

40e6 This choice applies a 40 MHz baseband filter.

THRough This choice bypasses filtering.

***RST** THR

Key Entry **40.000 MHz** **Through**

:IQ:EXTernal:FILTer:AUTO

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:DMODulation:ARB:IQ:EXTernal:FILTer:AUTO ON|OFF|1|0
[:SOURce]:RADio:DMODulation:ARB:IQ:EXTernal:FILTer:AUTO?
```

This command enables or disables the automatic selection of the filters for I/Q signals routed to the rear-panel I/Q outputs.

ON(1) This choice will automatically select a digital modulation filter optimized for the current signal generator settings.

OFF(0) This choice disables the auto feature which lets you select a digital modulation filter or through path. Refer to “:IQ:EXTernal:FILTer” on page 236 for selecting a filter or through path.

***RST** 1

Key Entry **I/Q Output Filter Manual Auto**

:FILTer

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:DMODulation:ARB:FILTer RNYQuist|NYQuist|GAUSsian|
RECTangle|AC4Fm|UGGaussian|"<user FIR>"
[:SOURce]:RADio:DMODulation:ARB:FILTer?
```

This command specifies the pre-modulation filter type.

`RNYQuist`	This choice selects a Root Nyquist (root raised cosine) filter. This filter is adjusted using Alpha.
`NYQuist`	This choice selects a Nyquist (raised cosine) filter. This filter is adjusted using Alpha.
`GAUSsian`	This choice selects a Gaussian Filter which is adjusted using Bbt values.
`Rectangle`	This choice selects a one symbol wide rectangular filter.
`AC4Fm`	This choice selects a predefined Association of Public Safety Communications officials (APCO) specified compatible 4-level frequency modulation (C4FM) filter.
`UGGAUSsian`	This choice selects a UN3/4 delay-compatible, GSM, 0.300 Bbt Gaussian filter. The Bbt value is not adjustable.
`"<User FIR>"`	This variable is any filter file that you have stored in memory. For information on the file name syntax, see "File Name Variables" on page 12.

***RST** RNYQuist

Key Entry **Root Nyquist** **Nyquist** **Gaussian** **Rectangle** **APCO 25 C4FM** **UN3/4 GSM Gaussian** **User FIR**

:FILTer:ALPHa

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:DMODulation:ARB:FILTer:ALPHa <val>
[:SOURce]:RADio:DMODulation:ARB:FILTer:ALPHa?
```

This command changes the Nyquist or root Nyquist filter alpha value.

The filter alpha value can be set to the minimum level (0), the maximum level (1), or in between by using fractional numeric values (0.001–0.999).

To change the current filter type, refer to ":FILTer" on page 237.

***RST** +3.50000000E–001

Range	0.000–1.000
Key Entry	**Filter Alpha**

:FILTer:BBT

Supported	E8267D with Option 601or 602

```
[:SOURce]:RADio:DMODulation:ARB:FILTer:BBT <val>
[:SOURce]:RADio:DMODulation:ARB:FILTer:BBT?
```

This command changes the bandwidth-multiplied-by-bit-time (BbT) filter parameter for a Gaussian filter. It has no effect on other types of filters.

The filter BbT value can be set to the minimum level (0), the maximum level (1), or in between by using fractional numeric values (0.001–0.999).

To change the current filter type, refer to “:FILTer” on page 237.

***RST**	+5.00000000E–001
Range	0.000–1.000
Key Entry	**Filter BbT**

:FILTer:CHANnel

Supported	E8267D with Option 601or 602

```
[:SOURce]:RADio:DMODulation:ARB:FILTer:CHANnel EVM|ACP
[:SOURce]:RADio:DMODulation:ARB:FILTer:CHANnel?
```

This command optimizes the Nyquist and root Nyquist filters to minimize error vector magnitude (EVM) or to minimize adjacent channel power (ACP). To change the current filter type, refer to “:FILTer” on page 237.

EVM	This choice provides the most ideal passband.
ACP	This choice improves stopband rejection.
***RST**	EVM
Key Entry	**Optimize FIR For EVM ACP**

:HEADer:CLEar

Supported	E8267D with Option 601or 602

```
[:SOURce]:RADio:DMODulation:ARB:HEADer:CLEar
```

This command clears the header information from the header file used by this modulation format.

Header information consists of signal generator settings and marker routings associated with the waveform file. Refer to the *PSG User's Guide* for information on header files.

For this command to function, Dmodulation must be on. To turn Dmodulation on, see '[:STATe]' on page 257.

***RST** N/A

Key Entry **Clear Header**

:HEADer:SAVE

Supported E8267D with Option 601or 602

`[:SOURce]:RADio:DMODulation:ARB:HEADer:SAVE`

This command saves the header information to the header file used by this modulation format. Header information consists of signal generator settings and marker routings associated with the waveform file. Refer to the *PSG User's Guide* for information on header files.

For this command to function, Dmodulation must be on. To turn Dmodulation on, see '[:STATe]' on page 257.

***RST** N/A

Key Entry **Save Setup To Header**

:IQ:MODulation:ATTen

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:DMODulation:ARB:IQ:MODulation:ATTen <val><unit>
[:SOURce]:RADio:DMODulation:ARB:IQ:MODulation:ATTen?
```

This command sets the attenuation level of the I/Q signals being modulated through the signal generator RF path. The variable `<val>` is expressed in units of decibels (dB).

***RST** +2.00000000E+000

Range 0–40 dB

Key Entry **Modulator Atten Manual Auto**

:IQ:MODulation:ATTen:AUTO

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:DMODulation:ARB:IQ:MODulation:ATTen:AUTO ON|OFF|1|0
[:SOURce]:RADio:DMODulation:ARB:IQ:MODulation:ATTen:AUTO?
```

This command enables or disables the I/Q attenuation auto mode.

`ON (1)`	This choice enables the attenuation auto mode which optimizes the modulator attenuation for the current conditions.
`OFF (0)`	This choice holds the attenuator at its current setting or at a selected value. Refer to ‘:IQ:MODulation:ATTen’ on page 239 for setting the attenuation value.
***RST**	1
Key Entry	**Modulator Atten Manual Auto**

:IQ:MODulation:FILTer

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:DMODulation:ARB:IQ:MODulation:FILTer 40e6|THRough
[:SOURce]:RADio:DMODulation:ARB:IQ:MODulation:FILTer?
```

This command enables you to select a filter or through path for I/Q signals modulated onto the RF carrier. Selecting a filter using this command will automatically set ‘:IQ:MODulation:FILTer:AUTO’ on page 241 to OFF(0) mode.

40E6	This choice applies a 40 MHz baseband filter to the I/Q signals.
THRough	This choice bypasses filtering.
***RST**	THR
Key Entry	**40.000 MHz** **Through**

:IQ:MODulation:FILTer:AUTO

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:DMODulation:ARB:IQ:MODulation:FILTer:AUTO ON|OFF|1|0
[:SOURce]:RADio:DMODulation:ARB:IQ:MODulation:FILTer:AUTO?
```

This command enables or disables the automatic selection of the filters for I/Q signals modulated onto the RF carrier.

`ON` (`1`) This choice automatically selects a digital modulation filter.

`OFF` (`0`) This choice disables the auto feature, enabling you to select a digital modulation filter or through path (see ":IQ:MODulation:FILTer" on page 215).

***RST** 1

Key Entry **I/Q Mod Filter Manual Auto**

:MDEStination:ALCHold

Supported E8267D with Option 601 or 602

CAUTION Incorrect ALC sampling can create a sudden unleveled condition that may create a spike in the RF output potentially damaging a DUT or connected instrument. Ensure that you set markers to let the ALC sample over an amplitude that accounts for the high power levels within the signal.

```
[:SOURce]:RADio:DMODulation:ARB:MDEStination:ALCHold NONE|M1|M2|M3|M4
[:SOURce]:RADio:DMODulation:ARB:MDEStination:ALCHold?
```

This command disables the marker ALC hold function, or it enables the marker hold function for the selected marker.

Use the ALC hold function when you have a waveform signal that uses idle periods, or when the increased dynamic range encountered with RF blanking is not desired. The ALC circuitry responds to the marker signal during the marker pulse (marker signal high), averaging the modulated signal level during this period.

The ALC hold function operates during the low periods of the marker signal. The marker polarity determines when the marker signal is high. For a positive polarity, this is during the marker points. For a negative polarity, this is when there are no marker points. To set a marker's polarity, see ':MPOLarity:MARKer1|2|3|4' on page 245. For more information on markers, see ':MARKer:[SET]' on page 218.

NOTE	Do not use the ALC hold for more than 100 ms, because it can affect the waveform's output amplitude.

The marker signal has a minimum of a two-sample delay in its response relative to the waveform signal response. To compensate for the marker signal delay, offset marker points from the waveform sample point at which you want the ALC sampling to begin.

The ALC hold setting is part of the file header information, so saving the setting to the file header saves the current marker routing for the waveform file.

NOTE	A waveform file that has unspecified settings in the file header uses the previous waveform's routing settings.

For more information on the marker ALC hold function, see the *PSG User's Guide*. To configure marker points, refer to the following sections located in the Dual ARB subsystem:

- For clearing a single marker point or a range of marker points, see ':MARKer:CLEar' on page 216.
- For clearing all marker points, see ':MARKer:CLEar:ALL' on page 217.
- For shifting marker points, see ':MARKer:ROTate' on page 217.
- For setting marker points, see ':MARKer:[SET]' on page 218.

NONE	This terminates the marker ALC hold function.
M1–M4	These are the marker choices. The ALC hold feature uses only one marker at a time.
***RST**	NONE
Key Entry	**None** **Marker 1** **Marker 2** **Marker 3** **Marker 4**
Remarks	N/A

:MDEStination:PULSe

Supported E8267D with Option 601 or 602

CAUTION	The pulse function incorporates ALC hold. Incorrect ALC sampling can create a sudden unleveled condition that may create a spike in the RF output potentially damaging a DUT or connected instrument. Ensure that you set markers to let the ALC sample over an amplitude that accounts for the high power levels within the signal.

```
[:SOURce]:RADio:DMODulation:ARB:MDEStination:PULSe NONE|M1|M2|M3|M4
[:SOURce]:RADio:DMODulation:ARB:MDEStination:PULSe?
```

This command disables the marker RF blanking/pulse function, or it enables the marker RF blanking/pulse function for the selected marker.

This function automatically incorporates the ALC hold function, so there is no need to select both functions for the same marker.

NOTE	Do not use ALC hold for more than 100 ms, because it can affect the waveform's output amplitude.

The signal generator blanks the RF output when the marker signal goes low. The marker polarity determines when the marker signal is low. For a positive polarity, this is during the marker points. For a negative polarity, this is when there are no marker points. To set a marker's polarity, see ':MPOLarity:MARKer1|2|3|4' on page 245. For more information on markers, see ':MARKer:[SET]' on page 218.

NOTE	Set marker points prior to using this function. Enabling this function without setting marker points may create a continuous low or high marker signal, depending on the marker polarity. This creates the condition where there is either no RF output or a continuous RF output.

To configure marker points, refer to the following sections located in the Dual ARB subsystem:

- For clearing a single marker point or a range of marker points, see':MARKer:CLEar' on page 216.
- For clearing all marker points, see ':MARKer:CLEar:ALL' on page 217.
- For shifting marker points, see ':MARKer:ROTate' on page 217.
- For setting marker points, see ':MARKer:[SET]' on page 218.

The marker signal has a minimum of a two-sample delay in its response relative to the waveform signal response. To compensate for the marker signal delay, offset marker points from the waveform sample point at which you want the RF blanking to begin.

The RF blanking setting is part of the file header information, so saving the setting to the file header saves the current marker routing for the waveform file.

NOTE	A waveform file that has unspecified settings in the file header uses the previous waveform's routing settings. This could create the situation where there is no RF output signal, because the previous waveform used RF blanking

For more information on the marker RF blanking function, see the *PSG User's Guide*.

NONE	This terminates the marker RF blanking/pulse function.
M1–M4	These are the marker choices. The RF blanking/pulse feature uses only one marker at a time.
***RST**	NONE
Key Entry	**None** **Marker 1** **Marker 2** **Marker 3** **Marker 4**

:MODulation:FSK[:DEViation]

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:DMODulation:ARB:MODulation:FSK[:DEViation] <val><units>
[:SOURce]:RADio:DMODulation:ARB:MODulation:FSK[:DEViation]?
```

This command sets the symmetric FSK frequency deviation value.

The variable `<val>` is a numeric expression with a maximum range equal to the current symbol rate value multiplied by ten, limited to 20 MHz. The variable `<units>` is expressed in units of Hertz.

To change the modulation type, refer to the command ":MODulation[:TYPE]" on page 244. Refer to the command ":SRATe" on page 250 for a list of the minimum and maximum symbol rate values.

For more information on setting an asymmetric FSK deviation value, refer to the *PSG User's Guide*.

***RST**	+4.00000000E+002
Range	0–2E7
Key Entry	**Freq Dev**

:MODulation[:TYPE]

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:DMODulation:ARB:MODulation[:TYPE] BPSK|QPSK|IS95QPSK|
GRAYQPSK|OQPSK|IS95OQPSK|P4DQPSK|PSK8|PSK16|D8PSK|EDGE|MSK|FSK2|FSK4|
FSK8|FSK16|C4FM|QAM4|QAM16|QAM32|QAM64|QAM128|QAM256
[:SOURce]:RADio:DMODulation:ARB:MODulation[:TYPE]?
```

This command sets the modulation type for the digital modulation personality.

***RST**	P4DQPSK
Key Entry	**BPSK** **QPSK** **IS-95 QPSK** **Gray Coded QPSK** **OQPSK**
	IS-95 OQPSK **π/4 DQPSK** **8PSK** **16PSK** **D8PSK** **EDGE** **MSK**
	2-Lvl FSK **4-Lvl FSK** **8-Lvl FSK** **16-Lvl FSK** **C4FM** **4QAM** **16QAM**

32QAM **64QAM** **128QAM** **256QAM** **User I/Q** **User FSK**

:MPOLarity:MARKer1|2|3|4

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:DMODulation:ARB:MPOLarity:MARKer1|2|3|4 NEGative|
POSitive
[:SOURce]:RADio:DMODulation:ARB:MPOLarity:MARKer1|2|3|4?
```

This command sets the polarity for the selected marker.

For a positive marker polarity, the marker signal is high during the marker points. For a negative marker polarity, the marker signal is high during the period of no marker points. To configure marker points, refer to the following sections located in the Dual ARB subsystem:

- For clearing a single marker point or a range of marker points, see‘:MARKer:CLEar’ on page 216.
- For clearing all marker points, see ‘:MARKer:CLEar:ALL’ on page 217.
- For shifting marker points, see ‘:MARKer:ROTate’ on page 217.
- For information on markers and setting marker points, see ‘:MARKer:[SET]’ on page 218.

***RST** POS

Key Entry **Marker 1 Polarity Neg Pos** **Marker 2 Polarity Neg Pos** **Marker 3 Polarity Neg Pos** **Marker 4 Polarity Neg Pos**

:REFerence:EXTernal:FREQuency

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:DMODulation:ARB:REFerence:EXTernal:FREQuency <val>
[:SOURce]:RADio:DMODulation:ARB:REFerence:EXTernal:FREQuency?
```

This command sets or retrieves the reference frequency value of an externally applied reference to the signal generator. The variable `<val>` is expressed in Hertz (Hz–MHz).

The value specified by this command is effective only when you are using an external ARB reference applied to the BASEBAND GEN REF IN rear-panel connector.

To specify external as the ARB reference source type, refer to “:REFerence[:SOURce]” on page 246.

***RST** +1.00000000E+007

Range 2.5E5–1E8

Key Entry **Reference Freq**

:REFerence[:SOURce]

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:DMODulation:ARB:REFerence[:SOURce] INTernal|EXTernal
[:SOURce]:RADio:DMODulation:ARB:REFerence[:SOURce]?
```

This command selects either an internal or external reference for the waveform clock.

If the EXTernal choice is selected, the external frequency value *must* be entered and the signal must be applied to the BASEBAND GEN REF IN rear-panel connector.

Refer to ":REFerence:EXTernal:FREQuency" on page 245 to enter the external reference frequency.

***RST** INT

Key Entry **ARB Reference Ext Int**

:RETRigger

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:DMODulation:ARB:RETRigger ON|OFF|IMMediate
[:SOURce]:RADio:DMODulation:ARB:RETRigger?
```

This commands selects the waveform's response to a trigger signal while using the single trigger mode.

When the PSG receives multiple trigger occurrences when only one is required, the signal generator uses the first trigger and ignores the rest. For more information on triggering and to select the single trigger mode, see ':TRIGger:TYPE' on page 251.

The following list describes the waveform's response to each of the command choices:

ON — The waveform waits for a trigger before play begins and accepts a subsequent trigger during playback. If there is a subsequent trigger during playback, the waveform completes its current playback and then plays one more time. If there is no subsequent trigger, the waveform plays once and stops until it receives another trigger.

OFF — The waveform waits for a trigger before play begins and ignores triggers during playback. To restart the waveform, you must send a trigger after the playback completes.

IMMediate — The waveform waits for a trigger before play begins and accepts a subsequent trigger during playback. Upon receipt of the subsequent trigger, the waveform immediately resets and begins playing from the beginning of the file. For a waveform sequence, this means to the beginning of the first segment in the sequence.

***RST** ON

Key Entry **On** **Off** **Immediate**

:SCLock:RATE

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:DMODulation:ARB:SCLock:RATE <1.0 kHz – 1.0e8 kHz {1.0e8
kHz}>
[:SOURce]:RADio:DMODulation:ARB:SCLock:RATE?
```

This command sets the sample clock rate.

The modulation format should be active before executing this command. If this command is executed before the modulation format is active, the entered value will be overridden by a calculated factory default value. Refer to ‘[:STATe]’ on page 257 to activate the modulation format.

***RST** +1.00000000E+008

Range 1–1E8

Key Entry **ARB Sample Clock**

:SETup

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:DMODulation:ARB:SETup GSM|NADC|PDC|PHS|DECT|AC4Fm|
ACQPsk|CDPD|PWT|EDGE|TETRa|MCARrier|"<file name>"
[:SOURce]:RADio:DMODulation:ARB:SETup?
```

This command selects the digital modulation format type.

For information on the file name syntax, see “File Name Variables” on page 12

***RST** NADC

Key Entry **GSM** **NADC** **PDC** **PHS** **DECT** **APCO 25 w/C4FM** **APCO w/CQPSK**
CDPD **PWT** **EDGE** **TETRA** **Multicarrier Off On** **Select File**

:SETup:MCARrier

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:DMODulation:ARB:SETup:MCARrier (GSM|NADC|PDC|PHS|DECT|
AC4Fm|ACQPsk|CDPD|PWT|EDGE|TETRa,<num carriers>,<freq spacing>)|
"<file name>"
[:SOURce]:RADio:DMODulation:ARB:SETup:MCARrier?
```

This command builds a table with the specified number of carriers and frequency spacing or retrieves the setup stored in the specified user file. The carrier type, number of carriers, and frequency spacing value are returned when a query is initiated. The output format is as follows:

```
<carrier type>,<num carriers>,<freq spacing>
```

If a specific file is loaded and then queried, only the file name is returned. For information on the file name syntax, see “File Name Variables” on page 12. To store a multicarrier setup refer to ‘SETup:MCARrier:STORe’ on page 248.

The variable <freq spacing> is expressed in units of Hertz (kHz–MHz).

***RST** *Carrier*: NADC
<num carriers>:2
<freq spacing>: +1.0000000000000E+06

Range *<num carriers>*:2–100vb
<freq spacing>: 2 ÷ (<num carriers> – 1) × 80 MHz

Key Entry **GSM** **NADC** **PDC** **PHS** **DECT** **APCO 25 w/C4FM** **APCO w/CQPSK** **CDPD** **PWT** **EDGE** **TETRA** **# of Carriers** **Freq Spacing** **Custom Digital Mod State**

:SETup:MCARrier:PHASe

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:DMODulation:ARB:SETup:MCARrier:PHASe FIXed|RANDom
[:SOURce]:RADio:DMODulation:ARB:SETup:MCARrier:PHASe?
```

This command sets the phase difference between carriers for multicarrier digital modulation.

FIXed This choice sets the phase of all carriers to 0.

RANDom This choice sets random phase values for all of the carriers.

***RST** FIX

Key Entry **Carrier Phases Fixed Random**

SETup:MCARrier:STORe

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:DMODulation:ARB:SETup:MCARrier:STORe "<file name>"
```

This command stores the current multicarrier setup information.

The stored file contains information that includes the digital modulation format, number of carriers, frequency spacing, and power settings for the multicarrier setup.

The setting enabled by this command is not affected by signal generator power-on, preset, or *RST. For information on the file name syntax, see "File Name Variables" on page 12.

***RST** N/A

Key Entry **Load / Store**

:SETup:MCARrier:TABLe

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:DMODulation:ARB:SETup:MCARrier:TABLe INIT|APPend|
<carrier_num>,GSM|NADC|PDC|PHS|DECT|AC4Fm|ACQPsk|CDPD|PWT|EDGE|TETRa|
"<file name>",<freq_offset>,<power>
[:SOURce]:RADio:DMODulation:ARB:SETup:MCARrier:TABLe? <carrier_num>
```

This command modifies the parameters of one of the available multicarrier digital modulation formats.

The variable <freq_offset> is expressed in Hertz (kHz–MHz).

The variable <power> is expressed in units of decibels (dB).

INIT — This choice clears the current information and creates a new one-row table, allowing for further definition using additional parameters.

APPend — This choice adds rows to an existing table.

<carrier_num> — This variable specifies the number of the carriers in the multicarrier table that will be modified.

The value of the variable <carrier_num> must be specified prior to selecting the digital modulation format.

For information on the file name syntax, see "File Name Variables" on page 12. To store a multicarrier setup refer to 'SETup:MCARrier:STORe' on page 248.

When a query is initiated, carrier type, frequency offset, and power level are returned in the following format: `<carrier type>,<freq_offset>,<power>`

***RST**

carrier type:	NADC
<*freq_offset*>:	5.00000000E+004
<*power*>:	+0.00000000E+000

Range

<*freq_offset*>: –1E5–1E6
<*power*>: –40–0

Key Entry	**Initialize Table** **Insert Row** **GSM** **NADC** **PDC** **PHS** **DECT** **APCO 25 w/C4FM** **APCO w/CQPSK** **CDPD** **PWT** **EDGE** **TETRA** **Custom Digital Mod State**

:SETup:MCARrier:TABLe:NCARriers

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:DMODulation:ARB:SETup:MCARrier:TABLe:NCARriers?
```

This query returns the number of carriers in the current multicarrier setup.

***RST**	+2
Range	1–100
Key Entry	**# of Carriers**

:SETup:STORe

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:DMODulation:ARB:SETup:STORe "<file name>"
```

This command stores the current custom digital modulation state.

The saved file contains information that includes the modulation type, filter and symbol rate for the custom modulation setup.

For information on the file name syntax, see "File Name Variables" on page 12.

***RST**	N/A
Range	N/A
Key Entry	**Store Custom Dig Mod State**

:SRATe

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:DMODulation:ARB:SRATe <val>
[:SOURce]:RADio:DMODulation:ARB:SRATe?
```

This command sets the transmission symbol rate. The variable `<val>` is expressed in units of bits per second (bps–Mbps) and the maximum range value is dependent upon the source of data (internal or external), the modulation type, and filter.

When user-defined filters are selected using the command in section ":FILTer" on page 237, the upper bit rate will be restricted using the following criteria:

- FIR filter length > 32 symbols: upper limit is 12.5 Msps
- FIR filter length > 16 symbols: upper limit is 25 Msps

When internal FIR filters are used, these limit restrictions always apply. For higher symbol rates, the FIR filter length will be truncated as follows:

- Above 12.5 Msps, the FIR length is truncated to 32 symbols
- Above 25 Msps, the FIR length is truncated to 16 symbols

  This impacts the relative timing of the modulated data, as well as the actual filter response.

To change the modulation type, refer to ":MODulation[:TYPE]" on page 244.

***RST**	+2.43000000E+004
Range	1 ksps–50 Msps
Key Entry	**Symbol Rate**

:TRIGger:TYPE

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:DMODulation:ARB:TRIGger:TYPE CONTinuous|SINGle|GATE
[:SOURce]:RADio:DMODulation:ARB:TRIGger:TYPE?
```

This command sets the trigger mode (type) that controls the waveform's playback.

Triggers control the playback by telling the PSG when to play the modulating signal (waveform). Depending on the trigger settings for the PSG, the waveform playback can occur once, continuously, or the PSG may start and stop playing the waveform repeatedly (GATE mode).

A trigger signal comprises both positive and negative signal transitions (states), which are also called high and low periods. You can configure the PSG to trigger on either state of the trigger signal. It is common to have multiple triggers, also referred to as trigger occurrences or events, occur when the signal generator requires only a single trigger. In this situation, the PSG recognizes the first trigger and ignores the rest.

When you select a trigger mode, you may lose the signal (carrier plus modulating) from the RF output until you trigger the waveform. This is because the PSG sets the I and Q signals to zero volts prior to the first trigger event, which suppresses the carrier. After the first trigger event, the waveform's final I and Q levels determine whether you will see the carrier signal or not (zero = no carrier, other values = carrier visible). At the end of most files, the final I and Q points are set to a value other than zero.

There are four parts to configuring the trigger:

- Choosing the trigger type, which controls the waveform’s transmission.
- Setting the waveform’s response to triggers:
  - — CONTinuous, see ‘:TRIGger:TYPE:CONTinuous[:TYPE]’ on page 252
  - — SINGle, see ‘:RETRigger’ on page 246
  - — GATE, selecting the mode also sets the response
- Selecting the trigger source (see ‘:TRIGger[:SOURce]’ on page 254), which determines how the PSG receives its trigger signal, internally or externally. The GATE choice requires an external trigger.
- Setting the trigger polarity when using an external source:
  - — CONTinuous and SINGle see ‘:TRIGger[:SOURce]:EXTernal:SLOPe’ on page 256
  - — GATE, see “:TRIGger:TYPE:GATE:ACTive” on page 253

For more information on triggering, see the *PSG User’s Guide*.

The following list describes the trigger type command choices:

CONTinuous	Upon triggering, the waveform repeats continuously.
SINGle	Upon triggering, the waveform segment or sequence plays once.
GATE	An external trigger signal repeatedly starts and stops the waveform’s playback (transmission). The time duration for playback depends on the duty period of the trigger signal and the gate polarity selection (see “:TRIGger:TYPE:GATE:ACTive” on page 253). The waveform plays during the inactive state and stops during the active polarity selection state. The active state can be set high or low. The gate mode works only with an external trigger source.

NOTE The ARB gating behavior described above is opposite to the gating behavior for real-time custom mode.

***RST** CONT

Key Entry **Continuous** **Single** **Gated**

:TRIGger:TYPE:CONTinuous[:TYPE]

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:DMODulation:ARB:TRIGger:TYPE:CONTinuous[:TYPE] FREE|
TRIGger|RESet
[:SOURce]:RADio:DMODulation:ARB:TRIGger:TYPE:CONTinuous[:TYPE]?
```

This commands selects the waveform’s response to a trigger signal while using the continuous trigger mode.

For more information on triggering and to select the continuous trigger mode, see “:TRIGger:TYPE” on page 251.

The following list describes the waveform’s response to each of the command choices:

FREE	Turning the ARB format on immediately triggers the waveform. The waveform repeats until you turn the format off, select another trigger, or choose another waveform file.
TRIGger	The waveform waits for a trigger before play begins. When the waveform receives the trigger, it plays continuously until you turn the format off, select another trigger, or choose another waveform file.
RESet	The waveform waits for a trigger before play begins. When the waveform receives the trigger, it plays continuously. Subsequent triggers reset the waveform to the beginning. For a waveform sequence, this means to the beginning of the first segment in the sequence.
***RST**	FREE
Key Entry	**Free Run** **Trigger & Run** **Reset & Run**

:TRIGger:TYPE:GATE:ACTive

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:DMODulation:ARB:TRIGger:TYPE:GATE:ACTive LOW|HIGH
[:SOURce]:RADio:DMODulation:ARB:TRIGger:TYPE:GATE:ACTive?
```

This command selects the active state (gate polarity) of the gate while using the gating trigger mode.

The LOW and HIGH selections correspond to the low and high states of an external trigger signal. For example, when you select HIGH, the active state occurs during the high of the trigger signal. When the active state occurs, the PSG stops the waveform playback at the last played sample point, then restarts the playback at the next sample point when the inactive state occurs. For more information on triggering and to select gating as the trigger mode, see “:TRIGger:TYPE” on page 251.

The following list describes the PSG’s gating behavior for the polarity selections:

LOW	The waveform playback stops when the trigger signal goes low (active state) and restarts when the trigger signal goes high (inactive state).
HIGH	The waveform playback stops when the trigger signal goes high (active state) and restarts when the trigger signal goes low (inactive state).
***RST**	HIGH
Key Entry	**Gate Active Low High**

:TRIGger[:SOURce]

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:DMODulation:ARB:TRIGger[:SOURce] KEY|EXT|BUS
[:SOURce]:RADio:DMODulation:ARB:TRIGger[:SOURce]?
```

This command sets the trigger source.

For more information on triggering, see “:TRIGger:TYPE” on page 251. The following list describes the command choices:

KEY This choice enables manual triggering by pressing the front-panel **Trigger** hardkey.

EXT An externally applied signal triggers the waveform. This is the only choice that works with gating. The following conditions affect an external trigger:

- The input connector selected for the trigger signal. You have a choice between the rear-panel PATTERN TRIG IN connector or the PATT TRIG IN 2 pin on the rear-panel AUXILIARY I/O connector. To make the connector selection, see ‘:TRIGger[:SOURce]:EXTernal[:SOURce]’ on page 255.

  For more information on the connectors and on connecting the cables, see the *PSG User’s Guide*.

- The trigger signal polarity:
  - gating mode, see ‘:TRIGger:TYPE:GATE:ACTive’ on page 253
  - continuous and single modes, see ‘:TRIGger[:SOURce]:EXTernal:SLOPe’ on page 256

- The time delay between when the PSG receives a trigger and when the waveform responds to the trigger. There are two parts to setting the delay:
  - setting the amount of delay, see ‘:TRIGger[SOURce]:EXTernal:DELay’ on page 255
  - turning the delay on, see ‘:TRIGger[:SOURce]:EXTernal:DELay:STATe’ on page 256

BUS This choice enables triggering over the GPIB or LAN using the `*TRG` or `GET` commands or the AUXILIARY INTERFACE (RS-232) using the `*TRG` command.

***RST** EXT

Key Entry **Trigger Key** **Ext** **Bus**

:TRIGger[:SOURce]:EXTernal[:SOURce]

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:DMODulation:ARB:TRIGger[:SOURce]:EXTernal[:SOURce] EPT1|
EPT2|EPTRIGGER1|EPTRIGGER2
[:SOURce]:RADio:DMODulation:ARB:TRIGger[:SOURce]:EXTernal[:SOURce]?
```

This command selects which PATTERN TRIG IN connection the PSG uses to accept an externally applied trigger signal when external is the trigger source selection.

For more information on configuring an external trigger source and to select external as the trigger source, see ":TRIGger[:SOURce]" on page 254. For more information on the rear-panel connectors, see the *PSG User's Guide*.

The following list describes the command choices:

EPT1 This choice is synonymous with EPTRIGGER1 and selects the PATTERN TRIG IN rear-panel connector.

EPT2 This choice is synonymous with EPTRIGGER2 and selects the PATT TRIG IN 2 pin on the rear-panel AUXILIARY I/O connector.

EPTRIGGER1 This choice is synonymous with EPT1 and selects the PATTERN TRIG IN rear-panel connector.

EPTRIGGER2 This choice is synonymous with EPT2 and selects the PATT TRIG IN 2 pin on the rear-panel AUXILIARY I/O connector.

***RST** EPT1

Key Entry **Patt Trig In 1** **Patt Trig In 2**

:TRIGger[SOURce]:EXTernal:DELay

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:DMODulation:ARB:TRIGger[:SOURce]:EXTernal:DELay <val>
[:SOURce]:RADio:DMODulation:ARB:TRIGger[:SOURce]:EXTernal:DELay?
```

This command sets the amount of time to delay the PSG's response to an external trigger.

The delay is a path (time) delay between when the PSG receives the trigger and when it responds to the trigger. For example, configuring a trigger delay of two seconds, causes the PSG to wait two seconds after receipt of the trigger before the PSG plays the waveform.

The delay does not occur until you turn it on (see ":TRIGger[:SOURce]:EXTernal:DELay:STATe"). You can set the delay value either before or after turning it on.

For more information on configuring an external trigger source and to select external as the trigger source, see “:TRIGger[:SOURce]” on page 254.

The unit of measurement for the variable <val> is in seconds (nsec–sec).

***RST**	+1.00000000E–003
Range	1E–8 to 4E1
Key Entry	**Ext Delay Time**

:TRIGger[:SOURce]:EXTernal:DELay:STATe

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:DMODulation:ARB:TRIGger[:SOURce]:EXTernal:DELay:
STATe ON|OFF|1|0
[:SOURce]:RADio:DMODulation:ARB:TRIGger[:SOURce]:EXTernal:DELay:STATe?
```

This command turns the trigger delay on or off when using an external trigger source.

For setting the delay time, see “:TRIGger[SOURce]:EXTernal:DELay” , and for more information on configuring an external source, see“:TRIGger[:SOURce]” on page 254.

***RST**	0
Key Entry	**Ext Delay Off On**

:TRIGger[:SOURce]:EXTernal:SLOPe

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:DMODulation:ARB:TRIGger[:SOURce]:EXTernal:
SLOPe POSitive|NEGative
[:SOURce]:RADio:DMODulation:ARB:TRIGger[:SOURce]:EXTernal:SLOPe?
```

This command sets the polarity for an external trigger signal while using the continuous, single triggering mode. To set the polarity for gating, see ‘:TRIGger:TYPE:GATE:ACTive’ on page 253.

The POSitive and NEGative selections correspond to the high (positive) and low (negative) states of the external trigger signal. For example, when you select POSitive, the waveform responds (plays) during the high state of the trigger signal. When the PSG receives multiple trigger occurrences when only one is required, the signal generator uses the first trigger and ignores the rest.

For more information on configuring an external trigger source and to select external as the trigger source, see ‘:TRIGger[:SOURce]’ on page 254.

***RST**	NEG
Key Entry	**Ext Polarity Neg Pos**

[:STATe]

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:DMODulation:ARB[:STATe] ON|OFF|1|0
[:SOURce]:RADio:DMODulation:ARB[:STATe]?
```

This command enables or disables the digital modulation capability.

`ON` (`1`) This choice sets up the internal hardware to generate the currently selected digital modulation format. When ON is selected, the I/Q state is activated and the I/Q source is set to internal.

`OFF` (`0`) This choice disables the digital modulation capability.

***RST** 0

Key Entry **Digital Modulation Off On**

Multitone Subsystem–Option 601 or 602 ([:SOURce]:RADio:MTONe:ARB)

Creating a Multitone Waveform

Use the following steps to create a multitone waveform:

1. Initialize the phase for the multitone waveform (“:SETup:TABLe:PHASe:INITialize” on page 268).
2. Assign the frequency spacing between the tones (“:SETup:TABLe:FSPacing” on page 267).
3. Define the number of tones within the waveform (“:SETup:TABLe:NTONes” on page 267).
4. Modify the power level, phase, and state of any individual tones (“:ROW” on page 269).

:HEADer:CLEar

Supported E8267D with Option 601or 602

`[:SOURce]:RADio:MTONe:ARB:HEADer:CLEar`

This command clears the header information from the header file used by this modulation format. Header information consists of signal generator settings and marker routings associated with the waveform file. Refer to the *PSG User’s Guide* for information on header files.

For this command to function, multitone must be on. To turn multitone on, see ‘[:STATe]’ on page 269.

***RST** N/A

Key Entry **Clear Header**

:HEADer:SAVE

Supported E8267D with Option 601or 602

`[:SOURce]:RADio:MTONe:ARB:HEADer:SAVE`

This command saves the header information to the header file used by this modulation format. Header information consists of signal generator settings and marker routings associated with the waveform file. Refer to the *PSG User’s Guide* for information on header files.

For this command to function, multitone must be on. To turn multitone on, see ‘[:STATe]’ on page 269.

***RST** N/A

Key Entry **Save Setup To Header**

:IQ:EXTernal:FILTer

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:MTONe:ARB:IQ:EXTernal:FILTer 40e6|THRough
[:SOURce]:RADio:MTONe:ARB:IQ:EXTernal:FILTer?
```

This command selects the filter or through path for I/Q signals routed to the rear-panel I and Q outputs. Selecting a filter using this command will automatically set ‘:IQ:EXTernal:FILTer:AUTO’ on page 259 to OFF(0) mode.

40e6 This choice applies a 40 MHz baseband filter.

THRough This choice bypasses filtering.

***RST** THR

Key Entry **40.000 MHz** **Through**

:IQ:EXTernal:FILTer:AUTO

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:MTONe:ARB:IQ:EXTernal:FILTer:AUTO ON|OFF|1|0
[:SOURce]:RADio:MTONe:ARB:IQ:EXTernal:FILTer:AUTO?
```

This command enables or disables the automatic selection of the filters for I/Q signals routed to the rear-panel I/Q outputs.

`ON` (`1`) This choice will automatically select a digital modulation filter optimized for the current signal generator settings.

`OFF` (`0`) This choice disables the auto feature which lets you select a digital modulation filter or through path. Refer to “:IQ:EXTernal:FILTer” on page 259 for selecting a filter or through path.

***RST** 1

Key Entry **I/Q Output Filter Manual Auto**

:IQ:MODulation:ATTen

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:MTONe:ARB:IQ:MODulation:ATTen <val><unit>
[:SOURce]:RADio:MTONe:ARB:IQ:MODulation:ATTen?
```

This command attenuates the I/Q signals that modulate the signal generator’s RF output.

The variable `<val>` is expressed as a number. The variable `<unit>` is expressed in decibels (dB).

***RST** +2.00000000E+000

Range 0–40

Key Entry **Modulator Atten Manual Auto**

:IQ:MODulation:ATTen:AUTO

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:MTONe:ARB:IQ:MODulation:ATTen:AUTO ON|OFF|1|0
[:SOURce]:RADio:MTONe:ARB:IQ:MODulation:ATTen:AUTO?
```

This command enables or disables the I/Q attenuation auto mode.

`ON` (1) This choice enables the attenuation auto mode which optimizes the modulator attenuation for the current conditions.

`OFF` (0) This choice holds the attenuator at its current setting or at a selected value. Refer to ‘:IQ:MODulation:ATTen’ on page 259 for setting the attenuation value.

***RST** 1

Key Entry **Modulator Atten Manual Auto**

:IQ:MODulation:FILTer

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:MTONe:ARB:IQ:MODulation:FILTer 40e6|THRough
[:SOURce]:RADio:MTONe:ARB:IQ:MODulation:FILTer?
```

This command enables you to select a filter or through path for I/Q signals modulated onto the RF carrier. Selecting a filter using this command will automatically set ‘:IQ:MODulation:FILTer:AUTO’ on page 260 to OFF(0) mode.

40E6 This choice applies a 40 MHz baseband filter to the I/Q signals.

THRough This choice bypasses filtering.

***RST** THR

Key Entry **40.000 MHz** **Through**

:IQ:MODulation:FILTer:AUTO

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:MTONe:ARB:IQ:MODulation:FILTer:AUTO ON|OFF|1|0
[:SOURce]:RADio:MTONe:ARB:IQ:MODulation:FILTer:AUTO?
```

This enables or disables the automatic selection of filters for I/Q signals modulating the signal generator's RF carrier.

`ON (1)`	Automatically selects a digital modulation filter.
`OFF (0)`	Enables you to select a digital modulation filter or through path. Refer to ":IQ:MODulation:FILTer" on page 215 for selecting a filter or through path.
***RST**	1
Key Entry	**I/Q Mod Filter Manual Auto**

:MDEStination:ALCHold

Supported E8267D with Option 601 or 602

CAUTION Incorrect ALC sampling can create a sudden unleveled condition that may create a spike in the RF output potentially damaging a DUT or connected instrument. Ensure that you set markers to let the ALC sample over an amplitude that accounts for the high power levels within the signal.

```
[:SOURce]:RADio:MTONe:ARB:MDEStination:ALCHold NONE|M1|M2|M3|M4
[:SOURce]:RADio:MTONe:ARB:MDEStination:ALCHold?
```

This command disables the marker ALC hold function, or it enables the marker hold function for the selected marker.

Use the ALC hold function when you have a waveform signal that incorporates idle periods, or when the increased dynamic range encountered with RF blanking is not desired. The ALC circuitry responds to the marker signal during the marker pulse (marker signal high), averaging the modulated signal level during this period.

The ALC hold function operates during the low periods of the marker signal. The marker polarity determines when the marker signal is high. For a positive polarity, this is during the marker points. For a negative polarity, this is when there are no marker points. To set a marker's polarity, see ':MPOLarity:MARKer1|2|3|4' on page 264. For more information on markers, see ':MARKer:[SET]' on page 218.

NOTE Do not use the ALC hold for more than 100 ms, because it can affect the waveform's output amplitude.

The marker signal has a minimum of a two-sample delay in its response relative to the waveform signal response. To compensate for the marker signal delay, offset marker points from the waveform sample point at which you want the ALC sampling to begin.

The ALC hold setting is part of the file header information, so saving the setting to the file header saves the current marker routing for the waveform file.

NOTE	A waveform file that has unspecified settings in the file header uses the previous waveform's routing settings.

For more information on the marker ALC hold function, see the *PSG User's Guide*. To configure marker points, refer to the following sections located in the Dual ARB subsystem:

- For clearing a single marker point or a range of marker points, see ':MARKer:CLEar' on page 216.
- For clearing all marker points, see ':MARKer:CLEar:ALL' on page 217.
- For shifting marker points, see ':MARKer:ROTate' on page 217.
- For setting marker points, see ':MARKer:[SET]' on page 218.

NONE	This terminates the marker ALC hold function.
M1–M4	These are the marker choices. The ALC hold feature uses only one marker at a time.
***RST**	NONE
Key Entry	**None** **Marker 1** **Marker 2** **Marker 3** **Marker 4**

:MDEStination:PULSe

Supported	E8267D with Option 601 or 602

CAUTION	The pulse function incorporates ALC hold. Incorrect ALC sampling can create a sudden unleveled condition that may create a spike in the RF output potentially damaging a DUT or connected instrument. Ensure that you set markers to let the ALC sample over an amplitude that accounts for the high power levels within the signal.

```
[:SOURce]:RADio:MTONe:ARB:MDEStination:PULSe NONE|M1|M2|M3|M4
[:SOURce]:RADio:MTONe:ARB:MDEStination:PULSe?
```

This command disables the marker RF blanking/pulse function, or it enables the marker RF blanking/pulse function for the selected marker.

This function automatically incorporates the ALC hold function, so there is no need to select both functions for the same marker.

NOTE Do not use ALC hold for more than 100 ms, because it can affect the waveform's output amplitude.

The signal generator blanks the RF output when the marker signal goes low. The marker polarity determines when the marker signal is low. For a positive polarity, this is during the marker points. For a negative polarity, this is when there are no marker points. To set a marker's polarity, see ':MPOLarity:MARKer1|2|3|4' on page 264. For more information on markers, see ':MARKer:[SET]' on page 218.

NOTE Set marker points prior to using this function. Enabling this function without setting marker points may create a continuous low or high marker signal, depending on the marker polarity. This creates the condition where there is either no RF output or a continuous RF output.

To configure marker points, refer to the following sections located in the Dual ARB subsystem:

- For clearing a single marker point or a range of marker points, see':MARKer:CLEar' on page 216.
- For clearing all marker points, see ':MARKer:CLEar:ALL' on page 217.
- For shifting marker points, see ':MARKer:ROTate' on page 217.
- For setting marker points, see ':MARKer:[SET]' on page 218.

The marker signal has a minimum of a two-sample delay in its response relative to the waveform signal response. To compensate for the marker signal delay, offset marker points from the waveform sample point at which you want the RF blanking to begin.

The RF blanking setting is part of the file header information, so saving the setting to the file header saves the current marker routing for the waveform file.

NOTE A waveform file that has unspecified settings in the file header uses the previous waveform's routing settings. This could create the situation where there is no RF output signal, because the previous waveform used RF blanking

For more information on the marker RF blanking function, see the *PSG User's Guide*.

NONE	This terminates the marker RF blanking/pulse function.
M1–M4	These are the marker choices. The RF blanking/pulse feature uses only one marker at a time.

***RST** NONE

Key Entry **None** **Marker 1** **Marker 2** **Marker 3** **Marker 4**

:MPOLarity:MARKer1|2|3|4

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:MTONe:ARB:MPOLarity:MARKer1|2|3|4 NEGative|POSitive
[:SOURce]:RADio:MTONe:ARB:MPOLarity:MARKer1|2|3|4?
```

This command sets the polarity for the selected marker.

For a positive marker polarity, the marker signal is high during the marker points. For a negative marker polarity, the marker signal is high during the period of no marker points. To configure marker points, refer to the following sections located in the Dual ARB subsystem:

- For clearing a single marker point or a range of marker points, see‘:MARKer:CLEar’ on page 216.
- For clearing all marker points, see ‘:MARKer:CLEar:ALL’ on page 217.
- For shifting marker points, see ‘:MARKer:ROTate’ on page 217.
- For information on markers and setting marker points, see ‘:MARKer:[SET]’ on page 218.

***RST** POS

Key Entry **Marker 1 Polarity Neg Pos** **Marker 2 Polarity Neg Pos** **Marker 3 Polarity Neg Pos** **Marker 4 Polarity Neg Pos**

:REFerence:EXTernal:FREQuency

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:MTONe:ARB:REFerence:EXTernal:FREQuency <val>
[:SOURce]:RADio:MTONe:ARB:REFerence:EXTernal:FREQuency?
```

This command allows you to enter the frequency of the applied external reference. The variable `<val>` is expressed in Hertz (Hz–MHz). The value specified by this command is effective only when you are using an external ARB reference applied to the BASEBAND GEN REF IN rear-panel connector. To specify external as the ARB reference source type, refer to “:REFerence[:SOURce]” on page 265.

***RST** +1.00000000E+007

Range 2.5E5–1E8

Key Entry **Reference Freq**

:REFerence[:SOURce]

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:MTONe:ARB:REFerence[:SOURce] INTernal|EXTernal
[:SOURce]:RADio:MTONe:ARB:REFerence[:SOURce]?
```

This command selects either an internal or external reference for the waveform clock. If EXTernal is selected, the external frequency *value must* be entered and the clock signal must be applied to the BASEBAND GEN REF IN rear-panel connector. See “:REFerence:EXTernal:FREQuency” on page 264 to enter the external reference frequency.

***RST** INT

Key Entry **ARB Reference Ext Int**

:SCLock:RATE

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:MTONe:ARB:SCLock:RATE <1.0 kHz – 1.0e8 kHz {1.0e8 kHz}>
[:SOURce]:RADio:MTONE:ARB:SCLock:RATE?
```

This command sets the sample clock rate.

The multitone generator should be on before executing this command. If this command is executed before the multitone generator is active, the entered value will be overridden by a calculated factory default value. Refer to ‘[:STATe]’ on page 257 to activate the modulation format.

***RST** +1.00000000E+008

Range 1–1E8

Key Entry **ARB Sample Clock**

:SETup

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:MTONe:ARB:SETup "<file name>"
[:SOURce]:RADio:MTONe:ARB:SETup?
```

This command retrieves a multitone waveform file. The name of a multitone waveform file is stored in the signal generator file system of MTONE files. This information is held in memory until you send the command that turns the waveform on. For information on the file name syntax, see “File Name Variables” on page 12.

Key Entry **Load From Selected File**

:SETup:STORe

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:MTONe:ARB:SETup:STORe "<file name>"
```

This command stores the current multitone waveform setup in the signal generator file system of MTONE files.

Key Entry **Store To File**

:SETup:TABLe

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:MTONe:ARB:SETup:TABLe <freq_spacing>,<num_tones>,
{<phase>,<state>}
[:SOURce]:RADio:MTONe:ARB:SETup:TABLe?
```

This command creates and configures a multitone waveform. The frequency offset, power, phase, and state value are returned when a query is initiated. The parameter format is as follows:

<freq_spacing> Expressed in Hertz (Hz–MHz). Spacing is limited by the 80 MHz bandwidth of the arbitrary waveform generator and the number of tones desired.

<num_tones> There must be a minimum of two tones and a maximum of 64.

<phase> 0-359

<state> An enabled state is +1. A disabled state is 0.

NOTE Frequency offset is related to frequency spacing. Frequency offset between tones equals the frequency spacing.

The variable <power> is expressed in units of decibels (dB).
To set the frequency spacing, refer to “:SETup:TABLe:FSPacing” on page 267.

***RST**

Tone	<frequency offset>	<power>	<phase>	<state>
Tone 1	-35000	+0.00000000E+000	+0	+1
Tone 2	-25000	+0.00000000E+000	+0	+1
Tone 3	-15000	+0.00000000E+000	+0	+1
Tone 4	-5000	+0.00000000E+000	+0	+1
Tone 5	+5000	+0.00000000E+000	+0	+1
Tone 6	+15000	+0.00000000E+000	+0	+1
Tone 7	+25000	+0.00000000E+000	+0	+1
Tone 8	+35000	+0.00000000E+000	+0	+1

Range <*freq_spacing*> (2 tones): 1E4–8E7 <*num_tones*>: 2–64
<*freq_spacing*> (>2 tones): 1E4 to (80 MHz ÷ (num_tones – 1))
<*phase*>: 0–359

Key Entry **Freq Spacing** **Number Of Tones** **Toggle State**

:SETup:TABLe:FSPacing

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:MTONe:ARB:SETup:TABLe:FSPacing <freq_spacing>
[:SOURce]:RADio:MTONe:ARB:SETup:TABLe:FSPacing?
```

This command sets the frequency spacing between the tones. The variable <`freq_spacing`> is expressed in Hertz (Hz–MHz).

To set frequency spacing and additional parameters required to create or configure a multitone waveform, refer to ":SETup:TABLe" on page 266. This command is the second step in creating a multitone waveform. Refer to "Creating a Multitone Waveform" on page 258 for all four steps.

***RST** +1.00000000E+004

Range <*freq_spacing*> (2 tones): 100 Hz –80 MHz
<*freq_spacing*> (>2 tones): 1E2 to (80 MHz ÷ (num_tones – 1))

Key Entry **Freq Spacing**

:SETup:TABLe:NTONes

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:MTONe:ARB:SETup:TABLe:NTONes <num_tones>
[:SOURce]:RADio:MTONe:ARB:SETup:TABLe:NTONes?
```

This command defines the number of tones in the multitone waveform. To specify the number of tones and additional parameters required to create or configure a multitone waveform, refer to ":SETup:TABLe" on page 266. This command is the third step in creating a multitone waveform. Refer to "Creating a Multitone Waveform" on page 258 for all four steps.

***RST** +8

Range 2–64

Key Entry **Number Of Tones**

:SETup:TABLe:PHASe:INITialize

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:MTONe:ARB:SETup:TABLe:PHASe:INITialize FIXed|RANDom
[:SOURce]:RADio:MTONe:ARB:SETup:TABLe:PHASe:INITialize?
```

This command initializes the phase in the multitone waveform table.

FIXed This choice sets the phase of all tones to the fixed value of 0 degrees.

RANDom This choice sets the phase of all tones to random values based on the setting on the random seed generator.

To change the random number generator seed value, refer to ":SETup:TABLe:PHASe:INITialize:SEED" on page 268.

This command is the first step in creating a multitone waveform. Refer to "Creating a Multitone Waveform" on page 258 for all four steps.

***RST** FIX

Key Entry **Initialize Phase Fixed Random**

:SETup:TABLe:PHASe:INITialize:SEED

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:MTONe:ARB:SETup:TABLe:PHASe:INITialize:SEED FIXed|RANDom
[:SOURce]:RADio:MTONe:ARB:SETup:TABLe:PHASe:INITialize:SEED?
```

This command initializes the random number generator seed that is used to generate phase values for the multitone waveform tones.

FIXed This choice sets the random number generator seed to a fixed value. This selection will generator random and repeatable phase values: the same phase values will be generated with subsequent execution of the command.

RANDom This choice sets the random number generator seed to a random value. This changes the phase value after each initialization of the phase.

***RST** FIX

Key Entry **Random Seed Fixed Random**

:ROW

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:MTONe:ARB:SETup:TABLe:ROW <row_number>,<power>,
<phase>,<state>
[:SOURce]:RADio:MTONe:ARB:SETup:TABLe:ROW? <row_number>
```

This command modifies the indicated tone (row) of the multitone waveform.

<row_number> The number of rows for this variable is determined by the :SETup:TABLe command.

The variable <power> is expressed in units of decibels (dB).
The variable <phase> is expressed in units of degrees (deg).
Frequency offset, power, phase, and state value are returned when a query is initiated. The output format is as follows:

```
<frequency offset>,<power>,<phase>,<state>
```

Refer to ":SETup:TABLe" on page 266 for information on how to change the number of rows. This command is the final step in creating a multitone waveform. Refer to "Creating a Multitone Waveform" on page 258 for all four steps.

***RST** *frequency offset*: –3.50000000E+004 <*power*>: +0.00000000E+000

<*phase*>: +0.00000000E+000 <*state*>: 1

Range *frequency offset*: –4E7 to 4E7 <*power*>: –80 to 0 <*phase*>: 0–359

<*state*>: 1

Key Entry **Goto Row** **Toggle State**

[:STATe]

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:MTONe:ARB[:STATe] ON|OFF|1|0
[:SOURce]:RADio:MTONe:ARB[:STATe]?
```

This command enables or disables the operating state of the multitone waveform generator.

***RST** 0

Key Entry **Multitone Off On**

Two Tone Subsystem ([:SOURce]:RADio:TTONe:ARB)

:ALIGnment

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:TTONe:ARB:ALIGnment LEFT|CENTer|RIGHt
[:SOURce]:RADio:TTONe:ARB:ALIGnment?
```

This command will align the two tones either left, center or right of the carrier frequency.

Key Entry **Alignment Left Cent Right**

:APPly

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:TTONe:ARB:APPLy
```

This command will cause the two-tone waveform to be regenerated using the current settings.

This command has no effect unless the two-tone waveform generator is enabled and a change has been made to the frequency spacing setting.

Key Entry **Apply Settings**

:FSPacing

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:TTONe:ARB:FSPacing <freq_spacing>
[:SOURce]:RADio:TTONe:ARB:FSPacing?
```

This command sets the frequency spacing between the tones.

The variable <freq_spacing> is expressed in units of Hertz (Hz–MHz).

***RST** +1.00000000E+004

Range 1E2–8E7

Key Entry **Freq Separation**

:HEADer:CLEar

Supported E8267D with Option 601 or 602

`[:SOURce]:RADio:TTONe:ARB:HEADer:CLEar`

This command clears the header information from the header file used for the two-tone waveform format. Header information consists of signal generator settings and marker routings associated with the waveform file. Refer to the *PSG User's Guide* for information on header files.

For this command to function, two tone must be on. To turn two tone on, see ':STATe' on page 280.

***RST** N/A

Key Entry **Clear Header**

:HEADer:SAVE

Supported E8267D with Option 601 or 602

`[:SOURce]:RADio:TTONe:ARB:HEADer:SAVE`

This command saves the header information to the header file used for the two-tone waveform format. Header information consists of signal generator settings and marker routings associated with the waveform file. Refer to the *PSG User's Guide* for information on header files.

For this command to function, two tone must be on. To turn two tone on, see '[:STATe]' on page 278.

***RST** N/A

Key Entry **Save Setup To Header**

:IQ:EXTernal:FILTer

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:TTONe:ARB:IQ:EXTernal:FILTer 40e6|THRough
[:SOURce]:RADio:TTONe:ARB:IQ:EXTernal:FILTer?
```

This command selects the filter or through path for I/Q signals routed to the rear-panel I and Q outputs. Selecting a filter with this command automatically sets ':IQ:EXTernal:FILTer:AUTO' on page 259 to OFF.

40e6 This choice applies a 40 MHz baseband filter.

THRough This choice bypasses filtering.

***RST** THR

Key Entry **40.000 MHz** **Through**

:IQ:EXTernal:FILTer:AUTO

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:TTONe:ARB:IQ:EXTernal:FILTer:AUTO ON|OFF|1|0
[:SOURce]:RADio:TTONe:ARB:IQ:EXTernal:FILTer:AUTO?
```

This command enables or disables the automatic selection of the filters for I/Q signals routed to the rear-panel I/Q outputs. A filter or through path will be selected for the two tone subsystem

`ON` (1) This choice will automatically select a digital modulation filter optimized for the current signal generator settings.

`OFF` (0) This choice disables the auto feature which lets you select a digital modulation filter or through path. Refer to ":IQ:EXTernal:FILTer" on page 259 for selecting a filter or through path.

***RST** 1

Key Entry **I/Q Output Filter Manual Auto**

:IQ:MODulation:ATTen

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:TTONe:ARB:IQ:MODulation:ATTen <val><unit>
[:SOURce]:RADio:TTONe:ARB:IQ:MODulation:ATTen?
```

This command attenuates the I/Q signals that modulate the signal generator's RF output.

The variable `<val>` is expressed as a number. The variable `<unit>` is expressed in decibels (dB).

***RST** +2.00000000E+000

Range 0–40 dB

Key Entry **Modulator Atten Manual Auto**

:IQ:MODulation:ATTen:AUTO

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:TTONe:ARB:IQ:MODulation:ATTen:AUTO ON|OFF|1|0
[:SOURce]:RADioTTONe:ARB:IQ:MODulation:ATTen:AUTO?
```

This command enables or disables the I/Q attenuation auto mode.

`ON` (1) This choice enables the attenuation auto mode which optimizes the modulator attenuation for the current conditions.

`OFF` (0) This choice holds the attenuator at its current setting or at a selected value. Refer to ‘:IQ:MODulation:ATTen’ on page 259 for setting the attenuation value.

***RST** 1

Key Entry **Modulator Atten Manual Auto**

:IQ:MODulation:FILTer

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:TTONe:ARB:IQ:MODulation:FILTer 40e6|THRough
[:SOURce]:RADio:TTONe:ARB:IQ:MODulation:FILTer?
```

This command enables you to select a filter or through path for I/Q signals modulated onto the RF carrier. Selecting a filter using this command will automatically set ‘:IQ:MODulation:FILTer:AUTO’ on page 260 to `OFF` (`0`) mode.

40E6 This choice applies a 40 MHz baseband filter to the I/Q signals.

THRough This choice bypasses filtering.

***RST** THR

Key Entry **40.000 MHz** **Through**

:IQ:MODulation:FILTer:AUTO

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:TTONe:ARB:IQ:MODulation:FILTer:AUTO ON|OFF|1|0
[:SOURce]:RADio:TTONe:ARB:IQ:MODulation:FILTer:AUTO?
```

This enables or disables the automatic selection of filters for I/Q signals modulating the signal generator’s RF carrier.

`ON` (`1`) Automatically selects a digital modulation filter.

`OFF` (`0`) Enables you to select a digital modulation filter or through path. Refer to “:IQ:MODulation:FILTer” on page 215 for selecting a filter or through path.

***RST** 1

Key Entry **I/Q Mod Filter Manual Auto**

:MDEStination:ALCHold

Supported E8267D with Option 601 or 602

CAUTION Incorrect ALC sampling can create a sudden unleveled condition that may create a spike in the RF output potentially damaging a DUT or connected instrument. Ensure that you set markers to let the ALC sample over an amplitude that accounts for the high power levels within the signal.

```
[:SOURce]:RADio:TTONe:ARB:MDEStination:ALCHold NONE|M1|M2|M3|M4
[:SOURce]:RADio:TTONe:ARB:MDEStination:ALCHold?
```

This command disables the marker ALC hold function, or it enables the marker hold function for the selected marker.

Use the ALC hold function when you have a waveform signal that incorporates idle periods, or when the increased dynamic range encountered with RF blanking is not desired. The ALC circuitry responds to the marker signal during the marker pulse (marker signal high), averaging the modulated signal level during this period.

The ALC hold function operates during the low periods of the marker signal. The marker polarity determines when the marker signal is high. For a positive polarity, this is during the marker points. For a negative polarity, this is when there are no marker points. To set a marker's polarity, see ':MPOLarity:MARKer1|2|3|4' on page 276. For more information on markers, see ':MARKer:[SET]' on page 218.

NOTE Do not use the ALC hold for more than 100 ms, because it can affect the waveform's output amplitude.

The marker signal has a minimum of a two-sample delay in its response relative to the waveform signal response. To compensate for the marker signal delay, offset marker points from the waveform sample point at which you want the ALC sampling to begin.

The ALC hold setting is part of the file header information, so saving the setting to the file header saves the current marker routing for the waveform file.

NOTE A waveform file that has unspecified settings in the file header uses the previous waveform's routing settings.

For more information on the marker ALC hold function, see the *PSG User's Guide*. To configure marker points, refer to the following sections located in the Dual ARB subsystem:

- For clearing a single marker point or a range of marker points, see‘:MARKer:CLEar’ on page 216.
- For clearing all marker points, see ‘:MARKer:CLEar:ALL’ on page 217.
- For shifting marker points, see ‘:MARKer:ROTate’ on page 217.
- For setting marker points, see ‘:MARKer:[SET]’ on page 218.

NONE	This terminates the marker ALC hold function.
M1–M4	These are the marker choices. The ALC hold feature uses only one marker at a time.
***RST**	NONE
Key Entry	**None** **Marker 1** **Marker 2** **Marker 3** **Marker 4**
Remarks	N/A

:MDEStination:PULSe

Supported E8267D with Option 601 or 602

CAUTION	The pulse function incorporates ALC hold. Incorrect ALC sampling can create a sudden unleveled condition that may create a spike in the RF output, potentially damaging a DUT or connected instrument. Ensure that you set markers to let the ALC sample over an amplitude that accounts for the high power levels within the signal.

```
[:SOURce]:RADio:TTONe:ARB:MDEStination:PULSe NONE|M1|M2|M3|M4
[:SOURce]:RADio:TTONe:ARB:MDEStination:PULSe?
```

This command disables the marker RF blanking/pulse function, or it enables the marker RF blanking/pulse function for the selected marker.

This function automatically incorporates the ALC hold function, so there is no need to select both functions for the same marker.

NOTE	Do not use ALC hold for more than 100 ms, because it can affect the waveform’s output amplitude.

The signal generator blanks the RF output when the marker signal goes low. The marker polarity determines when the marker signal is low. For a positive polarity, this is during the marker points. For a negative polarity, this is when there are no marker points. To set a marker’s polarity, see ‘:MPOLarity:MARKer1|2|3|4’ on page 276. For more information on markers, see ‘:MARKer:[SET]’ on page 218.

NOTE Set marker points prior to using this function. Enabling this function without setting marker points may create a continuous low or high marker signal, depending on the marker polarity. This creates the condition where there is either no RF output or a continuous RF output.

To configure marker points, refer to the following sections located in the Dual ARB subsystem:

- For clearing a single marker point or a range of marker points, see':MARKer:CLEar' on page 216.
- For clearing all marker points, see ':MARKer:CLEar:ALL' on page 217.
- For shifting marker points, see ':MARKer:ROTate' on page 217.
- For setting marker points, see ':MARKer:[SET]' on page 218.

The marker signal has a minimum of a two-sample delay in its response relative to the waveform signal response. To compensate for the marker signal delay, offset marker points from the waveform sample point at which you want the RF blanking to begin.

The RF blanking setting is part of the file header information, so saving the setting to the file header saves the current marker routing for the waveform file.

NOTE A waveform file that has unspecified settings in the file header uses the previous waveform's routing settings. This could create the situation where there is no RF output signal, because the previous waveform used RF blanking

For more information on the marker RF blanking function, see the *PSG User's Guide*.

NONE	This terminates the marker RF blanking/pulse function.
M1–M4	These are the marker choices. The RF blanking/pulse feature uses only one marker at a time.
***RST**	NONE
Key Entry	**None** **Marker 1** **Marker 2** **Marker 3** **Marker 4**

:MPOLarity:MARKer1|2|3|4

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:TTONe:ARB:MPOLarity:MARKer1|2|3|4 NEGative|POSitive
[:SOURce]:RADio:TTONe:ARB:MPOLarity:MARKer1|2|3|4?
```

This command sets the polarity for the selected marker.

For a positive marker polarity, the marker signal is high during the marker points. For a negative marker polarity, the marker signal is high during the period of no marker points. To configure marker points, refer to the following sections located in the Dual ARB subsystem:

- For clearing a single marker point or a range of marker points, see':MARKer:CLEar' on page 216.
- For clearing all marker points, see ':MARKer:CLEar:ALL' on page 217.
- For shifting marker points, see ':MARKer:ROTate' on page 217.
- For information on markers and setting marker points, see ':MARKer:[SET]' on page 218.

***RST** POS

Key Entry **Marker 1 Polarity Neg Pos** **Marker 2 Polarity Neg Pos** **Marker 3 Polarity Neg Pos** **Marker 4 Polarity Neg Pos**

:REFerence:EXTernal:FREQuency

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:TTONe:ARB:REFerence:EXTernal:FREQuency <val>
[:SOURce]:RADio:TTONe:ARB:REFerence:EXTernal:FREQuency?
```

This command allows you to enter the frequency of the external reference.

The variable `<val>` is expressed in units of Hertz (Hz–MHz).

The value specified by this command is effective only when you are using an external ARB reference applied to the BASEBAND GEN REF IN rear-panel connector.

To specify external as the ARB reference source type, refer to ":REFerence[:SOURce]" on page 223.

***RST** +1.00000000E+007

Range 2.5E5–1E8

Key Entry **Reference Freq**

:REFerence[:SOURce]

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:TTONe:ARB:REFerence[:SOURce] INTernal|EXTernal
[:SOURce]:RADio:TTONe:ARB:REFerence[:SOURce]?
```

This command selects either an internal or external reference for the waveform clock. If EXTernal is selected, the external frequency *value must* be entered and the clock signal must be applied to the BASEBAND GEN REF IN rear-panel connector. See ":REFerence:EXTernal:FREQuency" on page 264 to enter the external reference frequency.

***RST** INT

Key Entry **ARB Reference Ext Int**

:SCLock:RATE

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:TTONe:ARB:SCLock:RATE <1.0 kHz – 1.0e8 kHz {1.0e8 kHz}>
[:SOURce]:RADio:TTONE:ARB:SCLock:RATE?
```

This command sets the sample clock rate.

The multitone generator should be on before executing this command. If this command is executed before the multitone generator is active, the entered value will be overridden by a calculated factory default value. Refer to ‘[:STATe]’ on page 257 to activate the modulation format.

***RST** +1.00000000E+008

Range 1–1E8

Key Entry **ARB Sample Clock**

[:STATe]

Supported E8267D with Option 601or 602

```
[:SOURce]:RADio:TTONe:ARB[:STATe] ON|OFF|1|0
[:SOURce]:RADio:TTONe:ARB[:STATe]?
```

This command enables or disables the on/off operational state of the two-tone waveform generator function.

***RST** 0

Key Entry **Two Tone Off On**

Wideband Digital Modulation Subsystem ([:SOURce]:WDM)

:IQADjustment:IOFFset

Supported E8267D with Option 015

```
[:SOURce]:WDM:IQADjustment:IOFFset <val><unit>
[:SOURce]:WDM:IQADjustment:IOFFset?
```

This command sets the I channel offset value, as a percent of the full scale. 100% offset is equivalent to 500 mV DC at the input connector.

***RST** +0.00000000E+000

Range –5E1 to +5E1

Key Entry **I Offset**

:IQADjustment:QOFFset

Supported E8267D with Option 015

```
[:SOURce]:WDM:IQADjustment:QOFFset <val><unit>
[:SOURce]:WDM:IQADjustment:QOFFset?
```

This command sets the Q channel offset value, as a percent of the full scale. 100% offset is equivalent to 500 mV DC at the input connector

***RST** +0.00000000E+000

Range –5E1 to +5E1

Key Entry **Q Offset**

:IQADjustment:QSKew

Supported E8267D with Option 601or 602 and Option 015

```
[:SOURce]:WDM:IQADjustment:QSKew <val>
[:SOURce]:WDM:IQADjustment:QSKew?
```

This command adjusts the phase angle between the I and Q vectors.
The variable `<val>` is expressed in units of degrees with a minimum resolution of 0.1.

Positive skew increases the angle from 90 degrees while negative skew decreases the angle from 90 degrees. When the quadrature skew is zero, the phase angle is 90 degrees. If the signal generator is operating at frequencies greater than 3.3 GHz, quadrature skew settings greater than ±5 degrees will not be within specifications.

This command is effective only if the state of the I/Q adjustment function is set to ON. Refer to ":IQADjustment[:STATe]" on page 280.

***RST** +0.00000000E+000

Range –1E1 to +1E1

Key Entry **Quadrature Skew**

:IQADjustment[:STATe]

Supported E8267D with Option 015

```
[:SOURce]:WDM:IQADjustment[:STATe] ON|OFF|1|0
[:SOURce]:WDM:IQADjustment[:STATe]?
```

This command enables or disables the wideband I/Q adjustments.

***RST** 0

Key Entry **I/Q Adjustments Off On**

:STATe

Supported E8267D with Option 015

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

This command enables or disables the wideband I/Q modulator.
The I/Q modulator is enabled whenever a digital format is turned on.
The `I/Q` annunciator will be shown on the signal generator display whenever the I/Q modulator is on.

***RST** 0

Key Entry **I/Q Off On**

6 SCPI Command Compatibility

In the following sections, this chapter provides a comprehensive listing of SCPI commands and programming codes for signal generator models supported by Agilent PSG Signal Generators:

: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 PSG 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.

E8241A / 44A / 51A / 54A and the E8247C / 57C / 67C PSG Compatible SCPI Commands

All commands are fully supported. To use the commands, select *SCPI* as the remote language. See ":LANGuage" on page 81 for selecting the language type.

8340B / 41B and 8757D Compatible Commands

The tables in this section provide the following:

Table 6-1 on page 285: a comprehensive list of 8340B/41B and 8757D programming codes, listed in alphabetical order. The equivalent SCPI command sequence for each supported code is provided; codes that are *not* supported by the PSG family are indicated as such in the command column.

Table 6-2 on page 298: a list of the implemented 8340B/41B and 8757D programming codes that set the active function. This table also indicates which codes are compatible with the RB command (knob), and lists the operation active (OA) query, the operation prior (OP) query, and the increment (up), and the decrement (down) SCPI commands.

NOTE Compatibility is provided for GPIB only; RS-232 and LAN are *not* supported.

When using the programming codes in this section, you can:

- set the PSG system language to 8340 or 8757 for the current session:

  Utility > **GPIB / RS-232 LAN** > **Preset Language** > **8340B** (or **8757D**)

  or

  `:SYST:LANG “8340”` (or `“8757”`)

- set the PSG system language to 8340 or 8757 so that it does not reset with either preset or cycling power:

  Utility > **Power On / Preset** > **Preset Language** > **8340B** (or **8757D**)

  or

  `:SYST:PRESET:LANG “8340”` (or `“8757”`)

- set the `*IDN?` response to any 8340-like response you prefer:

  use the command :SYSTem:IDN on page 282.

Table 6-1 8340B / 41B Prog. Codes & Equivalent SCPI Sequences

Cmd	Description	8340	8757	Equivalent SCPI Command Sequence
A1	Internal leveling mode	Y	Y	`[:SOURce]:POWer:ALC:SOURce INTernal`
A2	External leveling mode with diode detector	Y	Y	`[:SOURce]:POWer:ALC:SOURce DIODe` `[:SOURce]:POWer:ALC:SOURce:EXTernal:COUPling <val> dB`
A3	External leveling mode with power meter	Y	Y	*supported, but has no effect on PSG*
AK0	Amplitude markers off	Y	Y	`[:SOURce]:MARKer:AMPLitude OFF|0`
AK1	Amplitude markers on	Y	Y	`[:SOURce]:MARKer:AMPLitude ON|1`
AL0	Alternate sweep mode off	Y	Y	`:SYSTem:ALTernate:STATe OFF`
AL1	Alternate sweep mode on	Y	Y	`:SYSTem:ALTernate:STATe ON` `:SYSTem:ALTernate n`
AM0	Amplitude modulation off	Y	N	`[:SOURce]:AM1:STATe OFF|0` `[:SOURce]:AM2:STATe OFF|0`

AM1	Amplitude modulation on	Y	N	`[:SOURce]:AM1:STATe OFF|0`
				`[:SOURce]:AM2:SOURce EXT[1]`
				`[:SOURce]:AM2:EXTernal[1]:COUPling DC`
				`[:SOURce]:AM2:DEPTh 100`
				`[:SOURce]:AM2:EXTernal[1]:IMPedance 600`
				`[:SOURce]:AM2:STATe ON|1`

AS0	Alternate state selection: select current front panel	Y	Y	*supported, but no equivalent SCPI command sequence*
AS1	Alternate state selection: select recalled state	Y	Y	*supported, but no equivalent SCPI command sequence*
AT	Set attenuator	Y	N	`[:SOURce]:POWer:ATTenuation <val><unit>`
AU	Auto-coupled mode to obtain shortest possible sweep time	Y	N	`[:SOURce]:SWEep:TIME:AUTO ON|1`
BC	Advance to next frequency bandcrossing	N	N	*not supported*
C1	1 MHz crystal marker frequency	N	Y	*supported, but has no effect on PSG*
C2	10 MHz crystal marker frequency	N	Y	*supported, but has no effect on PSG*
C3	50 MHz crystal marker frequency	N	Y	*supported, but has no effect on PSG*
C4	External crystal marker frequency	N	Y	*supported, but has no effect on PSG*

Table 6-1 8340B / 41B Prog. Codes & Equivalent SCPI Sequences (Continued)

Cmd	Description	8340	8757	Equivalent SCPI Command Sequence
CA0	Amplitude crystal markers off	N	Y	*supported, but has no effect on PSG*
CA1	Amplitude crystal markers on	N	Y	*supported, but has no effect on PSG*

CF	Center frequency (step sweep)	Y	Y	`[:SOURce]:SWEep:MODE AUTO`
				`[:SOURce]:FREQuency:MODE SWEep`
				`[:SOURce]:FREQuency:CENTer <val><unit>`

CL0	Intensity crystal markers off	N	Y	*supported, but has no effect on PSG*
CL1	Intensity crystal markers on	N	Y	*supported, but has no effect on PSG*
CS	Clear both status bytes	Y	Y	`*CLS`

CW	Set CW frequency	Y	Y	`[:SOURce]:SWEep:MODE AUTO`
				`[:SOURce]:FREQuency:MODE CW`
				`[:SOURce]:FREQuency[:CW] <val><unit>`

DB	dB(m) terminator	Y	Y	`DB`

DF	Delta frequency (step sweep)	Y	Y	`[:SOURce]:SWEep:MODE AUTO`
				`[:SOURce]:FREQuency:MODE SWEep`
				`[:SOURce]:FREQuency:SPAN <val> <unit>`

DM	dB(m) terminator	Y	Y	`DB`
DN	Step down (decrements active function by step value)	Y	Y	*supported, see Table 6-2 on page 234*
DP0	Display blanking off	N	Y	`DISPlay[:WINDow][:STATe] ON|1`
DP1	Display blanking on	N	Y	`DISPlay[:WINDow][:STATe] OFF|0`
DU0	Display update off	Y	Y	`DISPlay[:WINDow][:STATe] OFF|0`
DU1	Display update on	Y	Y	`DISPlay[:WINDow][:STATe] ON|1`
EF	Entry display off	Y	Y	`DISPlay[:WINDow][:STATe] ON|1`
EK	Enable knob	N	N	*not supported*
EM0	Extended marker mode off	N	Y	*supported, but* *no equivalent SCPI command sequence*
EM1	Extended marker mode on	N	Y	*supported, but* *no equivalent SCPI command sequence*
F1	20 MHz/V FM sensitivity	N	N	*not supported*
F2	6 MHz/V FM sensitivity	N	N	*not supported*

Table 6-1 8340B / 41B Prog. Codes & Equivalent SCPI Sequences (Continued)

<table>
<tr><th>Cmd</th><th>Description</th><th>8340</th><th>8757</th><th>Equivalent SCPI Command Sequence</th></tr>
<tr><td>FA</td><td>Start frequency (step sweep)</td><td>Y</td><td>Y</td><td>[:SOURce]:SWEep:MODE AUTO
[:SOURce]:FREQuency:MODE SWEep
[:SOURce]:FREQuency:STARt <val><unit></td></tr>
<tr><td>FB</td><td>Stop frequency (step sweep)</td><td>Y</td><td>Y</td><td>[:SOURce]:SWEep:MODE AUTO
[:SOURce]:FREQuency:MODE SWEep
[:SOURce]:FREQuency:STOP <val><unit></td></tr>
<tr><td>FL0</td><td>CW filter off</td><td>N</td><td>Y</td><td>supported, but has no effect on PSG</td></tr>
<tr><td>FL1</td><td>CW filter on</td><td>N</td><td>Y</td><td>supported, but has no effect on PSG</td></tr>
<tr><td>FM0</td><td>Frequency modulation off</td><td>Y</td><td>N</td><td>[:SOURce]:FM1:STATe OFF|0
[:SOURce]:FM2:STATe OFF|0</td></tr>
<tr><td>FM1</td><td>Frequency modulation on</td><td>Y</td><td>N</td><td>[:SOURce]:FM1:STATe OFF|0
[:SOURce]:FM2:SOURce EXT2
[:SOURce]:FM2:EXTernal2:COUPling DC
[:SOURce]:FM2:EXTernal2:IMPedance 50
[:SOURce]:FM2:STATe ON|1</td></tr>
<tr><td>FM1</td><td>Frequency modulation sensitivity</td><td>Y</td><td>N</td><td>[:SOURce]:FM2[:DEViation] <val><unit></td></tr>
<tr><td>FP</td><td>Fast phaselock</td><td>Y</td><td>N</td><td>supported, but has no effect on PSG</td></tr>
<tr><td>GZ</td><td>GHz terminator</td><td>Y</td><td>Y</td><td>GHZ</td></tr>
<tr><td>HZ</td><td>Hz terminator</td><td>Y</td><td>Y</td><td>HZ</td></tr>
<tr><td>IF</td><td>Increment frequency</td><td>Y</td><td>N</td><td>TRIGger[:SEQuence][:IMMediate]
or
[:SOURce]:FREQuency[:CW] UP</td></tr>
<tr><td>IL</td><td>Input learn string</td><td>Y</td><td>Y</td><td>supported, but
no equivalent SCPI command sequence</td></tr>
</table>

Table 6-1 8340B / 41B Prog. Codes & Equivalent SCPI Sequences (Continued)

Cmd	Description	8340	8757	Equivalent SCPI Command Sequence

IP	Instrument preset	Y	N	`SYSTem:PRESet`
				`[:SOURce]:FREQuency[:CW]:STEP [:INCRement] 1 GHZ`
				
				`[:SOURce]:FREQuency:MULTiplier <saved multiplier>`
				
				`[:SOURce]:SWEep:MODE AUTO`
				`[:SOURce]:FREQuency:MODE SWEep`
				`[:SOURce]:FREQuency:STARt 2 GHz or MIN`
				`[:SOURce]:FREQuency:STOP MAX`
				`[:SOURce]:POWer[:LEVel][:IMMediate] [:AMPLitude] 0 dB`
				
				`OUTput[:STATe] ON|1`

IP	Instrument preset	N	Y	`SYSTem:PRESet`
				`SYSTem:LANGuage "8757"`
				`[:SOURce]:SWEep:MODE AUTO`
				`[:SOURce]:FREQuency:MODE SWEep`
				`[:SOURce]:FREQuency:STARt 2 GHz or MIN`
				`[:SOURce]:FREQuency:STOP MAX`
				`[:SOURce]:POWer[:LEVel][:IMMediate] [:AMPLitude] 0 dB`
				
				`OUTput[:STATe] ON|1`

IX	Input micro learn string	N	Y	*supported, but has no effect on PSG*
KR	Key release	Y	Y	*supported, but* *no equivalent SCPI command sequence*
KZ	kHz terminator	Y	Y	`KHZ`
M0 MO	Frequency marker off	Y	Y	`[:SOURce]:MARKer[n]:[STATe] OFF|0`
MA	Turn on and set frequency marker 0	Y	Y	`[:SOURce]:MARKer0:[STATe] ON|1` `[:SOURce]:MARKer0:FREQuency <val><unit>`
M1	Turn on and set frequency marker 1	Y	Y	`[:SOURce]:MARKer1:[STATe] ON|1` `[:SOURce]:MARKer1:FREQuency <val><unit>`
M2	Turn on and set frequency marker 2	Y	Y	`[:SOURce]:MARKer2:[STATe] ON|1` `[:SOURce]:MARKer2:FREQuency <val><unit>`

Table 6-1 8340B / 41B Prog. Codes & Equivalent SCPI Sequences (Continued)

Cmd	Description	8340	8757	Equivalent SCPI Command Sequence
M3	Turn on and set frequency marker 3	Y	Y	[:SOURce]:MARKer3:[STATe] ON|1 [:SOURce]:MARKer3:FREQuency <val><unit>
M4	Turn on and set frequency marker 4	Y	Y	[:SOURce]:MARKer4:[STATe] ON|1 [:SOURce]:MARKer4:FREQuency <val><unit>
M5	Turn on and set frequency marker 5	Y	Y	[:SOURce]:MARKer5:[STATe] ON|1 [:SOURce]:MARKer5:FREQuency <val><unit>
M6	Turn on and set frequency marker 6	Y	Y	[:SOURce]:MARKer6:[STATe] ON|1 [:SOURce]:MARKer6:FREQuency <val><unit>
M7	Turn on and set frequency marker 7	Y	Y	[:SOURce]:MARKer7:[STATe] ON|1 [:SOURce]:MARKer7:FREQuency <val><unit>
M8	Turn on and set frequency marker 8	Y	Y	[:SOURce]:MARKer8:[STATe] ON|1 [:SOURce]:MARKer8:FREQuency <val><unit>
M9	Turn on and set frequency marker 9	Y	Y	[:SOURce]:MARKer9:[STATe] ON|1 [:SOURce]:MARKer9:FREQuency <val><unit>
MC	Active marker to center frequency	Y	Y	*supported, but* *no equivalent SCPI command sequence*
MD	Marker delta	N	N	*not supported*
MP0	Marker 1-2 sweep off	N	N	*not supported*
MP1	Marker 1-2 sweep on	N	N	*not supported*
MS	Milliseconds terminator	Y	Y	MS
MZ	MHz terminator	Y	Y	MHZ
NA	Network analyzer mode	N	Y	*supported, but* *no equivalent SCPI command sequence*
NT	Network analyzer trigger	N	Y	*supported, but has no effect on PSG*
OA	Output active parameter	Y	Y	*supported, see Table 6-2 on page 234*
OB	Output next bandcross frequency	N	N	*not supported*
OC	Output coupled parameters (start frequency, center frequency, sweep time)	Y	Y	[:SOURce]:FREQuency:STARt? [:SOURce]:FREQuency:CENTer? [:SOURce]:SWEep:TIME?
OD	Output diagnostic values	N	N	*not supported*

Table 6-1 8340B / 41B Prog. Codes & Equivalent SCPI Sequences (Continued)

Cmd	Description	8340	8757	Equivalent SCPI Command Sequence
OE	Output when executed	N	Y	*supported, but no equivalent SCPI command sequence*
OF	Output fault	Y	N	*supported, but no equivalent SCPI command sequence*
OI	Output identification	Y	Y	`*IDN?`
OK	Output last lock frequency	N	N	*not supported*
OL	Output learn string	Y	Y	*supported, but no equivalent SCPI command sequence*
OM	Output mode string	N	Y	*supported, but no equivalent SCPI command sequence*
OP	Output interrogated parameter	Y	Y	*supported, see Table 6-2 on page 234*
OPA2	Output external detector coupling factor	Y	Y	`[:SOURce]:POWer:ALC:SOURce:EXTernal :COUPling?`
OPAT	Output attenuator	Y	N	`[:SOURce]:POWer:ATTenuation?`
OPCF	Output center frequency	Y	Y	`[:SOURce]:FREQuency:CENTer?`
OPCW	Output CW frequency	Y	Y	`[:SOURce]:FREQuency:CW?`
OPDF	Output delta frequency	Y	Y	`[:SOURce]:FREQuency:SPAN?`
OPFA	Output start frequency	Y	Y	`[:SOURce]:FREQuency:STARt?`
OPFB	Output stop frequency	Y	Y	`[:SOURce]:FREQuency:STOP?`
OPFM1	Output FM sensitivity	Y	N	`[:SOURce]:FM2[:DEViation]?`
OPMA	Output marker 0 frequency	Y	Y	`[:SOURce]:MARKer0:FREQuency?`
OPM1	Output marker 1 frequency	Y	Y	`[:SOURce]:MARKer1:FREQuency?`
OPM2	Output marker 2 frequency	Y	Y	`[:SOURce]:MARKer2:FREQuency?`
OPM3	Output marker 3 frequency	Y	Y	`[:SOURce]:MARKer3:FREQuency?`
OPM4	Output marker 4 frequency	Y	Y	`[:SOURce]:MARKer4:FREQuency?`
OPM5	Output marker 5 frequency	Y	Y	`[:SOURce]:MARKer5:FREQuency?`
OPM6	Output marker 6 frequency	Y	Y	`[:SOURce]:MARKer6:FREQuency?`
OPM7	Output marker 7 frequency	Y	Y	`[:SOURce]:MARKer7:FREQuency?`

Table 6-1 8340B / 41B Prog. Codes & Equivalent SCPI Sequences (Continued)

Cmd	Description	8340	8757	Equivalent SCPI Command Sequence
OPM8	Output marker 8 frequency	Y	Y	`[:SOURce]:MARKer8:FREQuency?`
OPM9	Output marker 9 frequency	Y	Y	`[:SOURce]:MARKer9:FREQuency?`
OPPL	Output power level	Y	Y	`[:SOURce]:POWer[:LEVel][:IMMediate][:AMPLitude]?`
OPPS	Output power sweep span	Y	Y	`[:SOURce]:POWer:SPAN?`
OPSB	Output # of sweep buckets	N	N	*supported, but no equivalent SCPI command sequence*
OPSF	Output frequency step size	Y	Y	`[:SOURce]:FREQuency[:CW]:STEP[:INCRement]?`
OPSHA1	Output power level	Y	N	`[:SOURce]:POWer[:LEVel][:IMMediate][:AMPLitude]?`
OPSHA2	Output ALC level	Y	N	`[:SOURce]:POWer:ALC:LEVel?`
OPSHA3	Output ALC level	Y	N	`[:SOURce]:POWer:ALC:LEVel?`
OPSHAZ	Output ALC level	Y	N	`[:SOURce]:POWer:ALC:LEVel?`
OPSHCF	Output frequency step size	Y	N	`[:SOURce]:FREQuency[:CW]:STEP[:INCRement]?`
OPSHCW	Output swept CW frequency	Y	Y	`[:SOURce]:FREQuency:STARt?` or `[:SOURce]:FREQuency:STOP?`
OPSHFA	Output frequency multiplier	Y	Y	`[:SOURce]:FREQuency:MULTiplier?`
OPSHFB	Output frequency offset	Y	Y	`[:SOURce]:FREQuency:OFFSet?`
OPSHPL	Output power step size	Y	N	`[:SOURce]:POWer[:LEVel][:IMMediate][:AMPLitude]:STEP[:INCRement]?`
OPSHPS	Output ALC level	Y	Y	`[:SOURce]:POWer:ALC:LEVel?`
OPSHRF	Output power level	Y	N	`[:SOURce]:POWer[:LEVel][:IMMediate][:AMPLitude]?`
OPSHSL	Output attenuator	Y	N	`[:SOURce]:POWer:ATTenuation?`
OPSHSN	Output sweep step points	N	Y	`[:SOURce]:SWEep:POINts?`
OPSL	Output power slope	Y	Y	`[:SOURce]:POWer:SLOPe?`
OPSM	Output manual frequency	Y	Y	`[:SOURce]:FREQuency:MANual?`

Table 6-1 8340B / 41B Prog. Codes & Equivalent SCPI Sequences (Continued)

Cmd	Description	8340	8757	Equivalent SCPI Command Sequence
OPSN	Output sweep step points	Y	Y	`[:SOURce]:SWEep:POINts?`
OPSP	Output power step size	Y	Y	`[:SOURce]:POWer[:LEVel][:IMMediate]` `[:AMPLitude]:STEP[:INCRement]?`
OPST	Output sweep time	Y	Y	`[:SOURce]:SWEep:TIME?`
OPTL	Output sweep time limit	Y	Y	`[:SOURce]:SWEep:TIME:LLIMit?`
OR	Output internally measured power level	N	N	*not supported*
OS	Output status bytes	Y	Y	*supported, but* *no equivalent SCPI command sequence*
OX	Output micro learn string	N	Y	*supported, but has no effect on PSG*

PL	Set power level	Y	Y	`[:SOURce]:POWer:ATTenuation:AUTO ON|1`
				`[:SOURce]:POWer[:LEVel][:IMMediate]`
				`[:AMPLitude] <val><unit>`

PM0	Pulse modulation off	Y	Y	`[:SOURce]:PULM:STATe OFF|0`
PM1	Pulse modulation on	Y	N	`[:SOURce]:PULM:SOURce EXTernal` `[:SOURce]:PULM:STATe ON|1`
PM1	27.8 KHz square wave pulse modulation on	N	Y	`[:SOURce]:PULM:SOURce SCALar` `[:SOURce]:PULM:STATe ON|1`
PS0	Power sweep off	Y	Y	`[:SOURce]:POWer:MODE FIXed`
PS1	Power sweep on	Y	Y	`[:SOURce]:POWer:MODE SWEep` `[:SOURce]:POWer:SPAN <val> dB`
R2	Extended status byte #2 mask	N	Y	*supported, but has no effect on PSG*
RB	Control knob remotely	Y	Y	*supported, but* *no equivalent SCPI command sequence*
RC	Recall state	Y	Y	`*RCL <reg_num>[,<seq_num>]`
RE	Extended status byte mask	Y	Y	*supported, but* *no equivalent SCPI command sequence*
RF0	RF output off	Y	Y	`OUTPut[:STATe] OFF|0`
RF1	RF output on	Y	Y	`OUTPut[:STATe] ON|1`
RM	Status byte mask	Y	Y	`*SRE <mask>`

Table 6-1 8340B / 41B Prog. Codes & Equivalent SCPI Sequences (Continued)

Cmd	Description	8340	8757	Equivalent SCPI Command Sequence
RP0	RF peaking off	Y	N	*supported, but has no effect on PSG*
RP0	RF blanking off	N	Y	*supported, but has no effect on PSG*
RP1	RF peaking on	Y	N	*supported, but has no effect on PSG*
RP1	RF blanking on	N	Y	*supported, but has no effect on PSG*
RS	Reset sweep	Y	Y	*supported, but* *no equivalent SCPI command sequence*

S1	Continuous sweep mode	Y	Y	`[:SOURce]:SWEep:MODE AUTO`
				
				`[:SOURce]:SWEep:GENeration ANALog:TRIGger[:SEQuence]:SOURce IMMediate:INITiate:CONTinuous[:ALL] ON`

S2	Single sweep mode	Y	Y	`[:SOURce]:SWEep:MODE AUTO`
				
				`[:SOURce]:SWEep:GENeration ANALog:TRIGger[:SEQuence]:SOURce IMMediate:INITiate:CONTinuous[:ALL] OFF`

S3	Manual frequency sweep mode	Y	Y	`[:SOURce]:SWEep:MODE MANual`
				
				`[:SOURce]:SWEep:GENeration ANALog:TRIGger[:SEQuence]:SOURce IMMediate:INITiate:CONTinuous[:ALL] OFF`

SB	Number of sweep buckets	N	Y	*supported, but* *no equivalent SCPI command sequence*
SC	Seconds terminator	Y	Y	`S`
SF	Frequency step size	Y	Y	`[:SOURce]:FREQuency[:CW]:STEP[:INCRement] <val><unit>`

SG	Single sweep mode	Y	Y	`[:SOURce]:SWEep:MODE AUTO`
				
				`[:SOURce]:SWEep:GENeration ANALog :TRIGger[:SEQuence]:SOURce IMMediate :INITiate:CONTinuous[:ALL] OFF`

SH	Shift prefix	Y	Y	*supported, but* *no equivalent SCPI command sequence*
SH01	Blank display	N	Y	`DISPlay[:WINDow][:STATe] OFF|0`

Table 6-1 8340B / 41B Prog. Codes & Equivalent SCPI Sequences (Continued)

Cmd	Description	8340	8757	Equivalent SCPI Command Sequence
SHA1	Disable ALC and set power level	Y	N	`[:SOURce]:POWer:ALC[:STATe] OFF|0` `[:SOURce]:POWer[:LEVel][:IMMediate][:AMPLitude] <val><unit>`
SHA2	External leveling mode with millimeter head module	Y	N	`[:SOURce]:POWer:ALC:SOURce MMHead` `[:SOURce]:POWer:ALC:LEVel <val>dB`

SHA3	Directly control linear modulator circuit (bypassing ALC)	Y	N	`[:SOURce]:POWer:ATTenuation:AUTO OFF|0`
				`[:SOURce]:POWer:ALC[:STATe] OFF|0`
				`[:SOURce]:POWer:ALC:LEVel <val>dB`

SHAK	Immediate YTF peak	Y	N	*supported, but has no effect on PSG*
SHAL	Retain multiplication factor on power on/off and preset	Y	Y	*supported, but no equivalent SCPI command sequence*
SHAM	Pulse modulation enhancement	Y	N	*supported, but has no effect on PSG*
SHAZ	External leveling mode with millimeter head module	Y	N	`[:SOURce]:POWer:ALC:SOURce MMHead` `[:SOURce]:POWer:ALC:LEVel <val>dB`
SHCF	Frequency step size	Y	N	`[:SOURce]:FREQuency[:CW]:STEP[:INCRement] <val><unit>`
SHCF	Coarse CW resolution	N	Y	*supported, but has no effect on PSG*

SHCW	Swept CW	N	Y	`[:SOURce]:SWEep:MODE AUTO`
				`[:SOURce]:FREQuency:MODE SWEep`
				`[:SOURce]:FREQuency:STARt <val><unit>`
				`[:SOURce]:FREQuency:STOP <val><unit>`

SHDF	Fine CW resolution	N	Y	*supported, but has no effect on PSG*
SHEF	Restore cal. const. access function	N	N	*not supported*
SHFA	Frequency multiplier	Y	Y	`[:SOURce]:FREQuency:MULTiplier <val>`
SHFB	Frequency offset	Y	Y	`[:SOURce]:FREQuency:OFFSet <val><unit>`
SHIP	Reset multiplication factor to 1 and preset instrument	Y	Y	*supported, but no equivalent SCPI command sequence*
SHM0	All frequency markers off	Y	Y	`[:SOURce]:MARKer:AOFF`
SHM1	Turn on and set marker delta	N	Y	`[:SOURce]:MARKer:MODE DELTa`
SHM2	Enable counter interface	N	Y	*supported, but has no effect on PSG*
SHM3	Disable counter interface	N	Y	*supported, but has no effect on PSG*

Table 6-1 8340B / 41B Prog. Codes & Equivalent SCPI Sequences (Continued)

<table>
<tr><th>Cmd</th><th>Description</th><th>8340</th><th>8757</th><th>Equivalent SCPI Command Sequence</th></tr>
<tr><td>SHM4</td><td>Diagnostics: test/display results</td><td>N</td><td>N</td><td>not supported</td></tr>
<tr><td>SHMO</td><td>All frequency markers off</td><td>N</td><td>Y</td><td>[:SOURce]:MARKer:AOFF</td></tr>
<tr><td>SHMP</td><td>Set start frequency to marker 1 and set stop frequency to marker 2</td><td>Y</td><td>Y</td><td>[:SOURce]:SWEep:MARKer:XFER</td></tr>
<tr><td>SHPL</td><td>Power step size</td><td>Y</td><td>N</td><td>[:SOURce]:POWer[:LEVel][:IMMediate]
[:AMPLitude]:STEP[:INCRement] <val></td></tr>
<tr><td>SHPM</td><td>27.8 KHz square wave pulse modulation on</td><td>Y</td><td>Y</td><td>[:SOURce]:PULM:SOURce SCALar

[:SOURce]:PULM:STATe ON|1:OUTPut
:MODulation[:STATe] ON|1</td></tr>
<tr><td>SHPS</td><td>Decouple attenuator and ALC (control ALC independently)</td><td>Y</td><td>Y</td><td>[:SOURce]:POWer:ATTenuation:AUTO OFF|0
[:SOURce]:POWer:ALC[:STATe] ON|1
[:SOURce]:POWer:ALC:LEVel <val>dB</td></tr>
<tr><td>SHRC</td><td>Unlock save/recall</td><td>Y</td><td>Y</td><td>supported, but
no equivalent SCPI command sequence</td></tr>
<tr><td>SHRF</td><td>Disable ALC and set power level</td><td>Y</td><td>N</td><td>[:SOURce]:POWer:ALC[:STATe] OFF|0

[:SOURce]:POWer[:LEVel][:IMMediate]
[:AMPLitude] <val><unit></td></tr>
<tr><td>SHRP</td><td>Auto track</td><td>Y</td><td>N</td><td>supported, but has no effect on PSG</td></tr>
<tr><td>SHS10</td><td>Disable display update</td><td>Y</td><td>N</td><td>DISPlay[:WINDow][:STATe] OFF|0</td></tr>
<tr><td>SHS11</td><td>Re-enable display update</td><td>Y</td><td>N</td><td>DISPlay[:WINDow][:STATe] ON|1</td></tr>
<tr><td>SHS3</td><td>Display fault diagnostic</td><td>N</td><td>N</td><td>not supported</td></tr>
<tr><td>SHSL</td><td>Set attenuator from front panel</td><td>Y</td><td>Y</td><td>[:SOURce]:POWer:ATTenuation <val><unit></td></tr>
<tr><td>SHSN</td><td>Stepped sweep</td><td>N</td><td>Y</td><td>[:SOURce]:SWEep:MODE AUTO
[:SOURce]:SWEep:GENeration STEPped
[:SOURce]:LIST:TYPE STEP

[:SOURce]:LIST:TRIGger:SOURce IMMediate
:TRIGger[:SEQuence]:SOURce IMMediate
:INITiate:CONTinuous[:ALL] ON

[:SOURce]:SWEep:POINts <val></td></tr>
<tr><td>SHSS</td><td>Reset step sizes to default values</td><td>N</td><td>Y</td><td>supported, but has no effect on PSG</td></tr>
</table>

Table 6-1 8340B / 41B Prog. Codes & Equivalent SCPI Sequences (Continued)

Cmd	Description	8340	8757	Equivalent SCPI Command Sequence
SHST	Zoom function	N	N	*not supported*
SHSV	Lock save/recall	Y	Y	*supported, but* *no equivalent SCPI command sequence*
SHT1	Test displays	N	N	*not supported*
SHT2	Bandcrossing penlift	N	N	*not supported*
SHT3	Display unlock indicators	N	N	*not supported*
SHGZ	IO Channel	N	N	*not supported*
SHMZ	IO Subchannel	N	N	*not supported*
SHKZ	Write to IO	N	N	*not supported*
SHHZ	Read from IO	N	N	*not supported*
SHVR	Frequency offset	N	N	*not supported*
SL0	Power slope off	Y	Y	[:SOURce]:POWer:SLOPe:STATe OFF|0
SL1	Power slope on	Y	N	[:SOURce]:POWer:SLOPe:STATe ON|1 [:SOURce]:POWer:SLOPe <value> [DB/GHz]
SL1	Power slope on	N	Y	[:SOURce]:POWer:SLOPe:STATe ON|1 [:SOURce]:POWer:SLOPe <value> [DB/Hz]
SM	Manual frequency sweep mode	Y	Y	[:SOURce]:SWEep:MODE MANual [:SOURce]:FREQuency:MANual <val><unit>
SN	Number of points in a stepped sweep	Y	Y	[:SOURce]:SWEep:MODE AUTO [:SOURce]:SWEep:GENeration STEPped [:SOURce]:LIST:TYPE STEP [:SOURce]:LIST:TRIGger:SOURce BUS:TRIGger[:SEQuence]:SOURce IMMediate:INITiate:CONTinuous[:ALL] ON [:SOURce]:SWEep:POINts <val>
✻☆	Power step size	Y	Y	[:SOURce]:POWer[:LEVel][:IMMediate] [:AMPLitude]:STEP[:INCRement] <val>
ST	Sweep time	Y	Y	[:SOURce]:SWEep:MODE AUTO [:SOURce]:SWEep:TIME <val> <unit>
SV	Save state	Y	Y	*SAV <reg_num>[,<seq_num>]

Table 6-1 8340B / 41B Prog. Codes & Equivalent SCPI Sequences (Continued)

Cmd	Description	8340	8757	Equivalent SCPI Command Sequence
SW0	Swap network analyzer channels	Y	Y	*supported, but no equivalent SCPI command sequence*
SW1	Swap network analyzer channels	Y	Y	*supported, but no equivalent SCPI command sequence*
SX	External sweep type	N	Y	*supported, but has no effect on PSG*
T1	Free run sweep trigger mode	Y	Y	`:TRIGger[:SEQuence]:SOURce IMMediate` `:INITiate:CONTinuous[:ALL] ON`
T2	Line sweep trigger mode	N	N	*not supported*
T3	External sweep trigger mode	Y	Y	`:TRIGger[:SEQuence]:SOURce EXTernal` `:INITiate:CONTinuous[:ALL] ON`
T4	Single sweep trigger mode	N	Y	`:INITiate[:IMMediate][:ALL]`
TL	Sweep time limit	Y	Y	`[:SOURce]:SWEep:TIME:LLIMit <val> <unit>`
TS	Take sweep	Y	Y	`:TSWeep`
UP	Step up (increments active function by step value)	Y	Y	*supported, see Table 6-2 on page 234*
VR	CW vernier	N	Y	*supported, but has no effect on PSG*

Table 6-2 8340 and 8757 Code Compatibility

Code	Sets Active Function	Comp. with OA / OP	Comp. with UP / DN	Comp. with RB (Knob)	Equivalent SCPI Commands for OA / OP query and UP / DN command
A2	✓	✓	✓		[:SOURce]:POWer:ALC:SOURce:EXTernal:COUPling? [:SOURce]:POWer:ATTenuation UP [:SOURce]:POWer:ATTenuation DOWN
AT	✓	✓	✓		[:SOURce]:POWer:ATTenuation? [:SOURce]:POWer:ATTenuation UP [:SOURce]:POWer:ATTenuation DOWN
CF	✓	✓			[:SOURce]:FREQuency:CENTer?
CW	✓	✓	✓	✓	[:SOURce]:FREQuency[:CW]? [:SOURce]:FREQuency[:CW] UP [:SOURce]:FREQuency[:CW] DOWN
DF	✓	✓			[:SOURce]:FREQuency:SPAN?
FA	✓	✓			[:SOURce]:FREQuency:STARt?
FB	✓	✓			[:SOURce]:FREQuency:STOP?
FM1	✓	✓			[:SOURce]:FM2[:DEViation]?
MA	✓	✓			[:SOURce]:MARKer0:FREQuency?
M1	✓	✓			[:SOURce]:MARKer1:FREQuency?
M2	✓	✓			[:SOURce]:MARKer2:FREQuency?
M3	✓	✓			[:SOURce]:MARKer3:FREQuency?
M4	✓	✓			[:SOURce]:MARKer4:FREQuency?
M5	✓	✓			[:SOURce]:MARKer5:FREQuency?
M6	✓	✓			[:SOURce]:MARKer6:FREQuency?
M7	✓	✓			[:SOURce]:MARKer7:FREQuency?
M8	✓	✓			[:SOURce]:MARKer8:FREQuency?
M9	✓	✓			[:SOURce]:MARKer9:FREQuency?

Table 6-2 8340 and 8757 Code Compatibility (Continued)

Code	Sets Active Function	Comp. with OA / OP	Comp. with UP / DN	Comp. with RB (Knob)	Equivalent SCPI Commands for OA / OP query and UP / DN command

PL	✓	✓	✓	✓	`[:SOURce]:POWer[:LEVel][:IMMediate][:AMPLitude]?`
					`[:SOURce]:POWer[:LEVel][:IMMediate][:AMPLitude] UP`
					`[:SOURce]:POWer[:LEVel][:IMMediate][:AMPLitude] DOWN`

PS	✓	✓			`[:SOURce]:POWer:SPAN?`
RC	✓				*none*
SB	✓	✓			*supported, but no equivalent SCPI command sequence*
SF	✓	✓		✓	`[:SOURce]:FREQuency[:CW]:STEP[:INCRement]?`

SHA1	✓	✓	✓	✓	`[:SOURce]:POWer[:LEVel][:IMMediate][:AMPLitude]?`
					`[:SOURce]:POWer[:LEVel][:IMMediate][:AMPLitude] UP`
					`[:SOURce]:POWer[:LEVel][:IMMediate][:AMPLitude] DOWN`

SHA2	✓	✓		✓	`[:SOURce]:POWer:ALC:LEVel?`

SHA3	✓	✓	✓	✓	`[:SOURce]:POWer:ALC:LEVel?`
					`[:SOURce]:POWer:ATTenuation UP`
					`[:SOURce]:POWer:ATTenuation DOWN`

SHAZ	✓	✓		✓	`[:SOURce]:POWer:ALC:LEVel?`
SHCF	✓	✓		✓	`[:SOURce]:FREQuency[:CW]:STEP[:INCRement]?`

SHC W	✓	✓			`[:SOURce]:FREQuency:STARt?`
					or
					`[:SOURce]:FREQuency:STOP?`

SHFA	✓	✓		✓	`[:SOURce]:FREQuency:MULTiplier?`
SHFB	✓	✓		✓	`[:SOURce]:FREQuency:OFFSet?`

SHPL	✓	✓	✓	✓	`[:SOURce]:POWer[:LEVel][:IMMediate][:AMPLitude]:STEP [:INCRement]?`
					
					`[:SOURce]:POWer:ATTenuation UP`
					`[:SOURce]:POWer:ATTenuation DOWN`

SHPS	✓	✓	✓	✓	`[:SOURce]:POWer:ALC:LEVel?`
					`[:SOURce]:POWer:ATTenuation UP`
					`[:SOURce]:POWer:ATTenuation DOWN`

Table 6-2 8340 and 8757 Code Compatibility (Continued)

Code	Sets Active Function	Comp. with OA / OP	Comp. with UP / DN	Comp. with RB (Knob)	Equivalent SCPI Commands for OA / OP query and UP / DN command

SHRF	✓	✓	✓	✓	`[:SOURce]:POWer[:LEVel][:IMMediate][:AMPLitude]?`
					`[:SOURce]:POWer[:LEVel][:IMMediate][:AMPLitude] UP`
					`[:SOURce]:POWer[:LEVel][:IMMediate][:AMPLitude] DOWN`

SHSL	✓	✓			`[:SOURce]:POWer:ATTenuation?`
SHSN	✓	✓		✓	`[:SOURce]:SWEep:POINts?`
SL	✓	✓			`[:SOURce]:POWer:SLOPe?`
SM	✓	✓			`[:SOURce]:FREQuency:MANual?`
SN	✓	✓		✓	`[:SOURce]:SWEep:POINts?`
SP	✓	✓		✓	`[:SOURce]:POWer[:LEVel][:IMMediate][:AMPLitude]:STEP [:INCRement]?`
ST	✓	✓			`[:SOURce]:SWEep:TIME?`
SV	✓				*none*
TL	✓	✓			`[:SOURce]:SWEep:TIME:LLIMit?`

836xxB/L Compatible SCPI Commands

Table 6-3 is a comprehensive list of 836xxB/L SCPI commands arranged by subsystem. Commands that are supported by the PSG are identified, in addition to commands that are unsupported. Use the legend within the table to determine command compatibility.

The preset state of the PSG differs from that of the 836xxB/L. The RF output and sweep are turned off in the PSG, while in the 836xxB/L, these parameters are turned on. To optimize the benefit of using 836xxB/L compatible commands with a PSG, set up a user-defined preset state, emulating the preset state of the 836xxB/L.

NOTE Some of the PSG supported commands are a subset of the 836xxB/L commands. When this occurs, the syntax supported by the PSG is shown in addition to the syntax that is not supported.

Table 6-3 836xxB/L SCPI Commands

Y= Supported by PSG **N= Not supported by PSG**	**83620B & 83640B**	**83620L & 83640L**
IEEE Common Commands		
`*CLS`	Y	Y
`*ESE <data>`	Y	Y
`*ESE?`	Y	Y
`*ESR?`	Y	Y
`*IDN?`[a]	Y	Y
`*LRN?`	N	N
`*OPC`	Y	Y
`*OPC?`	Y	Y
`*OPT?`	N	N
`*RCL <reg_num>`	Y	Y

Table 6-3 836xxB/L SCPI Commands

Y= Supported by PSG N= Not supported by PSG	83620B & 83640B	83620L & 83640L
*RST	Y	Y
*SAV <reg_num>	Y	Y
*SRE <data>	Y	Y
*SRE?	Y	Y
*STB?	Y	Y
*TRG	Y	Y
*TST?	Y	Y
*WAI	Y	Y
Abort Subsystem		
:ABORt	Y	Y
Amplitude Modulation Subsystem		
:AM[:DEPTh] <num>[PCT]|MAXimum|MINimum|<num>DB	Y	
:AM[:DEPTh]? [MAXimum|MINimum]	Y	
:AM:INTernal:FREQuency <num>[<freq suffix>]|MAXimum|MINimum	Y	
:AM:INTernal:FREQuency? [MAXimum|MINimum]	Y	
:AM:INTernal:FUNCtion SINusoid|SQUare|TRIangle|RAMP|NOISe	Y	
:AM:INTernal:FUNCtion?	Y	
:AM:SOURce INTernal|EXTernal	Y	
:AM:SOURce?	Y	
:AM:MODE DEEP|NORMal	Y	
:AM:MODE?	Y	

Table 6-3 836xxB / L SCPI Commands

Y= Supported by PSG N= Not supported by PSG	83620B & 83640B	83620L & 83640L
:AM:STATe ON|OFF|1|0	Y	
:AM:STATe?	Y	
:AM:TYPE LINear|EXPonential	Y	
:AM:TYPE?	Y	
Calibration Subsystem		
:CALibration:AM:AUTO ON|OFF|1|0	N	
:CALibration:AM:AUTO?	N	
:CALibration:AM[:EXECute]	N	
:CALibration:PEAKing:AUTO ON|OFF|1|0	N	N
:CALibration:PEAKing:AUTO?	N	N
:CALibration:PEAKing[:EXECute]	N	N
:CALibration:PMETer:DETector:INITiate? IDETector|DIODe	N	N
:CALibration:PMETer:DETector:NEXT? <num>[<lvl suffix>]	N	N
:CALibration:PMETer:FLATness:INITiate? USER|DIODe|PMETER| MMHead	N	N
:CALibration:PMETer:FLATness:NEXT? <value>[<lvl suffix>]	N	N
:CALibration:SPAN:AUTO ON|OFF|1|0	N	N
:CALibration:SPAN:AUTO?	N	N
:CALibration:SPAN[:EXECute]	N	N
:CALibration:TRACk	N	N

Table 6-3 **836xxB/L SCPI Commands**

Y= Supported by PSG **N= Not supported by PSG**	**83620B & 83640B**	**83620L & 83640L**
Correction Subsystem		
`:CORRection:ARRay[i]{<value>[DB]}`	N	N
`:CORRection:ARRay[i]?`	N	N
`:CORRection:FLATness {<num>[freq suffix],<num>[DB]}2*801`	N	N
`:CORRection:FLATness?`	Y	Y
`:CORRection:SOURce[i] ARRay|FLATness`	N	N
`:CORRection:SOURce[i]?`	N	N
`:CORRection:FLATness:POINts? [MAXimum|MINimum]`	Y	Y
`:CORRection[:STATe] ON|OFF|1|0`	Y	Y
`:CORRection[:STATe]?`	Y	Y
Diagnostics Subsystem		
`:DIAGnostics:ABUS? <value>`	N	N
`:DIAGnostics:ABUS:AVERage <value>`	N	N
`:DIAGnostics:ABUS:AVERage?`	N	N
`:DIAGnostics:ABUS:STATus?`	N	N
`:DIAGnostics:INSTrument:PMETer:ADDRess <value>`	N	N
`:DIAGnostics:INSTrument:PMETer:ADDRess?`	N	N
`:DIAGnostics:INSTrument:PRINter:ADDRess <value>`	N	N
`:DIAGnostics:INSTrument:PRINter:ADDRess?`	N	N
`:DIAGnostics:IORW <value>,<value>`	N	N
`:DIAGnostics:IORW? <value>`	N	N

Table 6-3 836xxB / L SCPI Commands

Y= Supported by PSG N= Not supported by PSG	83620B & 83640B	83620L & 83640L
:DIAGnostics:OUTPut:FAULt?	N	N
:DIAGnostics:RESult?	N	N
:DIAGnostics:TEST:CONTinue	N	N
:DIAGnostics:TEST:DATA:DESC?	N	N
:DIAGnostics:TEST:DATA:MAXimum?	N	N
:DIAGnostics:TEST:DATA:MINimum?	N	N
:DIAGnostics:TEST:DATA:VALue?	N	N
:DIAGnostics:TEST:DISable {<num>}1*?|ALL	N	N
:DIAGnostics:TEST:ENABle {<num>}1*?|ALL	N	N
:DIAGnostics:TEST[:EXECute] <value>	N	N
:DIAGnostics:TEST:LOG:SOURce ALL|FAIL	N	N
:DIAGnostics:TEST:LOG:SOURce?	N	N
:DIAGnostics:TEST:LOG[:STATe]?	N	N
:DIAGnostics:TEST:LOG[:STATe] ON|OFF|1|0	N	N
:DIAGnostics:TEST:LOOP ON|OFF|1|0	N	N
:DIAGnostics:TEST:LOOP?	N	N
:DIAGnostics:TEST:NAME? [<value>]	N	N
:DIAGnostics:TEST:POINts?	N	N
:DIAGnostics:TEST:RESult? [<value>]	N	N
:DIAGnostics:TINT? <value>	N	N

Table 6-3 836xxB / L SCPI Commands

Y= Supported by PSG N= Not supported by PSG	83620B & 83640B	83620L & 83640L
Display Subsystem		
:DISPlay[:STATe] ON|OFF|1|0	Y	Y
:DISPlay[:STATe]?	Y	Y
Frequency Modulation Subsystem		
:FM:COUPling AC|DC	Y	
:FM:COUPling?	Y	
:FM[:DEViation] <val><unit>|MAXimum|MINimum	Y	
:FM[:DEViation]? [MAXimum|MINimum]	Y	
:FM:FILTer:HPASs <num>[<freq suffix>]|MAXimum|MINimum	N	
:FM:FILTer:HPASs? [MAXimum|MINimum]	N	
:FM:INTernal:FREQuency <num>[<freq suffix>]|MAXimum|MINimum	Y	
:FM:INTernal:FREQuency? [MAXimum|MINimum]	Y	
:FM:INTernal:FUNCtion SINusoid|SQUare|TRIangle|RAMP|NOISe	Y	
:FM:INTernal:FUNCtion?	Y	
:FM:SOURce INTernal|EXTernal	Y	
:FM:SOURce?	Y	
:FM:SENSitivity <val><freq suffix/V>|MAXimum|MINimum	Y	
:FM:SENSitivity? [MAXimum|MINimum]	Y	
:FM:STATe ON|OFF|1|0	Y	
:FM:STATe?	Y	

Table 6-3 836xxB / L SCPI Commands

Y= Supported by PSG N= Not supported by PSG	83620B & 83640B	83620L & 83640L
Frequency Subsystem		
:FREQuency:CENTer <num>[<freq suffix>]|MAXimum|MINimum|UP|DOWN	Y	Y
:FREQuency:CENTer? [MAXimum|MINimum]	Y	Y
:FREQuency[:CW|:FIXed] <num>[<freq suffix>]|MAXimum|MINimum|UP|DOWN	Y	Y
:FREQuency[:CW]? [MAXimum|MINimum]	Y	Y
:FREQuency[:FIXed]? [MAXimum|MINimum]	Y	Y
:FREQuency[:CW]:AUTO ON|OFF|1|0	N	N
:FREQuency[:CW]:AUTO?	N	N
:FREQuency[:FIXed]:AUTO ON|OFF|1|0	N	N
:FREQuency[:FIXed]:AUTO?	N	N
:FREQuency:MANual <num>[freq suffix]|MAXimum|MINimum|UP|DOWN	N	N
:FREQuency:MANual? [MAXimum|MINimum]	N	N
:FREQuency:MODE FIXed|CW|SWEep|LIST	Y	Y
:FREQuency:MODE?	Y	Y
:FREQuency:MULTiplier <num>|MAXimum|MINimum[b]	Y	Y
:FREQuency:MULTiplier? [MAXimum|MINimum]	Y	Y
:FREQuency:MULTiplier:STATe ON|OFF|1|0	N	N
:FREQuency:MULTiplier:STATe?	N	N
:FREQuency:OFFSet <num>|MAXimum|MINimum	Y	Y

Table 6-3 836xxB / L SCPI Commands

Y= Supported by PSG N= Not supported by PSG	83620B & 83640B	83620L & 83640L
:FREQuency:OFFSet? [MAXimum|MINimum]	Y	Y
:FREQuency:OFFSet:STATe ON|OFF|1|0	Y	Y
:FREQuency:OFFSet:STATe?	Y	Y
:FREQuency:SPAN <num>[<freq suffix>]|MAXimum|MINimum|UP|DOWN	Y	Y
:FREQuency:SPAN? [MAXimum|MINimum]	Y	Y
:FREQuency:STARt <num>[<freq suffix>]|MAXimum|MINimum|UP|DOWN	Y	Y
:FREQuency:STARt? [MAXimum|MINimum]	Y	Y
:FREQuency:STEP:AUTO ON|OFF|1|0	Y	Y
:FREQuency:STEP:AUTO?	Y	Y
:FREQuency:STEP[:INCRement] <num>[<freq suffix>]|MAXimum|MINimum	Y	Y
:FREQuency:STEP[:INCRement]?	Y	Y
:FREQuency:STOP <num>[<freq suffix>]|MAXimum|MINimum|UP|DOWN	Y	Y
:FREQuency:STOP? [MAXimum|MINimum]	Y	Y
Initiate Subsystem		
:INITiate:CONTinuous ON|OFF|1|0	Y	Y
:INITiate:CONTinuous?	Y	Y
:INITiate[:IMMediate]	Y	Y

Table 6-3 836xxB/L SCPI Commands

Y= Supported by PSG N= Not supported by PSG	83620B & 83640B	83620L & 83640L
List Subsystem		
:LIST:DWELl {<num>[<time suffix>]|MAXimum|MINimum}	Y	Y
:LIST:DWELl? [MAXimum|MINimum]	Y	Y
:LIST:DWELl:POINts? [MAXimum|MINimum]	Y	Y
:LIST:FREQuency {<value>[<freq suffix>]|MAXimum|MINimum}	Y	Y
:LIST:FREQuency?	Y	Y
:LIST:FREQuency:POINts? [MAXimum|MINimum]	Y	Y
:LIST:MANual <num>	Y	Y
:LIST:MANual?	Y	Y
:LIST:MODE AUTO|MANual	Y	Y
:LIST:MODE?	Y	Y
:LIST[:POWer]:CORRection {<value>[DB]|MAXimum|MINimum}	N	N
:LIST[:POWer]:CORRection?	N	N
:LIST[:POWer]:CORRection:POINts? [MAXimum|MINimum]	N	N
:LIST:TRIGger:SOURce IMMediate|BUS|EXTernal	Y	Y
:LIST:TRIGger:SOURce?	Y	Y
Marker Subsystem		
:MARKer[n]:AMPLitude[:STATe] ON|OFF|1|0	N	N
:MARKer[n]:AMPLitude[:STATe]?	N	N
:MARKer[n]:AMPLitude:VALue <value>[DB]|MAXimum|MINimum	N	N

Table 6-3 836xxB / L SCPI Commands

Y= Supported by PSG N= Not supported by PSG	83620B & 83640B	83620L & 83640L
:MARKer[n]:AMPLitude:VALue? [MAXimum|MINimum]	N	N
:MARKer[n]:AOFF	N	N
:MARKer[n]:DELTa? <value>,<value>	N	N
:MARKer[n]:FREQuency <value>[<freq suffix>]|MAXimum|MINimum	N	N
:MARKer[n]:FREQuency? [MAXimum|MINimum]	N	N
:MARKer[n]:MODE FREQuency|DELTa	N	N
:MARKer[n]:MODE?	N	N
:MARKer[n]:REFerence <n>	N	N
:MARKer[n]:REFerence?	N	N
:MARKer[n][:STATe] ON|OFF|1|0	N	N
:MARKer[n][:STATe]?	N	N
Measure Subsystem		
:MEASure:AM?	N	
:MEASure:FM?	N	
Modulation Subsystem		
:MODulation:OUTPut:SOURce AM|FM	N	
:MODulation:OUTPut:SOURce?	N	
:MODulation:OUTPut:STATe ON|OFF|1|0	Y	
:MODulation:OUTPut:STATe?	Y	
:MODulation:STATe?	Y	

Table 6-3 836xxB/L SCPI Commands

Y= Supported by PSG N= Not supported by PSG	83620B & 83640B	83620L & 83640L
Power Subsystem		
:POWer:ALC:BANDwidth|:BWIDth <value>[<freq suffix>]|MAXimum|MINimum	Y	Y
:POWer:ALC:BANDwidth?|:BWIDth? [MAXimum|MINimum]	Y	Y
:POWer:ALC:BANDwidth|:BWIDth:AUTO ON|OFF|1|0	Y	Y
:POWer:ALC:BANDwidth|:BWIDth:AUTO?	Y	Y
:POWer:ALC:CFACtor <value>[DB]|MAXimum|MINimum|UP|DOWN	Y	Y
:POWer:ALC:CFACtor? [MINimum|MAXimum]	Y	Y
:POWer:ALC:SOURce PMETer :POWer:ALC:SOURce INTernal|DIODe|MMHead	N Y	N Y
:POWer:ALC:SOURce?	Y	Y
:POWer:ALC[:STATe] ON|OFF|1|0	Y	Y
:POWer:ALC[:STATe]?	Y	Y
:POWer:AMPLifier:STATE ON|OFF|1|0	N	N
:POWer:AMPLifier:STATE?	N	N
:POWer:AMPLifier:STATE:AUTO ON|OFF|1|0	N	N
:POWer:AMPLifier:STATE:AUTO?	N	N
:POWer:ATTenuation <num>[DB]|MAXimum|MINimum|UP|DOWN	Y	Y
:POWer:ATTenuation? [MAXimum|MINimum]	Y	Y
:POWer:ATTenuation:AUTO ON|OFF|1|0	Y	Y
:POWer:ATTenuation:AUTO?	Y	Y

Table 6-3 836xxB / L SCPI Commands

Y= Supported by PSG **N= Not supported by PSG**	**83620B & 83640B**	**83620L & 83640L**
:POWer:CENTer <num>[<lvl suffix>]|MAXimum|MINimum|UP|DOWN	Y	Y
:POWer:CENTer? [MAXimum|MINimum]	Y	Y
:POWer[:LEVel] <num>[<lvl suffix>]|MAXimum|MINimum|UP|DOWN	Y	Y
:POWer[:LEVel]? [MAXimum|MINimum]	Y	Y
:POWer:MODE FIXed|SWEep	Y	Y
:POWer:MODE?	Y	Y
:POWer:OFFSet <num>[DB]|MAXimum|MINimum|UP|DOWN	Y	Y
:POWer:OFFSet? [MAXimum|MINimum]	Y	Y
:POWer:OFFSet:STATe ON|1[c] :POWer:OFFSet:STATe OFF|0[d]	N Y	N Y
:POWer:OFFSet:STATe?	Y	Y
:POWer:RANGe <value>[<lvl suffix>]|MAXimum|MINimum|UP|DOWN	N	N
:POWer:RANGe?	N	N
:POWer:SEARch ON|OFF|1|0|ONCE	Y	Y
:POWer:SEARch?	Y	Y
:POWer:SLOPe <value>[DB/<freq suffix>]|MIN|MAX|UP|DOWN	Y	Y
:POWer:SLOPe? [MAXimum|MINimum]	Y	Y
:POWer:SLOPe:STATe ON|OFF|1|0	Y	Y

Table 6-3 836xxB / L SCPI Commands

Y= Supported by PSG N= Not supported by PSG	83620B & 83640B	83620L & 83640L
`:POWer:SLOPe:STATe?`	Y	Y
`:POWer:SPAN <value>[DB]|MAXimum|MINimum|UP|DOWN`	Y	Y
`:POWer:SPAN? [MAXimum|MINimum]`	Y	Y
`:POWer:STARt <val><unit>|MAXimum|MINimum|UP|DOWN`	Y	Y
`:POWer:STARt? [MAXimum|MINimum]`	Y	Y
`:POWer:STATe ON|OFF|1|0`	Y	Y
`:POWer:STATe?`	Y	Y
`:POWer:STEP:AUTO ON|OFF|1|0`	Y	Y
`:POWer:STEP:AUTO?`	Y	Y
`:POWer:STEP[:INCRement] <num>[DB]|MAXimum|MINimum`	Y	Y
`:POWer:STEP[:INCRement]? [MAXimum|MINimum]`	Y	Y
`:POWer:STOP <val><unit>|MAXimum|MINimum|UP|DOWN`	Y	Y
`:POWer:STOP? [MAXimum|MINimum]`	Y	Y
Pulse Modulation Subsystem		
`:PULM:EXTernal:DELay <value>[<time suffix>]|MAXimum|MINimum`	N	
`:PULM:EXTernal:DELay? [MAXimum|MINimum]`	N	
`:PULM:EXTernal:POLarity NORMal|INVerted`	Y	
`:PULM:EXTernal:POLarity?`	Y	
`:PULM:INTernal:FREQuency <num>[<freq suffix>]|MAXimum|MINimum`	Y	
`:PULM:INTernal:FREQuency? [MAXimum|MINimum]`	Y	

Table 6-3 836xxB / L SCPI Commands

Y= Supported by PSG N= Not supported by PSG	83620B & 83640B	83620L & 83640L
:PULM:INTernal:GATE ON|OFF|1|0	N	
:PULM:INTernal:GATE?	N	
:PULM:INTernal:PERiod <num>[<time suffix>]|MAXimum|MINimum	Y	
:PULM:INTernal:PERiod? [MAXimum|MINimum]	Y	
:PULM:INTernal:TRIGger:SOURce INTernal|EXTernal	Y	
:PULM:INTernal:TRIGger:SOURce? [INTernal|EXTernal]	Y	
:PULM:INTernal:WIDTh <num>[<time suffix>]|MAXimum|MINimum	Y	
:PULM:INTernal:WIDTh? [MAXimum|MINimum]	Y	
:PULM:SLEW <value>[<time suffix>]|MAXimum|MINimum	N	
:PULM:SLEW? [MAXimum|MINimum]	N	
:PULM:SLEW:AUTO ON|OFF|1|0	N	
:PULM:SLEW:AUTO?	N	
:PULM:SOURce SCALar :PULM:SOURce INTernal|EXTernal	N Y	
:PULM:SOURce?	Y	
:PULM:STATe ON|OFF|1|0	Y	
:PULM:STATe?	Y	
Pulse Subsystem		
:PULSe:FREQuency <num>[<freq suffix>]|MAXimum|MINimum	Y	
:PULSe:FREQuency? [MAXimum|MINimum]	Y	
:PULSe:PERiod <num>[<time suffix>]|MAXimum|MINimum	Y	

Table 6-3 836xxB / L SCPI Commands

Y= Supported by PSG N= Not supported by PSG	83620B & 83640B	83620L & 83640L
:PULSe:PERiod? [MAXimum|MINimum]	Y	
:PULSe:WIDTh <num>[<time suffix>]|MAXimum|MINimum	Y	
:PULSe:WIDTh? [MAXimum|MINimum]	Y	
Reference Oscillator Subsystem		
:ROSCillator:SOURce?	Y	Y
:ROSCillator:SOURce:AUTO ON|OFF|1|0	Y	Y
:ROSCillator:SOURce:AUTO?	Y	Y
:ROSCillator:SOURce INTernal|EXTernal|NONE	Y	Y
Status Subsystem		
:STATus:OPERation:CONDition?	Y	Y
:STATus:OPERation:ENABle <value>	Y	Y
:STATus:OPERation:ENABle?	Y	Y
:STATus:OPERation[:EVENt]?	Y	Y
:STATus:OPERation:NTRansition <value>	Y	Y
:STATus:OPERation:NTRansition?	Y	Y
:STATus:OPERation:PTRansition <value>	Y	Y
:STATus:OPERation:PTRansition?	Y	Y
:STATus:PRESet	Y	Y
:STATus:QUEStionable:CONDition?	Y	Y
:STATus:QUEStionable:ENABle <value>	Y	Y
:STATus:QUEStionable:ENABle?	Y	Y
:STATus:QUEStionable[:EVENt]?	Y	Y

Table 6-3 836xxB/L SCPI Commands

Y= Supported by PSG N= Not supported by PSG	83620B & 83640B	83620L & 83640L
:STATus:QUEStionable:NTRansition <value>	Y	Y
:STATus:QUEStionable:NTRansition?	Y	Y
:STATus:QUEStionable:PTRansition <value>	Y	Y
:STATus:QUEStionable:PTRansition?	Y	Y
Sweep Subsystem		
:SWEep:CONTrol:STATe ON|OFF|1|0	N	N
:SWEep:CONTrol:STATe?	N	N
:SWEep:CONTrol:TYPE MASTer|SLAVe	N	N
:SWEep:CONTrol:TYPE?	N	N
:SWEep:DWELl <num>[<time suffix>]|MAXimum|MINimum	Y	Y
:SWEep:DWELl? [MAXimum|MINimum]	Y	Y
:SWEep:DWELl:AUTO ON|OFF|1|0	N	N
:SWEep:DWELl:AUTO?	N	N
:SWEep:GENeration STEPped|ANALog	N	N
:SWEep:GENeration?	N	N
:SWEep:MANual:POINt <num>|MAXimum|MINimum	Y	Y
:SWEep:MANual:POINt? [MAXimum|MINimum]	Y	Y
:SWEep:MANual[:RELative] <value>	N	N
:SWEep:MANual[:RELative]?	N	N
:SWEep:MARKer:STATe ON|OFF|1|0	N	N
:SWEep:MARKer:STATe?	N	N
:SWEep:MARKer:XFER	N	N

Table 6-3 836xxB/L SCPI Commands

Y= Supported by PSG N= Not supported by PSG	83620B & 83640B	83620L & 83640L
:SWEep:MODE AUTO|MANual	Y	Y
:SWEep:MODE?	Y	Y
:SWEep:POINts <num>|MAXimum|MINimum	Y	Y
:SWEep:POINts? [MAXimum|MINimum]	Y	Y
:SWEep:STEP <value>[<freq suffix>]|MAXimum|MINimum	N	N
:SWEep:STEP? [MAXimum|MINimum]	N	N
:SWEep:TIME <value>[<time suffix>]|MAXimum|MINimum	N	N
:SWEep:TIME? [MAXimum|MINimum]	N	N
:SWEep:TIME:AUTO ON|OFF|1|0	N	N
:SWEep:TIME:AUTO?	N	N
:SWEep:TIME:LLIMit <value>[<time suffix>]|MAXimum|MINimum	N	N
:SWEep:TIME:LLIMit? [MAXimum|MINimum]	N	N
:SWEep:TRIGger:SOURce IMMediate|BUS|EXTernal	Y	Y
:SWEep:TRIGger:SOURce?	Y	Y
System Subsystem		
:SYSTem:ALTernate <value>|MAXimum|MINimum	N	N
:SYSTem:ALTernate? [MAXimum|MINimum]	N	N
:SYSTem:ALTernate:STATe ON|OFF|1|0	N	N
:SYSTem:ALTernate:STATe?	N	N
:SYSTem:COMMunicate:GPIB:ADDRess <number>	Y	Y
:SYSTem:DUMP:PRINter?	N	N

Table 6-3 836xxB/L SCPI Commands

<table>
<tr><th>Y= Supported by PSG
N= Not supported by PSG</th><th>83620B & 83640B</th><th>83620L & 83640L</th></tr>
<tr><td>:SYSTem:ERRor?</td><td>Y</td><td>Y</td></tr>
<tr><td>:SYSTem:LANGuage CIIL|COMPatible
:SYSTem:LANGuage SCPI</td><td>N
Y</td><td>N
Y</td></tr>
<tr><td>:SYSTem:MMHead:SELect:AUTO ON|OFF|1|0</td><td>Y</td><td>Y</td></tr>
<tr><td>:SYSTem:MMHead:SELect:AUTO?</td><td>Y</td><td>Y</td></tr>
<tr><td>:SYSTem:MMHead:SELect FRONt|REAR|NONE[e]</td><td>Y</td><td>Y</td></tr>
<tr><td>:SYSTem:MMHead:SELect?</td><td>Y</td><td>Y</td></tr>
<tr><td>:SYSTem:PRESet[:EXECute]</td><td>Y</td><td>Y</td></tr>
<tr><td>:SYSTem:PRESet:SAVE</td><td>Y</td><td>Y</td></tr>
<tr><td>:SYSTem:PRESet:TYPE FACTory|USER</td><td>Y</td><td>Y</td></tr>
<tr><td>:SYSTem:PRESet:TYPE?</td><td>Y</td><td>Y</td></tr>
<tr><td>:SYSTem:SECurity:COUNt <value>[fg]</td><td>Y</td><td>Y</td></tr>
<tr><td>:SYSTem:SECurity:COUNt? [MINimum|MAXimum]</td><td>Y</td><td>Y</td></tr>
<tr><td>:SYSTem:SECurity[:STATe] ON|OFF|1|0[e]</td><td>Y</td><td>Y</td></tr>
<tr><td>:SYSTem:SECurity[:STATe]?</td><td>Y</td><td>Y</td></tr>
<tr><td>:SYSTem:VERSion?</td><td>Y</td><td>Y</td></tr>
<tr><td colspan="3">Trigger Subsystem</td></tr>
<tr><td>:TRIGger[:IMMediate]</td><td>Y</td><td>Y</td></tr>
<tr><td>:TRIGger:ODELay <value>[time suffix]|MAXimum|MINimum</td><td>N</td><td>N</td></tr>
<tr><td>:TRIGger:ODELay? [MAXimum|MINimum]</td><td>N</td><td>N</td></tr>
<tr><td>:TRIGger:SOURce IMMediate|BUS|EXTernal</td><td>Y</td><td>Y</td></tr>
<tr><td>:TRIGger:SOURce?</td><td>Y</td><td>Y</td></tr>
</table>

Table 6-3 836xxB/L SCPI Commands

Y= Supported by PSG N= Not supported by PSG	83620B & 83640B	83620L & 83640L
Tsweep Subsystem		
`:TSWeep`	N	N
Unit Subsystem		
`:UNIT:AM DB|PCT`	N	
`:UNIT:AM?`	N	
`:UNIT:POWer {<lvl suffix>}`	Y	Y
`:UNIT:POWer?`	Y	Y

a. The identification information can be modified for the PSG to reflect the signal generator that is being replaced. Refer to ":SYSTem:IDN" on page 282 for more information.
b. A multiplier of zero is not allowed.
c. The PSG will accept this command, but it has no effect.
d. This command resets the power offset level to 0dBm. It does not turn off or disable the power offset feature.
e. Since the PSG does not have a front panel millimeter head (source module) interface connector, the "FRONT" suffix defaults to the rear connector.
f. Flash memory allows only a limited number of "writes and erasures", excessive use of this command will reduce the memory lifetime.
g. This command can take several hours to execute because the PSG memory size is much larger than the HP 836xx memory.

8373xB and 8371xB Compatible SCPI Commands

Table 6-4 is a comprehensive list of 8373xB and 8371xB SCPI commands arranged by subsystem. Commands that are supported by the PSG are identified, in addition to commands that are unsupported. Use the legend within the table to determine command compatibility.

NOTE Some of the PSG supported commands are subsets of the 8373xB and 8371xB commands. When this occurs, the syntax supported by the PSG is shown in addition to the syntax that is not supported.

Table 6-4 8373xB and 8371xB SCPI Commands

Y= Supported by PSG N= Not supported by PSG	83731B & 83732B	83711B & 83712B
IEEE Common Commands		
`*CLS`	Y	Y
`*DMC`	N	N
`*EMC`	N	N
`*EMC?`	N	N
`*ESE <data>`	Y	Y
`*ESE?`	Y	Y
`*ESR?`	Y	Y
`*GMC?`	N	N
`*IDN?`[a]	Y	Y
`*LMC?`	N	N
`*LRN?`	N	N
`*OPC`	Y	Y
`*OPC?`	Y	Y
`*OPT?`	N	N

Table 6-4 8373xB and 8371xB SCPI Commands

Y= Supported by PSG N= Not supported by PSG	83731B & 83732B	83711B & 83712B
`*PMC`	N	N
`*PSC`	Y	Y
`*PSC?`	Y	Y
`*RCL <reg_num>`	Y	Y
`*RMC`	N	N
`*RST`	Y	Y
`*SAV <reg_num>`	Y	Y
`*SRE <data>`	Y	Y
`*SRE?`	Y	Y
`*STB?`	Y	Y
`*TST?`	Y	Y
`*WAI`	Y	Y
Abort Subsystem		
`:ABORt`	Y	
Amplitude Modulation Subsystem		
`[:SOURce]:AM[:DEPTh] <val><unit>`[b]	Y	
`[:SOURce]:AM[:DEPTh] <num>[<PCT>]|<num>DB`	Y	
`[:SOURce]:AM[:DEPTh]:STEP[:INCRement] incr|MINimum|MAXimum|DEFault`	Y	
`[:SOURce]:AM:INTernal:FREQuency <num>[<freq suffix>] incr|MINimum|MAXimum|DEFault`	Y	
`[:SOURce]:AM:INTernal:FREQuency:STEP[:INCRement]`	Y	
`[:SOURce]:AM:INTernal:FUNCtion SINusoid|SQUare|TRIangle|RAMP|NOISe|UNIForm|GAUSsian`	Y	
`[:SOURce]:AM:SENSitivity <val>|MIN|MAX|DEF`	N	

Table 6-4 8373xB and 8371xB SCPI Commands

Y= Supported by PSG N= Not supported by PSG	83731B & 83732B	83711B & 83712B
[:SOURce]:AM:SOURce FEED [:SOURce]:AM:SOURce INTernal|EXTernal	N Y	
[:SOURce]:AM:SOURce?	Y	
[:SOURce]:AM:STATe ON|OFF	Y	
[:SOURce]:AM:STATe?	Y	
[:SOURce]:AM:TYPE LINear|EXPonential	Y	
[:SOURce]:AM:TYPE?	Y	
Display Subsystem		
:DISPlay[:WINDow][:STATe] ON|OFF|1|0	Y	Y
:DISPlay[:WINDow][:STATe]?	Y	Y
Initiate Subsystem		
:INITiate:CONTinuous ON|OFF|1|0	Y	
:INITiate:CONTinuous?	Y	
Correction Subsystem		
[:SOURce]:CORRection:FLATness[:DATA] <freq>,<corr.>,... <freq>,<corr.>	Y	Y
[:SOURce]:CORRection:FLATness:POINts <points>	Y	Y
[:SOURce]:CORRection[:STATe] ON|OFF	Y	Y
[:SOURce]:CORRection[:STATe]?	Y	Y
[:SOURce]:CORRection:CSET[:SELect] tableno	N	N
[:SOURce]:CORRection:CSET[:SELect]?	N	N
[:SOURce]:CORRection:CSET:STATe ON|OFF|1|0	N	N
[:SOURce]:CORRection:CSET:STATe?	N	N
Frequency Modulation Subsystem		
[:SOURce]:FM:COUPling AC|DC	Y	

Table 6-4 8373xB and 8371xB SCPI Commands

Y= Supported by PSG N= Not supported by PSG	83731B & 83732B	83711B & 83712B
[:SOURce]:FM:COUPling?	Y	
[:SOURce]:FM[:DEViation] <val><unit>	Y	
[:SOURce]:FM[:DEViation]:STEP[:INCRement] <val> [<freq suffix>]	Y	
[:SOURce]:FM:INTernal:FREQuency <num>[<freq suffix>]	Y	
[:SOURce]:FM:INTernal:FREQuency:STEP[:INCRement] incr| MINimum|MAXimum|DEFault	N	
[:SOURce]:FM:INTernal:FUNCtion SINusoid|SQUAre|TRIAngle| RAMP|UNIForm|GAUSsian	N	
[:SOURce]:FM:SENSitivity?	Y	
[:SOURce]:FM:SOURce FEED [:SOURce]:FM:SOURce INTernal|EXTernal	N Y	
[:SOURce]:FM:STATe ON|OFF|1|0	Y	
[:SOURce]:FM:STATe?	Y	
Frequency Subsystem		
[:SOURce]:FREQuency[:CW|:FIXed] <num>[<freq suffix>]|UP| DOWN|DEFault	Y	Y
[:SOURce]:FREQuency[:CW|:FIXed] [MAXimum|MINimum|DEFault]	Y	Y
[:SOURce]:FREQuency[:CW|:FIXed]:STEP <val><unit>	Y	Y
[:SOURce]:FREQuency[:CW|:FIXed]:STEP?	Y	Y
[:SOURce]:FREQuency:MULTiplier <val>|UP|DOWN|DEFault[c]	Y	Y
[:SOURce]:FREQuency:MULTiplier?	Y	Y
[:SOURce]:FREQuency:MULTiplier:STEP[:INCRement] incr| MINimum|MAXimum|DEFault	N	N
[:SOURce]:FREQuency:MULTiplier:STEP[:INCRement]?	N	N
Memory Subsystem		

Table 6-4 8373xB and 8371xB SCPI Commands

Y= Supported by PSG N= Not supported by PSG	83731B & 83732B	83711B & 83712B
:MEMory:CATalog[:ALL]?	Y	Y
:MEMory:CATalog:TABLe?	N	N
:MEMory:CATalog:MACRo	N	N
:MEMory:RAM:INITialize	N	N
:MEMory:TABLe:FREQuency freq,...freq|MINimum|MAXimum	N	N
:MEMory:TABLe:FREQuency? MINimum|MAXimum	N	N
:MEMory:TABLe:FREQuency:POINts?	N	N
:MEMory:TABLe:LOSS[:MAGNitude] cf,...cf|MINimum|MAXimum	N	N
:MEMory:TABLe:LOSS[:MAGNitude]?	N	N
:MEMory:TABLe:LOSS[:MAGNitude]:POINts?	N	N
:MEMory:TABLe:SELect tableno	N	N
:MEMory:TABLe:SELect?	N	N
Modulation Subsystem		
[:SOURce]:MODulation:AOFF	Y	
[:SOURce]:MODulation:STATe ON|OFF	N	
[:SOURce]:MODulation:STATe?	Y	
Output Subsystem		
:OUTPut:IMPedance?	N	N
:OUTPut:PROTection[:STATe] ON|OFF	Y	Y
:OUTPut:PROTection[:STATe]?	Y	Y
:OUTPut[:STATe] ON|OFF|1|0	Y	Y
:OUTPut[:STATe]?	Y	Y
Phase Modulation Subsystem		
[:SOURce]:PM:COUPling AC|DC	Y	

Table 6-4 8373xB and 8371xB SCPI Commands

Y= Supported by PSG N= Not supported by PSG	83731B & 83732B	83711B & 83712B
[:SOURce]:PM[:DEViation] <val><unit>	Y	
[:SOURce]:PM[:DEViation]:STEP[:INCRement]	Y	
[:SOURce]:PM:INTernal:FREQuency <val><unit>	Y	
[:SOURce]:PM:INTernal:FREQuency:STEP[:INCRement]	Y	
[:SOURce]:PM:INTernal:FUNCtion SINusoid|SQUare|TRIAngle| RAMP|UNIForm|GAUSsian	Y	
[:SOURce]:PM:RANGe AUTO|LOW|HIGH	Y	
[:SOURce]:PM:SENSitivity sens|MINimum|MAXimum|DEFault	N	
[:SOURce]:PM:SOURce INTernal|FEED|EXTernal[d]	Y	
[:SOURce]:PM:STATe ON|OFF|1|0	Y	
Power Subsystem		
[:SOURce]:POWer:ALC:PMETer pmeter|MINimum|MAXimum|DEFault	N	N
[:SOURce]:POWer:ALC:PMETer?	N	N
[:SOURce]:POWer:ALC:PMETer:STEP incr|MINimum|MAXimum| DEFault	N	N
[:SOURce]:POWer:ALC:PMETer:STEP?	N	N
[:SOURce]:POWer:ALC:SOURce PMETer [:SOURce]:POWer:ALC:SOURce INTernal|DIODe	N Y	N Y
[:SOURce]:POWer:ALC:SOURce?	Y	Y
[:SOURce]:POWer:ATTenuation:AUTO ONCE [:SOURce]:POWer:ATTenuation:AUTO ON|OFF	N Y	N Y
[:SOURce]:POWer:ATTenuation:AUTO?	Y	Y
[:SOURce]:POWer[:LEVel] ampl|MINimum|MAXimum|UP|DOWN| DEFault	Y	Y
[:SOURce]:POWer[:LEVel]?	Y	Y
[:SOURce]:POWer[:LEVel]:STEP incr|MINimum|MAXimum|DEFault	Y	Y

Table 6-4 8373xB and 8371xB SCPI Commands

Y= Supported by PSG N= Not supported by PSG	83731B & 83732B	83711B & 83712B
[:SOURce]:POWer[:LEVel]:STEP?	Y	Y
[:SOURce]:POWer:PROTection:STATe ON|OFF	Y	Y
[:SOURce]:POWer:PROTection:STATe?	Y	Y
Pulse Modulation Subsystem		
[:SOURce]:PULM:EXTernal:POLarity NORMal|INVerted	Y	
[:SOURce]:PULM:EXTernal:POLarity?	Y	
[:SOURce]:PULM:SOURce INTernal|EXTernal	Y	
[:SOURce]:PULM:SOURce?	Y	
[:SOURce]:PULM:STATe ON|OFF|1|0	Y	
[:SOURce]:PULM:STATe?	Y	
Pulse Subsystem		
[:SOURce]:PULSe:DELay delay|MINimum|MAXimum|UP|DOWN| DEFault	Y	
[:SOURce]:PULSe:DELay?	Y	
[:SOURce]:PULSe:DELay:STEP <num>[<time suffix>][DEFault]	Y	
[:SOURce]:PULSe:DELay:STEP? [DEFault]	Y	
[:SOURce]:PULSe:DOUBle[:STATE] ON|OFF	N	
[:SOURce]:PULSe:DOUBle[:STATE]?	N	
[:SOURce]:PULSe:FREQuency freq|MINimum|MAXimum|UP|DOWN| DEFault	Y	
[:SOURce]:PULSe:FREQuency?	Y	
[:SOURce]:PULSe:FREQuency:STEP freq|DEFault	Y	
[:SOURce]:PULSe:FREQuency:STEP? [MIN|MAX|DEF]	Y	
[:SOURce]:PULSe:PERiod <num>[<time suffix>]|UP|DOWN	Y	
[:SOURce]:PULSe:PERiod?	Y	

Table 6-4 8373xB and 8371xB SCPI Commands

Y= Supported by PSG N= Not supported by PSG	83731B & 83732B	83711B & 83712B
[:SOURce]:PULSe:PERiod:STEP <num>[<time suffix>]	Y	
[:SOURce]:PULSe:PERiod:STEP?	Y	
[:SOURce]:PULse:TRANSition[:LEADing] SLOW|MEDium|FAST	N	
[:SOURce]:PULSe:TRANsition[:LEADing]?	N	
[:SOURce]:PULSe:TRANsition:STATe ON|OFF	N	
[:SOURce]:PULSe:TRANsition:STATe?	N	
[:SOURce]:PULSe:WIDTh MAXimum|MINimum|UP|DOWN|DEFault	Y	
[:SOURce]:PULSe:WIDTh? [MAXimum|MINimum|DEFault]	Y	
[:SOURce]:PULSe:WIDTh:STEP <num>[<time suffix>]|DEFault	Y	
[:SOURce]:PULSe:WIDTh:STEP? [MINimum|MAXimum|DEFault]	Y	
Reference Oscillator Subsystem		
[:SOURce]:ROSCillator:SOURce?	Y	Y
Status Subsystem		
:STATus:OPERation:CONDition?	Y	Y
:STATus:OPERation:ENABle <value>	Y	Y
:STATus:OPERation:ENABle?	Y	Y
:STATus:OPERation[:EVENt]?	Y	Y
:STATus:OPERation:NTRansition <value>	Y	Y
:STATus:OPERation:NTRansition?	Y	Y
:STATus:OPERation:PTRansition <value>	Y	Y
:STATus:OPERation:PTRansition?	Y	Y
:STATus:PRESet	Y	Y
:STATus:QUEStionable:CONDition?	Y	Y
:STATus:QUEStionable:ENABle <value>	Y	Y

Table 6-4 8373xB and 8371xB SCPI Commands

Y= Supported by PSG N= Not supported by PSG	83731B & 83732B	83711B & 83712B
:STATus:QUEStionable:ENABle?	Y	Y
:STATus:QUEStionable[:EVENt]?	Y	Y
:STATus:QUEStionable:NTRansition <value>	Y	Y
:STATus:QUEStionable:NTRansition?	Y	Y
:STATus:QUEStionable:PTRansition <value>	Y	Y
:STATus:QUEStionable:PTRansition?	Y	Y
System Subsystem		
:SYSTem:COMMunicate:GPIB:ADDRess <number>	Y	Y
:SYSTem:COMMunicate:GPIB:ADDRess?	Y	Y
:SYSTem:COMMunicate:PMETer:ADDRess	Y	Y
:SYSTem:COMMunicate:PMETer:ADDRess?	Y	Y
:SYSTem:ERRor?	Y	Y
:SYSTem:KEY keycode|MINimum|MAXimum	N	N
:SYSTem:KEY?	N	N
:SYSTem:LANGuage "COMP=8673"|"COMPatibility=8673" :SYSTem:LANGuage "SCPI"	N Y	N Y
:SYSTem:LANGuage?	Y	Y
:SYSTem:PRESet	Y	Y
:SYSTem:VERSion?	Y	Y
Trigger Subsystem		
:TRIGger[:SEQuence|:STARt]:SOURce IMMediate|EXTernal	N	
:TRIGger[:SEQuence|:STARt]:SOURce?	N	
:TRIGger:SEQuence2:STOP:SOURce IMMediate|EXTernal	N	
:TRIGger:SEQuence2:STOP:SOURce?	N	

Table 6-4 8373xB and 8371xB SCPI Commands

Y= Supported by PSG **N= Not supported by PSG**	**83731B & 83732B**	**83711B & 83712B**
:TRIGger:SEQuence2:SLOPe	N	
Unit Subsystem		
:UNIT:FREQuency {<freq suffix>}	Y	Y
:UNIT:FREQuency?	Y	Y
:UNIT:POWer {<lvl suffix>}	Y	Y
:UNIT:POWer?	Y	Y
:UNIT:TIME	N	N
:UNIT:TIME?	N	N
:UNIT:VOLTage {<lvl suffix>}	N	N
:UNIT:VOLTage?	N	N

a. The identification information can be modified for the PSG to reflect the signal generator that is being replaced. Refer to ":SYSTem:IDN" on page 282 for more information.
b. In linear mode, % cannot be used to select percent as the unit. Use PCT to specify percent as the unit.
c. A multiplier of zero is not allowed.
d. If FEED is selected, the query returns INT. FEED and INTernal are synonymous.

8375xB Compatible SCPI Commands

Table 6-5 is a comprehensive list of 83751B and 83752B SCPI commands, arranged by subsystem. Commands that are supported by the PSG are identified, in addition to commands that are unsupported. Use the legend within the table to determine command compatibility.

NOTE Some supported commands require the installation of hardware or firmware options.

Table 6-5 8375xB SCPI Commands

Y= Supported by PSG **N= Not supported by PSG**	**83751B & 83752B**
IEEE Common Commands	
`*CLS`	Y
`*DMC`	N
`*EMC`	N
`*EMC?`	N
`*ESE <value>`	Y
`*ESE?`	Y
`*ESR?`	Y
`*GMC? <label>`	N
`*IDN?`	Y
`*LMC?`	N
`*LRN?`	N
`*OPC`	Y
`*OPC?`	Y
`*OPT?`	N

Table 6-5 8375xB SCPI Commands (Continued)

Y= Supported by PSG N= Not supported by PSG	83751B & 83752B
`*PMC`	N
`*PSC ON|OFF|1|0`	Y
`*PSC?`	Y
`*RCL <reg_num>`	Y
`*RMC <label>`	N
`*RST`	Y
`*SAV <reg_num>`	Y
`*SRE <value>`	Y
`*SRE?`	Y
`*STB?`	Y
`*TRG`	Y
`*TST?`	Y
`*WAI`	Y
Abort Subsystem	
`:ABORt`	Y
Amplitude Modulation Subsystem	
`:AM:SOURce1 INTernal|EXTernal` `:AM:SOURce INTernal|EXTernal`	N Y
`:AM:SOURce1?` `:AM:SOURce?`	N Y
`:AM:STATe ON|OFF|1|0`	Y
`:AM:STATe?`	Y
Calibration Subsystem	

Table 6-5 8375xB SCPI Commands (Continued)

Y= Supported by PSG N= Not supported by PSG	83751B & 83752B
:CALibration:PEAKing[:EXECute]	N
:CALibration:PEAKing[:EXECute]? <dac_va>	N
:CALibration:PMETer:FLATness:INITiate? USER	N
:CALibration:PMETer:FLATness:NEXT? <value>[<lvlsuffix>]	N
:CALibration:SECurity:CODE <old> <new>	N
:CALibration:SECurity:PASSword <passwd>	N
:CALibration:TRACk	N
Correction Subsystem	
:CORRection:FLATness:AMPL <value>[DB],<value>[DB]...	N
:CORRection:FLATness:AMPL?	N
:CORRection:FLATness:FREQ <value>[<freqsuffix>],<value>[<freqsuffix>]...	N
:CORRection:FLATness:FREQ?	N
:CORRection:FLATness:POINts? MAXimum|MINimum	N
:CORRection:VOLTs:OFFSet	N
:CORRection:VOLTs:OFFSet?	N
:CORRection:VOLTs:SCALe	N
:CORRection:VOLTs:SCALe?	N
:CORRection[:STATe] ON|OFF|1|0	Y
:CORRection[:STATe]?	Y
Diagnostics Subsystem	
:DIAG:LRNS?	N
:DIAGnostic:TEST:FULLtest:REPort?	N

Table 6-5 8375xB SCPI Commands (Continued)

Y= Supported by PSG N= Not supported by PSG	83751B & 83752B
:DIAGnostic:TEST:FULLtest?	N

Display Subsystem	
:DISPlay[:STATe] ON|OFF|1|0	Y
:DISPlay[:STATe]?	Y
Frequency Modulation Subsystem	
:FM:COUPling AC|DC	Y
:FM:COUPling?	Y
:FM:SENSitivity <value><freqsuffix/V>	Y
:FM:SENSitivity?	Y
:FM:SOURce1 EXTernal :FM:SOURce EXTernal	N
:FM:SOURce1? :FM:SOURce?	N Y
:FM:STATe ON|OFF|1|0	Y
:FM:STATe?	Y
Frequency Subsystem	
:FREQuency:CENTer <value>[<freqsuffix>]|UP|DOWN	Y
:FREQuency:CENTer?	Y
:FREQuency:MANual <value><unit>|UP|DOWN	N
[:SOURce[1]]:FREQuency:MANual? [:SOURce]:FREQuency:MANual?	N Y
:FREQuency:MODE FIXed|CW|SWEep|SWCW	N
:FREQuency:MODE?	Y

Table 6-5 8375xB SCPI Commands (Continued)

<table>
<tr><th>Y= Supported by PSG
N= Not supported by PSG</th><th>83751B & 83752B</th></tr>
<tr><td>:FREQuency:MULTiplier <value></td><td>Y</td></tr>
<tr><td>:FREQuency:MULTiplier:STATe ON|OFF|1|0</td><td>N</td></tr>
<tr><td>:FREQuency:MULTiplier:STATe?</td><td>N</td></tr>
<tr><td>:FREQuency:MULTiplier?</td><td>Y</td></tr>
<tr><td>:FREQuency:OFFSet <value></td><td>Y</td></tr>
<tr><td>:FREQuency:OFFSet:STATe ON|OFF|1|0</td><td>Y</td></tr>
<tr><td>:FREQuency:OFFSet:STATe?</td><td>Y</td></tr>
<tr><td>:FREQuency:OFFSet?</td><td>Y</td></tr>
<tr><td>:FREQuency:SPAN <value>[<freqsuffix>]|UP|DOWN</td><td>Y</td></tr>
<tr><td>:FREQuency:SPAN?</td><td>Y</td></tr>
<tr><td>:FREQuency:STARt <value>[<freqsuffix>]|UP|DOWN</td><td>Y</td></tr>
<tr><td>:FREQuency:STARt?</td><td>Y</td></tr>
<tr><td>:FREQuency:STEP[:INCRement] <value>[<freqsuffix>]</td><td>Y</td></tr>
<tr><td>:FREQuency:STEP[:INCRement]?</td><td>Y</td></tr>
<tr><td>:FREQuency:STOP <value>[<freqsuffix>]|UP|DOWN</td><td>Y</td></tr>
<tr><td>:FREQuency:STOP?</td><td>Y</td></tr>
<tr><td>:FREQuency[:CW|:FIXed] <value>[<freqsuffix>]|UP|DOWN</td><td>Y</td></tr>
<tr><td>:FREQuency[:CW|:FIXed]:AUTO ON|OFF|1|0</td><td>N</td></tr>
<tr><td>:FREQuency[:CW|:FIXed]:AUTO?</td><td>N</td></tr>
<tr><td>:FREQuency[:CW|:FIXed]?</td><td>Y</td></tr>
<tr><td colspan="2">Initiate Subsystem</td></tr>
<tr><td>:INITiate:CONTinuous ON|OFF|1|0</td><td>Y</td></tr>
</table>

Table 6-5 8375xB SCPI Commands (Continued)

Y= Supported by PSG N= Not supported by PSG	83751B & 83752B
:INITiate:CONTinuous?	Y
:INITiate[:IMMediate]	Y
Marker Subsystem	
[:SOURce[1]]:MARKer[n]:AMPLitude[:STATe] ON|OFF|1|0 [:SOURce]:MARKer[n]:AMPLitude[:STATe] ON|OFF|1|0	N Y
[:SOURce[1]]:MARKer[n]:AMPLitude[:STATe]? [:SOURce]:MARKer[n]:AMPLitude[:STATe]?	N Y
:MARKer[n]:AOFF	Y
:MARKer[n]:FREQuency <value><unit>	Y
:MARKer[n]:FREQuency?	N
:MARKer[n]:MODE FREQuency|DELTa	Y
:MARKer[n]:MODE?	Y
:MARKer[n]:REFerence <n>	Y
:MARKer[n]:REFerence?	Y
:MARKer[n][:STATe] ON|OFF|1|0	N
:MARKer[n][:STATe]?	N
Memory Subsystem	
:MEMory:RAM:INITialize[:ALL]	N
Output Subsystem	
:OUTPut:IMPedance?	N
:OUTPut[:STATe] ON|OFF|1|0	Y
:OUTPut[:STATe]?	Y

Table 6-5 8375xB SCPI Commands (Continued)

Y= Supported by PSG **N= Not supported by PSG**	**83751B & 83752B**
Power Subsystem	
:POWer:ALC:CFACtor <value>[DB]|UP|DOWN	Y
:POWer:ALC:CFACtor?	Y
:POWer:ALC:SOURce1 INTernal|DIODe|PMETer|MMHead :POWer:ALC:SOURce INTernal|DIODe|PMETer|MMHead	N
:POWer:ALC:SOURce1? :POWer:ALC:SOURce?	N Y
:POWer:ALC[:STATe] ON|OFF|1|0	Y
:POWer:ALC[:STATe]?	Y
:POWer:ATTenuation <value>[DB]|UP|DOWN	Y
:POWer:ATTenuation:AUTO ON|OFF|1|0	Y
:POWer:ATTenuation:AUTO?	Y
:POWer:ATTenuation?	Y
:POWer:CENTer <value>[<lvlsuffix>]|UP|DOWN	Y
:POWer:CENTer?	Y
:POWer:MODE FIXed|SWEep	Y
:POWer:MODE?	Y
:POWer:OFFSet <value>[DB]|UP|DOWN	Y
:POWer:OFFSet:STATe ON|OFF|1|0	Y
:POWer:OFFSet:STATe?	Y
:POWer:OFFSet?	Y

Table 6-5 8375xB SCPI Commands (Continued)

Y= Supported by PSG N= Not supported by PSG	83751B & 83752B
:POWer:SLOPe <value>[DB/freqsuffix]|UP|DOWN	N
:POWer:SLOPe:STATe ON|OFF|1|0	N
:POWer:SLOPe:STATe?	N
:POWer:SLOPe?	Y
:POWer:SPAN <value>[DB]|UP|DOWN	Y
:POWer:SPAN?	Y
:POWer:STARt <value>[<lvlsuffix>]|UP|DOWN	Y
:POWer:STARt?	Y
:POWer:STATe ON|OFF|1|0	Y
:POWer:STATe?	Y
:POWer:STEP[:INCRement] <value>[DB]	Y
:POWer:STEP[:INCRement]?	Y
:POWer:STOP <value>[<lvlsuffix>]|UP|DOWN	Y
:POWer:STOP?	Y
:POWer[:LEVel] <value>[<lvlsuffix>]|UP|DOWN	Y
:POWer[:LEVel]?	Y
Pulse Modulation Subsystem	
:PULM:SOURce1 INTernal|EXTernal|SCALar|SQ1K :PULM:SOURce INTernal|EXTernal|SCALar|SQ1K	N
:PULM:SOURce1? :PULM:SOURce?	N Y
:PULM:STATe ON|OFF|1|0	Y
:PULM:STATe?	Y

Table 6-5 8375xB SCPI Commands (Continued)

Y= Supported by PSG N= Not supported by PSG	83751B & 83752B
Pulse Subsystem	
:PULSe:FREQuency <value>[<freqsuffix>]	Y
:PULSe:FREQuency?	Y
:PULSe:PERiod <value>[<timesuffix>]	Y
:PULSe:PERiod?	Y
:PULSe:WIDTh <value>[<timesuffix>]	Y
:PULSe:WIDTh?	Y
Reference Oscillator Subsystem	
:ROSCillator:SOURce1 INTernal|EXTernal|NONE :ROSCillator:SOURce INTernal|EXTernal|NONE	N Y
:ROSCillator:SOURce1:AUTO ON|OFF|1|0 :ROSCillator:SOURce:AUTO ON|OFF|1|0	N Y
:ROSCillator:SOURce1:AUTO? :ROSCillator:SOURce:AUTO?	N Y
:ROSCillator:SOURce1? :ROSCillator:SOURce?	N Y
Status Subsystem	
:STATus:OPERation:CONDition?	Y
:STATus:OPERation:ENABle <value>	Y
:STATus:OPERation:ENABle?	Y
:STATus:OPERation:NTRansition <value>	Y
:STATus:OPERation:NTRansition?	Y

Table 6-5 8375xB SCPI Commands (Continued)

Y= Supported by PSG N= Not supported by PSG	83751B & 83752B
:STATus:OPERation:PTRansition <value>	Y
:STATus:OPERation:PTRansition?	Y
:STATus:OPERation[:EVENt]?	Y
:STATus:PRESet	Y
:STATus:QUEStionable:CONDition?	Y
:STATus:QUEStionable:ENABle <value>	Y
:STATus:QUEStionable:ENABle?	Y
:STATus:QUEStionable:NTRansition <value>	Y
:STATus:QUEStionable:NTRansition?	Y
:STATus:QUEStionable:PTRansition <value>	Y
:STATus:QUEStionable:PTRansition?	Y
:STATus:QUEStionable[:EVENt]?	Y
Sweep Subsystem	
:SWEep:CONTrol:TYPE MASTer|SLAVe	Y
:SWEep:CONTrol:TYPE?	Y
:SWEep:DWELl <value>[<timesuffix>]	Y
:SWEep:DWELl:AUTO ON|OFF|1|0	N
:SWEep:DWELl:AUTO?	N
:SWEep:DWELl?	Y
:SWEep:GENeration ANALog|STEPped	Y
:SWEep:GENeration?	Y
:SWEep:MANual:POINt <value>	Y

Table 6-5 8375xB SCPI Commands (Continued)

Y= Supported by PSG N= Not supported by PSG	83751B & 83752B
:SWEep:MANual:POINt?	Y
:SWEep:MANual[:RELative] <value>	N
:SWEep:MANual[:RELative]?	N
:SWEep:MARKer:STATe ON|OFF|1|0	N
:SWEep:MARKer:STATe?	N
:SWEep:MARKer:XFER	N
:SWEep:MODE AUTO|MANual	Y
:SWEep:MODE?	Y
:SWEep:POINts <value>	Y
:SWEep:POINts?	Y
:SWEep:POWer:STEP <value>[<lvlsuffix>]|UP|DOWN	N
:SWEep:POWer:STEP?	N
:SWEep:TIME <value>[<timesuffix>]	N
:SWEep:TIME:AUTO ON|OFF|1|0	N
:SWEep:TIME:AUTO?	Y
:SWEep:TIME:LLIMit <value>[<timesuffix>]	Y
:SWEep:TIME:LLIMit?	Y
:SWEep:TIME?	Y
:SWEep[:FREQuency]:STEP <value>[<freqsuffix>]|UP|DOWN	N
:SWEep[:FREQuency]:STEP?	N
:SWEep[:POINts]:TRIGger:SOURce IMMediate|BUS|EXTernal :SWEep[:POINts]:TRIGger:SOURce IMMediate|BUS|EXTernal	N

Table 6-5 8375xB SCPI Commands (Continued)

Y= Supported by PSG **N= Not supported by PSG**	**83751B & 83752B**
:SWEep[:POINts]:TRIGger:SOURce? :SWEep[:POINts]:TRIGger:SOURce?	N
:SWEep[:POINts]:TRIGger[:IMMediate]	N
System Subsystem	
:SYSTem:ALTernate <reg num>	Y
:SYSTem:ALTernate:STATe ON|OFF|1|0	Y
:SYSTem:ALTernate:STATe?	Y
:SYSTem:ALTernate?	Y
:SYSTem:COMMunicate:GPIB:ADDRess <value>	Y
:SYSTem:COMMunicate:PMETer:ADDRess <value>	Y
:SYSTem:COMMunicate:PMETer:ADDRess?	Y
:SYSTem:COMMunicate:PMETer:TYPE SCPI|70100A|437B|438A	N
:SYSTem:COMMunicate:PMETer:TYPE?	N
:SYSTem:ERRor?	Y
:SYSTem:KEY:DISable SAVE	N
:SYSTem:KEY:DISable? SAVE	N
:SYSTem:KEY:ENABle SAVE	N
:SYSTem:KEY:ENABle? SAVE	N
:SYSTem:KEY[:CODE] <value>	N
:SYSTem:KEY[:CODE]?	N
:SYSTem:LANGuage "SCPI"|"TMSL"|"COMP"	N
:SYSTem:LANGuage?	Y
:SYSTem:PRESet:TYPE FACTory|USER	Y

Table 6-5 8375xB SCPI Commands (Continued)

Y= Supported by PSG N= Not supported by PSG	83751B & 83752B
:SYSTem:PRESet:TYPE?	Y
:SYSTem:PRESet[:EXECute]	Y
:SYSTem:PRESet[:USER]:SAVE	Y
:SYSTem:SECurity:CLEar	N
:SYSTem:SECurity:COUNt <value>	Y
:SYSTem:SECurity:KLOCk ON|OFF|0|1	N
:SYSTem:SECurity:ZERO ON|OFF|0|1	N
:SYSTem:VERSion?	Y
Trigger Subsystem	
:TRIGger:SOURce1 IMMediate|BUS|EXTernal|HOLD :TRIGger:SOURce IMMediate|BUS|EXTernal|HOLD	N
:TRIGger:SOURce1? :TRIGger:SOURce?	N Y
:TRIGger[:IMMediate]	Y
Tsweep Subsystem	
:TSWeep	Y

8662A / 63A Compatible Commands

The tables in this section provide the following:

Table 6-6 on page 343: 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 PSG family.

Table 6-7 on page 352: 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.

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 PSG List Sweep features.

Table 6-6 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
AM	AM modulation *See also*: Table 6-7 on page 352	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 6-7 on page 352

Table 6-6 8662A / 63A Commands & Equivalent SCPI Sequences (Continued)

Command	Description	8662	8663	Equivalent SCPI Command Sequence
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 6-7 on page 352
FA	Start frequency	Y	Y	*See W2, W3, W4,* *and* Table 6-7 on page 352
FB	Stop frequency	Y	Y	*See W2, W3, W4,* *and* Table 6-7 on page 352

FM	FM modulation
	See also: Table 6-7 on page 352
		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 6-7 on page 352
FS	Span frequency	Y	Y	*See W2, W3, W4,* *and* Table 6-7 on page 352
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`
L1	Learn front panel	N	N	*not supported*
L2	Fast learn	N	N	*not supported*

Table 6-6 8662A / 63A Commands & Equivalent SCPI Sequences (Continued)

Command	Description	8662	8663	Equivalent SCPI Command Sequence

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`

Table 6-6 8662A / 63A Commands & Equivalent SCPI Sequences (Continued)

Command	Description	8662	8663	Equivalent SCPI Command Sequence

MF	Modulation frequency
	
	<mod> = FM, or PM, depending on which is on.
	
	Also see: *M1, M2, and* Table 6-7 on page 352
			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*
N2	Linear 1000 steps	Y	Y	*See W2, W3, and W4*
N3	Step size	Y	Y	*See W2, W3,W4,* *and* Table 6-7 on page 352
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 6-7 on page 352
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`

Table 6-6 8662A/63A Commands & Equivalent SCPI Sequences (Continued)

Command	Description	8662	8663	Equivalent SCPI Command Sequence
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 PSG 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 PSG 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
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*

Table 6-6 8662A / 63A Commands & Equivalent SCPI Sequences (Continued)

Command	Description	8662	8663	Equivalent SCPI Command Sequence

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 6-6 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 PSG 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 PSG range limit; is set to 1ms.*

Table 6-6 8662A / 63A Commands & Equivalent SCPI Sequences (Continued)

Command	Description	8662	8663	Equivalent SCPI Command Sequence
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`
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 6-7 on page 352
UV	µV	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:DWELl: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:DWELl: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:DWELl:TYPE STEP`
				`LIST:TYPE LIST`
				`FREQ:MODE LIST`
				`INIT`

X1	Marker 1	N	N	*not supported*

Table 6-6 8662A / 63A Commands & Equivalent SCPI Sequences (Continued)

Command	Description	8662	8663	Equivalent SCPI Command Sequence
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`

Y1|Y2	Remote stepped sweep on	Y	Y	`INIT:CONT ON`
				`SWEEP:MODE AUTO`
				`LIST:DWELl:TYPE STEP`
				`LIST:TYPE LIST`
				`FREQ:MODE LIST`
				`LIST:TRIG:SOUR BUS`

Y3	Execute remote stepped sweep	Y	Y	`*TRG`

Table 6-7 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`

Symbols

Numerics

A

B

C

E

G

M

P

Q