RSA6100A Series
Real-Time Spectrum Analyzers

Programmer Manual

www.tektronix.com
071-1913-04

Tektronix

Table of Contents

Getting Started

Syntax and Commands

Status and Events

Appendices

Glossary

Preface

This programmer manual covers the RSA6100A Series Real-Time Spectrum Analyzers. It provides information on operating your analyzer using the General Purpose Interface Bus (GPIB).

This manual is composed of the following sections

- *Getting Started* outlines how to use the GPIB interface.
- *Syntax and Commands* defines the syntax used in command descriptions, presents a list of all command subsystems, and presents detailed descriptions of all programming commands.
- *Status and Events* describes how the status and Events Reporting system operates and presents a list of all system errors.
- *Appendices* provides additional information including character charts, GPIB interface specification, and factory initialization settings.

Related Documentation

- *RSA6100A Series Quick Start User Manual*
  (Tektronix part number 071-1909-XX)
  This manual contains general information about how to put your instrument into service, guides to user interface controls, and application examples.
- *RSA6100A Series Online Help*
  The online help contains detailed information about how to operate the instrument.
- *TekVISA Programmer Manual*
  (Tektronix part number 071-1101-XX)
  This manual is available as a printable PDF file on the Tektronix Web site (www.tektronix.com). The manual describes TekVISA, the Tektronix implementation of the VISA Application Programming Interface (API). TekVISA is industry-compliant software for writing interoperable instrument drivers in a variety of Application Development Environments (ADEs).

Getting Started

Getting Started

You can write computer programs that remotely set the analyzer front panel controls or that take measurements and read those measurements for further analysis or storage. To help you get started with programming the analyzer, this section includes the following subsections

- *Overview of the Manual*
  Summarizes each major section of this manual.
- *Connecting the Interface*
  Describes how to physically connect the analyzer to a controller.
- *Using GPIB Ports*
  Describes how to use the GPIB port.
- *Setting the GPIB Address*
  Describes how to set the GPIB parameters from the front panel.
- *Using TekVISA*
  Describes how to use the TekVISA communication protocol.

Overview of the Manual

The information contained in each major section of this manual is described below.

Syntax and Commands

Syntax and Commands, describes the structure and content of the messages your program sends to the analyzer. The following figure shows command parts as described in the *Command Syntax* subsection.

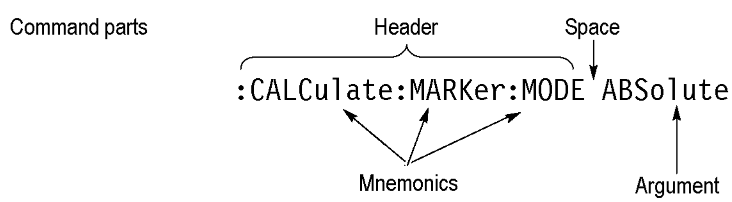


Figure 1-1: Command parts

Section 2 also describes the effect of each command and provides examples of how you might use it. The *Command Groups* subsection provides lists by functional areas. The commands are listed alphabetically in the *Command Descriptions* section.

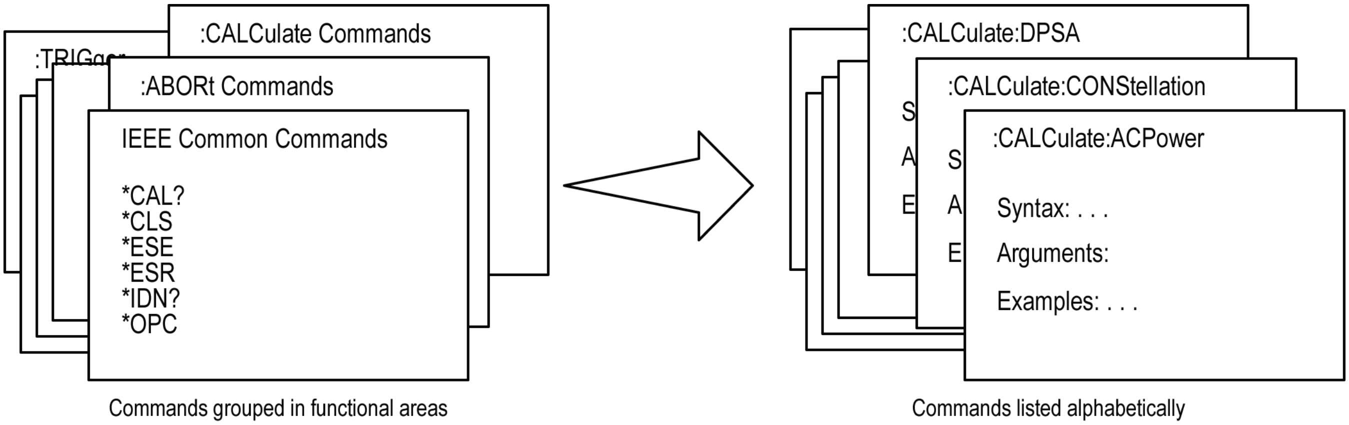


Figure 1-2: Functional groupings and an alphabetical list of commands

Status and Events

The program may request information from theinstrument. Theinstrument provides information in the form of status and error messages. The following figure illustrates the basic operation of this system. Section 3, *Status and Events*, describes how to get status or event information from the program and details the event and error messages.

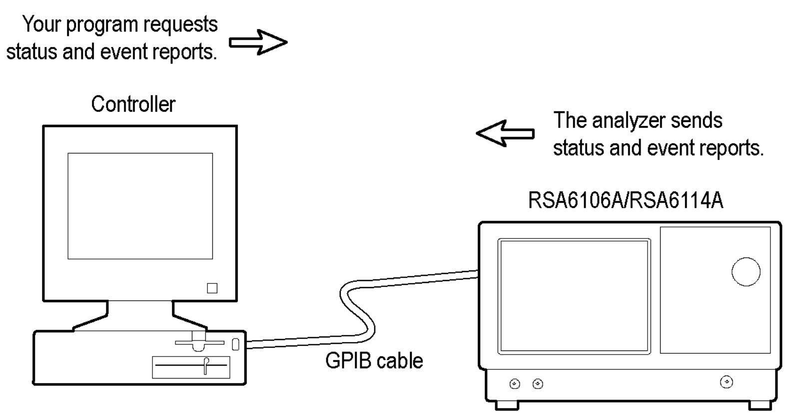


Figure 1-3: Event-driven program

Connecting the Interface

The instrument has a 24-pin GPIB connector on its rear panel, as shown in the following figure. This connector has a D-type shell and conforms to IEEE Std 488.1-1987. Attach an IEEE Std 488.1-1987 GPIB cable (Tektronix part number 012-0991-00) to this connector.

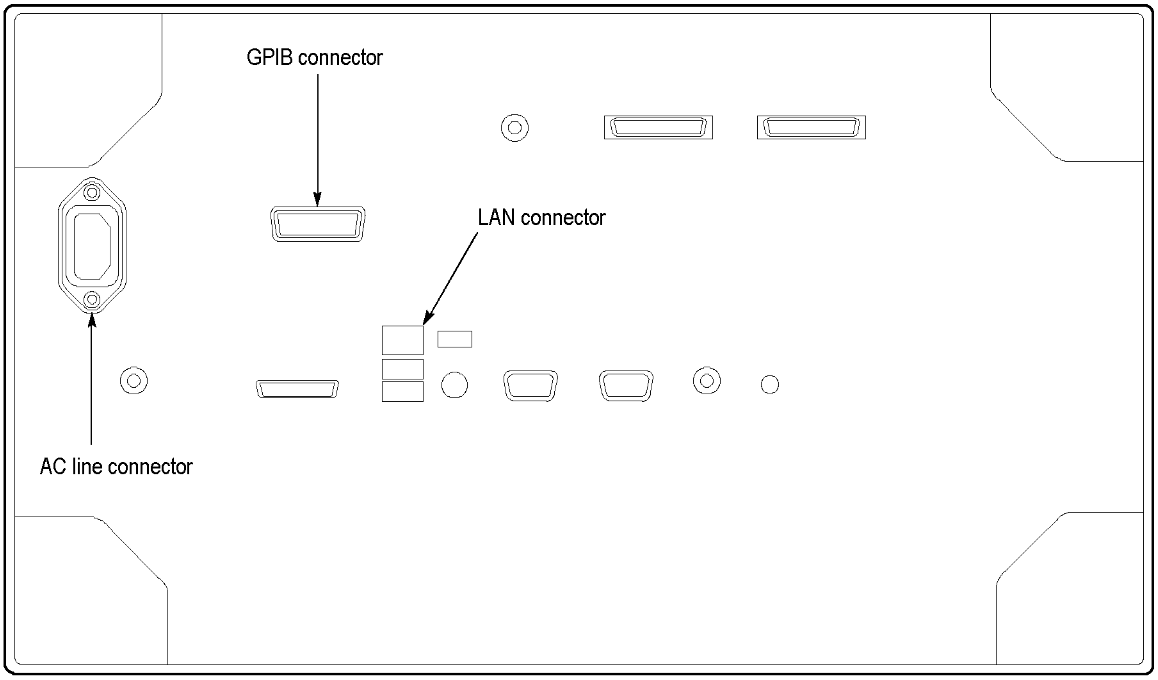


Figure 1-4: GPIB connector (rear panel)

Appendix B: GPIB Interface Specifications gives more information on the GPIB configuration of the analyzer. For the other interfaces, refer to the *RSA6100A Series Quick Start User Manual*.

Using the GPIB Port

The analyzer has Talker/Listener functions through which it can communicate with other devices, as well as the external controller, located on the bus.

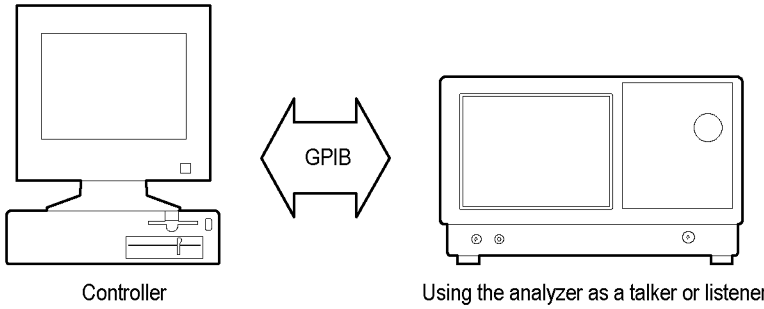


Figure 1-5: GPIB connection

GPIB Requirements

Observe the following rules when you use your analyzer with a GPIB network

- Assign a unique device address to each device on the bus. No two devices can share the same device address.
- Do not connect more than 15 devices to any one bus.
- Connect one device for every 2 m (6 ft) of cable used.
- Do not use more than 20 m (65 ft) of cable to connect devices to a bus.
- Turn on at least 2/3 of the devices on the network while using the network.
- Connect the devices on the network in a star or linear configuration, as shown in the following figure. Do not use loop or parallel configurations.

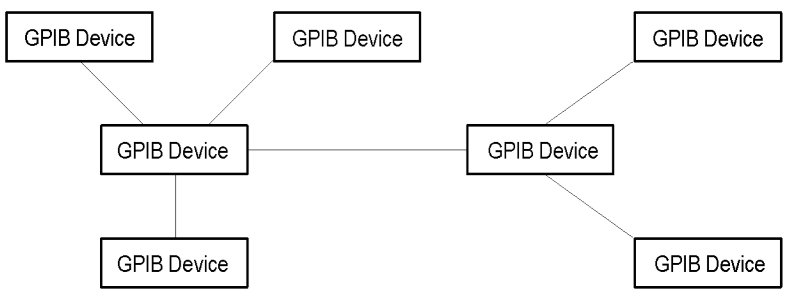
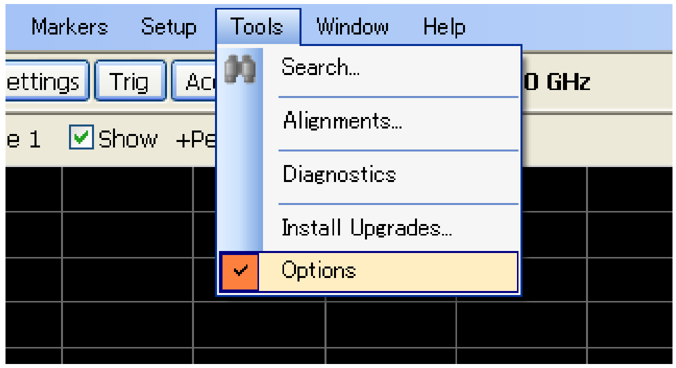


Figure 1-6: Typical GPIB network configurations

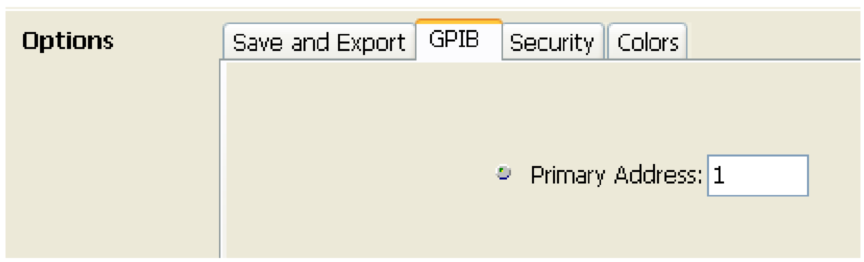
Setting the GPIB Address

When you use the GPIB port to communicate with an external controller, follow these steps to set the address of the analyzer.

1. From the **Tools** menu, select **Options** to open the Options control panel.



2. Click the **GPIB** tab and set the primary address. Range: 0 to 30 (default: 1)



***NOTE.** The GPIB address cannot be initialized by the *RST command.*

Using TekVISA

TekVISA is Tektronix implementation of VISA (Virtual Instrument Software Architecture), an industry-standard communication protocol. VISA provides a common standard for software developers so that software from multiple vendors, such as instrument drivers, can run on the same platform. TekVISA is industry-compliant software, available with selected Tektronix instruments. You can use this software to write (or draw) interoperable instrument drivers in a variety of Application Development Environments (ADEs). It implements a subset of Version 2.2 of the VISA specification for controlling GPIB and serial (RS-232) instrument interfaces locally or remotely via an Ethernet LAN connection.

Installation

Use an internet browser to access the Tektronix Web site (www.tektronix.com) and download the current TekVISA to your PC. Unzip the downloaded file in a temporary directory of your choice and run *Setup.exe*.

NOTE. *The details on TekVISA concepts and operations are explained in the TekVISA Programmer Manual that can be also found on the Tektronix Web site.*

Syntax and Commands

Command Syntax

This section contains information on the Standard Commands for Programmable Instruments (SCPI) and IEEE 488.2 Common Commands you can use to program your RSA6106A/RSA6114A analyzer. The information is organized in the following subsections

- Backus-Naur Form Definition
- SCPI Commands and Queries
- IEEE 488.2 Common Commands
- Constructed Mnemonics

Backus-Naur Form Definition

This manual may describe commands and queries using the Backus-Naur Form (BNF) notation. The following table defines the standard BNF symbols.

Table 2-1: BNF symbols and meanings

Symbol	Meaning
< >	Defined element
:=	Is defined as
|	Exclusive OR
{ }	Group; one element is required
[ ]	Optional; can be omitted
. . .	Previous element(s) may be repeated
( )	Comment

SCPI Commands and Queries

SCPI is a standard created by a consortium that provides guidelines for remote programming of instruments. These guidelines provide a consistent programming environment for instrument control and data transfer. This environment uses defined programming messages, instrument responses, and data format across all SCPI instruments, regardless of manufacturer. The analyzer uses a command language based on the SCPI standard.

The SCPI language is based on a hierarchical or tree structure as shown in the following figure that represents a subsystem. The top level of the tree is the root node; it is followed by one or more lower-level nodes.

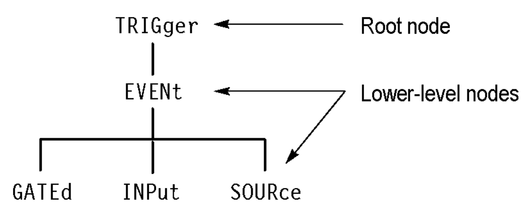


Figure 2-1: Example of SCPI subsystem hierarchy tree

You can create commands and queries from these subsystem hierarchy trees. Commands specify actions for the instrument to perform. Queries return measurement data and information about parameter settings.

Creating Commands

SCPI commands are created by stringing together the nodes of a subsystem hierarchy and separating each node by a colon.

In the figure above, TRIGger is the root node and EVENt, GATed, INPut, and SOURce are lower-level nodes. To create a SCPI command, start with the root node TRIGger and move down the tree structure adding nodes until you reach the end of a branch. Most commands and some queries have parameters; you must include a value for these parameters. If you specify a parameter value that is out of range, the parameter will be set to a default value. The command descriptions, list the valid values for all parameters.

For example, TRIGgerEVENt:SOURce EXTRear is a valid SCPI command created from the hierarchy tree. (See Figure 2-1.)

Creating Queries

To create a query, start at the root node of a tree structure, move down to the end of a branch, and add a question mark. TRIGgerEVENt:SOURce? is an example of a valid SCPI query using the hierarchy tree in the figure. (See Figure 2-1.)

Query Responses

The query causes the analyzer to return information about its status or settings. When a query is sent to the analyzer, only the values are returned. When the returned value is a mnemonic, it is noted in abbreviated format, as shown in the following table.

Table 2-2: Query response examples

Query	Response
CALCulate:SPECtrum:MARKer:X	7.50E+9
SENSe:SPECtrum:FFT:WINDow	BH4B

A few queries also initiate an operation action before returning information. For example, the *CAL? query runs a calibration.

Parameter Types

Every parameter in the command and query descriptions is of a specified type. The parameters are enclosed in brackets, such as <value>. The parameter type is listed after the parameter and is enclosed in parentheses, for example, (boolean). Some parameter types are defined specifically for the RSA6100A Series command set and some are defined by ANSI/IEEE 488.2-1987 as shown in the following table.

Table 2-3: Parameter types used in syntax descriptions

Parameter type	Description	Example
arbitrary block [1]	A specified length of arbitrary data	#512234xxxxx . . . where 5 indicates that the following 5 digits (12234) specify the length of the data in bytes; xxxxx ... indicates the data
boolean	Boolean numbers or values	ON or 1; OFF or 0
binary	Binary numbers	#B0110
octal	Octal numbers	#Q57, #Q3
hexadecimal [2]	Hexadecimal numbers (0-9, A, B, C, D, E, F)	#HAA, #H1
NR1 [2] numeric	Integers	0, 1, 15, -1
NR2 [2,3] numeric	Decimal numbers	1.2, 3.141516, -6.5
NR3 [2] numeric	Floating point numbers	3.1415E-9, -16.1E5
NRf [2] numeric	Flexible decimal number that may be type NR1, NR2 or NR3	See NR1, NR2, and NR3 examples
string [4]	Alphanumeric characters (must be within quotation marks)	"Testing 1, 2, 3"

1 Defined in ANSI/IEEE 488.2 as "Definite Length Arbitrary Block Response Data."

2 An ANSI/IEEE 488.2-1992-defined parameter type.

3 Some commands and queries will accept an octal or hexadecimal value even though the parameter type is defined as NR1.

4 Defined in ANSI/IEEE 488.2 as "String Response Data."

Special Characters

The Line Feed (LF) character (ASCII 10), and all characters in the range of ASCII 127-255 are defined as special characters. These characters are used in arbitrary block arguments only; using these characters in other parts of any command yields unpredictable results.

Abbreviating Commands, Queries, and Parameters

You can abbreviate most SCPI commands, queries, and parameters to an accepted short form. This manual shows these short forms as a combination of upper and lower case letters. The upper case letters indicate the accepted short form of a command. As shown in the following figure, you can create a short form by using only the upper case letters. The accepted short form and the long form are equivalent and request the same action of the instrument.

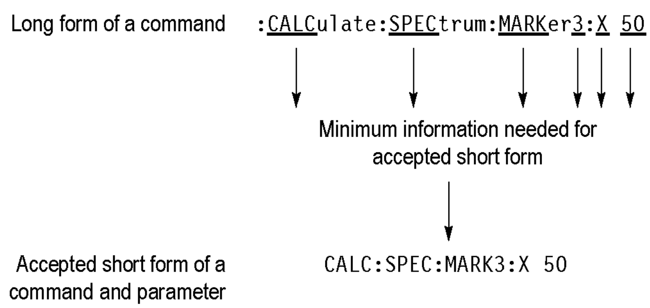


Figure 2-2: Example of abbreviating a command

NOTE. *The numeric suffix of a command or query may be included in either the long form or short form; the analyzer will default to "1" if no suffix is used. In the above figure, the "3" of "MARKer3" indicates that the command is directed to Marker 3.*

Chaining Commands and Queries

You can chain several commands or queries together into a single message. To create a chained message, first create a command or query, add a semicolon (;), and then add more commands or queries and semicolons until the message is complete. If the command following a semicolon is a root node, precede it with a colon (). The following figure illustrates a chained message consisting of several commands and queries. The single chained message should end in a command or query, not a semicolon. Responses to any queries in your message are separated by semicolons.

```
:CALC:SPEC:MARK:X 100;:SPEC:BAND:AUTO ON;:DISP:DDEM:X?;:READ:EVM:PEAK?
```

First command | Second command | First query | Second query

The response from this chained message might be `100;1.2E-2`

Response from first query | Response from second query

Figure 2-3: Example of chaining commands and queries

If a command or query has the same root and lower-level nodes as the previous command or query, you can omit these nodes. In the following figure, the second command has the same root node (TRIGgerEVENt) as the first command, so these nodes can be omitted.

```
:TRIG:EVEN:SOUR EXTR;:TRIG:EVEN:GAT HIG;:TRIG:EVEN:SLOP RIS
```

Identical root and lower-level nodes

```
:TRIG:EVEN:SOUR EXTR;:GAT HIG;:SLOP RIS
```

First command | Additional commands (omitted the root nodes)

Figure 2-4: Example of omitting root and lower-level nodes in a chained message

Unit and SI Prefix

If the decimal numeric argument refers to amplitude, frequency, or time, you can express it using SI units instead of using the scaled explicit point input value format <NR3>. (SI units are units that conform to the Systeme International d'Unites standard.) For example, you can use the input format 200 mV or 1.0 MHz instead of 200.0E-3 or 1.0E+6, respectively, to specify voltage or frequency.

The following table lists the available units.

Table 2-4: Available units

Symbol	Meaning
dB	decibel (relative amplitude)
dBm	decibel (absolute amplitude)
DEG	degree (phase)
Hz	hertz (frequency)
PCT	percent (%)
s	second (time)
V	volt

The available SI prefixes are shown in the following table.

Table 2-5: Available SI prefixes

SI prefix	Z	A	F	P	N	U	M	K	MA[1]	G	T	PE	EX
Corresponding power	10^{-21}	10^{-18}	10^{-15}	10^{-12}	10^{-9}	10^{-6}	10^{-3}	10^{+3}	10^{+6}	10^{+9}	10^{+12}	10^{+15}	10^{+18}

1 When the unit is "Hz", "M" may be used instead of "MA" so that the frequency can be represented by "MHz".

You can omit a unit in a command, but you must include the unit when using a SI prefix. For example, frequency of 15 MHz can be described as follows

15.0E6, 1.5E7Hz, 15000000, 15000000Hz, 15MHz, etc.
("15M" is not allowed.)

Note that you can use either lower or upper case units and prefixes. The following examples have the same result, respectively.

170mhz, 170mHz, 170MHz, etc.
250mv, 250mV, 250MV, etc.

General Rules

Here are three general rules for using SCPI commands, queries, and parameters:

- You can use single (‘ ’) or double (“ ”) quotation marks for quoted strings, but you cannot use both types of quotation marks for the same string.

correct	"This string uses quotation marks correctly."
correct	‘This string also uses quotation marks correctly.’
incorrect	"This string does not use quotation marks correctly.’

- You can use upper case, lower case, or a mixture of both cases for all commands, queries, and parameters.

```
SENSE:SPECTRUM:FFT:LENGTH 1024
```

is the same as

```
sense:spectrum:fft:length 1024
```

and

```
SENSE:spectrum:FFT:length 1024
```

NOTE. *Literal strings (quoted) are case sensitive, for example, file names.*

- No embedded spaces are allowed between or within nodes.

correct	`SENSE:SPECTRUM:FFT:LENGTH 1024`
incorrect	`SENSE: SPECTRUM: FFT: LEN GTH 1024`

IEEE 488.2 Common Commands

Description

ANSI/IEEE Standard 488.2 defines the codes, formats, protocols, and usage of common commands and queries used on the interface between the controller and the instruments. The analyzer complies with this standard.

Command and Query Structure

The syntax for an IEEE 488.2 common command is an asterisk (*) followed by a command and, optionally, a space and parameter value. The syntax for an IEEE 488.2 common query is an asterisk (*) followed by a query and a question mark. All of the common commands and queries are listed in the last part of the *Syntax and Commands* section. The following are examples of common commands:

- *ESE 16
- *CLS

The following are examples of common queries

- *ESR
- *IDN

Constructed Mnemonics

Some header mnemonics specify one of a range of mnemonics. For example, a trace mnemonic can be either TRACe1, TRACe2, TRACe3, or TRACe4. You use these mnemonics in the command just as you do any other mnemonic. For example, there is a TRACe1:SPECtrum:FUNCtion command, and there is also a TRACe2:SPECtrum:FUNCtion command. In the command descriptions, this list of choices is abbreviated as TRACe<x>. The value of <x> is the upper range of valid suffixes. If the numeric suffix is omitted, the analyzer uses the default value of "1".

Table 2-6: Constructed mnemonics

Symbol	Meaning
MARKer<x>	A marker specifier where <x> = 0, 1, 2, 3, or 4. Refer to *Marker Mnemonics*
RANGe<x>	A range specifier where <x> = 1 to 20. Refer to *[SENSe]:SPURious Subgroup* for details.
SPUR<x>	A spurious specifier where <x> = 1 to the number of spurious signals. Refer to *FETCh|:READ:SPURious Subgroup* for details, respectively.
TRACe<x>	A trace specifier where <x> = 0, 1, 2, 3, 4, or 5. Refer to *TRACe Commands* for details.

Command Groups

This section lists the RSA6100A Series analyzer commands in two ways. It first presents them by functional groups. It then lists them alphabetically. The functional group list starts below. The alphabetical list provides more detail on each command.

The RSA6100A Series analyzers conform to the Standard Commands for Programmable Instruments (SCPI) 1999.0 and IEEE Std 488.2-1987 except where noted.

Items followed by question marks are queries; items without question marks are commands. Some items in this section have a question mark in parentheses () in the command header section; this indicates that the item can be both a command and a query.

For the conventions of notation in this manual, refer to *Command Syntax* and following pages.

Measurement Views

The measurement views in the RSA6100A Series analyzers are categorized into the following four groups

- General signal viewing
- General purpose digital modulation (Option 21 only)
- RF measurements
- Pulsed RF (Option 20 only)

Each group contains the measurement views as shown in the following table. Each command works in particular measurement view(s) which are specified under the *Conditions* heading in the command descriptions.

NOTE. *If you send a command for the measurement view that is not displayed on screen, an execution error will occur.*

Table 2-7: Measurement views

Display group	Measurement view
General signal viewing	Spectrum
	DPX (Digital Phosphor) spectrum
	Amplitude versus Time
	Frequency versus Time
	Phase versus Time
	RF I&Q versus Time
	Spectrogram
	Time overview
General purpose digital modulation (Option 21 only)	Constellation
	EVM versus Time
	Magnitude error versus Time
	Phase error versus Time
	Signal quality
	Symbol table
RF measurements	CCDF
	Channel power and ACPR (Adjacent Channel Power Ratio)
	MCPR (Multiple Carrier Power Ratio)
	Occupied Bandwidth (OBW)
	Phase noise (Option 11 only)
	Spurious
Pulsed RF (Option 20 only)	Pulse statistics
	Pulse table
	Pulse trace

Functional Groups

All commands are divided into groups as shown in the following table.

Table 2-8: List of command group

Command group	Function
IEEE common	Conforms to the IEEE Std 488.2.
ABORt	Resets the trigger system and stops measurements.
CALCulate	Controls the markers and the search operations.
CALibration	Controls the external correction.
DISPlay	Controls the display of measurement results and waveforms.
FETCh	Retrieves the measurements from the last acquired data.
INITiate	Controls data acquisition.
INPut	Controls the characteristics of the signal input.
MMEMory	Provides mass storage capabilities for the analyzer.
OUTPut	Controls the characteristics of the signal output.
READ	Obtains the measurement results with acquiring data.
SENSe	Sets up detailed conditions for each measurement.
STATus	Controls the status and event registers.
SYSTem	Sets or queries system parameters for operation.
TRACe	Controls trace activation and math operations.
TRIGger	Controls triggering.
UNIT	Specifies fundamental units for measurement.

Programming Hints

Here are some basic tips for using the RSA6100A Series GPIB commands:

- *Selecting a measurement item*
  Use Display commands to select or display the measurement view.
  [Example] `DISPlay:GENeral:MEASview:NEW SPECtrum`
  Displays the spectrum view on the screen.

- *Setting measurement parameters*
  Use Sense commands to set conditions for the measurement session.
  [Example] `SENSe:SPECtrum:FREQuency:CENTer 1.5GHz`
  Sets the center frequency to 1.5 GHz in the spectrum view.

- *Acquiring an input signal*
  Use an Initiate or Abort command to start or stop data acquisition.
  [Example] `INITiate:CONTinuous ON;INITiate:IMMediate`
  Starts data acquisition in the continuous mode.

- *Processing waveforms arithmetically*
  Use Trace commands for math operation on waveforms.
  [Example] `TRACe1:SPECtrum:FUNCtion`
  AVERage Averages the spectrum waveform.

- *Measuring with the markers*
  Use Calculate commands to measure some quantity using the markers.
  [Example] `CALCulate:SPECtrum:MARKer1:MAXimum`
  Positions the marker at the highest peak signal on the spectrum.

- *Obtaining the measurement results*
  Use a Fetch or Read command to get the results.
  [Example] `FETCh:SPECtrum:TRACe1`
  Returns the spectrum trace data.

- *Scaling the waveform*
  Use Display commands to change the waveform portion on screen.
  [Example] `DISPlay:IQVTime:Y:SCALe 1.5`
  Sets the vertical range to 1.5 V in the IQ versus Time graph.

Refer to *Appendix C* for the default settings of the commands. (See page 3-1.)

The following sections list the commands by group.

IEEE Common Commands

The IEEE 488.2 common commands have a "*" prefix.

Table 2-9: Status and error commands

Header	Description
*CAL	Performs an internal self-calibration.
*CLS	Clears status.
*ESE	Sets or queries the bits in the ESER register.
*ESR?	Returns the contents of the SESR register.
*IDN?	Returns the instrument identification code.
*OPC	Synchronizes commands.
*OPT?	Returns a list of options installed in your analyzer.
*RST	Returns the instrument settings to the factory defaults.
*SRE	Sets or queries the bits in the SRER register.
*STB?	Returns the contents of the SBR using the MSS bit.
*TRG	Generates a trigger.
*WAI	Prevents the analyzer from executing further commands.

Abort Commands

Use the Abort commands to reset the trigger system and to stop measurements.

Table 2-10: Abort commands

Header	Description
ABORt	Resets the trigger system and stops measurements.

Calculate Commands

Use the Calculate commands to control the markers and the search operations.

Table 2-11: Calculate commands

Header	Description
CALCulate basic command subgroup	**General marker control**
CALCulate:MARKer:ADD	Adds a marker.
CALCulate:MARKer:AOFF	Turns off all markers.
CALCulate:MARKer:DELete	Deletes the last marker added.
CALCulate:MARKer:DENSity:EXCursion	Sets or queries the minimum excursion of DPX signal density.
CALCulate:MARKer:DENSity:SMOothing	Sets or queries the number of pixels squared for smoothing the density.
CALCulate:MARKer:DENSity:THReshold	Sets or queries the threshold of DPX signal density to detect peaks.
CALCulate:MARKer:MODE	Selects or queries the marker mode.
CALCulate:MARKer:PEAK:EXCursion	Sets or queries the minimum excursion level.
CALCulate:MARKer:PEAK:THReshold	Sets or queries the threshold level to detect peaks.
CALCulate:SEARch:LIMit:FAIL?	Queries whether the waveform cuts across the limit or not.
CALCulate:SEARch:LIMit:MATCh:BEEP[:STATe]	Selects or queries whether to beep when a match occurs.
CALCulate:SEARch:LIMit:MATCh:SACQuire[:STATe]	Selects or queries whether to stop acquiring data on match.
CALCulate:SEARch:LIMit:MATCh:SDATa[:STATe]	Selects or queries whether to save the acquisition data automatically.
CALCulate:SEARch:LIMit:MATCh:SPICture[:STATe]	Selects or queries whether to save the whole screen automatically.
CALCulate:SEARch:LIMit:MATCh:STRace[:STATe]	Selects or queries whether to save the spectrum trace automatically.
CALCulate:SEARch:LIMit:OPERation	Selects or queries the limit operation in the search function.
CALCulate:SEARch:LIMit:OPERation:FEED	Sets or queries the data flow to be fed in the search operation.
CALCulate:SEARch:LIMit:OPERation:MASK:LOAD	Loads the limit mask from a specified file for the search operation.
CALCulate:SEARch:LIMit:OPERation:MASK:STORE	Stores the limit mask to a specified file for the search operation.
CALCulate:SEARch:LIMit:OPERation:SLIMit	Sets or queries the limit value in the search operation.
CALCulate:SEARch:LIMit:REPort:DATA?	Returns the frequency range(s) that satisfy the search condition.
CALCulate:SEARch:LIMit:REPort:POINts?	Returns the number of range(s) that satisfy the search condition.
CALCulate:SEARch:LIMit:STATe	Selects or queries whether to enable or disable the search function.
CALCulate:ACPower subgroup	**Channel power and ACPR measurement**
CALCulate:ACPower:MARKer<x>:DELTa:X?	Returns the delta marker frequency for the selected marker.
CALCulate:ACPower:MARKer<x>:DELTa:Y?	Returns the delta marker amplitude for the selected marker.
CALCulate:ACPower:MARKer<x>:MAXimum	Moves the marker to the highest peak on the trace.
CALCulate:ACPower:MARKer<x>:PEAK:LEFT	Moves the marker to the next peak to the left on the trace.
CALCulate:ACPower:MARKer<x>:PEAK:RIGHt	Moves the marker to the next peak to the right on the trace.
CALCulate:ACPower:MARKer<x>:X	Sets or queries the horizontal position of the marker.
CALCulate:ACPower:MARKer<x>:Y?	Queries the vertical position of the marker.

Table 2-11: Calculate commands (cont.)

Header	Description
CALCulate:AVTime subgroup	**Frequency versus Time measurement**
CALCulate:AVTime:MARKer<x>:DELTa:X?	Returns the delta marker time for the selected marker.
CALCulate:AVTime:MARKer<x>:DELTa:Y?	Returns the delta marker amplitude for the selected marker.
CALCulate:AVTime:MARKer<x>:MAXimum	Moves the marker to the highest peak on the trace.
CALCulate:AVTime:MARKer<x>:PEAK:HIGHer	Moves the marker to the next peak higher in amplitude.
CALCulate:AVTime:MARKer<x>:PEAK:LEFT	Moves the marker to the next peak to the left on the trace.
CALCulate:AVTime:MARKer<x>:PEAK:LOWer	Moves the marker to the next peak lower in amplitude.
CALCulate:AVTime:MARKer<x>:PEAK:RIGHt	Moves the marker to the next peak to the right on the trace.
CALCulate:AVTime:MARKer<x>:TRACe	Selects or queries the trace on which the marker is placed.
CALCulate:AVTime:MARKer<x>:X	Sets or queries the horizontal position of the marker.
CALCulate:AVTime:MARKer<x>:Y?	Queries the vertical position of the marker.
CALCulate:CONSte subgroup (Option 21 only)	**Constellation measurement**
CALCulate:CONSte:MARKer<x>:DELTa:X[:TIME]?	Returns the delta marker time for the selected marker.
CALCulate:CONSte:MARKer<x>:MAGNitude?	Queries the magnitude readout of the marker.
CALCulate:CONSte:MARKer<x>:MAXimum	Positions the marker at the symbol in the center of the time record.
CALCulate:CONSte:MARKer<x>:PEAK:LEFT	Moves the marker in the time domain to the next lower symbol number.
CALCulate:CONSte:MARKer<x>:PEAK:RIGHt	Moves the marker in the time domain to the next higher symbol number.
CALCulate:CONSte:MARKer<x>:PHASe?	Queries the phase readout of the marker.
CALCulate:CONSte:MARKer<x>:SYMBol?	Queries the symbol readout of the marker.
CALCulate:CONSte:MARKer<x>:VALue?	Queries the value readout of the marker.
CALCulate:CONSte:MARKer<x>:X	Sets or queries the time position of the marker on the trace.
CALCulate:DPSA subgroup	**DPX spectrum measurement**
CALCulate:DPSA:MARKer<x>:DELTa:X?	Returns the delta marker frequency for the selected marker.
CALCulate:DPSA:MARKer<x>:DELTa:Y?	Returns the delta marker amplitude for the selected marker.
CALCulate:DPSA:MARKer<x>:MAXimum	Moves the marker to the highest peak on the trace.
CALCulate:DPSA:MARKer<x>:PEAK:HIGHer	Moves the marker to the next peak higher in amplitude.
CALCulate:DPSA:MARKer<x>:PEAK:LEFT	Moves the marker to the next peak to the left on the trace.
CALCulate:DPSA:MARKer<x>:PEAK:LOWer	Moves the marker to the next peak lower in amplitude.
CALCulate:DPSA:MARKer<x>:PEAK:RIGHt	Moves the marker to the next peak to the right on the trace.
CALCulate:DPSA:MARKer<x>[:SET]:CENTer	Sets the center frequency to the marker frequency.
CALCulate:DPSA:MARKer<x>:TRACe	Selects or queries the trace on which the marker is placed.
CALCulate:DPSA:MARKer<x>:X:AMPLitude	Sets or queries the amplitude position of the marker.
CALCulate:DPSA:MARKer<x>:X[:FREQuency]	Sets or queries the frequency position of the marker.
CALCulate:DPSA:MARKer<x>:Y?	Queries the vertical position of the marker.
CALCulate:EVM subgroup (Option 21 only)	**EVM versus Time measurement**
CALCulate:EVM:MARKer<x>:DELTa:X?	Returns the delta marker time for the selected marker.
CALCulate:EVM:MARKer<x>:DELTa:Y?	Returns the delta marker amplitude for the selected marker.

Table 2-11: Calculate commands (cont.)

Header	Description
CALCulate:EVM:MARKer<x>:MAXimum	Moves the marker to the highest peak on the trace.
CALCulate:EVM:MARKer<x>:PEAK:HIGHer	Moves the marker to the next peak higher in amplitude.
CALCulate:EVM:MARKer<x>:PEAK:LEFT	Moves the marker to the next peak to the left on the trace.
CALCulate:EVM:MARKer<x>:PEAK:LOWer	Moves the marker to the next peak lower in amplitude.
CALCulate:EVM:MARKer<x>:PEAK:RIGHt	Moves the marker to the next peak to the right on the trace.
CALCulate:EVM:MARKer<x>:X	Sets or queries the horizontal position of the marker.
CALCulate:EVM:MARKer<x>:Y?	Queries the vertical position of the marker.
CALCulate:FVTime subgroup	**Frequency versus Time measurement**
CALCulate:FVTime:MARKer<x>:DELTa:X?	Returns the delta marker time for the selected marker.
CALCulate:FVTime:MARKer<x>:DELTa:Y?	Returns the delta marker frequency for the selected marker.
CALCulate:FVTime:MARKer<x>:MAXimum	Moves the marker to the highest peak on the trace.
CALCulate:FVTime:MARKer<x>:PEAK:HIGHer	Moves the marker to the next peak higher in amplitude.
CALCulate:FVTime:MARKer<x>:PEAK:LEFT	Moves the marker to the next peak to the left on the trace.
CALCulate:FVTime:MARKer<x>:PEAK:LOWer	Moves the marker to the next peak lower in amplitude.
CALCulate:FVTime:MARKer<x>:PEAK:RIGHt	Moves the marker to the next peak to the right on the trace.
CALCulate:FVTime:MARKer<x>:X	Sets or queries the horizontal position of the marker.
CALCulate:FVTime:MARKer<x>:Y?	Queries the vertical position of the marker.
CALCulate:IQVTime subgroup	**RF I&Q versus Time measurement**
CALCulate:IQVTime:MARKer<x>:DELTa:X?	Returns the delta marker time for the selected marker.
CALCulate:IQVTime:MARKer<x>:DELTa:Y?	Returns the delta marker amplitude for the selected marker.
CALCulate:IQVTime:MARKer<x>:MAXimum	Moves the marker to the highest peak on the trace.
CALCulate:IQVTime:MARKer<x>:PEAK:HIGHer	Moves the marker to the next peak higher in amplitude.
CALCulate:IQVTime:MARKer<x>:PEAK:LEFT	Moves the marker to the next peak to the left on the trace.
CALCulate:IQVTime:MARKer<x>:PEAK:LOWer	Moves the marker to the next peak lower in amplitude.
CALCulate:IQVTime:MARKer<x>:PEAK:RIGHt	Moves the marker to the next peak to the right on the trace.
CALCulate:IQVTime:MARKer<x>:TRACe	Selects or queries the trace (I or Q) to place the marker.
CALCulate:IQVTime:MARKer<x>:X	Sets or queries the horizontal position of the marker.
CALCulate:IQVTime:MARKer<x>:Y?	Queries the vertical position of the marker.
CALCulate:MCPower subgroup	**MCPR measurement**
CALCulate:MCPower:MARKer<x>:DELTa:X?	Returns the delta marker frequency for the selected marker.
CALCulate:MCPower:MARKer<x>:DELTa:Y?	Returns the delta marker amplitude for the selected marker.
CALCulate:MCPower:MARKer<x>:MAXimum	Moves the marker to the highest peak on the trace.
CALCulate:MCPower:MARKer<x>:PEAK:LEFT	Moves the marker to the next peak to the left on the trace.
CALCulate:MCPower:MARKer<x>:PEAK:RIGHt	Moves the marker to the next peak to the right on the trace.
CALCulate:MCPower:MARKer<x>:X	Sets or queries the horizontal position of the marker.
CALCulate:MCPower:MARKer<x>:Y?	Queries the vertical position of the marker.

Table 2-11: Calculate commands (cont.)

Header	Description
CALCulate:MERRor subgroup (Option 21 only)	**Magnitude error versus Time measurement**
CALCulate:MERRor:MARKer<x>:DELTa:X?	Returns the delta marker time for the selected marker.
CALCulate:MERRor:MARKer<x>:DELTa:Y?	Returns the delta marker amplitude for the selected marker.
CALCulate:MERRor:MARKer<x>:MAXimum	Moves the marker to the highest peak on the trace.
CALCulate:MERRor:MARKer<x>:PEAK:HIGHer	Moves the marker to the next peak higher in amplitude.
CALCulate:MERRor:MARKer<x>:PEAK:LEFT	Moves the marker to the next peak to the left on the trace.
CALCulate:MERRor:MARKer<x>:PEAK:LOWer	Moves the marker to the next peak lower in amplitude.
CALCulate:MERRor:MARKer<x>:PEAK:RIGHt	Moves the marker to the next peak to the right on the trace.
CALCulate:MERRor:MARKer<x>:X	Sets or queries the horizontal position of the marker.
CALCulate:MERRor:MARKer<x>:Y?	Queries the vertical position of the marker.
CALCulate:OBWidth subgroup	**Occupied Bandwidth measurement**
CALCulate:OBWidth:MARKer<x>:DELTa:X?	Returns the delta marker frequency for the selected marker.
CALCulate:OBWidth:MARKer<x>:DELTa:Y?	Returns the delta marker amplitude for the selected marker.
CALCulate:OBWidth:MARKer<x>:MAXimum	Moves the marker to the highest peak on the trace.
CALCulate:OBWidth:MARKer<x>:PEAK:HIGHer	Moves the marker to the next peak higher in amplitude.
CALCulate:OBWidth:MARKer<x>:PEAK:LEFT	Moves the marker to the next peak to the left on the trace.
CALCulate:OBWidth:MARKer<x>:PEAK:LOWer	Moves the marker to the next peak lower in amplitude.
CALCulate:OBWidth:MARKer<x>:PEAK:RIGHt	Moves the marker to the next peak to the right on the trace.
CALCulate:OBWidth:MARKer<x>[:SET]:CENTer	Sets the center frequency to the value at the marker position.
CALCulate:OBWidth:MARKer<x>:X	Sets or queries the horizontal position of the marker.
CALCulate:OBWidth:MARKer<x>:Y?	Queries the vertical position of the marker.
CALCulate:PERRor subgroup (Option 21 only)	**Phase error versus Time measurement**
CALCulate:PERRor:MARKer<x>:DELTa:X?	Returns the delta marker time for the selected marker.
CALCulate:PERRor:MARKer<x>:DELTa:Y?	Returns the delta marker phase for the selected marker.
CALCulate:PERRor:MARKer<x>:MAXimum	Moves the marker to the highest peak on the trace.
CALCulate:PERRor:MARKer<x>:PEAK:HIGHer	Moves the marker to the next peak higher in amplitude.
CALCulate:PERRor:MARKer<x>:PEAK:LEFT	Moves the marker to the next peak to the left on the trace.
CALCulate:PERRor:MARKer<x>:PEAK:LOWer	Moves the marker to the next peak lower in amplitude.
CALCulate:PERRor:MARKer<x>:PEAK:RIGHt	Moves the marker to the next peak to the right on the trace.
CALCulate:PERRor:MARKer<x>:X	Sets or queries the horizontal position of the marker.
CALCulate:PERRor:MARKer<x>:Y?	Queries the vertical position of the marker.
CALCulate:PHVTime subgroup	**Phase versus Time measurement**
CALCulate:PHVTime:MARKer<x>:DELTa:X?	Returns the delta marker time for the selected marker.
CALCulate:PHVTime:MARKer<x>:DELTa:Y?	Returns the delta marker phase for the selected marker.
CALCulate:PHVTime:MARKer<x>:MAXimum	Moves the marker to the highest peak on the trace.
CALCulate:PHVTime:MARKer<x>:PEAK:HIGHer	Moves the marker to the next peak higher in amplitude.
CALCulate:PHVTime:MARKer<x>:PEAK:LEFT	Moves the marker to the next peak to the left on the trace.

Table 2-11: Calculate commands (cont.)

Header	Description
CALCulate:PHVTime:MARKer<x>:PEAK:LOWer	Moves the marker to the next peak lower in amplitude.
CALCulate:PHVTime:MARKer<x>:PEAK:RIGHt	Moves the marker to the next peak to the right on the trace.
CALCulate:PHVTime:MARKer<x>:X	Sets or queries the horizontal position of the marker.
CALCulate:PHVTime:MARKer<x>:Y?	Queries the vertical position of the marker.
CALCulate:PULSe subgroup (Option 20 only)	**Pulsed RF measurements**
CALCulate:PULSe:STATistics:MARKer<x>:DELTa:X?	Returns the delta marker frequency for the selected marker.
CALCulate:PULSe:STATistics:MARKer<x>:DELTa:Y?	Returns the delta marker amplitude for the selected marker.
CALCulate:PULSe:STATistics:MARKer<x>:MAXimum	Moves the marker to the highest peak on the statistics trace.
CALCulate:PULSe:STATistics:MARKer<x>:PEAK:HIGHer	Moves the marker to the next peak higher in amplitude.
CALCulate:PULSe:STATistics:MARKer<x>:PEAK:LEFT	Moves the marker to the next peak to the left on the statistics trace.
CALCulate:PULSe:STATistics:MARKer<x>:PEAK:LOWer	Moves the marker to the next peak lower in amplitude.
CALCulate:PULSe:STATistics:MARKer<x>:PEAK:RIGHt	Moves the marker to the next peak to the right on the statistics trace.
CALCulate:PULSe:STATistics:MARKer<x>:X	Sets or queries the horizontal position of the marker.
CALCulate:PULSe:STATistics:MARKer<x>:Y?	Queries the vertical position of the marker.
CALCulate:PULSe:TRACe:MARKer<x>:DELTa:X?	Returns the delta marker time for the selected marker.
CALCulate:PULSe:TRACe:MARKer<x>:DELTa:Y?	Returns the delta marker amplitude for the selected marker.
CALCulate:PULSe:TRACe:MARKer<x>:MAXimum	Moves the marker to the highest peak on the pulse trace.
CALCulate:PULSe:TRACe:MARKer<x>:PEAK:HIGHer	Moves the marker to the next peak higher in amplitude.
CALCulate:PULSe:TRACe:MARKer<x>:PEAK:LEFT	Moves the marker to the next peak to the left on the pulse trace.
CALCulate:PULSe:TRACe:MARKer<x>:PEAK:LOWer	Moves the marker to the next peak lower in amplitude.
CALCulate:PULSe:TRACe:MARKer<x>:PEAK:RIGHt	Moves the marker to the next peak to the right on the pulse trace.
CALCulate:PULSe:TRACe:MARKer<x>:X	Sets or queries the horizontal position of the marker.
CALCulate:PULSe:TRACe:MARKer<x>:Y?	Queries the vertical position of the marker.
CALCulate:SGRam subgroup	**Spectrogram measurement**
CALCulate:SGRam:MARKer<x>:DELTa:X:FREQuency?	Returns the delta marker frequency for the selected marker.
CALCulate:SGRam:MARKer<x>:DELTa:X[:TIME]?	Returns the delta marker time for the selected marker.
CALCulate:SGRam:MARKer<x>:DELTa:Y?	Returns the delta marker amplitude for the selected marker.
CALCulate:SGRam:MARKer<x>:MAXimum	Moves the marker to the highest peak on the trace.
CALCulate:SGRam:MARKer<x>:PEAK:HIGHer	Moves the marker to the next peak higher in amplitude.
CALCulate:SGRam:MARKer<x>:PEAK:LEFT	Moves the marker to the next peak to the left on the trace.
CALCulate:SGRam:MARKer<x>:PEAK:LOWer	Moves the marker to the next peak lower in amplitude.
CALCulate:SGRam:MARKer<x>:PEAK:RIGHt	Moves the marker to the next peak to the right on the trace.
CALCulate:SGRam:MARKer<x>[:SET]:CENTer	Sets the center frequency to the marker frequency.
CALCulate:SGRam:MARKer<x>:X:FREQuency	Sets or queries the marker frequency.
CALCulate:SGRam:MARKer<x>:X[:TIME]	Sets or queries the marker time.
CALCulate:SGRam:MARKer<x>:Y?	Queries the marker amplitude.

Table 2-11: Calculate commands (cont.)

Header	Description
CALCulate:SPECtrum subgroup	**Spectrum measurement**
CALCulate:SPECtrum:MARKer<x>:DELTa:X?	Returns the delta marker frequency for the selected marker.
CALCulate:SPECtrum:MARKer<x>:DELTa:Y?	Returns the delta marker amplitude for the selected marker.
CALCulate:SPECtrum:MARKer<x>:MAXimum	Moves the marker to the highest peak on the trace.
CALCulate:SPECtrum:MARKer<x>:PEAK:HIGHer	Moves the marker to the next peak higher in amplitude.
CALCulate:SPECtrum:MARKer<x>:PEAK:LEFT	Moves the marker to the next peak to the left on the trace.
CALCulate:SPECtrum:MARKer<x>:PEAK:LOWer	Moves the marker to the next peak lower in amplitude.
CALCulate:SPECtrum:MARKer<x>:PEAK:RIGHt	Moves the marker to the next peak to the right on the trace.
CALCulate:SPECtrum:MARKer<x>[:SET]:CENTer	Sets the center frequency to the marker frequency.
CALCulate:SPECtrum:MARKer<x>:TRACe	Selects or queries the trace on which the marker is placed.
CALCulate:SPECtrum:MARKer<x>:X	Sets or queries the horizontal position of the marker.
CALCulate:SPECtrum:MARKer<x>:Y?	Queries the vertical position of the marker.
CALCulate:SPURious subgroup	**Spurious measurement**
CALCulate:SPURious:MARKer<x>:DELTa:X?	Returns the delta marker frequency for the selected marker.
CALCulate:SPURious:MARKer<x>:DELTa:Y?	Returns the delta marker amplitude for the selected marker.
CALCulate:SPURious:MARKer<x>:MAXimum	Moves the marker to the highest peak on the trace.
CALCulate:SPURious:MARKer<x>:PEAK:HIGHer	Moves the marker to the next peak higher in amplitude.
CALCulate:SPURious:MARKer<x>:PEAK:LEFT	Moves the marker to the next peak to the left on the trace.
CALCulate:SPURious:MARKer<x>:PEAK:LOWer	Moves the marker to the next peak lower in amplitude.
CALCulate:SPURious:MARKer<x>:PEAK:RIGHt	Moves the marker to the next peak to the right on the trace.
CALCulate:SPURious:MARKer<x>[:SET]:CENTer	Sets the center frequency to the marker frequency.
CALCulate:SPURious:MARKer<x>:X	Sets or queries the horizontal position of the marker.
CALCulate:SPURious:MARKer<x>:Y?	Queries the vertical position of the marker.
CALCulate:TOVerview subgroup	**Time overview**
CALCulate:TOVerview:MARKer<x>:DELTa:X?	Returns the delta marker time for the selected marker.
CALCulate:TOVerview:MARKer<x>:DELTa:Y?	Returns the delta marker amplitude for the selected marker.
CALCulate:TOVerview:MARKer<x>:MAXimum	Moves the marker to the highest peak on the trace.
CALCulate:TOVerview:MARKer<x>:PEAK:HIGHer	Moves the marker to the next peak higher in amplitude.
CALCulate:TOVerview:MARKer<x>:PEAK:LEFT	Moves the marker to the next peak to the left on the trace.
CALCulate:TOVerview:MARKer<x>:PEAK:LOWer	Moves the marker to the next peak lower in amplitude.
CALCulate:TOVerview:MARKer<x>:PEAK:RIGHt	Moves the marker to the next peak to the right on the trace.
CALCulate:TOVerview:MARKer<x>:X	Sets or queries the horizontal position of the marker.
CALCulate:TOVerview:MARKer<x>:Y?	Queries the vertical position of the marker.

Marker Mnemonics

Up to five markers can be used. In commands, these are named MARKer<x>, where <x> can be 0, 1, 2, 3, or 4 as shown in the following table.

Table 2-12: Marker mnemonics

Mnemonic	Description
MARKer0	Reference marker (MR)
MARKer1	Marker 1 (M1)
MARKer2	Marker 2 (M2)
MARKer3	Marker 3 (M3)
MARKer4	Marker 4 (M4)

NOTE. *If you omit the numeric suffix, the marker control defaults to Marker 1.*

Before operating the marker, you have to enable it using the CALCulate basic commands.

If you attempt to use a marker other than above in a CALCulate command, the suffix error (error code -130) will occur.

Calibration Commands

Use the CALibration commands to control the external correction.

Table 2-13: Calibration commands

Header	Description
CALibration:ABORt	Aborts any actions related to the alignments in progress.
CALibration:AUTO	Selects or queries whether or not to run alignments automatically.
CALibration:CORRection:EXTernal:EDIT<x>:LABel	Sets or queries the name of the external loss table.
CALibration:CORRection:EXTernal:EDIT<x>:STATe	Sets or queries whether to enable or disable the external loss table.
CALibration:CORRection:EXTernal:GAIN[:MAGNitude]	Sets or queries the external gain value.
CALibration:CORRection:EXTernal:GAIN:STATe	Selects or queries whether to enable or disable the external gain value.
CALibration:CORRection:EXTernal:PROBe:CONNect?	Queries whether the external probe is connected to the analyzer or not.
CALibration:CORRection:EXTernal:PROBe[:MAGNitude]?	Queries the external probe attenuation.
CALibration:CORRection:EXTernal:PROBe:STATe	Determines whether to correct data for the external probe attenuation.

Display Commands

Use the DISPlay commands to control the display of measurement waveforms and results on the screen.

Table 2-14: Display commands

Header	Description
DISPlay basic command subgroup	**General window control**
DISPlay:WINDow:ACTive:MEASurement?	Queries the active measurement views.
DISPlay:WINDow:COLor:SCHeme	Selects or queries the color scheme for traces and background.
DISPlay:WINDow:OPTimized:MEASurement?	Queries the measurement views that are optimized.
DISPlay:ACPower subgroup	**Channel power and ACPR measurement**
DISPlay:ACPower:MARKer:SHOW:STATe	Determines whether to show the readout for the selected marker.
DISPlay:ACPower:PLEVel:SHOW:STATe	Determines whether to show the power levels.
DISPlay:ACPower:RESet:SCALe	Resets the horizontal and vertical scale to the default values.
DISPlay:ACPower:WINDow:TRACe:GRATicule:GRID:STATe	Determines whether to show the graticule grid on screen.
DISPlay:ACPower:X[:SCALe]	Sets or queries the horizontal range.
DISPlay:ACPower:X[:SCALe]:AUTO	Rescales the horizontal axis automatically.
DISPlay:ACPower:X[:SCALe]:OFFSet	Sets or queries the minimum horizontal value (left edge).
DISPlay:ACPower:Y[:SCALe]	Sets or queries the vertical range.
DISPlay:ACPower:Y[:SCALe]:AUTO	Rescales the vertical axis automatically.
DISPlay:ACPower:Y[:SCALe]:OFFSet	Sets or queries the vertical offset.
DISPlay:AVTime subgroup	**Amplitude versus Time measurement**
DISPlay:AVTime:MARKer:SHOW:STATe	Determines whether to show the readout for the selected marker.
DISPlay:AVTime:RESet	Resets the horizontal and vertical scale to the default values.
DISPlay:AVTime:TRIGger:LEVel:STATe	Determines whether to show the power trigger level line on screen.
DISPlay:AVTime:WINDow:TRACe:GRATicule:GRID:STATe	Determines whether to show the graticule grid on screen.
DISPlay:AVTime:X:RSCale	Rescales the horizontal axis automatically.
DISPlay:AVTime:X[:SCALe]:AUTO	Sets the horizontal scale automatically.
DISPlay:AVTime:X[:SCALe]:AUTO:STATe	Determines whether to set the horizontal scale automatically or manually.
DISPlay:AVTime:X[:SCALe]:FULL	Sets or queries the horizontal scale.
DISPlay:AVTime:X[:SCALe]:MAXimum?	Queries the upper limit of the horizontal scale setting range.
DISPlay:AVTime:X[:SCALe]:MINimum?	Queries the lower limit of the horizontal scale setting range.
DISPlay:AVTime:X[:SCALe]:OFFSet	Sets or queries the minimum horizontal value (left edge).
DISPlay:AVTime:X[:SCALe]:OFFSet:MAXimum?	Queries the upper limit of the horizontal offset setting range.
DISPlay:AVTime:X[:SCALe]:OFFSet:MINimum?	Queries the lower limit of the horizontal offset setting range.
DISPlay:AVTime:Y:RSCale	Rescales the vertical axis automatically.

Table 2-14: Display commands (cont.)

Header	Description
DISPlay:AVTime:Y[:SCALe]:FULL	Sets or queries the vertical scale.
DISPlay:AVTime:Y[:SCALe]:OFFSet	Sets or queries the vertical offset.
DISPlay:CONSte subgroup (Option 21 only)	**Constellation measurement**
DISPlay:CONSte:WINDow:TRACe:GRATicule:GRID:STATe	Selects or queries whether to show the graticule grid on the screen.
DISPlay:DDEMod subgroup (Option 21 only)	**General purpose digital modulation measurements**
DISPlay:DDEMod:MEASview:DELete	Deletes the measurement view.
DISPlay:DDEMod:MEASview:NEW	Displays a new measurement view.
DISPlay:DDEMod:MEASview:SELect	Selects or queries the measurement view.
DISPlay:DDEMod:RADix	Selects or queries the base of symbols.
DISPlay:DDEMod:X[:SCALe]	Sets or queries the horizontal scale.
DISPlay:DDEMod:X[:SCALe]:AUTO	Sets the horizontal scale automatically.
DISPlay:DDEMod:X[:SCALe]:AUTO:STATe	Determines whether to set the horizontal scale automatically or manually.
DISPlay:DDEMod:X[:SCALe]:MAXimum?	Queries the upper limit of the horizontal scale setting range.
DISPlay:DDEMod:X[:SCALe]:MINimum?	Queries the lower limit of the horizontal scale setting range.
DISPlay:DDEMod:X[:SCALe]:OFFSet	Sets or queries the minimum horizontal value (left edge).
DISPlay:DDEMod:X[:SCALe]:OFFSet:MAXimum?	Queries the upper limit of the horizontal offset setting range.
DISPlay:DDEMod:X[:SCALe]:OFFSet:MINimum?	Queries the lower limit of the horizontal offset setting range.
DISPlay:DDEMod:X[:SCALe]:RESet	Presets the horizontal scale to the default value.
DISPlay:DPSA subgroup	**DPX spectrum measurement**
DISPlay:DPSA:WINDow:TRACe:GRATicule:GRID:STATe	Selects or queries whether to show the graticule grid on the screen.
DISPlay:DPSA:Y[:SCALe]:PDIVision	Sets or queries the vertical scale (per division).
DISPlay:EVM subgroup (Option 21 only)	**EVM versus Time measurement**
DISPlay:EVM:Y[:SCALe]	Sets or queries the vertical scale.
DISPlay:EVM:Y[:SCALe]:AUTO	Sets the vertical scale automatically.
DISPlay:EVM:Y[:SCALe]:OFFSet	Sets or queries the minimum vertical value (bottom edge).
DISPlay:FVTime subgroup	**Frequency versus Time measurement**
DISPlay:FVTime:WINDow:TRACe:GRATicule:GRID:STATe	Selects or queries whether to show the graticule grid on the screen.
DISPlay:FVTime:X[:SCALe]	Sets or queries the horizontal scale.
DISPlay:FVTime:X[:SCALe]:AUTO	Sets the horizontal scale automatically.
DISPlay:FVTime:X[:SCALe]:AUTO:STATe	Determines whether to set the horizontal scale automatically or manually.
DISPlay:FVTime:X[:SCALe]:MAXimum?	Queries the upper limit of the horizontal scale setting range.
DISPlay:FVTime:X[:SCALe]:MINimum?	Queries the lower limit of the horizontal scale setting range.
DISPlay:FVTime:X[:SCALe]:OFFSet	Sets or queries the minimum horizontal value (left edge).
DISPlay:FVTime:X[:SCALe]:OFFSet:MAXimum?	Queries the upper limit of the horizontal offset setting range.

Table 2-14: Display commands (cont.)

Header	Description
DISPlay:FVTime:X[:SCALe]:OFFSet:MINimum?	Queries the lower limit of the horizontal offset setting range.
DISPlay:FVTime:Y[:SCALe]	Sets or queries the vertical scale.
DISPlay:FVTime:Y[:SCALe]:AUTO	Sets the vertical scale automatically.
DISPlay:FVTime:Y[:SCALe]:OFFSet	Sets or queries the vertical offset.
DISPlay:GENeral subgroup	**General signal viewing**
DISPlay:GENeral:MEASview:DELete	Deletes the measurement view.
DISPlay:GENeral:MEASview:NEW	Displays a new measurement view.
DISPlay:GENeral:MEASview:SELect	Selects or queries the measurement view.
DISPlay:GPRF subgroup	**General purpose RF measurements**
DISPlay:GPRF:MEASview:DELete	Deletes the measurement view.
DISPlay:GPRF:MEASview:NEW	Displays a new measurement view.
DISPlay:GPRF:MEASview:SELect	Selects or queries the measurement view.
DISPlay:IQVTime subgroup	**RF I&Q versus Time measurement**
DISPlay:IQVTime:WINDow:TRACe:GRATicule:GRID:STATe	Selects or queries whether to show the graticule grid on the screen.
DISPlay:IQVTime:X[:SCALe]	Sets or queries the horizontal scale.
DISPlay:IQVTime:X[:SCALe]:AUTO	Sets the horizontal scale automatically.
DISPlay:IQVTime:X[:SCALe]:AUTO:STATe	Determines whether to set the horizontal scale automatically or manually.
DISPlay:IQVTime:X[:SCALe]:MAXimum?	Queries the upper limit of the horizontal scale setting range.
DISPlay:IQVTime:X[:SCALe]:MINimum?	Queries the lower limit of the horizontal scale setting range.
DISPlay:IQVTime:X[:SCALe]:OFFSet	Sets or queries the minimum horizontal value (left edge).
DISPlay:IQVTime:X[:SCALe]:OFFSet:MAXimum?	Queries the upper limit of the horizontal offset setting range.
DISPlay:IQVTime:X[:SCALe]:OFFSet:MINimum?	Queries the lower limit of the horizontal offset setting range.
DISPlay:IQVTime:Y[:SCALe]	Sets or queries the vertical scale.
DISPlay:IQVTime:Y[:SCALe]:AUTO	Sets the vertical scale automatically.
DISPlay:IQVTime:Y[:SCALe]:OFFSet	Sets or queries the vertical offset.
DISPlay:IQVTime:Y[:SCALe]:RESCale	Rescales the vertical scale.
DISPlay:MCPower subgroup	**MCPR measurement**
DISPlay:MCPower:MARKer:SHOW:STATe	Determines whether to show the readout for the selected marker.
DISPlay:MCPower:PLEVel:SHOW:STATe	Determines whether to show the power levels.
DISPlay:MCPower:RESet:SCALe	Resets the horizontal and vertical scale to the default values.
DISPlay:MCPower:WINDow:TRACe:GRATicule:GRID:STATe	Determines whether to show the graticule grid on screen.
DISPlay:MCPower:X[:SCALe]	Sets or queries the horizontal range.
DISPlay:MCPower:X[:SCALe]:AUTO	Rescales the horizontal axis automatically.
DISPlay:MCPower:X[:SCALe]:OFFSet	Sets or queries the minimum horizontal value (left edge).
DISPlay:MCPower:Y[:SCALe]	Sets or queries the vertical range.

Table 2-14: Display commands (cont.)

Header	Description
DISPlay:MCPower:Y[:SCALe]:AUTO	Rescales the vertical axis automatically.
DISPlay:MCPower:Y[:SCALe]:OFFSet	Sets or queries the vertical offset.
DISPlay:MERRor subgroup (Option 21 only)	**Magnitude error versus Time measurement**
DISPlay:MERRor:Y[:SCALe]	Sets or queries the vertical scale.
DISPlay:MERRor:Y[:SCALe]:AUTO	Sets the vertical scale automatically.
DISPlay:MERRor:Y[:SCALe]:OFFSet	Sets or queries the minimum vertical value (bottom edge).
DISPlay:OBWidth subgroup	**Occupied Bandwidth measurement**
DISPlay:OBWidth:MARKer:SHOW:STATe	Determines whether to show the readout for the selected marker.
DISPlay:OBWidth:RESet:SCALe	Resets the horizontal and vertical scale to the default values.
DISPlay:OBWidth:SELected:BANDwidth	Selects or queries the bandwidth (OBW or x dB BW) to measure.
DISPlay:OBWidth:WINDow:TRACe:GRATicule:GRID:STATe	Determines whether to show the graticule grid on screen.
DISPlay:OBWidth:X[:SCALe]	Sets or queries the horizontal range.
DISPlay:OBWidth:X[:SCALe]:AUTO	Rescales the horizontal axis automatically.
DISPlay:OBWidth:X[:SCALe]:OFFSet	Sets or queries the minimum horizontal value (left edge).
DISPlay:OBWidth:Y[:SCALe]	Sets or queries the vertical range.
DISPlay:OBWidth:Y[:SCALe]:AUTO	Rescales the vertical axis automatically.
DISPlay:OBWidth:Y[:SCALe]:OFFSet	Sets or queries the vertical offset.
DISPlay:PERRor subgroup (Option 21 only)	**Phase error versus Time measurement**
DISPlay:PERRor:Y[:SCALe]	Sets or queries the vertical scale.
DISPlay:PERRor:Y[:SCALe]:AUTO	Sets the vertical scale automatically.
DISPlay:PERRor:Y[:SCALe]:OFFSet	Sets or queries the minimum vertical value (bottom edge).
DISPlay:PHVTime subgroup	**Phase versus Time measurement**
DISPlay:PHVTime:WINDow:TRACe:GRATicule:GRID:STATe	Selects or queries whether to show the graticule grid on the screen.
DISPlay:PHVTime:X[:SCALe]	Sets or queries the horizontal scale.
DISPlay:PHVTime:X[:SCALe]:AUTO	Sets the horizontal scale automatically.
DISPlay:PHVTime:X[:SCALe]:AUTO:STATe	Determines whether to set the horizontal scale automatically or manually.
DISPlay:PHVTime:X[:SCALe]:MAXimum?	Queries the upper limit of the horizontal scale setting range.
DISPlay:PHVTime:X[:SCALe]:MINimum?	Queries the lower limit of the horizontal scale setting range.
DISPlay:PHVTime:X[:SCALe]:OFFSet	Sets or queries the minimum horizontal value (left edge).
DISPlay:PHVTime:X[:SCALe]:OFFSet:MAXimum?	Queries the upper limit of the horizontal offset setting range.
DISPlay:PHVTime:X[:SCALe]:OFFSet:MINimum?	Queries the lower limit of the horizontal offset setting range.
DISPlay:PHVTime:Y[:SCALe]	Sets or queries the vertical scale.
DISPlay:PHVTime:Y[:SCALe]:AUTO	Sets the vertical scale automatically.
DISPlay:PHVTime:Y[:SCALe]:AXIS	Selects or queries the vertical axis representation.
DISPlay:PHVTime:Y[:SCALe]:AXIS:REFerence	Sets or queries the reference time for phase.

Table 2-14: Display commands (cont.)

Header	Description
DISPlay:PHVTime:Y[:SCALe]:OFFSet	Sets or queries the vertical offset.
DISPlay:PHVTime:Y[:SCALe]:RESCale	Rescales the vertical scale.
DISPlay:PNOise subgroup (Option 11 only)	**Phase noise measurements**
DISPlay:PNOise:MARKer:SHOW:STATe	Determines whether to show the readout for the selected marker.
DISPlay:PNOise:RESet:SCALe	Resets the horizontal and vertical scale to the default values.
DISPlay:PNOise:WINDow:TRACe:GRATicule:GRID:STATe	Selects or queries whether to show the graticule grid on the screen.
DISPlay:PNOise:X[:SCALe]:AUTO	Rescales the horizontal axis automatically.
DISPlay:PNOise:X[:SCALe]:STARt	Sets or queries the start frequency of the graph.
DISPlay:PNOise:X[:SCALe]:STOP	Sets or queries the stop frequency of the graph.
DISPlay:PNOise:Y[:SCALe]	Sets or queries the vertical scale.
DISPlay:PNOise:Y[:SCALe]:AUTO	Rescales the vertical axis automatically.
DISPlay:PNOise:Y[:SCALe]:OFFSet	Sets or queries the vertical offset.
DISPlay:PNOise:Y[:SCALe]:PDIVision	Sets or queries the vertical scale (per division).
DISPlay:PULSe subgroup (Option 20 only)	**Pulsed RF measurements**
DISPlay:PULSe:MEASview:DELete	Deletes the measurement view.
DISPlay:PULSe:MEASview:NEW	Displays a new measurement view.
DISPlay:PULSe:MEASview:SELect	Selects or queries the measurement view.
DISPlay:PULSe:RESult:ATX	Selects or queries whether to show the average transmitted power result.
DISPlay:PULSe:RESult:AVERage	Selects or queries whether to show the average on power result.
DISPlay:PULSe:RESult:DROop	Selects or queries whether to show the droop in the results table.
DISPlay:PULSe:RESult:DUTPct	Selects or queries whether to show the duty factor (%) result.
DISPlay:PULSe:RESult:DUTRatio	Selects or queries whether to show the duty factor (ratio) result.
DISPlay:PULSe:RESult:FALL	Selects or queries whether to show the fall time in the results table.
DISPlay:PULSe:RESult:FRDeviation	Selects or queries whether to show the frequency deviation result.
DISPlay:PULSe:RESult:MFReqerror	Selects or queries whether to show the maximum frequency error result.
DISPlay:PULSe:RESult:MPHerror	Selects or queries whether to show the maximum phase error result.
DISPlay:PULSe:RESult:PHDeviation	Selects or queries whether to show the phase deviation result.
DISPlay:PULSe:RESult:PPFRequency	Selects or queries whether to show the pulse-pulse frequency result.
DISPlay:PULSe:RESult:PPOWer	Selects or queries whether to show the peak power in the results table.
DISPlay:PULSe:RESult:PPPHase	Selects or queries whether to show the pulse-pulse carrier phase result.
DISPlay:PULSe:RESult:RINTerval	Selects or queries whether to show the repetition interval result.
DISPlay:PULSe:RESult:RIPPle	Selects or queries whether to show the ripple in the results table.
DISPlay:PULSe:RESult:RISE	Selects or queries whether to show the rise time in the results table.
DISPlay:PULSe:RESult:RMSFreqerror	Selects or queries whether to show the RMS frequency error result.
DISPlay:PULSe:RESult:RMSPherror	Selects or queries whether to show the RMS phase error result.
DISPlay:PULSe:RESult:RRATe	Selects or queries whether to show the repetition rate result.

Table 2-14: Display commands (cont.)

Header	Description
DISPlay:PULSe:RESult:TIME	Selects or queries whether to show the time in the results table.
DISPlay:PULSe:RESult:WIDTh	Selects or queries whether to show the pulse width in the results table.
DISPlay:PULSe:SELect:NUMBer	Selects or queries the pulse to measure.
DISPlay:PULSe:SELect:RESult	Selects or queries which result is shown in the trace and statistics views.
DISPlay:PULSe:STATistics:MARKer:SHOW:STATe	Selects or queries whether to show the marker readout in the graph.
DISPlay:PULSe:STATistics:PLOT	Selects or queries how to show the statistics graph.
DISPlay:PULSe:STATistics:WINDow:TRACe:GRATicule:GRID:STATe	Selects or queries whether to show the graticule grid in the statistics view.
DISPlay:PULSe:STATistics:X:RSCale	Rescales the horizontal axis of the statistics graph.
DISPlay:PULSe:STATistics:X[:SCALe]:NUMBer	Sets or queries the horizontal scale (the number of pulses per division).
DISPlay:PULSe:STATistics:X[:SCALe]:OFFSet	Sets or queries the minimum horizontal value in the statistics view.
DISPlay:PULSe:STATistics:Y:RSCale	Rescales the vertical axis of the statistics graph.
DISPlay:PULSe:STATistics:Y[:SCALe]:FULL	Sets or queries the vertical full-scale in the statistics view.
DISPlay:PULSe:STATistics:Y[:SCALe]:OFFSet	Sets or queries the vertical offset in the statistics view.
DISPlay:PULSe:STATistics:Y[:SCALe]:STOP?	Queries the minimum vertical value in the statistics view.
DISPlay:PULSe:TRACe:MARKer:SHOW:STATe	Selects or queries whether to show the marker readout in the trace view.
DISPlay:PULSe:TRACe:POINt:SHOW	Selects or queries whether to show the measurement points and lines.
DISPlay:PULSe:TRACe:WINDow:TRACe:GRATicule:GRID:STATe	Selects or queries whether to show the graticule grid in the trace view.
DISPlay:PULSe:TRACe:X:RSCale	Rescales the horizontal axis of the pulse trace view.
DISPlay:PULSe:TRACe:X[:SCALe]	Sets or queries the horizontal full scale in the pulse trace view.
DISPlay:PULSe:TRACe:X[:SCALe]:FULL	Selects or queries the full-scale reference for the horizontal rescale.
DISPlay:PULSe:TRACe:X[:SCALe]:OFFSet	Sets or queries the minimum horizontal value in the pulse trace view.
DISPlay:PULSe:TRACe:X[:SCALe]:PDIVision	Sets or queries the horizontal full scale in the pulse trace view.
DISPlay:PULSe:TRACe:Y:RSCale	Rescales the vertical axis of the pulse trace view.
DISPlay:PULSe:TRACe:Y[:SCALe]:FULL	Sets or queries the vertical full scale in the pulse trace view.
DISPlay:PULSe:TRACe:Y[:SCALe]:OFFSet	Sets or queries the vertical offset in the pulse trace view.
DISPlay:PULSe:TRACe:Y[:SCALe]:STOP?	Queries the minimum vertical value in the pulse trace view.
DISPlay:SGRam subgroup	**Spectrogram measurement**
DISPlay:SGRam:FREQuency:AUTO	Rescales the horizontal (frequency) axis automatically.
DISPlay:SGRam:FREQuency:OFFSet	Sets or queries the horizontal (frequency) offset.
DISPlay:SGRam:FREQuency:SCAle	Sets or queries the horizontal (frequency) range.
DISPlay:SGRam:TIME:AUTO	Rescales the vertical axis automatically.
DISPlay:SGRam:TIME:OFFSet	Sets or queries the vertical axis (time) offset (bottom line number).
DISPlay:SGRam:TIME:OVERlap	Determines whether or not to allow overlap between adjacent FFT frames.
DISPlay:SGRam:TIME:SCAle	Sets or queries the vertical scale (the amount of time in each line).

Table 2-14: Display commands (cont.)

Header	Description
DISPlay:SPECtrum subgroup	**Spectrum measurement**
DISPlay:SPECtrum:MARKer:NOISe:MODE	Selects or queries whether to enable the marker noise mode.
DISPlay:SPECtrum:FREQuency:AUTO	Rescales the horizontal (frequency) axis automatically.
DISPlay:SPECtrum:FREQuency:OFFSet	Sets or queries the horizontal (frequency) offset.
DISPlay:SPECtrum:FREQuency[:SCAle]	Sets or queries the horizontal (frequency) range.
DISPlay:SPECtrum:SCALe:LOG:STATe	Selects or queries whether to set the horizontal axis logarithmic.
DISPlay:SPECtrum:WINDow:TRACe:GRATicule:GRID:STATe	Selects or queries whether to show the graticule grid.
DISPlay:SPECtrum:WINDow:TRACe:LEGend:STATe	Selects or queries whether to show the trace legend.
DISPlay:SPECtrum:X:LABel	Selects or queries the labels for the horizontal axis.
DISPlay:SPECtrum:Y[:SCALe]	Sets or queries the vertical range.
DISPlay:SPECtrum:Y[:SCALe]:AUTO	Rescales the vertical axis automatically.
DISPlay:SPECtrum:Y[:SCALe]:OFFSet	Sets or queries the vertical offset.
DISPlay:SPECtrum:Y[:SCALe]:PDIVision	Sets or queries the vertical scale (per division).
DISPlay:SPECtrum:Y[:SCALe]:RESet	Resets the vertical scale to the default values.
DISPlay:SPURious subgroup	**Spurious measurement**
DISPlay:SPURious:MARKer:SHOW:STATe	Selects or queries whether to show the readout for the selected marker.
DISPlay:SPURious:RESet:SCALe	Resets the horizontal and vertical scale to the default values.
DISPlay:SPURious:SCALe:LOG:STATe	Selects or queries whether to set the horizontal axis logarithmic.
DISPlay:SPURious:SELect:NUMBer	Selects or queries the spurious number.
DISPlay:SPURious:SHOW:LIMit	Selects or queries how to display the limits.
DISPlay:SPURious:WINDow:TRACe:GRATicule:GRID:STATe	Determines whether to show the graticule grid on screen.
DISPlay:SPURious:X[:SCALe]:AUTO	Rescales the horizontal axis automatically.
DISPlay:SPURious:X[:SCALe]:STARt	Sets or queries the minimum horizontal value of the spectrum graph.
DISPlay:SPURious:X[:SCALe]:STOP	Sets or queries the maximum horizontal value of the spectrum graph.
DISPlay:SPURious:Y[:SCALe]	Sets or queries the vertical range of the spectrum graph.
DISPlay:SPURious:Y[:SCALe]:AUTO	Rescales the vertical axis automatically.
DISPlay:SPURious:Y[:SCALe]:OFFSet	Sets or queries the vertical offset of the spectrum graph.
DISPlay:TOVerview subgroup	**Time overview**
DISPlay:TOVerview:WINDow:TRACe:GRATicule:GRID:STATe	Selects or queries whether to show the graticule grid in the time overview.
DISPlay:TOVerview:X[:SCALe]	Sets or queries the horizontal scale.
DISPlay:TOVerview:X[:SCALe]:AUTO	Sets the horizontal scale and offset automatically.
DISPlay:TOVerview:X[:SCALe]:OFFSet	Sets or queries the minimum horizontal value (left edge).
DISPlay:TOVerview:Y[:SCALe]	Sets or queries the vertical scale.
DISPlay:TOVerview:Y[:SCALe]:AUTO	Sets the vertical scale and offset automatically.

Table 2-14: Display commands (cont.)

Header	Description
DISPlay:TOVerview:Y[:SCALe]:OFFSet	Sets or queries the vertical offset.
DISPlay:TOVerview:Y[:SCALe]:RESCale	Rescales the vertical scale.

Fetch Commands

The FETCh commands retrieve the measurements from the data taken by the latest INITiate command.

To perform a FETCh operation on fresh data, use the READ commands, which acquire a new input signal and fetch the measurement results from that data.

Table 2-15: Fetch commands

Header	Description
FETCh basic command subgroup	**General fetch control**
FETCh:RFIN:IQ?	Returns time-domain IQ data for a specific acquisition data record.
FETCh:RFIN:IQ:HEADer?	Returns the header information for a specific acquisition data record.
FETCh:RFIN:IQ:SCALe?	Returns the scaling factor contained in the .tiq file header.
FETCh:RFIN:RECord:IDS?	Returns the beginning and end ID numbers of acquisition data.
FETCh:ACPower subgroup	**Channel power and ACPR measurement**
FETCh:ACPower?	Returns the ACPR measurement results.
FETCh:ACPower:CHANnel:POWer?	Returns the average power of the main channel.
FETCh:ACPower:SPECtrum?	Returns spectrum trace data of the ACPR measurement.
FETCh:AVTime subgroup	**Amplitude versus Time measurement**
FETCh:AVTime:AVERage?	Returns the RMS value.
FETCh:AVTime:{FIRSt|SECond|THIRd|FOURth}?	Returns the trace data.
FETCh:AVTime:MAXimum?	Returns the maximum value.
FETCh:AVTime:MAXLocation?	Returns the time at the maximum.
FETCh:AVTime:MINimum?	Returns the minimum value.
FETCh:AVTime:MINLocation?	Returns the time at the minimum.
FETCh:AVTime:RESult?	Returns the measurement results.
FETCh:CCDF subgroup	**CCDF measurement**
FETCh:CCDF?	Returns the CCDF measurement results.
FETCh:CCDF:{FIRSt|SECond|THIRd}:X?	Returns the horizontal values of the specified trace.
FETCh:CCDF:{FIRSt|SECond|THIRd}:XY?	Returns the horizontal and vertical value pairs of the specified trace.
FETCh:CCDF:{FIRSt|SECond|THIRd}[:Y]?	Returns the vertical values of the specified trace.
FETCh:CONSte subgroup (Option 21 only)	**Constellation measurement**
FETCh:CONSte:RESults?	Returns the constellation measurement results.
FETCh:CONSte:TRACe?	Returns the constellation trace data.
FETCh:DDEMod subgroup (Option 21 only)	**General purpose digital modulation measurements**
FETCh:DDEMod:STABle?	Returns the symbol table data.
FETCh:DDEMod:SYNCh:WORD:LENGth?	Returns the length of the synch word in the symbol table.
FETCh:DDEMod:SYNCh:WORD:POSition?	Returns the position of the synch word in the symbol table.

Table 2-15: Fetch commands (cont.)

Header	Description
FETCh:DPSA subgroup	**DPX spectrum measurement**
FETCh:DPSA:TRACe:AVERage?	Returns waveform data of the average trace.
FETCh:DPSA:TRACe:MATH?	Returns waveform data of the math trace.
FETCh:DPSA:TRACe:MAXimum?	Returns waveform data of the maximum trace.
FETCh:DPSA:TRACe:MINimum?	Returns waveform data of the minimum trace.
FETCh:EVM subgroup (Option 21 only)	**EVM versus Time measurement**
FETCh:EVM:FERRor?	Returns the frequency error.
FETCh:EVM:PEAK?	Returns the peak value.
FETCh:EVM:PINDex?	Returns the time at the EVM peak.
FETCh:EVM:RMS?	Returns the RMS value.
FETCh:EVM:TRACe?	Returns the EVM versus Time trace data.
FETCh:FVTime subgroup	**Frequency versus Time measurement**
FETCh:FVTime?	Returns the Frequency versus Time trace data.
FETCh:FVTime:MAXimum?	Returns the maximum value.
FETCh:FVTime:MAXLocation?	Returns the time at which the frequency drift is maximum.
FETCh:FVTime:MINimum?	Returns the minimum value.
FETCh:FVTime:MINLocation?	Returns the time at which the frequency drift is minimum.
FETCh:FVTime:RESult?	Returns the measurement results.
FETCh:IQVTime subgroup	**RF I&Q versus Time measurement**
FETCh:IQVTime:I?	Returns the I versus Time trace data.
FETCh:IQVTime:MAXimum?	Returns the maximum value.
FETCh:IQVTime:MAXLocation?	Returns the time at which the I or Q level is maximum.
FETCh:IQVTime:MINimum?	Returns the minimum value.
FETCh:IQVTime:MINLocation?	Returns the time at which the I or Q level is minimum.
FETCh:IQVTime:Q?	Returns the Q versus Time trace data.
FETCh:IQVTime:RESult?	Returns the measurement results.
FETCh:MCPower subgroup	**MCPR measurement**
FETCh:MCPower:ADJacent:CHANnels?	Returns the power of adjacent channels.
FETCh:MCPower:CHANnel:POWer?	Returns the reference power.
FETCh:MCPower:MAIN:CHANnels?	Returns the power of main channels.
FETCh:MCPower:SPECtrum?	Returns spectrum trace data.
FETCh:MERRor subgroup (Option 21 only)	**Magnitude error versus Time measurement**
FETCh:MERRor:FERRor?	Returns the frequency error.
FETCh:MERRor:PEAK?	Returns the peak value.
FETCh:MERRor:PINDex?	Returns the time at the magnitude error peak.
FETCh:MERRor:RMS?	Returns the RMS value.
FETCh:MERRor:TRACe?	Returns the Magnitude error versus Time trace data.

Table 2-15: Fetch commands (cont.)

Header	Description
FETCh:OBWidth subgroup	**Occupied Bandwidth measurement**
FETCh:OBWidth:FREQuency:ERRor?	Returns the frequency error.
FETCh:OBWidth:OBWidth:BANDwidth?	Returns the occupied bandwidth.
FETCh:OBWidth:OBWidth:LEFT:FREQuency?	Returns the left (lower) frequency of the occupied bandwidth.
FETCh:OBWidth:OBWidth:LEFT:LEVel?	Returns the level at the left frequency of the occupied bandwidth.
FETCh:OBWidth:OBWidth:POWer?	Returns the reference power in the Occupied Bandwidth measurement.
FETCh:OBWidth:OBWidth:RIGHt:FREQuency?	Returns the right (higher) frequency of the occupied bandwidth.
FETCh:OBWidth:OBWidth:RIGHt:LEVel?	Returns the level at the right frequency of the occupied bandwidth.
FETCh:OBWidth:SPECtrum?	Returns spectrum trace data of the Occupied Bandwidth measurement.
FETCh:OBWidth:XDBBandwidth:BANDwidth?	Returns the x dB bandwidth.
FETCh:OBWidth:XDBBandwidth:LEFT:FREQuency?	Returns the left (lower) frequency of the x dB bandwidth.
FETCh:OBWidth:XDBBandwidth:LEFT:LEVel?	Returns the level at the left frequency of the x dB bandwidth.
FETCh:OBWidth:XDBBandwidth:POWer?	Returns the reference power in the x dB bandwidth measurement.
FETCh:OBWidth:XDBBandwidth:RIGHt:FREQuency?	Returns the right (higher) frequency of the x dB bandwidth.
FETCh:OBWidth:XDBBandwidth:RIGHt:LEVel?	Returns the level at the right frequency of the x dB bandwidth.
FETCh:PERRor subgroup (Option 21 only)	**Phase error versus Time measurement**
FETCh:PERRor:FERRor?	Returns the frequency error.
FETCh:PERRor:PEAK?	Returns the peak value.
FETCh:PERRor:PINDex?	Returns the time at the phase error peak.
FETCh:PERRor:RMS?	Returns the RMS value.
FETCh:PERRor:TRACe?	Returns the Phase error versus Time trace data.
FETCh:PHVTime subgroup	**Phase versus Time measurement**
FETCh:PHVTime?	Returns the Phase versus Time trace data.
FETCh:PHVTime:MAXimum?	Returns the maximum value.
FETCh:PHVTime:MAXLocation?	Returns the time at which the phase is maximum.
FETCh:PHVTime:MINimum?	Returns the minimum value.
FETCh:PHVTime:MINLocation?	Returns the time at which the phase is minimum.
FETCh:PHVTime:RESult?	Returns the results.
FETCh:PNOise subgroup (Option 11 only)	**Phase noise measurements**
FETCh:PNOise:ALL?	Returns all the measurement results.
FETCh:PNOise:CARRier:FERRor?	Returns the carrier frequency error.
FETCh:PNOise:CARRier:POWer?	Returns the carrier power.
FETCh:PNOise:RESidual:FM?	Returns the residual FM.
FETCh:PNOise:RMS:PNOise?	Returns the RMS phase noise.
FETCh:PNOise:SPECtrum<x>:X?	Returns the frequencies of the specified trace.
FETCh:PNOise:SPECtrum<x>:XY?	Returns the frequency and phase noise pairs of the specified trace.
FETCh:PNOise:SPECtrum<x>[:Y]?	Returns the phase noise values of the specified trace.

Table 2-15: Fetch commands (cont.)

Header	Description
FETCh:PNOise:JITTer?	Returns the jitter.
FETCh:PULSe subgroup (Option 20 only)	**Pulsed RF measurements**
FETCh:PULSe[:RESult]:ATX?	Returns the average transmitted power in the results table.
FETCh:PULSe[:RESult]:AVERage?	Returns the average on power in the results table.
FETCh:PULSe[:RESult]:DROop?	Returns the droop in the results table.
FETCh:PULSe[:RESult]:DUTPct?	Returns the duty factor (%) in the results table.
FETCh:PULSe[:RESult]:DUTRatio?	Returns the duty factor (ratio) in the results table.
FETCh:PULSe[:RESult]:FALL?	Returns the fall time in the results table.
FETCh:PULSe[:RESult]:FRDeviation?	Returns the frequency deviation in the results table.
FETCh:PULSe[:RESult]:MFReqerror?	Returns the maximum frequency error in the results table.
FETCh:PULSe[:RESult]:MPHerror?	Returns the maximum phase error in the results table.
FETCh:PULSe[:RESult]:PHDeviation?	Returns the phase deviation in the results table.
FETCh:PULSe[:RESult]:PPFRequency?	Returns the pulse-pulse carrier frequency in the results table.
FETCh:PULSe[:RESult]:PPOWer?	Returns the peak power in the results table.
FETCh:PULSe[:RESult]:PPPHase?	Returns the pulse-pulse carrier phase in the results table.
FETCh:PULSe[:RESult]:RINTerval?	Returns the repetition interval in the results table.
FETCh:PULSe[:RESult]:RIPPle?	Returns the ripple in the results table.
FETCh:PULSe[:RESult]:RISE?	Returns the rise time in the results table.
FETCh:PULSe[:RESult]:RMSFreqerror?	Returns the RMS frequency error in the results table.
FETCh:PULSe[:RESult]:RMSPherror?	Returns the RMS phase error in the results table.
FETCh:PULSe[:RESult]:RRATe?	Returns the repetition rate in the results table.
FETCh:PULSe[:RESult]:TIME?	Returns the time in the results table.
FETCh:PULSe[:RESult]:WIDTh?	Returns the pulse width in the results table.
FETCh:PULSe:STATistics?	Returns the trace data of the pulse statistics measurement.
FETCh:PULSe:STATistics:ATX?	Returns the average transmitted power of the statistics.
FETCh:PULSe:STATistics:AVERage?	Returns the average on power of the statistics.
FETCh:PULSe:STATistics:DROop?	Returns the droop of the statistics.
FETCh:PULSe:STATistics:DUTPct?	Returns the duty factor (%) of the statistics.
FETCh:PULSe:STATistics:DUTRatio?	Returns the duty factor (ratio) of the statistics.
FETCh:PULSe:STATistics:FALL?	Returns the fall time of the statistics.
FETCh:PULSe:STATistics:FRDeviation?	Returns the frequency deviation of the statistics.
FETCh:PULSe:STATistics:MFReqerror?	Returns the maximum frequency error of the statistics.
FETCh:PULSe:STATistics:MPHerror?	Returns the maximum phase error of the statistics.
FETCh:PULSe:STATistics:PHDeviation?	Returns the phase deviation of the statistics.
FETCh:PULSe:STATistics:PPFRequency?	Returns the pulse-pulse carrier frequency of the statistics.
FETCh:PULSe:STATistics:PPOWer?	Returns the peak power of the statistics.
FETCh:PULSe:STATistics:PPPHase?	Returns the pulse-pulse carrier phase of the statistics.

Table 2-15: Fetch commands (cont.)

Header	Description
FETCh:PULSe:STATistics:RINTerval?	Returns the repetition interval of the statistics.
FETCh:PULSe:STATistics:RIPPle?	Returns the ripple of the statistics.
FETCh:PULSe:STATistics:RISE?	Returns the rise time of the statistics.
FETCh:PULSe:STATistics:RMSFreqerror?	Returns the RMS frequency error of the statistics.
FETCh:PULSe:STATistics:RMSPherror?	Returns the RMS phase error of the statistics.
FETCh:PULSe:STATistics:RRATe?	Returns the repetition rate of the statistics.
FETCh:PULSe:STATistics:WIDTh?	Returns the pulse width of the statistics.
FETCh:PULSe:TRACe:X?	Returns the time values of the pulse trace.
FETCh:PULSe:TRACe:XY?	Returns the horizontal (time) and vertical value pairs of the pulse trace.
FETCh:PULSe:TRACe[:Y]?	Returns the vertical values of the pulse trace.
FETCh:SGRam subgroup	**Spectrogram measurement**
FETCh:SGRam?	Returns the spectrogram trace data.
FETCh:SPECtrum subgroup	**Spectrum measurement**
FETCh:SPECtrum:TRACe<x>?	Returns the trace data in the Spectrum Analyzer measurement.
FETCh:SPURious subgroup	**Spurious measurement**
FETCh:SPURious:CARRier:POWer?	Returns the carrier power.
FETCh:SPURious:COUNt?	Returns the number of spurious signals.
FETCh:SPURious:PASS?	Returns the pass/fail limit test result.
FETCh:SPURious:SPECtrum:X?	Returns the frequencies of the spectrum trace.
FETCh:SPURious:SPECtrum:XY?	Returns the frequency and amplitude pairs of the spectrum trace.
FETCh:SPURious:SPECtrum[:Y]?	Returns the amplitudes of the spectrum trace.
FETCh:SPURious:SPUR<x>:AMPLitude:ABSolute?	Returns the absolute amplitude of the specified spurious signal.
FETCh:SPURious:SPUR<x>:AMPLitude:RELative?	Returns the relative amplitude of the specified spurious signal.
FETCh:SPURious:SPUR<x>:FREQuency:ABSolute?	Returns the absolute frequency of the specified spurious signal.
FETCh:SPURious:SPUR<x>:FREQuency:RELative?	Returns the relative frequency of the specified spurious signal.
FETCh:SPURious:SPUR<x>:LIMit:ABSolute?	Returns the absolute amplitude of the limit for a spurious signal.
FETCh:SPURious:SPUR<x>:LIMit:RELative?	Returns the relative amplitude of the limit for a spurious signal.
FETCh:SPURious:SPUR<x>:LIMit:VIOLation?	Returns whether the specified spurious signal exceeds the limit or not.
FETCh:SPURious:SPUR<x>:RANGe?	Returns the frequency range in which the spurious signal occurred.
FETCh:SQUality subgroup (Option 21 only)	**Signal quality measurement**
FETCh:SQUality:FREQuency:ERRor?	Returns the frequency error.
FETCh:SQUality:GAIN:IMBalance?	Returns the gain imbalance.
FETCh:SQUality:ORIGin:OFFSet?	Returns the origin offset.
FETCh:SQUality:PEAK:EVM?	Returns the peak EVM (%).
FETCh:SQUality:PEAK:EVM:DB?	Returns the peak EVM (dB).
FETCh:SQUality:PEAK:EVM:LOCation?	Returns the time at which the EVM is peak.
FETCh:SQUality:PEAK:MERRor?	Returns the peak magnitude error (%).

Table 2-15: Fetch commands (cont.)

Header	Description
FETCh:SQUality:PEAK:MERRor:DB?	Returns the peak magnitude error (dB).
FETCh:SQUality:PEAK:MERRor:LOCation?	Returns the time at which the magnitude error is peak.
FETCh:SQUality:PEAK:PERRor?	Returns the peak phase error.
FETCh:SQUality:PEAK:PERRor:LOCation?	Returns the time at which the phase error is peak.
FETCh:SQUality:QUADrature:ERRor?	Returns the quadrature error.
FETCh:SQUality:RHO?	Returns the r (waveform quality).
FETCh:SQUality:RMS:EVM?	Returns the RMS EVM (%).
FETCh:SQUality:RMS:EVM:DB?	Returns the RMS EVM (dB).
FETCh:SQUality:RMS:MER:DB?	Returns the RMS MER (dB).
FETCh:SQUality:RMS:MERRor?	Returns the RMS magnitude error (%).
FETCh:SQUality:RMS:MERRor:DB?	Returns the RMS magnitude error (dB).
FETCh:SQUality:RMS:PERRor?	Returns the RMS phase error.
FETCh:TOVerview subgroup	**Time overview**
FETCh:TOVerview?	Returns the trace data.

Initiate Commands

Use the INITiate commands to control the acquisition of data.

Table 2-16: Initiate commands

Header	Description
INITiate:CONTinuous	Selects or queries whether to acquire data continuously.
INITiate[:IMMediate]	Starts data acquisition.

Input Commands

Use the INPut commands to control the characteristics of the signal input.

Table 2-17: Input commands

Header	Description
INPut[:RF]:ATTenuation	Sets or queries the input attenuation.
INPut[:RF]:ATTenuation:AUTO	Selects or queries whether to set the attenuation automatically.
INPut[:RF]:ATTenuation:MONitor:STATe	Selects or queries whether to enable to monitor attenuator use.
INPut[:RF]:GAIN:STATe (Option 01 only)	Selects or queries whether to enable the internal preamplifier.
INPut:{MLEVel|RLEVel}	Sets or queries the reference level.

Mass Memory Commands

Use the MMEMory commands to manipulate files on the mass memory devices.

For the trace specifier TRACe<x>, refer to *Trace Mnemonics.* (See page 2-63.)

Table 2-18: Mass memory (MMEMory) commands

Header	Description
MMEMory basic command subgroup	**General file control**
MMEMory:CALibration:LOAD:CORRection:EXTernal:EDIT<x>	Loads an external loss table from a specified file.
MMEMory:CALibration:STORe:CORRection:EXTernal:EDIT<x>	Stores an external loss table to a specified file.
MMEMory:LOAD:IQ	Loads time-domain IQ waveform into the acquisition memory.
MMEMory:LOAD:STATe	Loads the instrument setup from a specified file.
MMEMory:LOAD:TRACe	Loads trace data from the specified file.
MMEMory:STORe:IQ	Stores time-domain IQ waveform in the acquisition memory to a file.
MMEMory:STORe:IQ:CSV	Stores time-domain IQ waveform to a file in the CSV format.
MMEMory:STORe:IQ:MAT	Stores time-domain IQ waveform to a file in the MATLAB format.
MMEMory:STORe:MSTate	Stores the measurement parameters in a specified file.
MMEMory:STORe:RESults	Stores the measurement results in a specified file.
MMEMory:STORe:STATe	Stores the instrument setup in a specified file.
MMEMory:STORe:TRACe	Stores trace data in a specified file.
MMEMory:AVTime subgroup	**Amplitude versus Time measurement**
MMEMory:AVTime:LOAD:TRACe<x>	Loads trace data from the specified file.
MMEMory:AVTime:STORe:TRACe<x>	Stores trace data in the specified file.
MMEMory:CCDF subgroup	**CCDF measurement**
MMEMory:CCDF:LOAD:TRACe<x>	Loads trace data from the specified file.
MMEMory:CCDF:STORe:TRACe<x>	Stores trace data in the specified file.
MMEMory:DPSA subgroup	**DPX spectrum measurement**
MMEMory:DPSA:LOAD:TRACe<x>	Loads trace data from the specified file.
MMEMory:DPSA:STORe:TRACe<x>	Stores trace data in the specified file.
MMEMory:FVTime subgroup	**Frequency versus Time measurement**
MMEMory:FVTime:LOAD:TRACe	Loads trace data from the specified file.
MMEMory:FVTime:STORe:TRACe	Stores trace data in the specified file.
MMEMory:IQVTime subgroup	**RF I&Q versus Time measurement**
MMEMory:IQVTime:LOAD:TRACe:I	Loads I trace data from the specified file.
MMEMory:IQVTime:LOAD:TRACe:Q	Loads Q trace data from the specified file.
MMEMory:IQVTime:STORe:TRACe:I	Stores I trace data in the specified file.
MMEMory:IQVTime:STORe:TRACe:Q	Stores Q trace data in the specified file.

Table 2-18: Mass memory (MMEMory) commands (cont.)

Header	Description
MMEMory:PHVTime subgroup	**Phase versus Time measurement**
MMEMory:PHVTime:LOAD:TRACe	Loads trace data from the specified file.
MMEMory:PHVTime:STORe:TRACe	Stores trace data in the specified file.
MMEMory:SPECtrum subgroup	**Spectrum measurement**
MMEMory:SPECtrum:LOAD:TRACe<x>	Loads trace data from the specified file.
MMEMory:SPECtrum:STORe:TRACe<x>	Stores trace data in the specified file.
MMEMory:SPURious subgroup	**Spurious measurement**
MMEMory:SPURious:LOAD:TABLe	Loads the spurious table from the specified file.
MMEMory:SPURious:STORe:TABLe	Stores the spurious table in the specified file.

Specifying the File

For loading and storing a file, specify the file following these rules

- You can omit the file extension to load and store data. The measurement-specific extension is automatically added.
- You can use the absolute path to specify the file name. For example, specify the *SAMPLE1* file in the *My Documents* folder on the C drive as `"C:\My Documents\SAMPLE1"`.
- If you omit the directory path, the default path is used, which is *C:\Program Files\Tektronix\RSA6100A* initially.

Once a file is saved to a different directory, the new directory will be used as the default for all load and store operations.

Output Commands

Use the OUTPut commands to control the characteristics of the signal output.

Table 2-19: Output commands

Header		Description
OUTPut:IF:{BANDwidth|BWIDth}	(Option 05 only)	Selects or queries the IF output filter.
OUTPut:IF[:STATe]	(Option 05 only)	Selects or queries whether to turn on or off IF output.
OUTPut:IQ[:STATe]	(Option 05 only)	Selects or queries whether to turn on or off IQ output.
OUTPut:NOISe[:STATe]		Selects or queries whether to turn on or off 28 V DC power.

Read Commands

The READ commands acquire an input signal once in the single mode and obtain the measurement results from that data.

To fetch the measurement results from the data currently residing in the memory without acquiring the input signal, use the FETCh commands.

Table 2-20: Read commands

Header	Description
READ:ACPower subgroup	**Channel power and ACPR measurement**
READ:ACPower?	Returns the ACPR measurement results.
READ:ACPower:CHANnel:POWer?	Returns the average power of the main channel.
READ:ACPower:SPECtrum?	Returns spectrum trace data of the ACPR measurement.
READ:AVTime subgroup	**Amplitude versus Time measurement**
READ:AVTime:AVERage?	Returns the RMS value.
READ:AVTime:{FIRSt|SECond|THIRd|FOURth}?	Returns the trace data.
READ:AVTime:MAXimum?	Returns the maximum value.
READ:AVTime:MAXLocation?	Returns the time at the maximum.
READ:AVTime:MINimum?	Returns the minimum value.
READ:AVTime:MINLocation?	Returns the time at the minimum.
READ:AVTime:RESult?	Returns the measurement results.
READ:CCDF subgroup	**CCDF measurement**
READ:CCDF?	Returns the CCDF measurement results.
READ:CCDF:{FIRSt|SECond|THIRd}:X?	Returns the horizontal values of the specified trace.
READ:CCDF:{FIRSt|SECond|THIRd}:XY?	Returns the horizontal and vertical value pairs of the specified trace.
READ:CCDF:{FIRSt|SECond|THIRd}[:Y]?	Returns the vertical values of the specified trace.
READ:CONSte subgroup (Option 21 only)	**Constellation measurement**
READ:CONSte:RESults?	Returns the constellation measurement results.
READ:CONSte:TRACe?	Returns the constellation trace data.
READ:DDEMod subgroup (Option 21 only)	**General purpose digital modulation measurements**
READ:DDEMod:STABle?	Returns the symbol table data.
READ:DPSA subgroup	**DPX spectrum measurement**
READ:DPSA:TRACe:AVERage?	Returns waveform data of the average trace.
READ:DPSA:TRACe:MATH?	Returns waveform data of the math trace.
READ:DPSA:TRACe:MAXimum?	Returns waveform data of the maximum trace.
READ:DPSA:TRACe:MINimum?	Returns waveform data of the minimum trace.
READ:EVM subgroup (Option 21 only)	**EVM versus Time measurement**
READ:EVM:FERRor?	Returns the frequency error.
READ:EVM:PEAK?	Returns the peak value.

Table 2-20: Read commands (cont.)

Header	Description
READ:EVM:PINDex?	Returns the time at the EVM peak.
READ:EVM:RMS?	Returns the RMS value.
READ:EVM:TRACe?	Returns the EVM versus Time trace data.
READ:FVTime subgroup	**Frequency versus Time measurement**
READ:FVTime?	Returns the Frequency versus Time trace data.
READ:FVTime:MAXimum?	Returns the maximum value.
READ:FVTime:MAXLocation?	Returns the time at which the frequency drift is maximum.
READ:FVTime:MINimum?	Returns the minimum value.
READ:FVTime:MINLocation?	Returns the time at which the frequency drift is minimum.
READ:FVTime:RESult?	Returns the measurement results.
READ:IQVTime subgroup	**RF I&Q versus Time measurement**
READ:IQVTime:I?	Returns the I versus Time trace data.
READ:IQVTime:MAXimum?	Returns the maximum value.
READ:IQVTime:MAXLocation?	Returns the time at which the I or Q level is maximum.
READ:IQVTime:MINimum?	Returns the minimum value.
READ:IQVTime:MINLocation?	Returns the time at which the I or Q level is minimum.
READ:IQVTime:Q?	Returns the Q versus Time trace data.
READ:IQVTime:RESult?	Returns the measurement results.
READ:MCPower subgroup	**MCPR measurement**
READ:MCPower:ADJacent:CHANnels?	Returns the power of adjacent channels.
READ:MCPower:CHANnel:POWer?	Returns the reference power.
READ:MCPower:MAIN:CHANnels?	Returns the power of main channels.
READ:MCPower:SPECtrum?	Returns spectrum trace data.
READ:MERRor subgroup (Option 21 only)	**Magnitude error versus Time measurement**
READ:MERRor:FERRor?	Returns the frequency error.
READ:MERRor:PEAK?	Returns the peak value.
READ:MERRor:PINDex?	Returns the time at the magnitude error peak.
READ:MERRor:RMS?	Returns the RMS value.
READ:MERRor:TRACe?	Returns the Magnitude error versus Time trace data.
READ:PERRor subgroup (Option 21 only)	**Phase error versus Time measurement**
READ:PERRor:FERRor?	Returns the frequency error.
READ:PERRor:PEAK?	Returns the peak value.
READ:PERRor:PINDex?	Returns the time at the phase error peak.
READ:PERRor:RMS	Returns the RMS value.
READ:PERRor:TRACe?	Returns the Phase error versus Time trace data.
READ:OBWidth subgroup	**Occupied Bandwidth measurement**
READ:OBWidth:FREQuency:ERRor?	Returns the frequency error.

Table 2-20: Read commands (cont.)

Header	Description
READ:OBWidth:OBWidth:BANDwidth?	Returns the occupied bandwidth.
READ:OBWidth:OBWidth:LEFT:FREQuency?	Returns the left (lower) frequency of the occupied bandwidth.
READ:OBWidth:OBWidth:LEFT:LEVel?	Returns the level at the left frequency of the occupied bandwidth.
READ:OBWidth:OBWidth:POWer?	Returns the reference power in the Occupied Bandwidth measurement.
READ:OBWidth:OBWidth:RIGHt:FREQuency?	Returns the right (higher) frequency of the occupied bandwidth.
READ:OBWidth:OBWidth:RIGHt:LEVel?	Returns the level at the right frequency of the occupied bandwidth.
READ:OBWidth:SPECtrum?	Returns spectrum trace data of the Occupied Bandwidth measurement.
READ:OBWidth:XDBBandwidth:BANDwidth?	Returns the x dB bandwidth.
READ:OBWidth:XDBBandwidth:LEFT:FREQuency?	Returns the left (lower) frequency of the x dB bandwidth.
READ:OBWidth:XDBBandwidth:LEFT:LEVel?	Returns the level at the left frequency of the x dB bandwidth.
READ:OBWidth:XDBBandwidth:POWer?	Returns the reference power in the x dB bandwidth measurement.
READ:OBWidth:XDBBandwidth:RIGHt:FREQuency?	Returns the right (higher) frequency of the x dB bandwidth.
READ:OBWidth:XDBBandwidth:RIGHt:LEVel?	Returns the level at the right frequency of the x dB bandwidth.
READ:PHVTime subgroup	**Phase versus Time measurement**
READ:PHVTime?	Returns the Phase versus Time trace data.
READ:PHVTime:MAXimum?	Returns the maximum value.
READ:PHVTime:MAXLocation?	Returns the time at which the phase is maximum.
READ:PHVTime:MINimum?	Returns the minimum value.
READ:PHVTime:MINLocation?	Returns the time at which the phase is minimum.
READ:PHVTime:RESult?	Returns the results.
READ:PNOise subgroup (Option 11 only)	**Phase noise measurements**
READ:PNOise:ALL?	Returns all the measurement results.
READ:PNOise:CARRier:FERRor?	Returns the carrier frequency error.
READ:PNOise:CARRier:POWer?	Returns the carrier power.
READ:PNOise:RESidual:FM?	Returns the residual FM.
READ:PNOise:RMS:PNOise?	Returns the RMS phase noise.
READ:PNOise:SPECtrum<x>:X?	Returns the frequencies of the specified trace.
READ:PNOise:SPECtrum<x>:XY?	Returns the frequency and phase noise pairs of the specified trace.
READ:PNOise:SPECtrum<x>[:Y]?	Returns the phase noise values of the specified trace.
READ:PNOise:JITTer?	Returns the jitter.
READ:PULSe subgroup (Option 20 only)	**Pulsed RF measurements**
READ:PULSe[:RESult]:ATX?	Returns the average transmitted power in the results table.
READ:PULSe[:RESult]:AVERage?	Returns the average on power in the results table.
READ:PULSe[:RESult]:DROop?	Returns the droop in the results table.
READ:PULSe[:RESult]:DUTPct?	Returns the duty factor (%) in the results table.
READ:PULSe[:RESult]:DUTRatio?	Returns the duty factor (ratio) in the results table.
READ:PULSe[:RESult]:FALL?	Returns the fall time in the results table.

Table 2-20: Read commands (cont.)

Header	Description
READ:PULSe[:RESult]:FRDeviation?	Returns the frequency deviation in the results table.
READ:PULSe[:RESult]:MFReqerror?	Returns the maximum frequency error in the results table.
READ:PULSe[:RESult]:MPHerror?	Returns the maximum phase error in the results table.
READ:PULSe[:RESult]:PHDeviation?	Returns the phase deviation in the results table.
READ:PULSe[:RESult]:PPFRequency?	Returns the pulse-pulse carrier frequency in the results table.
READ:PULSe[:RESult]:PPOWer?	Returns the peak power in the results table.
READ:PULSe[:RESult]:PPPHase?	Returns the pulse-pulse carrier phase in the results table.
READ:PULSe[:RESult]:RINTerval?	Returns the repetition interval in the results table.
READ:PULSe[:RESult]:RIPPle?	Returns the ripple in the results table.
READ:PULSe[:RESult]:RISE?	Returns the rise time in the results table.
READ:PULSe[:RESult]:RMSFreqerror?	Returns the RMS frequency error in the results table.
READ:PULSe[:RESult]:RMSPherror?	Returns the RMS phase error in the results table.
READ:PULSe[:RESult]:RRATe?	Returns the repetition rate in the results table.
READ:PULSe[:RESult]:TIME?	Returns the time in the results table.
READ:PULSe[:RESult]:WIDTh?	Returns the pulse width in the results table.
READ:PULSe:STATistics?	Returns the trace data of the pulse statistics measurement.
READ:PULSe:STATistics:ATX?	Returns the average transmitted power of the statistics.
READ:PULSe:STATistics:AVERage?	Returns the average on power of the statistics.
READ:PULSe:STATistics:DROop?	Returns the droop of the statistics.
READ:PULSe:STATistics:DUTPct?	Returns the duty factor (%) of the statistics.
READ:PULSe:STATistics:DUTRatio?	Returns the duty factor (ratio) of the statistics.
READ:PULSe:STATistics:FALL?	Returns the fall time of the statistics.
READ:PULSe:STATistics:FRDeviation?	Returns the frequency deviation of the statistics.
READ:PULSe:STATistics:MFReqerror?	Returns the maximum frequency error of the statistics.
READ:PULSe:STATistics:MPHerror?	Returns the maximum phase error of the statistics.
READ:PULSe:STATistics:PHDeviation?	Returns the phase deviation of the statistics.
READ:PULSe:STATistics:PPFRequency?	Returns the pulse-pulse carrier frequency of the statistics.
READ:PULSe:STATistics:PPOWer?	Returns the peak power of the statistics.
READ:PULSe:STATistics:PPPHase?	Returns the pulse-pulse carrier phase of the statistics.
READ:PULSe:STATistics:RINTerval?	Returns the repetition interval of the statistics.
READ:PULSe:STATistics:RIPPle?	Returns the ripple of the statistics.
READ:PULSe:STATistics:RISE?	Returns the rise time of the statistics.
READ:PULSe:STATistics:RMSFreqerror?	Returns the RMS frequency error of the statistics.
READ:PULSe:STATistics:RMSPherror?	Returns the RMS phase error of the statistics.
READ:PULSe:STATistics:RRATe?	Returns the repetition rate of the statistics.
READ:PULSe:STATistics:WIDTh?	Returns the pulse width of the statistics.
READ:PULSe:TRACe:X?	Returns the time values of the pulse trace.

Table 2-20: Read commands (cont.)

Header	Description
READ:PULSe:TRACe:XY?	Returns the horizontal (time) and vertical value pairs of the pulse trace.
READ:PULSe:TRACe[:Y]?	Returns the vertical values of the pulse trace.
READ:SGRam subgroup	**Spectrogram measurement**
READ:SGRam?	Returns the spectrogram trace data.
READ:SPECtrum subgroup	**Spectrum measurement**
READ:SPECtrum:TRACe<x>?	Returns the trace data in the Spectrum Analyzer measurement.
READ:SPURious subgroup	**Spurious measurement**
READ:SPURious:CARRier:POWer?	Returns the carrier power.
READ:SPURious:COUNt?	Returns the number of spurious signals.
READ:SPURious:PASS?	Returns the pass/fail limit test result.
READ:SPURious:SPECtrum:X?	Returns the frequencies of the spectrum trace.
READ:SPURious:SPECtrum:XY?	Returns the frequency and amplitude pairs of the spectrum trace.
READ:SPURious:SPECtrum[:Y]?	Returns the amplitudes of the spectrum trace.
READ:SPURious:SPUR<x>:AMPLitude:ABSolute?	Returns the absolute amplitude of the specified spurious signal.
READ:SPURious:SPUR<x>:AMPLitude:RELative?	Returns the relative amplitude of the specified spurious signal.
READ:SPURious:SPUR<x>:FREQuency:ABSolute?	Returns the absolute frequency of the specified spurious signal.
READ:SPURious:SPUR<x>:FREQuency:RELative?	Returns the relative frequency of the specified spurious signal.
READ:SPURious:SPUR<x>:LIMit:ABSolute?	Returns the absolute amplitude of the limit for a spurious signal.
READ:SPURious:SPUR<x>:LIMit:RELative?	Returns the relative amplitude of the limit for a spurious signal.
READ:SPURious:SPUR<x>:LIMit:VIOLation?	Returns whether the specified spurious signal exceeds the limit or not.
READ:SPURious:SPUR<x>:RANGe?	Returns the frequency range in which the spurious signal occurred.
READ:SQUality subgroup (Option 21 only)	**Signal quality measurement**
READ:SQUality:FREQuency:ERRor?	Returns the frequency error.
READ:SQUality:GAIN:IMBalance?	Returns the gain imbalance.
READ:SQUality:ORIGin:OFFSet?	Returns the origin offset.
READ:SQUality:PEAK:EVM?	Returns the peak EVM (%).
READ:SQUality:PEAK:EVM:DB?	Returns the peak EVM (dB).
READ:SQUality:PEAK:EVM:LOCation?	Returns the time at which the EVM is peak.
READ:SQUality:PEAK:MERRor?	Returns the peak magnitude error (%).
READ:SQUality:PEAK:MERRor:DB?	Returns the peak magnitude error (dB).
READ:SQUality:PEAK:MERRor:LOCation?	Returns the time at which the magnitude error is peak.
READ:SQUality:PEAK:PERRor?	Returns the peak phase error.
READ:SQUality:PEAK:PERRor:LOCation?	Returns the time at which the phase error is peak.
READ:SQUality:QUADrature:ERRor?	Returns the quadrature error.
READ:SQUality:RHO?	Returns the r (waveform quality).
READ:SQUality:RMS:EVM?	Returns the RMS EVM (%).
READ:SQUality:RMS:EVM:DB?	Returns the RMS EVM (dB).

Table 2-20: Read commands (cont.)

Header	Description
READ:SQUality:RMS:MER:DB?	Returns the RMS MER (dB).
READ:SQUality:RMS:MERRor?	Returns the RMS magnitude error (%).
READ:SQUality:RMS:MERRor:DB?	Returns the RMS magnitude error (dB).
READ:SQUality:RMS:PERRor?	Returns the RMS phase error.
READ:TOVerview subgroup	**Time overview**
READ:TOVerview?	Returns the trace data.

Sense Commands

Use the SENSe commands to set up detailed measurement conditions.

Table 2-21: Sense commands

Header	Description
[SENSe] basic command subgroup	**General analysis parameter control**
[SENSe]:ACQuisition:{BANDwidth|BWIDth}	Sets or queries the acquisition bandwidth.
[SENSe]:ACQuisition:FFRame:ACTual?	Queries the actual number of Fast Frames.
[SENSe]:ACQuisition:FFRame:LIMit	Sets or queries the limit number of Fast Frames.
[SENSe]:ACQuisition:FFRame:STATe	Determines whether to enable or disable the Fast Frame.
[SENSe]:ACQuisition:MEMory:AVAilable:SAMPles?	Returns the amount of acquisition memory available in the instrument.
[SENSe]:ACQuisition:MEMory:CAPacity[:TIME]?	Returns the acquisition memory capacity.
[SENSe]:ACQuisition:MEMory:USED[:PERCent]?	Returns the percentage of the capacity used.
[SENSe]:ACQuisition:MODE	Selects or queries the acquisition mode.
[SENSe]:ACQuisition:SAMPles	Sets or queries the acquisition samples.
[SENSe]:ACQuisition:SEConds	Sets or queries the acquisition length.
[SENSe]:ANALysis:ADVanced:DITHer	Determines whether to enable or disable dithering.
[SENSe]:ANALysis:LENGth	Sets or queries the analysis length.
[SENSe]:ANALysis:LENGth:ACTual?	Queries the actual analysis length.
[SENSe]:ANALysis:LENGth:AUTO	Selects or queries whether to set the analysis length automatically.
[SENSe]:ANALysis:REFerence	Selects or queries the analysis time reference.
[SENSe]:ANALysis:STARt	Sets or queries the analysis offset time.
[SENSe]:ANALysis:STARt:AUTO	Selects or queries whether to set the analysis offset automatically.
[SENSe]:MEASurement:FREQuency	Sets or queries the measurement frequency.
[SENSe]:POWer:UNITs	Selects or queries the unit of power.
[SENSe]:REANalyze	Have all measurements reanalyze the current acquisition record.
[SENSe]:ROSCillator:SOURce	Selects or queries the reference oscillator source.
[SENSe]:SPECtrum:LENGth	Sets or queries the spectrum length.
[SENSe]:SPECtrum:LENGth:ACTual?	Queries the actual spectrum length.
[SENSe]:SPECtrum:LENGth:AUTO	Selects or queries whether to set the spectrum length automatically.
[SENSe]:SPECtrum:STARt	Sets or queries the spectrum offset time.
[SENSe]:SPECtrum:TIMe:MODe	Selects or queries whether to set the spectrum time automatically.
[SENSe]:USETtings	Updates the analyzer settings.
[SENSe]:ACPower subgroup	**Channel power and ACPR measurement**
[SENSe]:ACPower:AVERage	Selects or queries how to average waveform.
[SENSe]:ACPower:AVERage:COUNt	Sets or queries the number of traces for averaging.
[SENSe]:ACPower:{BANDwidth|BWIDth}[:RESolution]	Sets or queries the resolution bandwidth (RBW).

Table 2-21: Sense commands (cont.)

Header	Description
[SENSe]:ACPower:{BANDwidth|BWIDth}[:RESolution]:ACTual?	Queries the actual resolution bandwidth (RBW).
[SENSe]:ACPower:{BANDwidth|BWIDth}[:RESolution]:AUTO	Selects or queries whether to set the RBW automatically.
[SENSe]:ACPower:{BANDwidth|BWIDth}:VIDeo	Sets or queries the video bandwidth (VBW).
[SENSe]:ACPower:{BANDwidth|BWIDth}:VIDeo:STATe	Selects or queries whether to enable the video bandwidth (VBW).
[SENSe]:ACPower:CHANnel:{BANDwidth|BWIDth}	Sets or queries the channel bandwidth.
[SENSe]:ACPower:CHANnel:FILTer	Selects or queries the adjacent channel filter.
[SENSe]:ACPower:CHANnel:PAIRs	Sets or queries the number of adjacent channel pairs.
[SENSe]:ACPower:CHANnel:SPACing	Sets or queries the channel-to-channel spacing.
[SENSe]:ACPower:CHIPrate	Sets or queries the chip rate.
[SENSe]:ACPower:CLEar:RESults	Restarts the average trace.
[SENSe]:ACPower:FREQuency	Sets or queries the center frequency.
[SENSe]:ACPower:FREQuency:STEP	Sets or queries the frequency step size.
[SENSe]:ACPower:FREQuency:STEP:AUTO	Selects or queries whether to set the frequency step size automatically.
[SENSe]:ACPower:NFLoor:STATe	Selects or queries whether to enable the correction for noise floor.
[SENSe]:ACPower:OPTimize:SPAN	Selects or queries the optimization method.
[SENSe]:ACPower:RRCRolloff	Sets or queries the filter parameter for the Root Raised Cosine filter.
[SENSe]:AVTime subgroup	**Amplitude versus Time measurement**
[SENSe]:AVTime:{BANDwidth|BWIDth}	Sets or queries the time-domain bandwidth filter.
[SENSe]:AVTime:{BANDwidth|BWIDth}:ACTual?	Queries the actual resolution bandwidth (RBW).
[SENSe]:AVTime:CLEar:RESults	Restarts multi-trace functions (Average and Max/Min Hold).
[SENSe]:AVTime:MAXTracepoints	Selects or queries the maximum trace points.
[SENSe]:AVTime:METHod	Selects or queries the method to set the measurement bandwidth.
[SENSe]:AVTime:SPAN	Sets or queries the frequency span.
[SENSe]:CCDF subgroup	**CCDF measurement**
[SENSe]:CCDF:{BANDwidth|BWIDth}	Sets or queries the CCDF measurement bandwidth.
[SENSe]:CCDF:CLEar	Clears the CCDF accumulator and restarts the measurement.
[SENSe]:CCDF:TIME:TOTal:LENGth	Sets or queries the CCDF measurement time.
[SENSe]:CCDF:TIME:TYPE	Selects or queries how to repeat the CCDF measurement.
[SENSe]:DDEMod subgroup (Option 21 only)	**General purpose digital modulation measurements**
[SENSe]:DDEMod:ANALysis:LENGth	Sets or queries the analysis length.
[SENSe]:DDEMod:ANALysis:LENGth:ACTual?	Queries the actual analysis length.
[SENSe]:DDEMod:ANALysis:LENGth:AUTO	Selects or queries whether to set the analysis length automatically.
[SENSe]:DDEMod:BURSt:DETect	Selects or queries how to detect bursts.
[SENSe]:DDEMod:BURSt:THReshold	Sets or queries the threshold level to determine a burst.
[SENSe]:DDEMod:CARRier:OFFSet	Sets or queries the carrier frequency offset.

Table 2-21: Sense commands (cont.)

Header	Description
[SENSe]:DDEMod:FILTer:ALPHa	Sets or queries the filter factor (a/BT).
[SENSe]:DDEMod:FILTer:MEASurement	Selects or queries the measurement filter.
[SENSe]:DDEMod:FILTer:REFerence	Selects or queries the reference filter.
[SENSe]:DDEMod:MAGNitude:NORMalize	Selects or queries the method for the magnitude normalization.
[SENSe]:DDEMod:MODulation:TYPE	Selects or queries the modulation type.
[SENSe]:DDEMod:SRATe	Sets or queries the symbol rate.
[SENSe]:DDEMod:SWAP:IQ	Selects or queries whether or not to swap I and Q data.
[SENSe]:DDEMod:SYMBol:POINts	Sets or queries the number of points per symbol.
[SENSe]:DDEMod:SYNCh:WORD	Selects or queries whether to enable the synchronization word.
[SENSe]:DDEMod:SYNCh:WORD:SYMBol	Sets or queries the synchronization word.
[SENSe]:DDEMod:TIMe:UNITs	Selects or queries the fundamental unit of time.
[SENSe]:DPSA subgroup	**DPX spectrum measurement**
[SENSe]:DPSA:AUDio:DEMod:GAIN	Sets or queries the gain for the audio demodulation.
[SENSe]:DPSA:AUDio:DEMod:RXBWidth	Sets or queries the receiver bandwidth for the audio demodulation.
[SENSe]:DPSA:AUDio:DEMod:RXFRequency?	Queries the receiver frequency for the audio demodulation.
[SENSe]:DPSA:AUDio:DEMod:STATe	Selects or queries whether to enable or disable the audio demodulation.
[SENSe]:DPSA:AUDio:DEMod:TUNE	Selects or queries how to determine the tuning frequency.
[SENSe]:DPSA:AUDio:DEMod:TYPE	Selects or queries the modulation type for the audio demodulation.
[SENSe]:DPSA:{BANDwidth|BWIDth}[:RESolution]	Sets or queries the resolution bandwidth (RBW).
[SENSe]:DPSA:{BANDwidth|BWIDth}[:RESolution]:AUTO	Selects or queries whether to set the RBW automatically.
[SENSe]:DPSA:CLEar:RESults	Restarts multi-trace functions (Average and Max/Min Hold).
[SENSe]:DPSA:COLor	Selects or queries the color palette of three-dimensional graphs.
[SENSe]:DPSA:COLor:MAXimum	Sets or queries the maximum value of the color axis.
[SENSe]:DPSA:COLor:MINimum	Sets or queries the minimum value of the color axis.
[SENSe]:DPSA:FREQuency:CENTer	Sets or queries the center frequency.
[SENSe]:DPSA:FREQuency:SPAN	Sets or queries the frequency span.
[SENSe]:DPSA:FREQuency:STARt	Sets or queries the measurement start frequency.
[SENSe]:DPSA:FREQuency:STEP	Sets or queries the frequency step size.
[SENSe]:DPSA:FREQuency:STEP:AUTO	Selects or queries whether to set the frequency step size automatically.
[SENSe]:DPSA:FREQuency:STOP	Sets or queries the measurement stop frequency.
[SENSe]:FVTime subgroup	**Frequency versus Time measurement**
[SENSe]:FVTime:CLEar:RESults	Restarts multi-trace functions (Average and Max/Min Hold).
[SENSe]:FVTime:FREQuency:CENTer	Sets or queries the center frequency.
[SENSe]:FVTime:FREQuency:SPAN	Sets or queries the frequency span.
[SENSe]:FVTime:FREQuency:STARt	Sets or queries the measurement start frequency.
[SENSe]:FVTime:FREQuency:STEP	Sets or queries the frequency step size.

Table 2-21: Sense commands (cont.)

Header	Description
[SENSe]:FVTime:FREQuency:STEP:AUTO	Selects or queries whether to set the frequency step size automatically.
[SENSe]:FVTime:FREQuency:STOP	Sets or queries the measurement stop frequency.
[SENSe]:FVTime:MAXTracepoints	Selects or queries the maximum trace points.
[SENSe]:IQVTime subgroup	**RF I&Q versus Time measurement**
[SENSe]:IQVTime:CLEar:RESults	Restarts multi-trace functions (Average and Max/Min Hold).
[SENSe]:IQVTime:FREQuency:CENTer	Sets or queries the center frequency.
[SENSe]:IQVTime:FREQuency:SPAN	Sets or queries the frequency span.
[SENSe]:IQVTime:FREQuency:STARt	Sets or queries the measurement start frequency.
[SENSe]:IQVTime:FREQuency:STEP	Sets or queries the frequency step size.
[SENSe]:IQVTime:FREQuency:STEP:AUTO	Selects or queries whether to set the frequency step size automatically.
[SENSe]:IQVTime:FREQuency:STOP	Sets or queries the measurement stop frequency.
[SENSe]:IQVTime:MAXTracepoints	Selects or queries the maximum trace points.
[SENSe]:MCPower subgroup	**MCPR measurement**
[SENSe]:MCPower:AVERage	Selects or queries how to average waveform.
[SENSe]:MCPower:AVERage:COUNt	Sets or queries the number of waveforms for average.
[SENSe]:MCPower:{BANDwidth|BWIDth}[:RESolution]	Sets or queries the resolution bandwidth (RBW).
[SENSe]:MCPower:{BANDwidth|BWIDth}[:RESolution]:ACTual?	Queries the actual resolution bandwidth (RBW).
[SENSe]:MCPower:{BANDwidth|BWIDth}[:RESolution]:AUTO	Selects or queries whether to set the RBW automatically.
[SENSe]:MCPower:{BANDwidth|BWIDth}:VIDeo	Sets or queries the video bandwidth (VBW).
[SENSe]:MCPower:{BANDwidth|BWIDth}:VIDeo:STATe	Selects or queries whether to enable the video bandwidth (VBW).
[SENSe]:MCPower:CHANnel:ADJacent:ADD	Adds a pair of upper and lower adjacent channels.
[SENSe]:MCPower:CHANnel:ADJacent:DELete	Deletes a selected adjacent channel.
[SENSe]:MCPower:CHANnel:FILTer	Selects or queries the measurement filter.
[SENSe]:MCPower:CHANnel:MAIN:{BANDwidth|BWIDth}	Sets or queries the frequency bandwidth of the main channels.
[SENSe]:MCPower:CHANnel:MAIN:COUNt	Sets or queries the number of main channels.
[SENSe]:MCPower:CHANnel:MAIN:INACtive	Makes a main channel inactive or queries the inactive main channels.
[SENSe]:MCPower:CHANnel:MAIN:SPACing	Sets or queries the main channel spacing.
[SENSe]:MCPower:CHIPrate	Sets or queries the chip rate.
[SENSe]:MCPower:CLEar:RESults	Restarts the average trace.
[SENSe]:MCPower:FREQuency	Sets or queries the center frequency.
[SENSe]:MCPower:FREQuency:STEP	Sets or queries the frequency step size.
[SENSe]:MCPower:FREQuency:STEP:AUTO	Selects or queries whether to set the frequency step size automatically.
[SENSe]:MCPower:NFLoor:STATe	Selects or queries whether to enable the correction for noise floor.
[SENSe]:MCPower:OPTimize:SPAN	Selects or queries the optimization method.

Table 2-21: Sense commands (cont.)

Header	Description
[SENSe]:MCPower:RCHannels?	Queries the power reference.
[SENSe]:MCPower:RCHannels:MAIN<x>	Sets the power reference to the main channel with the index (<x>).
[SENSe]:MCPower:RCHannels:TOTal	Sets the power reference to the total power of all the active channels.
[SENSe]:MCPower:RRCRolloff	Sets or queries the filter parameter for the Root Raised Cosine filter.
[SENSe]:OBWidth subgroup	**Occupied Bandwidth measurement**
[SENSe]:OBWidth:AVERage	Selects or queries whether to enable or disable averaging.
[SENSe]:OBWidth:AVERage:COUNt	Sets or queries the number of measurements for averaging.
[SENSe]:OBWidth:{BANDwidth|BWIDth}:MEASurement	Sets or queries the measurement bandwidth.
[SENSe]:OBWidth:{BANDwidth|BWIDth}[:RESolution]	Sets or queries the resolution bandwidth (RBW).
[SENSe]:OBWidth:{BANDwidth|BWIDth}[:RESolution]:ACTual?	Queries the actual resolution bandwidth (RBW).
[SENSe]:OBWidth:{BANDwidth|BWIDth}[:RESolution]:AUTO	Selects or queries whether to set the RBW automatically.
[SENSe]:OBWidth:{BANDwidth|BWIDth}:VIDeo	Sets or queries the video bandwidth (VBW).
[SENSe]:OBWidth:{BANDwidth|BWIDth}:VIDeo:STATe	Selects or queries whether to enable the video bandwidth (VBW).
[SENSe]:OBWidth:CLEar:RESults	Restarts the averaging.
[SENSe]:OBWidth:FREQuency:CENTer	Sets or queries the center frequency.
[SENSe]:OBWidth:FREQuency:STEP	Sets or queries the frequency step size.
[SENSe]:OBWidth:FREQuency:STEP:AUTO	Determines whether to set the frequency step size automatically.
[SENSe]:OBWidth:PERCent	Sets or queries the occupied bandwidth percent power.
[SENSe]:OBWidth:XDBLevel	Sets or queries the x dB level.
[SENSe]:PHVTime subgroup	**Phase versus Time measurement**
[SENSe]:PHVTime:CLEar:RESults	Restarts multi-trace functions (Average and Max/Min Hold).
[SENSe]:PHVTime:FREQuency:CENTer	Sets or queries the center frequency.
[SENSe]:PHVTime:FREQuency:SPAN	Sets or queries the frequency span.
[SENSe]:PHVTime:FREQuency:STARt	Sets or queries the measurement start frequency.
[SENSe]:PHVTime:FREQuency:STEP	Sets or queries the frequency step size.
[SENSe]:PHVTime:FREQuency:STEP:AUTO	Selects or queries whether to set the frequency step size automatically.
[SENSe]:PHVTime:FREQuency:STOP	Sets or queries the measurement stop frequency.
[SENSe]:PHVTime:MAXTracepoints	Selects or queries the maximum trace points.
[SENSe]:PNOise subgroup (Option 11 only)	**Phase noise measurement**
[SENSe]:PNOise:AVERage:COUNt	Sets or queries the number of traces to combine for averaging.
[SENSe]:PNOise:AVERage:ENABle	Selects or queries whether to enable or disable averaging trace.
[SENSe]:PNOise:CARRier:FREQuency:TRACk	Selects or queries whether to enable tracking the carrier frequency.
[SENSe]:PNOise:CARRier:THReshold	Sets or queries the threshold level to detect the carrier.
[SENSe]:PNOise:CLEar:RESults	Restarts the average process.

Table 2-21: Sense commands (cont.)

Header	Description
[SENSe]:PNOise:FREQuency:INTegration:OFFSet:STARt	Sets or queries the start offset frequency for integration.
[SENSe]:PNOise:FREQuency:INTegration:OFFSet:STOP	Sets or queries the stop offset frequency for integration.
[SENSe]:PNOise:FREQuency:PLOT:OFFSet:STARt	Sets or queries the start offset frequency for plot.
[SENSe]:PNOise:FREQuency:PLOT:OFFSet:STOP	Sets or queries the stop offset frequency for plot.
[SENSe]:PNOise:OPTimization	Selects or queries the method of optimization.
[SENSe]:PULSe subgroup (Option 20 only)	**Pulsed RF measurements**
[SENSe]:PULSe:ANALyze:LEVel	Selects or queries how to determine the 50% level.
[SENSe]:PULSe:ANALyze:LEVel:FIFTy	Selects or queries how to determine the 50% level.
[SENSe]:PULSe:ANALyze:LEVel:HUNDred	Selects or queries how to determine the 50% level.
[SENSe]:PULSe:ANALyze:MEASurement:TIME:AUTO	Selects or queries whether to set the measurement time automatically.
[SENSe]:PULSe:ANALyze:MEASurement:TIME:STARt	Sets or queries the measurement start time.
[SENSe]:PULSe:ANALyze:MEASurement:TIME:STOP	Sets or queries the measurement stop time.
[SENSe]:PULSe:ANALyze:PMLocation	Sets or queries the phase measurement location.
[SENSe]:PULSe:ANALyze:POINt:LOCation	Selects or queries the point location method.
[SENSe]:PULSe:ANALyze:RFALl	Selects or queries the threshold levels to measure the rise/fall time.
[SENSe]:PULSe:ANALyze:RIPPle	Sets or queries the ripple portion of the pulse top.
[SENSe]:PULSe:CARRier:OFFSet	Sets or queries the carrier frequency offset.
[SENSe]:PULSe:CARRier:SEARch	Selects or queries how to detect the carrier.
[SENSe]:PULSe:DETect:MEASurement	Selects or queries whether to set the maximum number of pulses.
[SENSe]:PULSe:DETect:NUMBer	Sets or queries the maximum number of pulses to detect.
[SENSe]:PULSe:DETect:POWer[:THReshold]	Sets or queries the power threshold to detect pulses.
[SENSe]:PULSe:DETect:TIMe[:THReshold]	Sets or queries the minimum off-time between pulses.
[SENSe]:PULSe:FILTer:{BANDwidth|BWIDth}	Sets or queries the the filter bandwidth for the Gaussian filter.
[SENSe]:PULSe:FILTer:MEASurement	Selects or queries the measurement filter.
[SENSe]:PULSe:FREFerence:AUTO	Determines whether to estimate the frequency reference automatically.
[SENSe]:PULSe:FREFerence:CHIRpbw	Sets or queries the chirp bandwidth.
[SENSe]:PULSe:FREFerence:OFFSet	Sets or queries the frequency reference offset.
[SENSe]:PULSe:MODulation:TYPE	Selects or queries the modulation type.
[SENSe]:PULSe:SIGNal:TYPE	Selects or queries the signal type.
[SENSe]:SGRam subgroup	**Spectrogram measurement**
[SENSe]:SGRam:{BANDwidth|BWIDth}:OPTimization	Selects or queries the method of optimizing gain and input bandwidth.
[SENSe]:SGRam:{BANDwidth|BWIDth}:RESolution	Sets or queries the resolution bandwidth (RBW).
[SENSe]:SGRam:{BANDwidth|BWIDth}[:RESolution]:ACTual?	Queries the actual resolution bandwidth (RBW).
[SENSe]:SGRam:{BANDwidth|BWIDth}[:RESolution]:AUTO	Selects or queries whether to set the RBW automatically.

Table 2-21: Sense commands (cont.)

Header	Description
[SENSe]:SGRam:{BANDwidth|BWIDth}[:RESolution]:MODE	Selects or queries whether to enable or disable the RBW processing.
[SENSe]:SGRam:{BANDwidth|BWIDth}:VIDeo	Sets or queries the video bandwidth (VBW).
[SENSe]:SGRam:{BANDwidth|BWIDth}:VIDeo:STATe	Selects or queries whether to enable the video bandwidth (VBW).
[SENSe]:SGRam:COLor	Selects or queries the color palette of three-dimensional graphs.
[SENSe]:SGRam:COLor:MAXimum	Sets or queries the maximum value of the color axis.
[SENSe]:SGRam:COLor:MINimum	Sets or queries the minimum value of the color axis.
[SENSe]:SGRam:FFT:WINDow	Selects or queries the FFT window.
[SENSe]:SGRam:FILTer[:SHAPe]	Selects or queries the filter shape.
[SENSe]:SGRam:FREQuency:CENTer	Sets or queries the center frequency.
[SENSe]:SGRam:FREQuency:SPAN	Sets or queries the frequency span.
[SENSe]:SGRam:FREQuency:SPAN:BANDwidth[:RESolution]:RATio	Sets or queries the ratio of span to RBW.
[SENSe]:SGRam:FREQuency:SPAN:MAXimum	Sets the frequency range to the maximum real-time span.
[SENSe]:SGRam:FREQuency:STARt	Sets or queries the measurement start frequency.
[SENSe]:SGRam:FREQuency:STEP	Sets or queries the frequency step size.
[SENSe]:SGRam:FREQuency:STEP:AUTO	Selects or queries whether to set the frequency step size automatically.
[SENSe]:SGRam:FREQuency:STOP	Sets or queries the measurement stop frequency.
[SENSe]:SPECtrum subgroup	**Spectrum measurement**
[SENSe]:SPECtrum:{BANDwidth|BWIDth}:OPTimization	Selects or queries the method of optimizing gain and input bandwidth.
[SENSe]:SPECtrum:{BANDwidth|BWIDth}[:RESolution]	Sets or queries the resolution bandwidth (RBW).
[SENSe]:SPECtrum:{BANDwidth|BWIDth}[:RESolution]:ACTual?	Queries the actual resolution bandwidth (RBW).
[SENSe]:SPECtrum:{BANDwidth|BWIDth}[:RESolution]:AUTO	Selects or queries whether to set the RBW automatically.
[SENSe]:SPECtrum:{BANDwidth|BWIDth}[:RESolution]:MODE	Selects or queries whether to enable or disable the RBW process.
[SENSe]:SPECtrum:{BANDwidth|BWIDth}:VIDeo	Sets or queries the video bandwidth (VBW).
[SENSe]:SPECtrum:{BANDwidth|BWIDth}:VIDeo:STATe	Selects or queries whether to enable the video bandwidth (VBW).
[SENSe]:SPECtrum:CLEar:RESults	Restarts multi-trace functions (Average and Max/Min Hold).
[SENSe]:SPECtrum:FFT:WINDow	Selects or queries the FFT window.
[SENSe]:SPECtrum:FILTer[:SHAPe]	Selects or queries the filter shape.
[SENSe]:SPECtrum:FREQuency:CENTer	Sets or queries the center frequency.
[SENSe]:SPECtrum:FREQuency:SPAN	Sets or queries the frequency span.
[SENSe]:SPECtrum:FREQuency:SPAN:BANDwidth[:RESolution]:RATio	Sets or queries the ratio of span to RBW.

Table 2-21: Sense commands (cont.)

Header	Description
[SENSe]:SPECtrum:FREQuency:STARt	Sets or queries the measurement start frequency.
[SENSe]:SPECtrum:FREQuency:STEP	Sets or queries the frequency step size.
[SENSe]:SPECtrum:FREQuency:STEP:AUTO	Selects or queries whether to set the frequency step size automatically.
[SENSe]:SPECtrum:FREQuency:STOP	Sets or queries the measurement stop frequency.
[SENSe]:SPECtrum:MAX:SPAN	Sets the frequency span to the maximum span.
[SENSe]:SPECtrum:POINts:COUNt	Sets or queries the number of sample points on the signal spectrum.
[SENSe]:SPURious subgroup	**Spurious measurement**
[SENSe]:SPURious:CARRier:{BANDwidth|BWIDth}	Sets or queries the channel width for the carrier as power reference.
[SENSe]:SPURious:CARRier:{BANDwidth|BWIDth}:INTegration	Sets or queries the integration bandwidth to calculate the carrier power.
[SENSe]:SPURious:CARRier:{BANDwidth|BWIDth}[:RESolution]	Sets or queries the resolution bandwidth to measure the carrier power.
[SENSe]:SPURious:CARRier:{BANDwidth|BWIDth}[:RESolution]:AUTO	Selects or queries whether to set the resolution bandwidth automatically.
[SENSe]:SPURious:CARRier:DETection	Selects or queries the carrier detection method.
[SENSe]:SPURious:CARRier:FREQuency	Sets or queries the carrier frequency.
[SENSe]:SPURious:CARRier:THReshold	Sets or queries the threshold level to detect the carrier.
[SENSe]:SPURious:CLEar:RESults	Restarts multi-trace functions (Average and Max Hold).
[SENSe]:SPURious[:FREQuency]:OVERlap?	Queries whether any of the frequency ranges (A to T) overlap.
[SENSe]:SPURious:LIST	Selects or queries how to list the spurious signals.
[SENSe]:SPURious:MODE	Selects or queries the frequency range mode (Multi or Single).
[SENSe]:SPURious:OPTimization	Selects or queries the method of optimization.
[SENSe]:SPURious:POINts:COUNt	Sets or queries the trace point count for the frequency range.
[SENSe]:SPURious:RANGe<x>:BANDwidth:VIDeo	Sets or queries the VBW for the specified frequency range.
[SENSe]:SPURious:RANGe<x>:BANDwidth:VIDeo:STATe	Selects or queries whether to enable the VBW for the frequency range.
[SENSe]:SPURious:RANGe<x>:DETection	Selects or queries the display detector for the frequency range.
[SENSe]:SPURious:RANGe<x>:EXCursion	Sets or queries the excursion level in the frequency range.
[SENSe]:SPURious:RANGe<x>:FILTer[:SHAPe]	Selects or queries the filter shape for the frequency range.
[SENSe]:SPURious:RANGe<x>:FILTer[:SHAPe]:BANDwidth	Sets or queries the filter bandwidth for the frequency range.
[SENSe]:SPURious:RANGe<x>:FILTer[:SHAPe]:BANDwidth:AUTO	Selects or queries whether to set the filter bandwidth automatically.
[SENSe]:SPURious:RANGe<x>:FREQuency:STARt	Sets or queries the start frequency of the range.
[SENSe]:SPURious:RANGe<x>:FREQuency:STOP	Sets or queries the stop frequency of the range.
[SENSe]:SPURious:RANGe<x>:LIMit:ABSolute:STARt	Sets or queries the absolute start amplitude of the limits for the range.
[SENSe]:SPURious:RANGe<x>:LIMit:ABSolute:STOP	Sets or queries the absolute stop amplitude of the limits for the range.
[SENSe]:SPURious:RANGe<x>:LIMit:MASK	Selects or queries the limit mask function mode for the frequency range.

Table 2-21: Sense commands (cont.)

Header	Description
[SENSe]:SPURious:RANGe<x>:LIMit:RELative:STARt	Sets or queries the relative start amplitude of the limits for the range.
[SENSe]:SPURious:RANGe<x>:LIMit:RELative:STOP	Sets or queries the relative stop amplitude of the limits for the range.
[SENSe]:SPURious:RANGe<x>:STATe	Selects or queries whether to enable or disable the frequency range.
[SENSe]:SPURious:RANGe<x>:THReshold	Sets or queries the threshold level to detect spurious signals in a range.
[SENSe]:SPURious:REFerence	Selects or queries the power reference.
[SENSe]:SPURious:REFerence:MANual:POWer	Sets or queries the reference power level.
[SENSe]:TOVerview subgroup	**Time overview**
[SENSe]:TOVerview:FREQuency:CENTer	Sets or queries the center frequency.
[SENSe]:TOVerview:MAXTracepoints	Selects or queries the maximum trace points.

Status Commands

Use the STATus commands to control registers defined in the SCPI status reporting structure.

Table 2-22: Status commands

Header	Description
STATus:ACPower:EVENts?	Returns the current events for the ACPR measurement.
STATus:AVTime:EVENts?	Returns the current events for the Amplitude versus Time measurement.
STATus:CCDF:EVENts?	Returns the current events for the CCDF measurement.
STATus:CONSte:EVENts?	Returns the current events for the Constellation measurement.
STATus:DPSA:EVENts?	Returns the current events for the DPX spectrum measurement.
STATus:EVM:EVENts?	Returns the current events for the EVM versus Time measurement.
STATus:FVTime:EVENts?	Returns the current events for the Frequency versus Time measurement.
STATus:IQVTime:EVENts?	Returns the current events for the RF I&Q versus Time measurement.
STATus:MCPower:EVENts?	Returns the current events for the MCPR measurement.
STATus:MERRor:EVENts?	Returns the current events for the Mag error versus Time measurement.
STATus:OBWidth:EVENts?	Returns the current events for the Occupied Bandwidth measurement.
STATus:OPERation:CONDition?	Queries the contents of the OCR.
STATus:OPERation:ENABle	Sets or queries the mask for the OENR.
STATus:OPERation[:EVENt]?	Queries the contents of the OEVR.
STATus:OPERation:NTRansition	Sets or queries the value of the negative transition filter.
STATus:OPERation:PTRansition	Sets or queries the value of the positive transition filter.
STATus:PERRor:EVENts?	Returns the current events for the Phase error measurement.
STATus:PHVTime:EVENts?	Returns the current events for the Phase versus Time measurement.
STATus:PNOise:EVENts?	Returns the current events for the phase noise measurement.
STATus:PRESet	Presets a status byte.
STATus:PULSe:RESult:EVENts?	Returns the current events for the pulse table measurement.
STATus:PULSe:STATistics:EVENts?	Returns the current events for the pulse statistics measurement.
STATus:PULSe:TRACe:EVENts?	Returns the current events for the pulse trace measurement.
STATus:QUEStionable:CONDition?	Queries the contents of the QCR.
STATus:QUEStionable:ENABle	Sets or queries the mask for the OENR.
STATus:QUEStionable[:EVENt]?	Queries the contents of the QER.
STATus:QUEStionable:NTRansition	Sets or queries the value of the negative transition filter.
STATus:QUEStionable:PTRansition	Sets or queries the value of the positive transition filter.
STATus:QUEStionable:CALibration:CONDition?	Queries the contents of the questionable calibration condition register.
STATus:QUEStionable:CALibration:ENABle	Sets or queries the mask for the questionable calibration enable register.
STATus:QUEStionable:CALibration[:EVENt]?	Queries the contents of the questionable calibration event register.
STATus:QUEStionable:CALibration:NTRansition	Sets or queries the value of the negative transition filter.

Table 2-22: Status commands (cont.)

Header	Description
STATus:QUEStionable:CALibration:PTRansition	Sets or queries the value of the positive transition filter.
STATus:QUEStionable:FREQuency:CONDition?	Queries the contents of the questionable frequency condition register.
STATus:QUEStionable:FREQuency:ENABle	Sets or queries the mask for the questionable frequency enable register.
STATus:QUEStionable:FREQuency[:EVENt]?	Queries the contents of the questionable frequency event register.
STATus:QUEStionable:FREQuency:NTRansition	Sets or queries the value of the negative transition filter.
STATus:QUEStionable:FREQuency:PTRansition	Sets or queries the value of the positive transition filter.
STATus:SGRAM:EVENts?	Returns the current events for the spectrogram measurement.
STATus:SPECtrum:EVENts?	Returns the current events for the spectrum measurement.
STATus:SPURious:EVENts?	Returns the current events for the spurious measurement.
STATus:SQUality:EVENts?	Returns the current events for the signal quality measurement.

System Commands

Use the SYSTem commands to set or query system parameters for operation.

Table 2-23: System commands

Header	Description
SYSTem:COMMunicate:GPIB[:SELF]:ADDRess	Sets or queries the GPIB address of the instrument.
SYSTem:DATE	Sets or queries the current date.
SYSTem:ERRor:ALL?	Queries all the error or event information.
SYSTem:ERRor:CODE:ALL?	Queries all the error or event codes.
SYSTem:ERRor:CODE[:NEXT]?	Queries the latest error or event information.
SYSTem:ERRor:COUNt?	Queries the number of errors or events.
SYSTem:ERRor[:NEXT]?	Queries the latest error or event information.
SYSTem:OPTions?	Queries optional information.
SYSTem:PRESet	Presets the analyzer.
SYSTem:TIME	Sets or queries the current time.
SYSTem:VERSion?	Queries the version of the SCPI.

Trace Commands

Use the TRACe commands to select trace type and to control trace arithmetic.

Table 2-24: Trace Commands

Header	Description
TRACe<x>:AVTime subgroup	**Amplitude versus Time measurement**
TRACe<x>:AVTime	Selects or queries whether or not to show the specified trace.
TRACe<x>:AVTime:AVERage:COUNt	Sets or queries the number of traces to combine for averaging.
TRACe<x>:AVTime:AVERage:RESet	Clears the average data and resets the average counter.
TRACe<x>:AVTime:COUNt	Sets or queries the count for the Max or Min Hold trace.
TRACe<x>:AVTime:COUNt:ENABle	Selects or queries whether or not to enable the count for Max/Min Hold.
TRACe<x>:AVTime:COUNt:RESet	Clears the Max or Min Hold data and counter, and restarts the process.
TRACe<x>:AVTime:FREeze	Selects or queries whether or not to freeze the display of the trace.
TRACe<x>:AVTime:FUNCtion	Selects or queries the trace function.
TRACe<x>:AVTime:LEFToperand	Selects or queries the left operand for the math trace.
TRACe<x>:AVTime:RIGHtoperand	Selects or queries the right operand for the math trace.
TRACe<x>:AVTime:SELect	Selects or queries the trace number to display the readout.
TRACe<x>:CCDF subgroup	**CCDF measurement**
TRACe<x>:CCDF:FREeze	Selects or queries whether or not to freeze the display of the trace.
TRACe<x>:CCDF:SELect	Selects or queries the trace number to display the readout.
TRACe<x>:CCDF:SHOW	Selects or queries whether to show or hide the trace.
TRACe<x>:CCDF:X	Sets or queries the horizontal position of the measurement pointer.
TRACe<x>:CCDF:Y?	Queries the vertical position (CCDF value) of the measurement pointer.
TRACe:CONSte subgroup (Option 21 only)	**Constellation measurement**
TRACe:CONSte:MODE	Selects or queries how to display the constellation trace.
TRACe<x>:DPSA subgroup	**DPX spectrum measurement**
TRACe<x>:DPSA	Selects or queries whether or not to show the waveform.
TRACe<x>:DPSA:AVERage:COUNt	Sets or queries the number of traces to combine for averaging.
TRACe<x>:DPSA:COLor:INTensity	Sets or queries the color intensity.
TRACe<x>:DPSA:DOT:PERSistent	Selects or queries whether to enable or disable the dot persistence.
TRACe<x>:DPSA:DOT:PERSistent:TYPE	Selects or queries the persistence type.
TRACe<x>:DPSA:DOT:PERSistent:VARiable	Sets or queries the length of time that data points are displayed.
TRACe<x>:DPSA:FREeze	Selects or queries whether or not to freeze the display of the trace.
TRACe<x>:DPSA:FUNCtion	Selects or queries the function.
TRACe<x>:DPSA:LEFToperand	Selects or queries the left operand for the math trace.
TRACe<x>:DPSA:RIGHtoperand	Selects or queries the right operand for the math trace.
TRACe<x>:DPSA:SELect	Selects or queries the trace number to display the readout.

Table 2-24: Trace Commands (cont.)

Header	Description
TRACe:FVTime subgroup	**Frequency versus Time measurement**
TRACe:FVTime	Selects or queries whether or not to show the trace.
TRACe:FVTime:AVERage:COUNt	Sets or queries the number of traces to combine for averaging.
TRACe:FVTime:COUNt	Sets or queries the count for the Max or Min Hold trace.
TRACe:FVTime:COUNt:ENABle	Selects or queries whether or not to enable the count for Max/Min Hold.
TRACe:FVTime:COUNt:RESet	Clears the Max or Min Hold data and counter, and restarts the process.
TRACe:FVTime:FREeze	Selects or queries whether or not to freeze the display of the trace.
TRACe:FVTime:FUNCtion	Selects or queries the trace function.
TRACe:IQVTime subgroup	**RF I&Q versus Time measurement**
TRACe:IQVTime:AVERage:COUNt	Sets or queries the number of traces to combine for averaging.
TRACe:IQVTime:COUNt	Sets or queries the count for the Max or Min Hold trace.
TRACe:IQVTime:COUNt:ENABle	Selects or queries whether or not to enable the count for Max/Min Hold.
TRACe:IQVTime:COUNt:RESet	Clears the Max or Min Hold data and counter, and restarts the process.
TRACe:IQVTime:ENABle:I	Selects or queries whether to show or hide the trace I.
TRACe:IQVTime:ENABle:Q	Selects or queries whether to show or hide the trace Q.
TRACe:IQVTime:FREeze	Selects or queries whether to freeze the IQ traces.
TRACe:IQVTime:FUNCtion	Selects or queries the trace function.
TRACe:IQVTime:SELect:I	Selects or queries whether to choose the I trace.
TRACe:IQVTime:SELect:Q	Selects or queries whether to choose the Q trace.
TRACe:OBWidth subgroup	**Occupied Bandwidth measurement**
TRACe:OBW:MAXHold	Determines whether to enable or disable the Max Hold trace.
TRACe:PHVTime subgroup	**Phase versus Time measurement**
TRACe:PHVTime	Selects or queries whether or not to show the trace.
TRACe:PHVTime:AVERage:COUNt	Sets or queries the number of traces to combine for averaging.
TRACe:PHVTime:COUNt	Sets or queries the count for the Max or Min Hold trace.
TRACe:PHVTime:COUNt:ENABle	Selects or queries whether or not to enable the count for Max/Min Hold.
TRACe:PHVTime:COUNt:RESet	Clears the Max or Min Hold data and counter, and restarts the process.
TRACe:PHVTime:FREeze	Selects or queries whether to freeze the trace.
TRACe:PHVTime:FUNCtion	Selects or queries the trace function.
TRACe:PNOise subgroup (Option 11 only)	**Phase noise measurement**
TRACe<x>:PNOise:DETection	Selects or queries the display detector.
TRACe<x>:PNOise:FREeze	Selects or queries whether or not to freeze the trace display.
TRACe<x>:PNOise:SELect	Selects a trace or queries the currently selected trace.
TRACe<x>:PNOise:SHOW	Selects or queries whether to show or hide the trace.
TRACe<x>:PNOise:SMOothing:COUNt	Sets or queries the number of data points for smoothing the trace.
TRACe<x>:PNOise:SMOothing:ENABle	Selects or queries whether to enable smoothing the specified trace.
TRACe<x>:PNOise:SMOothing:RESet	Restarts the smoothing process.

Table 2-24: Trace Commands (cont.)

Header	Description
TRACe:SGRam subgroup	**Spectrogram measurement**
TRACe:SGRam:DETection	Selects or queries the display detector.
TRACe:SGRam:FREeze	Selects or queries whether or not to freeze the spectrogram display.
TRACe:SGRam:FUNCtion	Selects or queries the trace function for the spectrogram.
TRACe:SGRam:FUNCtion:TIME	Sets or queries the number of traces to combine for the trace function.
TRACe:SGRam:SELect:LINE	Selects or queries the number of line to send to the spectrum display.
TRACe<x>:SPECtrum subgroup	**Spectrum measurement**
TRACe<x>:SPECtrum	Selects or queries whether to show or hide the specified trace.
TRACe<x>:SPECtrum:AVERage:COUNt	Sets or queries the number of traces to combine for averaging.
TRACe<x>:SPECtrum:AVERage:RESet	Clears the average data and resets the average counter.
TRACe<x>:SPECtrum:COUNt	Sets or queries the count for the Max or Min Hold trace.
TRACe<x>:SPECtrum:COUNt:ENABle	Selects or queries whether or not to enable the count for Max/Min Hold.
TRACe<x>:SPECtrum:COUNt:RESet	Clears the Max or Min Hold data and counter, and restarts the process.
TRACe<x>:SPECtrum:DETection	Selects or queries the display detector.
TRACe<x>:SPECtrum:FREeze	Selects or queries whether or not to freeze the display of the trace.
TRACe<x>:SPECtrum:FUNCtion	Selects or queries the trace function.
TRACe<x>:SPECtrum:LEFToperand	Selects or queries the left operand for the math trace.
TRACe<x>:SPECtrum:RIGHtoperand	Selects or queries the right operand for the math trace.
TRACe<x>:SPECtrum:SELect	Selects or queries the trace number to display the readout.
TRACe:SPURious subgroup	**Spurious measurement**
TRACe:SPURious:COUNt	Sets or queries the count for the Max Hold or Average trace.
TRACe:SPURious:COUNt:ENABle	Selects or queries whether to enable the count for Max Hold or Average.
TRACe:SPURious:COUNt:RESet	Clears Max Hold or Average data and counter, and restarts the process.
TRACe:SPURious:FREeze	Selects or queries whether or not to freeze the display of the trace.
TRACe:SPURious:FUNCtion	Selects or queries the trace function.

Trace Mnemonics

Multiple traces can be used in some measurement displays. The traces are specified by the trace specifier TRACe<x> (<x>=1 to 5) which is defined for each measurement display as follows.

Table 2-25: Trace mnemonics

Measurement display	TRACe1	TRACe2	TRACe3	TRACe4	TRACe5
Amplitude versus Time	Trace 1	Trace 2	Math trace	NA	NA
CCDF	Trace 1	Trace 2	Gaussian curve	NA	NA
DPX spectrum	+Peak trace	-Peak trace	Average trace	Math trace	Bitmap trace
Phase noise	Trace 1	Trace 2	NA	NA	NA
Spectrum	Trace 1	Trace 2	Trace 3	Math trace	Spectrogram

NOTE. *Valid traces depend on commands. Refer to each command description.*

Trigger commands

Use the TRIGger commands to set up the trigger system.

Table 2-26: Trigger commands

Header	Description
TRIGger:MASK:NEW	Loads a new frequency mask.
TRIGger:MASK:NEW:AUTO	Draws a new frequency mask automatically.
TRIGger:MASK:OPEN	Opens a trigger mask with a specified file.
TRIGger:MASK:SAVE	Saves the current trigger mask to a specified file.
TRIGger[:SEQuence]:ADVanced:SWEep:MODE	Selects or queries whether to trigger each segment in the swept mode.
TRIGger[:SEQuence]:EVENt:EXTFront:IMPedance	Selects or queries the impedance of the external trigger input (front).
TRIGger[:SEQuence]:EVENt:EXTFront:LEVel	Sets or queries the trigger level at the external trigger input (front).
TRIGger[:SEQuence]:EVENt:EXTFront:SLOPe	Selects or queries the trigger slope of the external trigger input (front).
TRIGger[:SEQuence]:EVENt:EXTRear:SLOPe	Selects or queries the trigger slope of the external trigger input (rear).
TRIGger[:SEQuence]:EVENt:GATed	Selects or queries the logic for the gated trigger input.
TRIGger[:SEQuence]:EVENt:INPut:FMASk:VIOLation	Selects or queries when the trigger occurs in the frequency mask trigger.
TRIGger[:SEQuence]:EVENt:INPut:LEVel	Sets or queries the trigger level for the RF input level trigger.
TRIGger[:SEQuence]:EVENt:INPut:SLOPe	Selects or queries the trigger slope for the RF input level trigger.
TRIGger[:SEQuence]:EVENt:INPut:TDBWidth	Sets or queries the time-domain bandwidth.
TRIGger[:SEQuence]:EVENt:INPut:TDBWidth:ACTual?	Queries the actual time-domain bandwidth.
TRIGger[:SEQuence]:EVENt:INPut:TDBWidth:STATe	Determines whether to set the time-domain bandwidth automatically.
TRIGger[:SEQuence]:EVENt:INPut:TYPE	Selects or queries the trigger type for the source of RF input.
TRIGger[:SEQuence]:EVENt:SOURce	Selects or queries the trigger event source.
TRIGger[:SEQuence]:FORCed	Selects or queries whether to causes a manual trigger.
TRIGger[:SEQuence]:IMMediate	Causes a trigger immediately.
TRIGger[:SEQuence]:STATus	Selects or queries the trigger mode (Free Run or Triggered).
TRIGger[:SEQuence]:TIME:DELay	Sets or queries the trigger delay time.
TRIGger[:SEQuence]:TIME:POSition	Sets or queries the trigger position.

Unit Commands

Specify fundamental units for measurement.

Table 2-27: Unit commands

Header	Description
UNIT:POWer	Selects or queries the unit of power.

Retrieving Response Message

When receiving a query command from the external controller, the analyzer puts the response message on the Output Queue. This message cannot be retrieved unless you perform retrieval operations through the external controller. (For example, call the IBRD subroutine included in the GPIB software of National Instruments.)

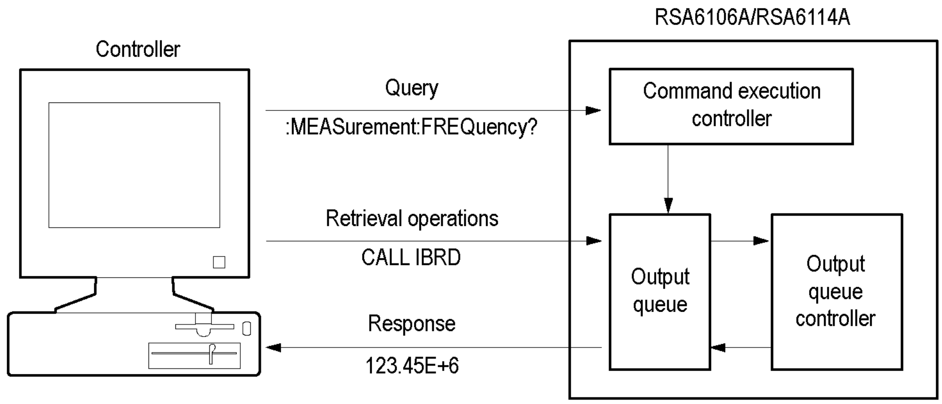


Figure 2-5: Retrieving response message

When the Output Queue contains a response message, sending another command from the external controller before retrieving this message deletes it from the queue. The Output Queue always contains the response message to the most recent query command.

You can use the MAV bit of the Status Byte Register (SBR) to check whether the Output Queue contains a response message. For details, refer to *Status Byte Register (SBR)*.

Command Descriptions

ABORt (No Query Form)

Resets the trigger system and places all trigger sequences in the idle state. Any actions related to the trigger system that are in progress, such as a sweep or acquiring a measurement is also aborted.

To start data acquisition, use the INITiate commands.

Conditions Measurement views: All

Group Abort commands

Syntax `ABORt`

Related Commands INITiate:CONTinuous, INITiate[:IMMediate]

Arguments None

Examples `ABORT` resets the trigger system and stops data acquisition.

*CAL (Query Only)

Instructs the analyzer to perform an internal self-alignment and return its status.

NOTE. *The self-alignment can take several minutes to respond. No other commands will be executed until alignment is complete.*

Conditions Measurement views: All

Group IEEE common commands

Syntax `*CAL`

Returns `<NR1>=1` indicates that the alignment was successful.

`<NR1>=0` indicates that the alignment was unsuccessful.

Examples `*CAL` performs an internal self-alignment and will return `1` if the alignment is successful.

CALCulate:ACPower:MARKer<x>:DELTa:X? (Query Only)

Returns the delta marker frequency for the selected marker in the Channel power and ACPR measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

Conditions Measurement views: Channel power and ACPR

Group Calculate commands

Syntax `CALCulate:ACPower:MARKer<x>:DELTa:X?`

Related Commands CALCulate:MARKer:ADD, CALCulate:ACPower:MARKer<x>:DELTa:Y?

Returns `<NRf>` Delta marker frequency for the selected marker.

Examples `CALCULATE:ACPOWER:MARKER1:DELTA:X?` might return `1.28E+6`, indicating that the delta marker frequency is 1.28 MHz.

CALCulate:ACPower:MARKer<x>:DELTa:Y? (Query Only)

Returns the delta marker amplitude for the selected marker in the Channel power and ACPR measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

Conditions Measurement views: Channel power and ACPR

Group Calculate commands

Syntax `CALCulate:ACPower:MARKer<x>:DELTa:Y?`

Related Commands CALCulate:MARKer:ADD, CALCulate:ACPower:MARKer<x>:DELTa:X?

Returns `<NRf>` Delta marker amplitude for the selected marker.

Examples `CALCULATE:ACPOWER:MARKER1:DELTA:Y?` might return `23.45`, indicating that the delta marker amplitude is 23.45 dB.

CALCulate:ACPower:MARKer<x>:MAXimum (No Query Form)

Moves the selected marker to the highest peak on the trace in the Channel power and ACPR measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

Conditions Measurement views: Channel power and ACPR

Group Calculate commands

Syntax `CALCulate:ACPower:MARKer<x>:MAXimum`

Arguments None

Examples `CALCULATE:ACPOWER:MARKER1:MAXIMUM` moves Marker 1 (M1) to the highest peak on the trace.

CALCulate:ACPower:MARKer<x>:PEAK:LEFT (No Query Form)

Moves the selected marker to the next peak to the left on the trace in the Channel power and ACPR measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

Conditions Measurement views: Channel power and ACPR

Group Calculate commands

Syntax `CALCulate:ACPower:MARKer<x>:PEAK:LEFT`

Related Commands CALCulate:ACPower:MARKer<x>:PEAK:RIGHt

Arguments None

Examples `CALCULATE:ACPOWER:MARKER1:PEAK:LEFT` moves Marker 1 (M1) to the next peak to the left on the trace.

CALCulate:ACPower:MARKer<x>:PEAK:RIGHt (No Query Form)

Moves the selected marker to the next peak to the right on the trace in the Channel power and ACPR measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

Conditions Measurement views: Channel power and ACPR

Group Calculate commands

Syntax `CALCulate:ACPower:MARKer<x>:PEAK:RIGHt`

Related Commands CALCulate:ACPower:MARKer<x>:PEAK:LEFT

Arguments None

Examples `CALCULATE:ACPOWER:MARKER1:PEAK:RIGHT` moves Marker 1 (M1) to the next peak to the right on the trace.

CALCulate:ACPower:MARKer<x>:X

Sets or queries the horizontal position of the selected marker in the Channel power and ACPR measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

Conditions Measurement views: Channel power and ACPR

Group Calculate commands

Syntax

```
CALCulate:ACPower:MARKer<x>:X <value>
CALCulate:ACPower:MARKer<x>:X?
```

Related Commands CALCulate:ACPower:MARKer<x>:Y?

Arguments `<value>::=<NRf>` specifies the horizontal position of the marker.
Range: (center frequency) ± (span)/2.

Examples `CALCULATE:ACPOWER:MARKER1:X 800MHz` places Marker 1 (M1) at 800 MHz on the trace.

CALCulate:ACPower:MARKer<x>:Y? (Query Only)

Queries the vertical position of the selected marker in the Channel power and ACPR measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

Conditions Measurement views: Channel power and ACPR

Group Calculate commands

Syntax

```
CALCulate:ACPower:MARKer<x>:Y?
```

Related Commands CALCulate:ACPower:MARKer<x>:X

Arguments None

Examples `CALCULATE:ACPOWER:MARKER1:Y` might return `-34.28`, indicating Marker 1 (M1) is at -34.28 dBm.

CALCulate:AVTime:MARKer<x>:DELTa:X? (Query Only)

Returns the delta marker time for the selected marker in the Amplitude versus Time measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

Conditions Measurement views: Amplitude versus Time

Group Calculate commands

Syntax `CALCulate:AVTime:MARKer<x>:DELTa:X?`

Related Commands CALCulate:MARKer:ADD, CALCulate:AVTime:MARKer<x>:DELTa:Y?

Returns `<NRf>` Delta marker time for the selected marker.

Examples `CALCULATE:AVTIME:MARKER1:DELTA:X?` might return `38.0E-9`, indicating that the delta marker time is 38.0 ns.

CALCulate:AVTime:MARKer<x>:DELTa:Y? (Query Only)

Returns the delta marker amplitude for the selected marker in the Amplitude versus Time measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

Conditions Measurement views: Amplitude versus Time

Group Calculate commands

Syntax `CALCulate:AVTime:MARKer<x>:DELTa:Y?`

Related Commands CALCulate:MARKer:ADD, CALCulate:AVTime:MARKer<x>:DELTa:X?

Returns <NRf> Delta marker amplitude for the selected marker.

Examples `CALCULATE:AVTIME:MARKER1:DELTA:Y?` might return `23.45`, indicating that the delta marker amplitude is 23.45 dB.

CALCulate:AVTime:MARKer<x>:MAXimum (No Query Form)

Moves the selected marker to the highest peak on the trace in the Amplitude versus Time measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

Conditions Measurement views: Amplitude versus Time

Group Calculate commands

Syntax `CALCulate:AVTime:MARKer<x>:MAXimum`

Arguments None

Examples `CALCULATE:AVTIME:MARKER1:MAXIMUM` moves Marker 1 (M1) to the highest peak on the trace.

CALCulate:AVTime:MARKer<x>:PEAK:HIGHer (No Query Form)

Moves the selected marker to the next peak higher in amplitude on the Amplitude versus Time trace.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

Conditions Measurement views: Amplitude versus Time

Group Calculate commands

Syntax `CALCulate:AVTime:MARKer<x>:PEAK:HIGHer`

Related Commands CALCulate:AVTime:MARKer<x>:PEAK:LOWer

Arguments None

Examples `CALCULATE:AVTIME:MARKER1:PEAK:HIGHER` moves Marker 1 (M1) to the next peak higher in amplitude on the trace.

CALCulate:AVTime:MARKer<x>:PEAK:LEFT (No Query Form)

Moves the selected marker to the next peak to the left on the trace in the Amplitude versus Time measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

Conditions Measurement views: Amplitude versus Time

Group Calculate commands

Syntax `CALCulate:AVTime:MARKer<x>:PEAK:LEFT`

Related Commands CALCulate:AVTime:MARKer<x>:PEAK:RIGHt

Arguments None

Examples `CALCulate:AVTime:MARKer1:PEAK:LEFT` moves Marker 1 (M1) to the next peak to the left on the trace.

CALCulate:AVTime:MARKer<x>:PEAK:LOWer (No Query Form)

Moves the selected marker to the next peak lower in amplitude on the Amplitude versus Time trace.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

Conditions Measurement views: Amplitude versus Time

Group Calculate commands

Syntax `CALCulate:AVTime:MARKer<x>:PEAK:LOWer`

Related Commands CALCulate:AVTime:MARKer<x>:PEAK:HIGHer

Arguments None

Examples `CALCULATE:AVTIME:MARKER1:PEAK:LOWER` moves Marker 1 (M1) to the next peak lower in amplitude on the trace.

CALCulate:AVTime:MARKer<x>:PEAK:RIGHt (No Query Form)

Moves the selected marker to the next peak to the right on the trace in the Amplitude versus Time measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

Conditions Measurement views: Amplitude versus Time

Group Calculate commands

Syntax `CALCulate:AVTime:MARKer<x>:PEAK:RIGHt`

Related Commands CALCulate:AVTime:MARKer<x>:PEAK:LEFT

Arguments None

Examples `CALCULATE:AVTIME:MARKER1:PEAK:RIGHT` moves Marker 1 (M1) to the next peak to the right on the trace.

CALCulate:AVTime:MARKer<x>:TRACe

Selects or queries the trace on which the specified marker is placed in the Amplitude versus Time measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

Conditions Measurement views: Amplitude versus Time

Group Calculate commands

Syntax
```
CALCulate:AVTime:MARKer<x>:TRACe { TRACE1 | TRACE2 | TRACE3
| TRACE4 }
CALCulate:AVTime:MARKer<x>:TRACe?
```

Arguments `TRACE1` places the selected marker on Trace 1.

`TRACE2` places the selected marker on Trace 2.

`TRACE3` places the selected marker on Trace 3.

`TRACE4` places the selected marker on Trace 4 (math trace).

Examples `CALCULATE:AVTIME:MARKER1:TRACE TRACE1` places Marker 1 (M1) on Trace 1.

CALCulate:AVTime:MARKer<x>:X

Sets or queries the horizontal position of the selected marker in the Amplitude versus Time measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

Conditions Measurement views: Amplitude versus Time

Group Calculate commands

Syntax
```
CALCulate:AVTime:MARKer<x>:X <value>
CALCulate:AVTime:MARKer<x>:X?
```

Related Commands CALCulate:AVTime:MARKer<x>:Y?

Arguments `<value>::=<NRf>` specifies the horizontal position of the marker.
Range: (analysis offset) to [(analysis offset) + (analysis length)].

Examples `CALCULATE:AVTIME:MARKER1:X 1.5U` places Marker 1 (M1) at 1.5 µs on the trace.

CALCulate:AVTime:MARKer<x>:Y? (Query Only)

Queries the vertical position of the selected marker in the Amplitude versus Time measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

Conditions Measurement views: Amplitude versus Time

Group Calculate commands

Syntax `CALCulate:AVTime:MARKer<x>:Y?`

Related Commands CALCulate:AVTime:MARKer<x>:X

Arguments None

Examples `CALCULATE:AVTIME:MARKER1:Y?` might return `-2.73`, indicating Marker 1 (M1) is at -2.73 dBm.

CALCulate:CONSte:MARKer<x>:DELTa:X[:TIME]? (Query Only)

Returns the delta marker time for the selected marker on the constellation trace.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

Conditions Measurement views: Constellation

Group Calculate commands

Syntax `CALCulate:CONSte:MARKer<x>:DELTa:X[:TIME]?`

Returns `<NRf>` Delta marker time for the selected marker.
The unit can be changed by the [SENSe]:DDEMod:TIMe:UNITs command.

Examples `CALCULATE:CONSTE:MARKER1:DELTA:X:TIME?` might return `-62.75`, indicating that the delta marker time is -62.75 symbols.

CALCulate:CONSte:MARKer<x>:MAGNitude? (Query Only)

Queries the magnitude readout of the selected marker in the constellation measurement.

Conditions Measurement views: Constellation

Group Calculate commands

Syntax `CALCulate:CONSte:MARKer<x>:MAGNitude?`

Related Commands CALCulate:CONSte:MARKer<x>:X

Arguments None

Returns `<NRf>` The magnitude readout for the selected marker.

Examples `CALCulate:CONSte:MARKer1:MAGNitude?` might return `0.713927`, indicating the magnitude readout of Marker 1 (M1) is 0.713927.

CALCulate:CONSte:MARKer<x>:MAXimum (No Query Form)

Positions the selected marker at the symbol in the center of the time record.

Conditions Measurement views: Constellation

Group Calculate commands

Syntax `CALCulate:CONSte:MARKer<x>:MAXimum`

Arguments None

Examples `CALCulate:CONSte:MARKer1:MAXimum` positions Marker 1 (M1) at the symbol in the center of the time record.

CALCulate:CONSte:MARKer<x>:PEAK:LEFT (No Query Form)

Moves the selected marker in the time domain to the next lower symbol number, relative to the previous marker position.

Conditions Measurement views: Constellation

Group Calculate commands

Syntax `CALCulate:CONSte:MARKer<x>:PEAK:LEFT`

Related Commands CALCulate:CONSte:MARKer<x>:PEAK:RIGHt

Arguments None

Examples `CALCulate:CONSte:MARKer1:PEAK:LEFT` moves Marker 1 (M1) in the time domain to the next lower symbol number.

CALCulate:CONSte:MARKer<x>:PEAK:RIGHt (No Query Form)

Moves the selected marker in the time domain to the next higher symbol number, relative to the previous marker position.

Conditions Measurement views: Constellation

Group Calculate commands

Syntax `CALCulate:CONSte:MARKer<x>:PEAK:RIGHt`

Related Commands CALCulate:CONSte:MARKer<x>:PEAK:LEFT

Arguments None

Examples `CALCulate:CONSte:MARKer1:PEAK:RIGHT` moves the Marker 1 (M1) in the time domain to the next higher symbol number.

CALCulate:CONSte:MARKer<x>:PHASe? (Query Only)

Queries the phase readout of the selected marker in the constellation measurement.

Conditions Measurement views: Constellation

Group Calculate commands

Syntax `CALCulate:CONSte:MARKer<x>:PHASe?`

Related Commands CALCulate:CONSte:MARKer<x>:X

Arguments None

Returns `<NRf>` The phase readout for the selected marker.

Examples `CALCulate:CONSte:MARKer1:PHASe` might return `35.74`, indicating the phase readout of Marker 1 (M1) is 35.74°.

CALCulate:CONSte:MARKer<x>:SYMBol? (Query Only)

Queries the symbol readout of the selected marker in the constellation measurement.

Conditions Measurement views: Constellation

Group Calculate commands

Syntax `CALCulate:CONSte:MARKer<x>:SYMBol?`

Related Commands CALCulate:CONSte:MARKer<x>:X

Arguments None

Returns `<NRf>` The symbol readout for the selected marker.

Examples `CALCULATE:CONSTE:MARKER1:SYMBOL?` might return `62.00`, indicating the symbol readout of Marker 1 (M1) is 62.

CALCulate:CONSte:MARKer<x>:VALue? (Query Only)

Queries the value readout of the selected marker in the constellation measurement.

Conditions Measurement views: Constellation

Group Calculate commands

Syntax `CALCulate:CONSte:MARKer<x>:VALue?`

Related Commands CALCulate:CONSte:MARKer<x>:X

Arguments None

Returns `<NRf>` The value readout for the selected marker.

Examples `CALCULATE:CONSTE:MARKER1:VALUE?` might return `2.00`, indicating the value readout of Marker 1 (M1) is 2.

CALCulate:CONSte:MARKer<x>:X

Sets or queries the time position of the selected marker on the constellation trace.

Conditions Measurement views: Constellation

Group Calculate commands

Syntax

```
CALCulate:CONSte:MARKer<x>:X <value>
CALCulate:CONSte:MARKer<x>:X?
```

Related Commands

CALCulate:CONSte:MARKer<x>:MAGNitude?, CALCulate:CONSte:MARKer<x>:PHASe?, CALCulate:CONSte:MARKer<x>:SYMBol?

Arguments

`<value>::=<NRf>` specifies the time position of the marker.
Range: (analysis offset) to [(analysis offset) + (analysis length)].

Examples

`CALCULATE:CONSTE:MARKER1:X -1.63875m` places the Marker 1 (M1) at -1.63875 ms on the constellation trace.

CALCulate:DPSA:MARKer<x>:DELTa:X? (Query Only)

Returns the delta marker frequency for the selected marker on the DPX spectrum trace.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

Conditions

Measurement views: DPX spectrum

Group

Calculate commands

Syntax

```
CALCulate:DPSA:MARKer<x>:DELTa:X?
```

Related Commands

CALCulate:DPSA:MARKer<x>:DELTa:Y?

Arguments

None

Returns

`<NRf>` Delta marker frequency for the selected marker.

Examples

`CALCULATE:DPSA:MARKER1:DELTA:X?` might return `1.28E+6`, indicating that the delta marker frequency is 1.28 MHz.

CALCulate:DPSA:MARKer<x>:DELTa:Y? (Query Only)

Returns the delta marker amplitude for the selected marker on the DPX spectrum trace.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

Conditions Measurement views: DPX spectrum

Group Calculate commands

Syntax `CALCulate:DPSA:MARKer<x>:DELTa:Y?`

Related Commands CALCulate:DPSA:MARKer<x>:DELTa:X?

Arguments None

Returns `<NRf>` Delta marker amplitude for the selected marker.

Examples `CALCULATE:DPSA:MARKER1:DELTA:Y?` might return `23.45`, indicating that the delta marker amplitude is 23.45 dB.

CALCulate:DPSA:MARKer<x>:MAXimum (No Query Form)

Moves the selected marker to the highest peak on the DPX spectrum trace.

Conditions Measurement views: DPX spectrum

Group Calculate commands

Syntax `CALCulate:DPSA:MARKer<x>:MAXimum`

Arguments None

Examples `CALCULATE:DPSA:MARKER1:MAXIMUM` moves Marker 1 (M1) to the highest peak on the trace.

CALCulate:DPSA:MARKer<x>:PEAK:HIGHer (No Query Form)

Moves the selected marker to the next peak higher in amplitude on the DPX spectrum trace.

Conditions Measurement views: DPX spectrum

Group Calculate commands

Syntax `CALCulate:DPSA:MARKer<x>:PEAK:HIGHer`

Related Commands CALCulate:DPSA:MARKer<x>:PEAK:LOWer

Arguments None

Examples `CALCULATE:DPSA:MARKER1:PEAK:HIGHER` moves Marker 1 (M1) to the next peak higher in amplitude on the trace.

CALCulate:DPSA:MARKer<x>:PEAK:LEFT (No Query Form)

Moves the selected marker to the next peak to the left on the DPX spectrum trace.

Conditions Measurement views: DPX spectrum

Group Calculate commands

Syntax `CALCulate:DPSA:MARKer<x>:PEAK:LEFT`

Related Commands CALCulate:DPSA:MARKer<x>:PEAK:RIGHt

Arguments None

Examples `CALCULATE:DPSA:MARKER1:PEAK:LEFT` moves Marker 1 (M1) to the next peak to the left on the trace.

CALCulate:DPSA:MARKer<x>:PEAK:LOWer (No Query Form)

Moves the selected marker to the next peak lower in amplitude on the DPX spectrum trace.

Conditions Measurement views: DPX spectrum

Group Calculate commands

Syntax `CALCulate:DPSA:MARKer<x>:PEAK:LOWer`

Related Commands CALCulate:DPSA:MARKer<x>:PEAK:HIGHer

Arguments None

Examples `CALCULATE:DPSA:MARKER1:PEAK:LOWER` moves Marker 1 (M1) to the next peak lower in amplitude on the trace.

CALCulate:DPSA:MARKer<x>:PEAK:RIGHt (No Query Form)

Moves the selected marker to the next peak to the right on the DPX spectrum trace.

Conditions Measurement views: DPX spectrum

Group Calculate commands

Syntax `CALCulate:DPSA:MARKer<x>:PEAK:RIGHt`

Related Commands CALCulate:DPSA:MARKer<x>:PEAK:LEFT

Arguments None

Examples `CALCULATE:DPSA:MARKER1:PEAK:RIGHT` moves Marker 1 (M1) to the next peak to the right on the trace.

CALCulate:DPSA:MARKer<x>[:SET]:CENTer (No Query Form)

Sets the center frequency to the marker frequency in the DPX spectrum view.

Conditions Measurement views: DPX spectrum

Group Calculate commands

Syntax
```
CALCulate:DPSA:MARKer<x>[:SET]:CENTer
```

Arguments None

Examples `CALCULATE:DPSA:MARKER1:SET:CENTER` sets the center frequency to the marker frequency in the DPX spectrum view.

CALCulate:DPSA:MARKer<x>:TRACe

Selects or queries the trace on which the specified marker is placed in the DPX spectrum measurement.

Conditions Measurement views: DPX spectrum

Group Calculate commands

Syntax
```
CALCulate:DPSA:MARKer<x>:TRACe { BITMAP | TRACE1 | TRACE2 |
TRACE3 | TRACE4 }
CALCulate:DPSA:MARKer<x>:TRACe?
```

Arguments

`BITMAP` places the specified marker on the bitmap trace.

`TRACE1` places the specified marker on the +peak trace.

`TRACE2` places the specified marker on the -peak trace.

`TRACE3` places the specified marker on the average trace.

`TRACE4` places the specified marker on the math trace.

Examples `CALCULATE:DPSA:MARKER1:TRACE TRACE1` places Marker 1 (M1) on the maximum trace.

CALCulate:DPSA:MARKer<x>:X:AMPLitude

Sets or queries the amplitude position of the selected marker in the DPX spectrum view. This command is valid for the marker on the bitmap trace (refer to the CALCulate:DPSA:MARKer<x>:TRACe command). The frequency position is set by the CALCulate:DPSA:MARKer<x>:X[:FREQuency] command.

Conditions Measurement views: DPX spectrum

Group Calculate commands

Syntax

```
CALCulate:DPSA:MARKer<x>:X:AMPLitude <value>
CALCulate:DPSA:MARKer<x>:X:AMPLitude
```

Arguments `<value>::=<NRf>` specifies the amplitude position of the marker.
Range: -100 to 0 dBm.

Examples `CALCULATE:DPSA:MARKER1:X:AMPLITUDE –34.5dBm` places Marker 1 (M1) at –34.5 dBm.

CALCulate:DPSA:MARKer<x>:X[:FREQuency]

Sets or queries the frequency position of the selected marker in the DPX spectrum view.

Conditions Measurement views: DPX spectrum

Group Calculate commands

Syntax

```
CALCulate:DPSA:MARKer<x>:X[:FREQuency] <value>
CALCulate:DPSA:MARKer<x>:X[:FREQuency]?
```

Related Commands CALCulate:DPSA:MARKer<x>:Y?

Arguments `<value>::=<NRf>` specifies the frequency position of the marker.
Range: (center frequency) ± (span)/2.

Examples `CALCULATE:DPSA:MARKER1:X:FREQUENCY 800MHz` places Marker 1 (M1) at 800 MHz on the trace.

CALCulate:DPSA:MARKer<x>:Y? (Query Only)

Queries the vertical position of the selected marker in the DPX spectrum view. The data occurence rate is returned for the bitmap trace, and the amplitude value for the +peak, -peak, average, and math traces. The horizontal position can be set by the CALCulate:DPSA:MARKer<x>:X:AMPLitude and CALCulate:DPSA:MARKer<x>:X[:FREQuency] commands.

Conditions Measurement views: DPX spectrum

Group Calculate commands

Syntax `CALCulate:DPSA:MARKer<x>:Y?`

Arguments None

Returns `<NRf>` The value type depends on which trace the marker is placed on (refer to the CALCulate:DPSA:MARKer<x>:TRACe command):

The amplitude value is returned in dBm for the marker on the +peak, -peak, average, or math trace. The data occurence rate is returned in percent (%) for the marker on the bitmap trace.

Examples `CALCULATE:DPSA:MARKER1:Y?` might return `-34.28` indicating Marker 1 (M1) is at -34.28 dBm when it is placed on the +peak, -peak, average, or math trace.

`CALCULATE:DPSA:MARKER1:Y?` might return `76.5` indicating Marker 1 (M1) is at 76.5% when it is placed on the bitmap trace.

CALCulate:EVM:MARKer<x>:DELTa:X? (Query Only)

Returns the delta marker time for the selected marker in the EVM versus Time measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

Conditions Measurement views: EVM versus Time

Group Calculate commands

Syntax `CALCulate:EVM:MARKer<x>:DELTa:X?`

Related Commands CALCulate:EVM:MARKer<x>:DELTa:Y?

Arguments None

Returns `<NRf>` Delta marker time for the selected marker.

The unit can be changed by the [SENSe]:DDEMod:TIMe:UNITs command.

Examples `CALCulate:EVM:MARKer1:DELTa:X?` might return `9.52`, indicating that the delta marker time is 9.52 symbols.

CALCulate:EVM:MARKer<x>:DELTa:Y? (Query Only)

Returns the delta marker amplitude for the selected marker in the EVM versus Time measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

Conditions Measurement views: EVM versus Time

Group Calculate commands

Syntax `CALCulate:EVM:MARKer<x>:DELTa:Y?`

Related Commands CALCulate:EVM:MARKer<x>:DELTa:X?

Arguments

Returns `<NRf>` Delta marker amplitude for the selected marker.

Examples `CALCULATE:EVM:MARKER1:DELTA:Y?` might return `1.62`, indicating that the delta marker amplitude is 1.62%.

CALCulate:EVM:MARKer<x>:MAXimum (No Query Form)

Moves the selected marker to the highest peak on the trace in the EVM versus Time measurement.

Conditions Measurement views: EVM versus Time

Group Calculate commands

Syntax `CALCulate:EVM:MARKer<x>:MAXimum`

Arguments None

Examples `CALCULATE:EVM:MARKER1:MAXIMUM` moves Marker 1 (M1) to the highest peak on the trace.

CALCulate:EVM:MARKer<x>:PEAK:HIGHer (No Query Form)

Moves the selected marker to the next peak higher in amplitude on the EVM versus Time trace.

Conditions Measurement views: EVM versus Time

Group Calculate commands

Syntax `CALCulate:EVM:MARKer<x>:PEAK:HIGHer`

Related Commands CALCulate:EVM:MARKer<x>:PEAK:LOWer

Arguments None

Examples `CALCULATE:EVM:MARKER1:PEAK:HIGHER` moves Marker 1 (M1) to the next peak higher in amplitude on the trace.

CALCulate:EVM:MARKer<x>:PEAK:LEFT (No Query Form)

Moves the selected marker to the next peak to the left on the trace in the EVM versus Time measurement.

Conditions Measurement views: EVM versus Time

Group Calculate commands

Syntax `CALCulate:EVM:MARKer<x>:PEAK:LEFT`

Related Commands CALCulate:EVM:MARKer<x>:PEAK:RIGHt

Arguments None

Examples `CALCULATE:EVM:MARKER1:PEAK:LEFT` moves Marker 1 (M1) to the next peak to the left on the trace.

CALCulate:EVM:MARKer<x>:PEAK:LOWer (No Query Form)

Moves the selected marker to the next peak lower in amplitude on the EVM versus Time trace.

Conditions Measurement views: EVM versus Time

Group Calculate commands

Syntax `CALCulate:EVM:MARKer<x>:PEAK:LOWer`

Related Commands CALCulate:EVM:MARKer<x>:PEAK:HIGHer

Arguments None

Examples `CALCULATE:EVM:MARKER1:PEAK:LOWER` moves Marker 1 (M1) to the next peak lower in amplitude on the trace.

CALCulate:EVM:MARKer<x>:PEAK:RIGHt (No Query Form)

Moves the selected marker to the next peak to the right on the trace in the EVM versus Time measurement.

Conditions Measurement views: EVM versus Time

Group Calculate commands

Syntax

```
CALCulate:EVM:MARKer<x>:PEAK:RIGHt
```

Related Commands CALCulate:EVM:MARKer<x>:PEAK:LEFT

Arguments None

Examples `CALCULATE:EVM:MARKER1:PEAK:RIGHT` moves Marker 1 (M1) to the next peak to the right on the trace.

CALCulate:EVM:MARKer<x>:X

Sets or queries the horizontal position of the selected marker in the EVM versus Time measurement.

Conditions Measurement views: EVM versus Time

Group Calculate commands

Syntax

```
CALCulate:EVM:MARKer<x>:X <value>
CALCulate:EVM:MARKer<x>:X?
```

Arguments `<value>::=<NRf>` specifies the horizontal position of the marker.
Range: (analysis offset) to (analysis offset) + (analysis length).
The unit can be changed by the [SENSe]:DDEMod:TIMe:UNITs command.

Examples `CALCULATE:EVM:MARKER1:X 1.5us` places Marker 1 (M1) at 1.5 μ s on the trace.

CALCulate:EVM:MARKer<x>:Y? (Query Only)

Queries the vertical position of the selected marker in the EVM versus Time measurement.

Conditions Measurement views: EVM versus Time

Group Calculate commands

Syntax `CALCulate:EVM:MARKer<x>:Y?`

Related Commands CALCulate:EVM:MARKer<x>:X

Arguments None

Examples `CALCULATE:EVM:MARKER1:Y?` might return `-15.34E+6`, indicating Marker 1 (M1) is at -15.34 MHz.

CALCulate:FVTime:MARKer<x>:DELTa:X? (Query Only)

Returns the delta marker time for the selected marker in the Frequency versus Time measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

Conditions Measurement views: Frequency versus Time

Group Calculate commands

Syntax `CALCulate:FVTime:MARKer<x>:DELTa:X?`

Related Commands CALCulate:FVTime:MARKer<x>:DELTa:Y?

Arguments None

Returns `<NRf>` Delta marker time for the selected marker.

Examples `CALCULATE:FVTIME:MARKER1:DELTA:X?` might return `120.0E-9`, indicating that the delta marker time is 120 ns.

CALCulate:FVTime:MARKer<x>:DELTa:Y? (Query Only)

Returns the delta marker frequency for the selected marker in the Frequency versus Time measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

Conditions Measurement views: Frequency versus Time

Group Calculate commands

Syntax `CALCulate:FVTime:MARKer<x>:DELTa:Y?`

Related Commands CALCulate:FVTime:MARKer<x>:DELTa:X?

Arguments None

Returns `<NRf>` Delta marker frequency for the selected marker.

Examples `CALCULATE:FVTIME:MARKER1:DELTA:Y?` might return `27.05E+3`, indicating that the delta marker frequency is 27.05 kHz.

CALCulate:FVTime:MARKer<x>:MAXimum (No Query Form)

Moves the selected marker to the highest peak on the trace in the Frequency versus Time measurement.

Conditions Measurement views: Frequency versus Time

Group Calculate commands

Syntax `CALCulate:FVTime:MARKer<x>:MAXimum`

Arguments None

Examples `CALCULATE:FVTIME:MARKER1:MAXIMUM` moves Marker 1 (M1) to the highest peak on the trace.

CALCulate:FVTime:MARKer<x>:PEAK:HIGHer (No Query Form)

Moves the selected marker to the next peak higher in amplitude on the Frequency versus Time trace.

Conditions Measurement views: Frequency versus Time

Group Calculate commands

Syntax `CALCulate:FVTime:MARKer<x>:PEAK:HIGHer`

Related Commands CALCulate:FVTime:MARKer<x>:PEAK:LOWer

Arguments None

Examples `CALCULATE:FVTIME:MARKER1:PEAK:HIGHER` moves Marker 1 (M1) to the next peak higher in amplitude on the trace.

CALCulate:FVTime:MARKer<x>:PEAK:LEFT (No Query Form)

Moves the selected marker to the next peak to the left on the trace in the Frequency versus Time measurement.

Conditions Measurement views: Frequency versus Time

Group Calculate commands

Syntax `CALCulate:FVTime:MARKer<x>:PEAK:LEFT`

Related Commands CALCulate:FVTime:MARKer<x>:PEAK:RIGHt

Arguments None

Examples `CALCULATE:FVTIME:MARKER1:PEAK:LEFT` moves Marker 1 (M1) to the next peak to the left on the trace.

CALCulate:FVTime:MARKer<x>:PEAK:LOWer (No Query Form)

Moves the selected marker to the next peak lower in amplitude on the Frequency versus Time trace.

Conditions Measurement views: Frequency versus Time

Group Calculate commands

Syntax `CALCulate:FVTime:MARKer<x>:PEAK:LOWer`

Related Commands CALCulate:FVTime:MARKer<x>:PEAK:HIGHer

Arguments None

Examples `CALCULATE:FVTIME:MARKER1:PEAK:LOWER` moves Marker 1 (M1) to the next peak lower in amplitude on the trace.

CALCulate:FVTime:MARKer<x>:PEAK:RIGHt (No Query Form)

Moves the selected marker to the next peak to the right on the trace in the Frequency versus Time measurement.

Conditions Measurement views: Frequency versus Time

Group Calculate commands

Syntax `CALCulate:FVTime:MARKer<x>:PEAK:RIGHt`

Related Commands CALCulate:FVTime:MARKer<x>:PEAK:LEFT

Arguments None

Examples `CALCULATE:FVTIME:MARKER1:PEAK:RIGHT` moves Marker 1 (M1) to the next peak to the right on the trace.

CALCulate:FVTime:MARKer<x>:X

Sets or queries the horizontal position of the selected marker in the Frequency versus Time measurement.

Conditions Measurement views: Frequency versus Time

Group Calculate commands

Syntax
```
CALCulate:FVTime:MARKer<x>:X <value>
CALCulate:FVTime:MARKer<x>:X?
```

Related Commands CALCulate:FVTime:MARKer<x>:Y?

Arguments `<value>::=<NRf>` specifies the horizontal position of the marker.
Range: (analysis offset) to [(analysis offset) + (analysis length)].

Examples `CALCULATE:FVTIME:MARKER1:X 1.5u` places Marker 1 (M1) at 1.5 µs on the trace.

CALCulate:FVTime:MARKer<x>:Y? (Query Only)

Queries the vertical position of the selected marker in the Frequency versus Time measurement.

Conditions Measurement views: Frequency versus Time

Group Calculate commands

Syntax
```
CALCulate:FVTime:MARKer<x>:Y?
```

Related Commands CALCulate:FVTime:MARKer<x>:X

Arguments None

Examples `CALCULATE:FVTIME:MARKER1:Y?` might return `-15.34E+6`, indicating Marker 1 (M1) is at -15.34 MHz.

CALCulate:IQVTime:MARKer<x>:DELTa:X? (Query Only)

Returns the delta marker time for the selected marker in the RF I&Q versus Time measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

Conditions Measurement views: RF I&Q versus Time

Group Calculate commands

Syntax `CALCulate:IQVTime:MARKer<x>:DELTa:X?`

Related Commands CALCulate:IQVTime:MARKer<x>:DELTa:Y?

Arguments None

Returns `<NRf>` Delta marker time for the selected marker.

Examples `CALCULATE:IQVTIME:MARKER1:DELTA:X?` might return `120.0E-9`, indicating that the delta marker time is 120 ns.

CALCulate:IQVTime:MARKer<x>:DELTa:Y? (Query Only)

Returns the delta marker amplitude for the selected marker in the RF I&Q versus Time measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

Conditions Measurement views: RF I&Q versus Time

Group Calculate commands

Syntax `CALCulate:IQVTime:MARKer<x>:DELTa:Y?`

Related Commands CALCulate:IQVTime:MARKer<x>:DELTa:X?

Arguments None

Returns `<NRf>` Delta marker amplitude for the selected marker.

Examples `CALCULATE:IQVTIME:MARKER1:DELTA:Y?` might return `-3.45E-3`, indicating that the delta marker amplitude is -3.45 mV.

CALCulate:IQVTime:MARKer<x>:MAXimum (No Query Form)

Moves the selected marker to the highest peak on the trace in the RF I&Q versus Time measurement.

Conditions Measurement views: RF I&Q versus Time

Group Calculate commands

Syntax `CALCulate:IQVTime:MARKer<x>:MAXimum`

Arguments None

Examples `CALCULATE:IQVTIME:MARKER1:MAXIMUM` moves Marker 1 (M1) to the highest peak on the trace.

CALCulate:IQVTime:MARKer<x>:PEAK:HIGHer (No Query Form)

Moves the selected marker to the next peak higher in amplitude on the RF I&Q versus Time trace.

Conditions Measurement views: RF I&Q versus Time

Group Calculate commands

Syntax `CALCulate:IQVTime:MARKer<x>:PEAK:HIGHer`

Arguments None

Examples `CALCULATE:IQVTIME:MARKER1:PEAK:HIGHER` moves Marker 1 (M1) to the next peak higher in amplitude on the trace.

CALCulate:IQVTime:MARKer<x>:PEAK:LEFT (No Query Form)

Moves the selected marker to the next peak to the left on the trace in the RF I&Q versus Time measurement.

Conditions Measurement views: RF I&Q versus Time

Group Calculate commands

Syntax `CALCulate:IQVTime:MARKer<x>:PEAK:LEFT`

Related Commands CALCulate:IQVTime:MARKer<x>:PEAK:RIGHt

Arguments None

Examples `CALCULATE:IQVTIME:MARKER1:PEAK:LEFT` moves Marker 1 (M1) to the next peak to the left on the trace.

CALCulate:IQVTime:MARKer<x>:PEAK:LOWer (No Query Form)

Moves the selected marker to the next peak lower in amplitude on the RF I&Q versus Time trace.

Conditions Measurement views: RF I&Q versus Time

Group Calculate commands

Syntax `CALCulate:IQVTime:MARKer<x>:PEAK:LOWer`

Related Commands CALCulate:IQVTime:MARKer<x>:PEAK:HIGHer

Arguments None

Examples `CALCULATE:IQVTIME:MARKER1:PEAK:LOWER` moves Marker 1 (M1) to the next peak lower in amplitude on the trace.

CALCulate:IQVTime:MARKer<x>:PEAK:RIGHt (No Query Form)

Moves the selected marker to the next peak to the right on the trace in the RF I&Q versus Time measurement.

Conditions Measurement views: RF I&Q versus Time

Group Calculate commands

Syntax `CALCulate:IQVTime:MARKer<x>:PEAK:RIGHt`

Related Commands CALCulate:IQVTime:MARKer<x>:PEAK:LEFT

Arguments None

Examples `CALCULATE:IQVTIME:MARKER1:PEAK:RIGHT` moves Marker 1 (M1) to the next peak to the right on the trace.

CALCulate:IQVTime:MARKer<x>:TRACe

Places the selected marker on the I or Q trace in the RF I&Q versus Time measurement. The query command returns which trace the selected marker is placed on.

Conditions Measurement views: RF I&Q versus Time

Group Calculate commands

Syntax
```
CALCulate:IQVTime:MARKer<x>:TRACe { TRACE1 | TRACE2 }
CALCulate:IQVTime:MARKer<x>:TRACe?
```

Arguments TRACE1 places the selected marker on the I trace.

TRACE2 places the selected marker on the Q trace.

Examples CALCULATE:IQVTIME:MARKER1:TRACE TRACE1 places Marker 1 (M1) on the I trace.

CALCulate:IQVTime:MARKer<x>:X

Sets or queries the horizontal position of the selected marker in the RF I&Q versus Time measurement.

Conditions Measurement views: RF I&Q versus Time

Group Calculate commands

Syntax
```
CALCulate:IQVTime:MARKer<x>:X <value>
CALCulate:IQVTime:MARKer<x>:X?
```

Related Commands CALCulate:IQVTime:MARKer<x>:Y?

Arguments <value>::=<NRf> specifies the horizontal position of the marker.
Range: (analysis offset) to [(analysis offset) + (analysis length)].

Examples CALCULATE:IQVTIME:MARKER1:X 1.5us places Marker 1 (M1) at 1.5 µs on the trace.

CALCulate:IQVTime:MARKer<x>:Y? (Query Only)

Queries the vertical position of the selected marker in the RF I&Q versus Time measurement.

Conditions Measurement views: RF I&Q versus Time

Group Calculate commands

Syntax `CALCulate:IQVTime:MARKer<x>:Y?`

Related Commands CALCulate:IQVTime:MARKer<x>:X

Arguments None

Examples `CALCULATE:IQVTIME:MARKER1:Y?` might return `25.803E-3`, indicating Marker 1 (M1) is at 25.803 mV.

CALCulate:MARKer:ADD (No Query Form)

Adds a marker. Every execution of this command adds a marker from MR, then M1 to M4, sequentially.

NOTE. *If all markers are already turned on, the error message "Cannot add another marker" (execution error -200) is returned.*

Conditions Measurement views: All

Group Calculate commands

Syntax `CALCulate:MARKer:ADD`

Related Commands CALCulate:MARKer:AOFF, CALCulate:MARKer:DELete

Arguments None

Examples `CALCULATE:MARKER:ADD` adds a marker.

CALCulate:MARKer:AOFF (No Query Form)

Turns off all markers.

Conditions Measurement views: All

Group Calculate commands

Syntax `CALCulate:MARKer:AOFF`

Related Commands CALCulate:MARKer:ADD, CALCulate:MARKer:DELete

Arguments None

Examples `CALCULATE:MARKER:AOFF` turns off all markers.

CALCulate:MARKer:DELete (No Query Form)

Deletes the last marker added.

NOTE. *If all markers are turned off, the error message "Cannot delete another marker" (execution error -200) is returned.*

Conditions Measurement views: All

Group Calculate commands

Syntax `CALCulate:MARKer:DELete`

Related Commands CALCulate:MARKer:ADD, CALCulate:MARKer:AOFF

Arguments None

Examples `CALCULATE:MARKER:DELETE` deletes the last marker added.

CALCulate:MARKer:DENSity:EXCursion

Sets or queries the minimum excursion of DPX signal density, or how far the density (hit count for pixels) must be above the surrounding noise to be detected as a peak. This command is effective for the DPX bitmap trace. For marker peak up and marker peak down operations, the signal is considered to be a peak if it exceeds the excursion and the threshold set by the CALCulate:MARKer:DENSity:THReshold command. For marker peak left and marker peak right operations, the signal is considered to be a peak if it first exceeds the amplitude excursion set by CALCulate:MARKer:PEAK:EXCursion, the amplitude threshold set by CALCulate:MARKer:PEAK:THReshold, the density excursion set by CALCulate:MARKer:DENSity:EXCursion and the density threshold set by CALCulate:MARKer:DENSity:THReshold.

Conditions Measurement views: All

Group Calculate commands

Syntax

```
CALCulate:MARKer:DENSity:EXCursion <number>
CALCulate:MARKer:DENSity:EXCursion?
```

Related Commands CALCulate:MARKer:DENSity:THReshold, CALCulate:MARKer:PEAK:EXCursion, CALCulate:MARKer:PEAK:THReshold

Arguments `<number>::=<NR1>` specifies the minimum excursion density.
Range: 1 to 50000.

Examples `CALCULATE:MARKER:DENSITY:EXCURSION 30` sets the minimum excursion density to 30.

CALCulate:MARKer:DENSity:SMOothing

Sets or queries the number of pixels squared for smoothing the DPX signal density. This command is effective for the DPX bitmap trace.

Conditions Measurement views: All

Group Calculate commands

Syntax

```
CALCulate:MARKer:DENSity:SMOothing <number>
CALCulate:MARKer:DENSity:SMOothing?
```

Arguments `<number>::=<NR1>` specifies the number of pixels squared for smoothing the DPX signal density. Range: 1 to 20.

Examples `CALCULATE:MARKER:DENSITY:SMOOTHING 5` sets the smoothing number to 5.

CALCulate:MARKer:DENSity:THReshold

Sets or queries the threshold of DPX signal density above which the density (hit count for pixels) is detected as a peak. This command is effective for the DPX bitmap trace.

Conditions Measurement views: All

Group Calculate commands

Syntax

```
CALCulate:MARKer:DENSity:THReshold <number>
CALCulate:MARKer:DENSity:THReshold?
```

Related Commands CALCulate:MARKer:DENSity:EXCursion

Arguments `<number>::=<NR1>` specifies the threshold density for detecting peaks. Range: 1 to 50000.

Examples `CALCULATE:MARKER:DENSITY:THRESHOLD 300` sets the threshold density to 300.

CALCulate:MARKer:MODE

Selects or queries the marker mode.

Conditions Measurement views: All

Group Calculate commands

Syntax

```
CALCulate:MARKer:MODE { ABSolute | DELTa }
CALCulate:MARKer:MODE?
```

Arguments `ABSolute` selects the absolute marker mode, in which the marker readout indicates the absolute value.

`DELTa` selects the delta marker mode, in which the marker readout indicates the relative value to the reference marker.

Examples `CALCULATE:MARKER:MODE DELTa` selects the delta marker mode.

CALCulate:MARKer:PEAK:EXCursion

Sets or queries the minimum excursion level (how far a signal must be above the surrounding noise to be detected as a peak). The signal is considered to be a peak if it exceeds the minimum excursion level and the threshold level set by the CALCulate:MARKer:PEAK:THReshold command.

Conditions Measurement views: All

Group Calculate commands

Syntax

```
CALCulate:MARKer:PEAK:EXCursion <value>
CALCulate:MARKer:PEAK:EXCursion?
```

Arguments `<value>::=<NRf>` specifies the excursion level. Range: 0 to 100 dB.

Examples `CALCULATE:MARKER:PEAK:EXCURSION 10` sets the excursion level to 10 dB.

CALCulate:MARKer:PEAK:THReshold

Sets or queries the threshold level above which a signal is detected as a peak.

Conditions Measurement views: All

Group Calculate commands

Syntax

```
CALCulate:MARKer:PEAK:THReshold <value>
CALCulate:MARKer:PEAK:THReshold?
```

Arguments `<value>::=<NRf>` specifies the threshold level for detecting peaks. Range: -170 to +130 dBm.

Examples `CALCULATE:MARKER:PEAK:THRESHOLD -50` sets the threshold level to -50 dBm.

CALCulate:MCPower:MARKer<x>:DELTa:X? (Query Only)

Returns the delta marker frequency for the selected marker in the MCPR measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

Conditions Measurement views: MCPR

Group Calculate commands

Syntax `CALCulate:MCPower:MARKer<x>:DELTa:X?`

Related Commands CALCulate:MCPower:MARKer<x>:DELTa:Y?

Arguments None

Returns `<NRf>` Delta marker frequency for the selected marker.

Examples `CALCULATE:MCPOWER:MARKER1:DELTA:X?` might return `1.28E+6`, indicating that the delta marker frequency is 1.28 MHz.

CALCulate:MCPower:MARKer<x>:DELTa:Y? (Query Only)

Returns the delta marker amplitude for the selected marker in the MCPR measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

Conditions Measurement views: MCPR

Group Calculate commands

Syntax `CALCulate:MCPower:MARKer<x>:DELTa:Y?`

Related Commands CALCulate:MCPower:MARKer<x>:DELTa:X?

Arguments None

Returns `<NRf>` Delta marker amplitude for the selected marker.

Examples `CALCULATE:MCPOWER:MARKER1:DELTA:Y?` might return `23.45`, indicating that the delta marker amplitude is 23.45 dB.

CALCulate:MCPower:MARKer<x>:MAXimum (No Query Form)

Moves the selected marker to the highest peak on the trace in the MCPR measurement.

Conditions Measurement views: MCPR

Group Calculate commands

Syntax `CALCulate:MCPower:MARKer<x>:MAXimum`

Arguments None

Examples `CALCULATE:MCPOWER:MARKER1:MAXIMUM` moves Marker 1 (M1) to the highest peak on the trace.

CALCulate:MCPower:MARKer<x>:PEAK:LEFT (No Query Form)

Moves the selected marker to the next peak to the left on the trace in the MCPR measurement.

Conditions Measurement views: MCPR

Group Calculate commands

Syntax `CALCulate:MCPower:MARKer<x>:PEAK:LEFT`

Related Commands CALCulate:MCPower:MARKer<x>:PEAK:RIGHt

Arguments None

Examples `CALCULATE:MCPOWER:MARKER1:PEAK:LEFT` moves Marker 1 (M1) to the next peak to the left on the trace.

CALCulate:MCPower:MARKer<x>:PEAK:RIGHt (No Query Form)

Moves the selected marker to the next peak to the right on the trace in the MCPR measurement.

Conditions Measurement views: MCPR

Group Calculate commands

Syntax `CALCulate:MCPower:MARKer<x>:PEAK:RIGHt`

Related Commands CALCulate:MCPower:MARKer<x>:PEAK:LEFT

Arguments None

Examples `CALCULATE:MCPOWER:MARKER1:PEAK:RIGHT` moves Marker 1 (M1) to the next peak to the right on the trace.

CALCulate:MCPower:MARKer<x>:X

Sets or queries the horizontal position of the selected marker in the MCPR measurement.

Conditions Measurement views: MCPR

Group Calculate commands

Syntax

```
CALCulate:MCPower:MARKer<x>:X <value>
CALCulate:MCPower:MARKer<x>:X?
```

Related Commands CALCulate:MCPower:MARKer<x>:Y?

Arguments `<value>::=<NRf>` specifies the horizontal position of the marker.
Range: (center frequency) ± (span)/2.

Examples `CALCULATE:MCPOWER:MARKER1:X 800MHz` places Marker 1 (M1) at 800 MHz on the trace.

CALCulate:MCPower:MARKer<x>:Y? (Query Only)

Queries the vertical position of the selected marker in the MCPR measurement.

Conditions Measurement views: MCPR

Group Calculate commands

Syntax `CALCulate:MCPower:MARKer<x>:Y?`

Related Commands CALCulate:MCPower:MARKer<x>:X

Arguments None

Examples `CALCULATE:MCPOWER:MARKER1:Y?` might return `-34.28`, indicating Marker 1 (M1) is at -34.28 dBm.

CALCulate:MERRor:MARKer<x>:DELTa:X? (Query Only)

Returns the delta marker time for the selected marker in the Magnitude error versus Time measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

Conditions Measurement views: Magnitude error versus Time

Group Calculate commands

Syntax `CALCulate:MERRor:MARKer<x>:DELTa:X?`

Related Commands CALCulate:MERRor:MARKer<x>:DELTa:Y?

Arguments None

Returns `<NRf>` Delta marker time for the selected marker.

The unit can be changed by the [SENSe]:DDEMod:TIMe:UNITs command.

Examples `CALCULATE:MERROR:MARKER1:DELTA:X?` might return `9.52`, indicating that the delta marker time is 9.52 symbols.

CALCulate:MERRor:MARKer<x>:DELTa:Y? (Query Only)

Returns the delta marker amplitude for the selected marker in the Magnitude error versus Time measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

Conditions Measurement views: Magnitude error versus Time

Group Calculate commands

Syntax `CALCulate:MERRor:MARKer<x>:DELTa:Y?`

Related Commands CALCulate:MERRor:MARKer<x>:DELTa:X?

Arguments None

Returns `<NRf>` Delta marker amplitude for the selected marker.

Examples `CALCULATE:MERROR:MARKER1:DELTA:Y?` might return `3.84`, indicating that the delta marker amplitude is 3.84%.

CALCulate:MERRor:MARKer<x>:MAXimum (No Query Form)

Moves the selected marker to the highest peak on the trace in the Magnitude error versus Time measurement.

Conditions Measurement views: Magnitude error versus Time

Group Calculate commands

Syntax `CALCulate:MERRor:MARKer<x>:MAXimum`

Arguments None

Examples `CALCULATE:MERROR:MARKER1:MAXIMUM` moves Marker 1 (M1) to the highest peak on the trace.

CALCulate:MERRor:MARKer<x>:PEAK:HIGHer (No Query Form)

Moves the selected marker to the next peak higher in amplitude on the Magnitude error versus Time trace.

Conditions Measurement views: Magnitude error versus Time

Group Calculate commands

Syntax `CALCulate:MERRor:MARKer<x>:PEAK:HIGHer`

Related Commands CALCulate:MERRor:MARKer<x>:PEAK:LOWer

Arguments None

Examples `CALCULATE:MERROR:MARKER1:PEAK:HIGHER` moves Marker 1 (M1) to the next peak higher in amplitude on the trace.

CALCulate:MERRor:MARKer<x>:PEAK:LEFT (No Query Form)

Moves the selected marker to the next peak to the left on the trace in the Magnitude error versus Time measurement.

Conditions Measurement views: Magnitude error versus Time

Group Calculate commands

Syntax `CALCulate:MERRor:MARKer<x>:PEAK:LEFT`

Related Commands CALCulate:MERRor:MARKer<x>:PEAK:RIGHt

Arguments None

Examples `CALCULATE:MERROR:MARKER1:PEAK:LEFT` moves Marker 1 (M1) to the next peak to the left on the trace.

CALCulate:MERRor:MARKer<x>:PEAK:LOWer (No Query Form)

Moves the selected marker to the next peak lower in amplitude on the Magnitude error versus Time trace.

Conditions Measurement views: Magnitude error versus Time

Group Calculate commands

Syntax `CALCulate:MERRor:MARKer<x>:PEAK:LOWer`

Related Commands CALCulate:MERRor:MARKer<x>:PEAK:HIGHer

Arguments None

Examples `CALCULATE:MERROR:MARKER1:PEAK:LOWER` moves Marker 1 (M1) to the next peak lower in amplitude on the trace.

CALCulate:MERRor:MARKer<x>:PEAK:RIGHt (No Query Form)

Moves the selected marker to the next peak to the right on the trace in the Magnitude error versus Time measurement.

Conditions Measurement views: Magnitude error versus Time

Group Calculate commands

Syntax

```
CALCulate:MERRor:MARKer<x>:PEAK:RIGHt
```

Related Commands CALCulate:MERRor:MARKer<x>:PEAK:LEFT

Arguments None

Examples `CALCULATE:MERROR:MARKER1:PEAK:RIGHT` moves Marker 1 (M1) to the next peak to the right on the trace.

CALCulate:MERRor:MARKer<x>:X

Sets or queries the horizontal position of the selected marker in the Magnitude error versus Time measurement.

Conditions Measurement views: Magnitude error versus Time

Group Calculate commands

Syntax

```
CALCulate:MERRor:MARKer<x>:X <value>
CALCulate:MERRor:MARKer<x>:X?
```

Arguments `<value>::=<NRf>` specifies the horizontal position of the marker.
Range: (analysis offset) to [(analysis offset) + (analysis length)].
The unit can be changed by the [SENSe]:DDEMod:TIMe:UNITs command.

Examples `CALCULATE:MERROR:MARKER1:X 1.5u` places Marker 1 (M1) at 1.5 µs on the trace.

CALCulate:MERRor:MARKer<x>:Y? (Query Only)

Queries the vertical position of the selected marker in the Magnitude error versus Time measurement.

Conditions Measurement views: Magnitude error versus Time

Group Calculate commands

Syntax `CALCulate:MERRor:MARKer<x>:Y?`

Related Commands CALCulate:MERRor:MARKer<x>:X

Arguments None

Examples `CALCULATE:MERROR:MARKER1:Y?` might return `-15.34E+6`, indicating Marker 1 (M1) is at -15.34 MHz.

CALCulate:OBWidth:MARKer<x>:DELTa:X? (Query Only)

Returns the delta marker frequency for the selected marker in the Occupied Bandwidth measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

Conditions Measurement views: Occupied Bandwidth

Group Calculate commands

Syntax `CALCulate:OBWidth:MARKer<x>:DELTa:X?`

Related Commands CALCulate:OBWidth:MARKer<x>:DELTa:Y?

Arguments None

Returns `<NRf>` Delta marker frequency for the selected marker.

Examples `CALCULATE:OBWIDTH:MARKER1:DELTA:X?` might return `1.28E+6`, indicating that the delta marker frequency is 1.28 MHz.

CALCulate:OBWidth:MARKer<x>:DELTa:Y? (Query Only)

Returns the delta marker amplitude for the selected marker in the Occupied Bandwidth measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

Conditions Measurement views: Occupied Bandwidth

Group Calculate commands

Syntax `CALCulate:OBWidth:MARKer<x>:DELTa:Y?`

Related Commands CALCulate:OBWidth:MARKer<x>:DELTa:X?

Arguments None

Returns `<NRf>` Delta marker amplitude for the selected marker.

Examples `CALCULATE:OBWIDTH:MARKER1:DELTA:Y?` might return `23.45`, indicating that the delta marker amplitude is 23.45 dB.

CALCulate:OBWidth:MARKer<x>:MAXimum (No Query Form)

Moves the selected marker to the highest peak on the trace in the Occupied Bandwidth measurement.

Conditions Measurement views: Occupied Bandwidth

Group Calculate commands

Syntax `CALCulate:OBWidth:MARKer<x>:MAXimum`

Arguments None

Examples `CALCULATE:OBWIDTH:MARKER1:MAXIMUM` moves Marker 1 (M1) to the highest peak on the trace.

CALCulate:OBWidth:MARKer<x>:PEAK:HIGHer (No Query Form)

Moves the selected marker to the next peak higher in amplitude on the Occupied Bandwidth trace.

Conditions Measurement views: Occupied Bandwidth

Group Calculate commands

Syntax `CALCulate:OBWidth:MARKer<x>:PEAK:HIGHer`

Related Commands CALCulate:OBWidth:MARKer<x>:PEAK:LOWer

Arguments None

Examples `CALCULATE:OBWIDTH:MARKER1:PEAK:HIGHER` moves Marker 1 (M1) to the next peak higher in amplitude on the trace.

CALCulate:OBWidth:MARKer<x>:PEAK:LEFT (No Query Form)

Moves the selected marker to the next peak to the left on the trace in the Occupied Bandwidth measurement.

Conditions Measurement views: Occupied Bandwidth

Group Calculate commands

Syntax `CALCulate:OBWidth:MARKer<x>:PEAK:LEFT`

Related Commands CALCulate:OBWidth:MARKer<x>:PEAK:RIGHt

Arguments None

Examples `CALCULATE:OBWIDTH:MARKER1:PEAK:LEFT` moves Marker 1 (M1) to the next peak to the left on the trace.

CALCulate:OBWidth:MARKer<x>:PEAK:LOWer (No Query Form)

Moves the selected marker to the next peak lower in amplitude on the Occupied Bandwidth trace.

Conditions Measurement views: Occupied Bandwidth

Group Calculate commands

Syntax `CALCulate:OBWidth:MARKer<x>:PEAK:LOWer`

Related Commands CALCulate:OBWidth:MARKer<x>:PEAK:HIGHer

Arguments None

Examples `CALCULATE:OBWIDTH:MARKER1:PEAK:LOWER` moves Marker 1 (M1) to the next peak lower in amplitude on the trace.

CALCulate:OBWidth:MARKer<x>:PEAK:RIGHt (No Query Form)

Moves the selected marker to the next peak to the right on the trace in the Occupied Bandwidth measurement.

Conditions Measurement views: Occupied Bandwidth

Group Calculate commands

Syntax `CALCulate:OBWidth:MARKer<x>:PEAK:RIGHt`

Related Commands CALCulate:OBWidth:MARKer<x>:PEAK:LEFT

Arguments None

Examples `CALCULATE:OBWIDTH:MARKER1:PEAK:RIGHT` moves Marker 1 (M1) to the next peak to the right on the trace.

CALCulate:OBWidth:MARKer<x>[:SET]:CENTer (No Query Form)

Sets the center frequency to the value at the marker position in the Occupied Bandwidth measurement.

Conditions Measurement views: Occupied Bandwidth

Group Calculate commands

Syntax `CALCulate:OBWidth:MARKer<x>[:SET]:CENTer`

Arguments None

Examples `CALCULATE:OBWIDTH:MARKER1:SET:CENTER` sets the center frequency to the value at the Marker 1 position.

CALCulate:OBWidth:MARKer<x>:X

Sets or queries the horizontal position of the selected marker in the Occupied Bandwidth measurement.

Conditions Measurement views: Occupied Bandwidth

Group Calculate commands

Syntax
```
CALCulate:OBWidth:MARKer<x>:X <value>
CALCulate:OBWidth:MARKer<x>:X?
```

Related Commands CALCulate:OBWidth:MARKer<x>:Y?

Arguments `<value>::=<NRf>` specifies the horizontal position of the marker.
Range: (center frequency) ± (span)/2.

Examples `CALCULATE:OBWIDTH:MARKER1:X 800MHz` places Marker 1 (M1) at 800 MHz on the trace.

CALCulate:OBWidth:MARKer<x>:Y? (Query Only)

Queries the vertical position of the selected marker in the Occupied Bandwidth measurement.

Conditions Measurement views: Occupied Bandwidth

Group Calculate commands

Syntax `CALCulate:OBWidth:MARKer<x>:Y?`

Related Commands CALCulate:OBWidth:MARKer<x>:X

Arguments None

Examples `CALCULATE:OBWIDTH:MARKER1:Y?` might return -34.28 indicating Marker 1 (M1) is at -34.28 dBm.

CALCulate:PERRor:MARKer<x>:DELTa:X? (Query Only)

Returns the delta marker time for the selected marker in the Phase error versus Time measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

Conditions Measurement views: Phase error versus Time

Group Calculate commands

Syntax `CALCulate:PERRor:MARKer<x>:DELTa:X?`

Related Commands CALCulate:PERRor:MARKer<x>:DELTa:Y?

Arguments None

Returns `<NRf>` Delta marker time for the selected marker.

The unit can be changed by the [SENSe]:DDEMod:TIMe:UNITs command.

Examples `CALCULATE:PERROR:MARKER1:DELTA:X?` might return `9.52`, indicating that the delta marker time is 9.52 symbols.

CALCulate:PERRor:MARKer<x>:DELTa:Y? (Query Only)

Returns the delta marker phase for the selected marker in the Phase error versus Time measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

Conditions Measurement views: Phase error versus Time

Group Calculate commands

Syntax `CALCulate:PERRor:MARKer<x>:DELTa:Y?`

Related Commands CALCulate:PERRor:MARKer<x>:DELTa:X?

Arguments None

Returns `<NRf>` Delta marker phase for the selected marker.

Examples `CALCULATE:PERROR:MARKER1:DELTA:Y?` might return `-7.93`, indicating that the delta marker phase is -7.93°.

CALCulate:PERRor:MARKer<x>:MAXimum (No Query Form)

Moves the selected marker to the highest peak on the trace in the Phase error versus Time measurement.

Conditions Measurement views: Phase error versus Time

Group Calculate commands

Syntax `CALCulate:PERRor:MARKer<x>:MAXimum`

Arguments None

Examples `CALCULATE:PERROR:MARKER1:MAXIMUM` moves Marker 1 (M1) to the highest peak on the trace.

CALCulate:PERRor:MARKer<x>:PEAK:HIGHer (No Query Form)

Moves the selected marker to the next peak higher in amplitude on the Phase error versus Time trace.

Conditions Measurement views: Phase error versus Time

Group Calculate commands

Syntax `CALCulate:PERRor:MARKer<x>:PEAK:HIGHer`

Related Commands CALCulate:PERRor:MARKer<x>:PEAK:LOWer

Arguments None

Examples `CALCULATE:PERROR:MARKER1:PEAK:HIGHER` moves Marker 1 (M1) to the next peak higher in amplitude on the trace.

CALCulate:PERRor:MARKer<x>:PEAK:LEFT (No Query Form)

Moves the selected marker to the next peak to the left on the trace in the Phase error versus Time measurement.

Conditions Measurement views: Phase error versus Time

Group Calculate commands

Syntax `CALCulate:PERRor:MARKer<x>:PEAK:LEFT`

Related Commands CALCulate:PERRor:MARKer<x>:PEAK:RIGHt

Arguments None

Examples `CALCULATE:PERROR:MARKER1:PEAK:LEFT` moves Marker 1 (M1) to the next peak to the left on the trace.

CALCulate:PERRor:MARKer<x>:PEAK:LOWer (No Query Form)

Moves the selected marker to the next peak lower in amplitude on the Phase error versus Time trace.

Conditions Measurement views: Phase error versus Time

Group Calculate commands

Syntax `CALCulate:PERRor:MARKer<x>:PEAK:LOWer`

Related Commands CALCulate:PERRor:MARKer<x>:PEAK:HIGHer

Arguments None

Examples `CALCULATE:PERROR:MARKER1:PEAK:LOWER` moves Marker 1 (M1) lower in amplitude to the next peak on the trace.

CALCulate:PERRor:MARKer<x>:PEAK:RIGHt (No Query Form)

Moves the selected marker to the next peak to the right on the trace in the Phase error versus Time measurement.

Conditions Measurement views: Phase error versus Time

Group Calculate commands

Syntax `CALCulate:PERRor:MARKer<x>:PEAK:RIGHt`

Related Commands CALCulate:PERRor:MARKer<x>:PEAK:LEFT

Arguments None

Examples `CALCULATE:PERROR:MARKER1:PEAK:RIGHT` moves Marker 1 (M1) to the next peak to the right on the trace.

CALCulate:PERRor:MARKer<x>:X

Sets or queries the horizontal position of the selected marker in the Phase error versus Time measurement.

Conditions Measurement views: Phase error versus Time

Group Calculate commands

Syntax
```
CALCulate:PERRor:MARKer<x>:X <value>
CALCulate:PERRor:MARKer<x>:X?
```

Arguments `<value>::=<NRf>` specifies the horizontal position of the marker.
Range: (analysis offset) to [(analysis offset) + (analysis length)].
The unit can be changed by the [SENSe]:DDEMod:TIMe:UNITs command.

Examples `CALCULATE:PERROR:MARKER1:X 1.5u` places Marker 1 (M1) at 1.5 µs on the trace.

CALCulate:PERRor:MARKer<x>:Y? (Query Only)

Queries the vertical position of the selected marker in the Phase error versus Time measurement.

Conditions Measurement views: Phase error versus Time

Group Calculate commands

Syntax `CALCulate:PERRor:MARKer<x>:Y?`

Related Commands CALCulate:PERRor:MARKer<x>:X

Arguments None

Examples `CALCULATE:PERROR:MARKER1:Y?` might return `-15.34E+6`, indicating Marker 1 (M1) is at -15.34 MHz.

CALCulate:PHVTime:MARKer<x>:DELTa:X? (Query Only)

Returns the delta marker time for the selected marker in the Phase versus Time measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

Conditions Measurement views: Phase versus Time

Group Calculate commands

Syntax `CALCulate:PHVTime:MARKer<x>:DELTa:X?`

Related Commands CALCulate:PHVTime:MARKer<x>:DELTa:Y?

Arguments None

Returns `<NRf>` Delta marker time for the selected marker.

Examples `CALCULATE:PHVTIME:MARKER1:DELTA:X?` might return `38.0E-9`, indicating that the delta marker time is 38.0 ns.

CALCulate:PHVTime:MARKer<x>:DELTa:Y? (Query Only)

Returns the delta marker phase for the selected marker in the Phase versus Time measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

Conditions Measurement views: Phase versus Time

Group Calculate commands

Syntax `CALCulate:PHVTime:MARKer<x>:DELTa:Y?`

Related Commands CALCulate:PHVTime:MARKer<x>:DELTa:X?

Arguments None

Returns `<NRf>` Delta marker phase for the selected marker.

Examples `CALCULATE:PHVTIME:MARKER1:DELTA:Y?` might return `162.38`, indicating that the delta marker phase is 162.38°.

CALCulate:PHVTime:MARKer<x>:MAXimum (No Query Form)

Moves the selected marker to the highest peak on the trace in the Phase versus Time measurement.

Conditions Measurement views: Phase versus Time

Group Calculate commands

Syntax `CALCulate:PHVTime:MARKer<x>:MAXimum`

Arguments None

Examples `CALCULATE:PHVTIME:MARKER1:MAXIMUM` moves Marker 1 (M1) to the highest peak on the trace.

CALCulate:PHVTime:MARKer<x>:PEAK:HIGHer (No Query Form)

Moves the selected marker to the next peak higher in amplitude on the Phase versus Time trace.

Conditions Measurement views: Phase versus Time

Group Calculate commands

Syntax `CALCulate:PHVTime:MARKer<x>:PEAK:HIGHer`

Related Commands CALCulate:PHVTime:MARKer<x>:PEAK:LOWer

Arguments None

Examples `CALCULATE:PHVTIME:MARKER1:PEAK:HIGHER` moves Marker 1 (M1) to the next peak higher in amplitude on the trace.

CALCulate:PHVTime:MARKer<x>:PEAK:LEFT (No Query Form)

Moves the selected marker to the next peak to the left on the trace in the Phase versus Time measurement.

Conditions Measurement views: Phase versus Time

Group Calculate commands

Syntax `CALCulate:PHVTime:MARKer<x>:PEAK:LEFT`

Related Commands CALCulate:PHVTime:MARKer<x>:PEAK:RIGHt

Arguments None

Examples `CALCULATE:PHVTIME:MARKER1:PEAK:LEFT` moves Marker 1 (M1) to the next peak to the left on the trace.

CALCulate:PHVTime:MARKer<x>:PEAK:LOWer (No Query Form)

Moves the selected marker to the next peak lower in amplitude on the Phase versus Time trace.

Conditions Measurement views: Phase versus Time

Group Calculate commands

Syntax `CALCulate:PHVTime:MARKer<x>:PEAK:LOWer`

Related Commands CALCulate:PHVTime:MARKer<x>:PEAK:HIGHer

Arguments None

Examples `CALCULATE:PHVTIME:MARKER1:PEAK:LOWER` moves Marker 1 (M1) to the next peak lower in amplitude on the trace.

CALCulate:PHVTime:MARKer<x>:PEAK:RIGHt (No Query Form)

Moves the selected marker to the next peak to the right on the trace in the Phase versus Time measurement.

Conditions Measurement views: Phase versus Time

Group Calculate commands

Syntax `CALCulate:PHVTime:MARKer<x>:PEAK:RIGHt`

Related Commands CALCulate:PHVTime:MARKer<x>:PEAK:LEFT

Arguments None

Examples `CALCULATE:PHVTIME:MARKER1:PEAK:RIGHT` moves Marker 1 (M1) to the next peak to the right on the trace.

CALCulate:PHVTime:MARKer<x>:X

Sets or queries the horizontal position of the selected marker in the Phase versus Time measurement.

Conditions Measurement views: Phase versus Time

Group Calculate commands

Syntax

```
CALCulate:PHVTime:MARKer<x>:X <value>
CALCulate:PHVTime:MARKer<x>:X?
```

Related Commands CALCulate:PHVTime:MARKer<x>:Y?

Arguments `<value>::=<NRf>` specifies the horizontal position of the marker.
Range: (analysis offset) to [(analysis offset) + (analysis length)].

Examples `CALCULATE:PHVTIME:MARKER1:X 1.5US` places Marker 1 (M1) at 1.5 μs on the trace.

CALCulate:PHVTime:MARKer<x>:Y? (Query Only)

Queries the vertical position of the selected marker in the Phase versus Time measurement.

Conditions Measurement views: Phase versus Time

Group Calculate commands

Syntax

```
CALCulate:PHVTime:MARKer<x>:Y?
```

Related Commands CALCulate:PHVTime:MARKer<x>:X

Arguments None

Examples `CALCULATE:PHVTIME:MARKER1:Y?` might return `-18.435`, indicating Marker 1 (M1) is at -18.435°.

CALCulate:PULSe:STATistics:MARKer<x>:DELTa:X? (Query Only)

Returns the delta marker frequency for the selected marker on the pulse trace. This command is valid when DISPlay:PULSe:STATistics:PLOT is set to FFT.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

Conditions Measurement views: Pulse statistics

Group Calculate commands

Syntax `CALCulate:PULSe:STATistics:MARKer<x>:DELTa:X?`

Related Commands DISPlay:PULSe:STATistics:PLOT, CALCulate:PULSe:STATistics:MARKer<x>:DELTa:Y?

Returns `<NRf>` Delta marker frequency for the selected marker.

Examples `CALCULATE:PULSE:STATISTICS:MARKER1:DELTA:X?` might return `614.2`, indicating that the delta marker frequency is 614.2 Hz.

CALCulate:PULSe:STATistics:MARKer<x>:DELTa:Y? (Query Only)

Returns the delta marker amplitude for the selected marker on the pulse trace. This command is valid when DISPlay:PULSe:STATistics:PLOT is set to FFT.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

Conditions Measurement views: Pulse statistics

Group Calculate commands

Syntax `CALCulate:PULSe:STATistics:MARKer<x>:DELTa:Y?`

Related Commands CALCulate:PULSe:STATistics:MARKer<x>:DELTa:X?

Arguments

Returns `<NRf>` Delta marker amplitude for the selected marker.

Examples `CALCULATE:PULSE:STATISTICS:MARKER1:DELTA:Y?` might return `2.345`, indicating that the delta marker amplitude is 2.345 dB.

CALCulate:PULSe:STATistics:MARKer<x>:MAXimum (No Query Form)

Moves the selected marker to the highest peak on the pulse statistics trace. This command is valid when DISPlay:PULSe:STATistics:PLOT is set to FFT.

Conditions Measurement views: Pulse statistics

Group Calculate commands

Syntax `CALCulate:PULSe:STATistics:MARKer<x>:MAXimum`

Arguments None

Examples `CALCULATE:PULSE:STATISTICS:MARKER1:MAXIMUM` moves Marker 1 (M1) to the highest peak on the trace.

CALCulate:PULSe:STATistics:MARKer<x>:PEAK:HIGHer (No Query Form)

Moves the selected marker to the next peak higher in amplitude on the pulse statistics trace. This command is valid when DISPlay:PULSe:STATistics:PLOT is set to FFT.

Conditions Measurement views: Pulse statistics

Group Calculate commands

Syntax `CALCulate:PULSe:STATistics:MARKer<x>:PEAK:HIGHer`

Related Commands CALCulate:PULSe:STATistics:MARKer<x>:PEAK:LOWer

Arguments None

Examples `CALCULATE:PULSE:STATISTICS:MARKER1:PEAK:HIGHER` moves Marker 1 (M1) to the next peak higher in amplitude on the trace.

CALCulate:PULSe:STATistics:MARKer<x>:PEAK:LEFT (No Query Form)

Moves the selected marker to the next peak to the left on the pulse statistics trace. This command is valid when DISPlay:PULSe:STATistics:PLOT is set to FFT.

Conditions Measurement views: Pulse statistics

Group Calculate commands

Syntax `CALCulate:PULSe:STATistics:MARKer<x>:PEAK:LEFT`

Related Commands CALCulate:PULSe:STATistics:MARKer<x>:PEAK:RIGHt

Arguments None

Examples `CALCULATE:PULSE:STATISTICS:MARKER1:PEAK:LEFT` moves Marker 1 (M1) to the next peak to the left on the trace.

CALCulate:PULSe:STATistics:MARKer<x>:PEAK:LOWer (No Query Form)

Moves the selected marker to the next peak lower in amplitude on the pulse statistics trace. This command is valid when DISPlay:PULSe:STATistics:PLOT is set to FFT.

Conditions Measurement views: Pulse statistics

Group Calculate commands

Syntax `CALCulate:PULSe:STATistics:MARKer<x>:PEAK:LOWer`

Related Commands CALCulate:PULSe:STATistics:MARKer<x>:PEAK:HIGHer

Arguments None

Examples `CALCULATE:PULSE:STATISTICS:MARKER1:PEAK:LOWER` moves Marker 1 (M1) to the next peak lower in amplitude on the trace.

CALCulate:PULSe:STATistics:MARKer<x>:PEAK:RIGHt (No Query Form)

Moves the selected marker to the next peak to the right on the pulse statistics trace. This command is valid when DISPlay:PULSe:STATistics:PLOT is set to FFT.

Conditions Measurement views: Pulse statistics

Group Calculate commands

Syntax `CALCulate:PULSe:STATistics:MARKer<x>:PEAK:RIGHt`

Related Commands CALCulate:PULSe:STATistics:MARKer<x>:PEAK:LEFT

Arguments None

Examples `CALCULATE:PULSE:STATISTICS:MARKER1:PEAK:RIGHT` moves Marker 1 (M1) to the next peak to the right on the trace.

CALCulate:PULSe:STATistics:MARKer<x>:X

Sets or queries the horizontal position of the selected marker in the pulse statistics view. This command is valid when DISPlay:PULSe:STATistics:PLOT is set to FFT.

Conditions Measurement views: Pulse statistics

Group Calculate commands

Syntax

```
CALCulate:PULSe:STATistics:MARKer<x>:X <value>
CALCulate:PULSe:STATistics:MARKer<x>:X?
```

Related Commands CALCulate:PULSe:STATistics:MARKer<x>:Y?

Arguments `<value>::=<NRf>` specifies the horizontal position of the marker.
Range: 0 to half of the average repetition rate for all detected pulses in Hz.

Examples `CALCULATE:PULSE:STATISTICS:MARKER1:X 12.5kHz` places Marker 1 (M1) at 12.5 kHz on the trace.

CALCulate:PULSe:STATistics:MARKer<x>:Y? (Query Only)

Queries the vertical position of the selected marker in the pulse statistics view. This command is valid when DISPlay:PULSe:STATistics:PLOT is set to FFT.

Conditions Measurement views: Pulse statistics

Group Calculate commands

Syntax

```
CALCulate:PULSe:STATistics:MARKer<x>:Y?
```

Related Commands CALCulate:PULSe:STATistics:MARKer<x>:X

Arguments None

Examples `CALCULATE:PULSE:STATISTICS:MARKER1:Y?` might return `-28.173`, indicating Marker 1 (M1) is at -28.173 dB.

CALCulate:PULSe:TRACe:MARKer<x>:DELTa:X? (Query Only)

Returns the delta marker time for the selected marker on the pulse trace.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

Conditions Measurement views: Pulse trace

Group Calculate commands

Syntax `CALCulate:PULSe:TRACe:MARKer<x>:DELTa:X?`

Related Commands CALCulate:PULSe:TRACe:MARKer<x>:DELTa:Y?

Arguments None

Returns `<NRf>` Delta marker time for the selected marker.

Examples `CALCULATE:PULSE:TRACE:MARKER1:DELTA:X?` might return `38.0E-9`, indicating that the delta marker time is 38.0 ns.

CALCulate:PULSe:TRACe:MARKer<x>:DELTa:Y? (Query Only)

Returns the delta marker amplitude for the selected marker on the pulse trace.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

Conditions Measurement views: Pulse trace

Group Calculate commands

Syntax `CALCulate:PULSe:TRACe:MARKer<x>:DELTa:Y?`

Related Commands CALCulate:PULSe:TRACe:MARKer<x>:DELTa:X?

Arguments None

Returns `<NRf>` Delta marker amplitude for the selected marker.

Examples `CALCULATE:PULSE:TRACE:MARKER1:DELTA:Y?` might return `23.45`, indicating that the delta marker amplitude is 23.45 dB.

CALCulate:PULSe:TRACe:MARKer<x>:MAXimum (No Query Form)

Moves the selected marker to the highest peak on the pulse trace.

Conditions Measurement views: Pulse trace

Group Calculate commands

Syntax `CALCulate:PULSe:TRACe:MARKer<x>:MAXimum`

Arguments None

Examples `CALCULATE:PULSE:TRACE:MARKER1:MAXIMUM` moves Marker 1 (M1) to the highest peak on the trace.

CALCulate:PULSe:TRACe:MARKer<x>:PEAK:HIGHer (No Query Form)

Moves the selected marker to the next peak higher in amplitude on the pulse trace.

Conditions Measurement views: Pulse trace

Group Calculate commands

Syntax `CALCulate:PULSe:TRACe:MARKer<x>:PEAK:HIGHer`

Related Commands CALCulate:PULSe:TRACe:MARKer<x>:PEAK:LOWer

Arguments None

Examples `CALCULATE:PULSE:TRACE:MARKER1:PEAK:HIGHER` moves Marker 1 (M1) to the next peak higher in amplitude on the trace.

CALCulate:PULSe:TRACe:MARKer<x>:PEAK:LEFT (No Query Form)

Moves the selected marker to the next peak to the left on the pulse trace.

Conditions Measurement views: Pulse trace

Group Calculate commands

Syntax `CALCulate:PULSe:TRACe:MARKer<x>:PEAK:LEFT`

Related Commands CALCulate:PULSe:TRACe:MARKer<x>:PEAK:RIGHt

Arguments None

Examples `CALCULATE:PULSE:TRACE:MARKER1:PEAK:LEFT` moves Marker 1 (M1) to the next peak to the left on the trace.

CALCulate:PULSe:TRACe:MARKer<x>:PEAK:LOWer (No Query Form)

Moves the selected marker to the next peak lower in amplitude on the pulse trace.

Conditions Measurement views: Pulse trace

Group Calculate commands

Syntax `CALCulate:PULSe:TRACe:MARKer<x>:PEAK:LOWer`

Related Commands CALCulate:PULSe:TRACe:MARKer<x>:PEAK:HIGHer

Arguments None

Examples `CALCULATE:PULSE:TRACE:MARKER1:PEAK:LOWER` moves Marker 1 (M1) to the next peak lower in amplitude on the trace.

CALCulate:PULSe:TRACe:MARKer<x>:PEAK:RIGHt (No Query Form)

Moves the selected marker to the next peak to the right on the pulse trace.

Conditions Measurement views: Pulse trace

Group Calculate commands

Syntax `CALCulate:PULSe:TRACe:MARKer<x>:PEAK:RIGHt`

Related Commands CALCulate:PULSe:TRACe:MARKer<x>:PEAK:LEFT

Arguments None

Examples `CALCULATE:PULSE:TRACE:MARKER1:PEAK:RIGHT` moves Marker 1 (M1) to the next peak to the right on the trace.

CALCulate:PULSe:TRACe:MARKer<x>:X

Sets or queries the horizontal position of the selected marker in the pulse trace view.

Conditions Measurement views: Pulse trace

Group Calculate commands

Syntax
```
CALCulate:PULSe:TRACe:MARKer<x>:X <value>
CALCulate:PULSe:TRACe:MARKer<x>:X?
```

Related Commands CALCulate:PULSe:TRACe:MARKer<x>:Y?

Arguments `<value>::=<NRf>` specifies the horizontal position of the marker.

***NOTE.** To set the horizontal position, the marker must be on the trace of the pulse selected using the DISPlay:PULSe:SELect:NUMBer command. You cannot put the marker out of the horizontal range of the pulse.*

Examples `CALCULATE:PULSE:TRACE:MARKER1:X 1.5us` places Marker 1 (M1) at 1.5 µs on the trace.

CALCulate:PULSe:TRACe:MARKer<x>:Y? (Query Only)

Queries the vertical position of the selected marker in the pulse trace view.

Conditions Measurement views: Pulse trace

Group Calculate commands

Syntax `CALCulate:PULSe:TRACe:MARKer<x>:Y?`

Related Commands CALCulate:PULSe:TRACe:MARKer<x>:X

Arguments None

Examples `CALCULATE:PULSE:TRACE:MARKER1:Y?` might return `228.858E-3`, indicating Marker 1 (M1) is at 228.858 mV.

CALCulate:SEARch:LIMit:FAIL? (Query Only)

Queries whether the waveform cuts across the limit in the search operation.

Conditions Measurement views: All

Group Calculate commands

Syntax `CALCulate:SEARCh:LIMit:FAIL?`

Arguments None

Returns `{ 0 | 1 }`

`0` represents Pass, indicating that the waveform does not cut across the limit.

`1` represents Fail, indicating that the waveform cuts across the limit.

Examples `CALCULATE:SEARCH:LIMIT:FAIL?` might return `1`, indicating that the waveform cuts across the limit (Fail).

CALCulate:SEARch:LIMit:MATCh:BEEP[:STATe]

Determines whether or not to beep on match during run or replay in the search operation.

Conditions Measurement views: All

Group Calculate commands

Syntax
```
CALCulate:SEARch:LIMit:MATCh:BEEP[:STATe] { OFF | ON | 0 |
1 }
CALCulate:SEARch:LIMit:MATCh:BEEP[:STATe]?
```

Arguments `OFF` or `0` disables to beep on match.

`ON` or `1` enables to beep on match.

Examples `CALCULATE:SEARCH:LIMIT:MATCH:BEEP:STATE ON` enables to beep on match.

CALCulate:SEARch:LIMit:MATCh:SACQuire[:STATe]

Determines whether or not to stop acquiring data on match during run or replay in the search operation.

Conditions Measurement views: All

Group Calculate commands

Syntax
```
CALCulate:SEARch:LIMit:MATCh:SACQuire[:STATe] { OFF | ON |
0 | 1 }
CALCulate:SEARch:LIMit:MATCh:SACQuire[:STATe]?
```

Arguments `OFF` or `0` disables to stop acquiring data on match.

`ON` or `1` enables to stop acquiring data on match.

Examples `CALCULATE:SEARCH:LIMIT:MATCH:SACQUIRE:STATE ON` enables to stop acquiring data on match.

CALCulate:SEARch:LIMit:MATCh:SDATa[:STATe]

Determines whether or not to save automatically (AutoSave) acquisition data on match during run in the search operation.

Conditions Measurement views: All

Group Calculate commands

Syntax
```
CALCulate:SEARCh:LIMit:MATCh:SDATa[:STATe] { OFF | ON | 0 |
1 }
CALCulate:SEARCh:LIMit:MATCh:SDATa[:STATe]?
```

Arguments OFF or 0 disables the AutoSave.

ON or 1 enables to save acquisition data automatically on match.
The data is saved to a file with the name:

<name>-yyyy.mm.dd.hh.mm.ss.sss.tiq

Where
<name> is the file name that was last specified.
yyyy.mm.dd and hh.mm.ss.sss represent date and 24 hour time.
The file extension is .tiq.
Example: SAVED-2007.03.20.12.34.567.tiq

For the directory of file, refer to *Specifying the File* (See page 2-40.) in the MMEMory command section.

Examples `CALCULATE:SEARCH:LIMIT:MATCH:SDATA:STATE ON` enables to save acquisition data automatically on match.

CALCulate:SEARCh:LIMit:MATCh:SPICture[:STATe]

Determines whether or not to save automatically (AutoSave) the whole screen on match during run in the search operation.

Conditions Measurement views: All

Group Calculate commands

Syntax
```
CALCulate:SEARCh:LIMit:MATCh:SPICture[:STATe] { OFF | ON |
0 | 1 }
CALCulate:SEARCh:LIMit:MATCh:SPICture[:STATe]?
```

Arguments OFF or 0 disables the AutoSave.

ON or 1 enables to save the whole screen automatically on match.

The picture is saved to a file with the name:

<name>-yyyy.mm.dd.hh.mm.ss.sss.png

Where
<name> is the file name that was last specified.
yyyy.mm.dd and hh.mm.ss.sss represent date and 24 hour time.
The file extension is .png.
Example: SAVED-2007.03.20.12.34.567.png

For the directory of file, refer to *Specifying the File* (See page 2-40.) in the MMEMory command section.

Examples

`CALCULATE:SEARCH:LIMIT:MATCH:SPICTURE:STATE ON` enables to save the whole screen automatically on match.

CALCulate:SEARch:LIMit:MATCh:STRace[:STATe]

Determines whether or not to save automatically (AutoSave) the spectrum trace on match during run in the search operation.

Conditions

Measurement views: All

Group

Calculate commands

Syntax

```
CALCulate:SEARch:LIMit:MATCh:STRace[:STATe] { OFF | ON | 0 |
1 }
CALCulate:SEARch:LIMit:MATCh:STRace[:STATe]?
```

Arguments

`OFF` or `0` disables the AutoSave.

`ON` or `1` enables to save the spectrum trace automatically on match.
The trace is saved to a file with the name:

<name>-yyyy.mm.dd.hh.mm.ss.sss.Specan

Where
<name> is the file name that was last specified.
yyyy.mm.dd and hh.mm.ss.sss represent date and 24 hour time.
The file extension is .Specan.
Example: SAVED-2007.03.20.12.34.567.Specan

For the directory of file, refer to *Specifying the File* (See page 2-40.) in the MMEMory command section.

Examples `CALCULATE:SEARCH:LIMIT:MATCH:STRACE:STATE ON` enables to save the spectrum trace automatically on match.

CALCulate:SEARch:LIMit:OPERation

Selects or queries the search condition.

Conditions Measurement views: All

Group Calculate commands

Syntax

```
CALCulate:SEARch:LIMit:OPERation { LT | GT | IMASk | OMASk }
CALCulate:SEARch:LIMit:OPERation?
```

Related Commands CALCulate:SEARch:LIMit:OPERation:SLIMit

Arguments The following table lists the arguments.

Search condition

Argument	Meaning
LT	The data is less than the limit value.
GT	The data is greater than the limit value.
IMASk	The data is inside the limit mask.
OMASk	The data is outside the limit mask.

You can select the data using the CALCulate:SEARch:LIMit:OPERation:FEED command.

You can set the limit value using the CALCulate:SEARch:LIMit:OPERation:SLIMit command.

You can store and load the limit mask using the CALCulate:SEARch:LIMit:OPERation:MASK:STORE and CALCulate:SEARch:LIMit:OPERation:MASK:LOAD commands.

Examples `CALCULATE:SEARCH:LIMIT:OPERATION GT` selects "the data is greater than the limit value" for the search condition.

CALCulate:SEARch:LIMit:OPERation:FEED

Sets or queries the data flow to be fed in the search operation.

Conditions Measurement views: All

Group Calculate commands

Syntax

```
CALCulate:SEARCh:LIMit:OPERation:FEED <view>,<trace>
CALCulate:SEARCh:LIMit:OPERation:FEED?
```

Arguments `<view>::=<string>` and `<trace>::=<string>` are listed in the following table.

Source data of the search operation

<view>	<trace>	Meaning
"Spectrum"	"Trace 1" [1]	Trace 1 in the Spectrum view.
	"Trace 2" [1]	Trace 2 in the Spectrum view.
	"Trace 3" [1]	Trace 3 in the Spectrum view.
	"Math Trace"	Math trace in the Spectrum view.
	"Spectrogram Trace"	Spectrogram trace in the Spectrum view.

[1] There is a space character between Trace and the number.

Examples `CALCULATE:SEARCH:LIMIT:OPERATION:FEED "Spectrum","Trace 1”` selects the Trace 1 in the Spectrum measurement view for the search operation.

`CALCULATE:SEARCH:LIMIT:OPERATION:FEED?` might return `"Spectrum","Math Trace"`, indicating that the math trace is used as the source data in the search operation.

CALCulate:SEARch:LIMit:OPERation:MASK:LOAD (No Query Form)

Loads the limit mask from a specified file for the search operation.

Conditions Measurement views: All

Group Calculate commands

Syntax

```
CALCulate:SEARCh:LIMit:OPERation:MASK:LOAD <file_name>
```

Arguments `<file_name>::=<string>` specifies the file to load the limit mask from. The file extension is .lmt. You can omit the extension.

For the directory of file, refer to *Specifying the File* (See page 2-40.) in the MMEMory command section.

Examples `CALCULATE:SEARCH:LIMIT:OPERATION:MASK:LOAD "Limit1"` loads the limit mask from the *Limit1.lmt* file.

CALCulate:SEARch:LIMit:OPERation:MASK:STORE (No Query Form)

Stores the limit mask to a specified file in the search operation.

Conditions Measurement views: All

Group Calculate commands

Syntax `CALCulate:SEARCH:LIMit:OPERation:MASK:STORE <file_name>`

Arguments `<file_name>::=<string>` specifies the file to store the limit mask to. The file extension is .lmt. You can omit the extension.

For the directory of file, refer to *Specifying the File* (See page 2-40.) in the MMEMory command section.

Examples `CALCULATE:SEARCH:LIMIT:OPERATION:MASK:STORE "Limit1"` stores the limit mask to the *Limit1.lmt* file.

CALCulate:SEARch:LIMit:OPERation:SLIMit

Sets or queries the limit value in the search operation.

Conditions Measurement views: All

Group Calculate commands

Syntax
```
CALCulate:SEARCH:LIMit:OPERation:SLIMit <value>
CALCulate:SEARCH:LIMit:OPERation:SLIMit?
```

Related Commands CALCulate:SEARch:LIMit:OPERation

Arguments `<value>::=<NRf>` specifies the limit value in the search operation. Range: -100 to +100 dBm.

Examples `CALCULATE:SEARCH:LIMIT:OPERATION:SLIMIT -20` sets the limit value to -20 dBm.

CALCulate:SEARch:LIMit:REPort:DATA? (Query Only)

Returns the frequency range(s) that satisfy the search condition.

Conditions Measurement views: All

Group Calculate commands

Syntax `CALCulate:SEARCH:LIMit:REPort:DATA?`

Arguments None

Returns `<num_range>,<range(1)>,<range(2)>,...,<range(n)>`

Where
`<num_range>::=<NR1>` is the number of ranges that satisfy the condition.
`<range(n)>::="<lower_freq(n)>, <upper_freq(n)>"` (string) represents the n^{th} frequency range that satisfy the search condition in ascending order. `<lower_freq(n)>` and `<upper_freq(n)>` are the lower and upper frequencies of the range #n, respectively.

Examples `CALCULATE:SEARCH:LIMIT:REPORT:DATA?` might return `2,"1.4800E+9, 1.5001E+9","1.5002E+9, 1.5200E+9"`, indicating that the search condition is satisfied in these two ranges 1.48 to 1.5001 GHz and 1.5002 to 1.52 GHz.

CALCulate:SEARch:LIMit:REPort:POINts? (Query Only)

Returns the number of frequency range(s) that satisfy the search condition.

Conditions Measurement views: All

Group Calculate commands

Syntax `CALCulate:SEARch:LIMit:REPort:POINts?`

Arguments None

Returns `<number>::=<NR1>` represents the number of frequency range(s) that satisfy the search condition.

Examples `CALCULATE:SEARCH:LIMIT:REPORT:POINTS?` might return `5`, indicating that five ranges satisfy the search condition.

CALCulate:SEARch:LIMit:STATe

Determines whether to enable or disable the search function.

Conditions Measurement views: All

Group Calculate commands

Syntax
```
CALCulate:SEARch:LIMit:STATe { OFF | ON | 0 | 1 }
CALCulate:SEARch:LIMit:STATe?
```

Arguments `OFF` or `0` disables the search function.

`ON` or `1` enables the search function.

Examples `CALCULATE:SEARCH:LIMIT:STATE ON` enables the search function.

CALCulate:SGRam:MARKer<x>:DELTa:X:FREQuency? (Query Only)

Returns the delta marker frequency for the selected marker in the spectrogram.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

Conditions Measurement views: Spectrogram

Group Calculate commands

Syntax `CALCulate:SGRam:MARKer<x>:DELTa:X:FREQuency?`

Related Commands CALCulate:SGRam:MARKer<x>:DELTa:Y?

Arguments None

Returns `<NRf>` Delta marker frequency for the selected marker.

Examples `CALCULATE:SGRAM:MARKER1:DELTA:X:FREQUENCY?` might return `5.95E+6`, indicating that the delta marker frequency is 5.95 MHz.

CALCulate:SGRam:MARKer<x>:DELTa:X[:TIME]? (Query Only)

Returns the delta marker time for the selected marker in the spectrogram.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

Conditions Measurement views: Spectrogram

Group Calculate commands

Syntax `CALCulate:SGRam:MARKer<x>:DELTa:X[:TIME]?`

Related Commands CALCulate:SGRam:MARKer<x>:DELTa:Y?

Arguments None

Returns `<NRf>` Delta marker time for the selected marker.

Examples `CALCULATE:SGRAM:MARKER1:DELTA:X:TIME?` might return `-1.84E-3`, indicating that the delta marker time is -1.84 ms.

CALCulate:SGRam:MARKer<x>:DELTa:Y? (Query Only)

Returns the delta marker amplitude for the selected marker in the spectrogram.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

Conditions Measurement views: Spectrogram

Group Calculate commands

Syntax `CALCulate:SGRam:MARKer<x>:DELTa:Y?`

Related Commands CALCulate:SGRam:MARKer<x>:DELTa:X[:TIME]?

Arguments None

Returns `<NRf>` Delta marker amplitude for the selected marker.

Examples `CALCULATE:SGRAM:MARKER1:DELTA:Y?` might return `-8.45`, indicating that the delta marker amplitude is -8.45 dB.

CALCulate:SGRam:MARKer<x>:MAXimum (No Query Form)

Moves the selected marker to the highest peak on a line in the spectrogram. The line is selected using the TRACe:SGRam:SELect:LINE command.

Conditions Measurement views: Spectrogram

Group Calculate commands

Syntax `CALCulate:SGRam:MARKer<x>:MAXimum`

Arguments None

Examples `CALCULATE:SGRAM:MARKER1:MAXIMUM` moves Marker 1 (M1) to the highest peak on the line.

CALCulate:SGRam:MARKer<x>:PEAK:HIGHer (No Query Form)

Moves the selected marker to the next peak higher in amplitude on a line in the spectrogram. The line is selected using the TRACe:SGRam:SELect:LINE command.

Conditions Measurement views: Spectrogram

Group Calculate commands

Syntax `CALCulate:SGRam:MARKer<x>:PEAK:HIGHer`

Related Commands CALCulate:SGRam:MARKer<x>:PEAK:LOWer

Arguments None

Examples `CALCULATE:SGRAM:MARKER1:PEAK:HIGHER` moves Marker 1 (M1) to the next peak higher in amplitude on the line.

CALCulate:SGRam:MARKer<x>:PEAK:LEFT (No Query Form)

Moves the selected marker to the next peak to the left on a line in the spectrogram. The line is selected using the TRACe:SGRam:SELect:LINE command.

Conditions Measurement views: Spectrogram

Group Calculate commands

Syntax `CALCulate:SGRam:MARKer<x>:PEAK:LEFT`

Related Commands CALCulate:SGRam:MARKer<x>:PEAK:RIGHt

Arguments None

Examples `CALCULATE:SGRAM:MARKER1:PEAK:LEFT` moves Marker 1 (M1) to the next peak to the left on the line.

CALCulate:SGRam:MARKer<x>:PEAK:LOWer (No Query Form)

Moves the selected marker to the next peak lower in amplitude on a line in the spectrogram. The line is selected using the TRACe:SGRam:SELect:LINE command.

Conditions Measurement views: Spectrogram

Group Calculate commands

Syntax `CALCulate:SGRam:MARKer<x>:PEAK:LOWer`

Related Commands CALCulate:SGRam:MARKer<x>:PEAK:HIGHer

Arguments None

Examples `CALCULATE:SGRAM:MARKER1:PEAK:LOWER` moves Marker 1 (M1) to the next peak lower in amplitude on the line.

CALCulate:SGRam:MARKer<x>:PEAK:RIGHt (No Query Form)

Moves the selected marker to the next peak to the right on a line in the spectrogram. The line is selected using the TRACe:SGRam:SELect:LINE command.

Conditions Measurement views: Spectrogram

Group Calculate commands

Syntax `CALCulate:SGRam:MARKer<x>:PEAK:RIGHt`

Related Commands CALCulate:SGRam:MARKer<x>:PEAK:LEFT

Arguments None

Examples `CALCULATE:SGRAM:MARKER1:PEAK:RIGHT` moves Marker 1 (M1) to the next peak to the right on the trace.

CALCulate:SGRam:MARKer<x>[:SET]:CENTer (No Query Form)

Sets the center frequency to the marker frequency in the spectrogram.

Conditions Measurement views: Spectrogram

Group Calculate commands

Syntax `CALCulate:SGRam:MARKer<x>[:SET]:CENTer`

Arguments None

Examples `CALCULATE:SGRAM:MARKER1:SET:CENTER` sets the center frequency to the Marker 1 frequency in the spectrogram.

CALCulate:SGRam:MARKer<x>:X:FREQuency

Sets or queries the marker frequency in the spectrogram.

Conditions Measurement views: Spectrogram

Group Calculate commands

Syntax
```
CALCulate:SGRam:MARKer<x>:X:FREQuency <value>
CALCulate:SGRam:MARKer<x>:X:FREQuency?
```

Related Commands CALCulate:SGRam:MARKer<x>:Y?

Arguments `<value>::=<NRf>` specifies the marker frequency.
Range: (center frequency) ± (span)/2.

Examples `CALCULATE:SGRAM:MARKER1:X 800MHz` places Marker 1 (M1) at 800 MHz on the trace.

CALCulate:SGRam:MARKer<x>:X[:TIME]

Sets or queries the marker time in the spectrogram.

Conditions Measurement views: Spectrogram

Group Calculate commands

Syntax

```
CALCulate:SGRam:MARKer<x>:X[:TIME] <value>
CALCulate:SGRam:MARKer<x>:X[:TIME]?
```

Related Commands CALCulate:SGRam:MARKer<x>:Y?

Arguments `<value>::=<NRf>` specifies the marker time.

Examples `CALCULATE:SGRAM:MARKER1:X:TIME -234.5us` places Marker 1 (M1) at -234.5 µs on the trace.

CALCulate:SGRam:MARKer<x>:Y? (Query Only)

Queries the marker amplitude in the spectrogram.

Conditions Measurement views: Spectrogram

Group Calculate commands

Syntax

```
CALCulate:SGRam:MARKer<x>:Y?
```

Related Commands CALCulate:SGRam:MARKer<x>:X[:TIME]

Arguments None

Examples `CALCULATE:SGRAM:MARKER1:Y?` might return `-34.28`, indicating Marker 1 (M1) is at -34.28 dBm.

CALCulate:SPECtrum:MARKer<x>:DELTa:X? (Query Only)

Returns the delta marker frequency for the selected marker on the spectrum trace.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

Conditions Measurement views: Spectrum

Group Calculate commands

Syntax `CALCulate:SPECtrum:MARKer<x>:DELTa:X?`

Related Commands CALCulate:SPECtrum:MARKer<x>:DELTa:Y?

Arguments None

Returns `<NRf>` Delta marker frequency for the selected marker.

Examples `CALCULATE:SPECTRUM:MARKER1:DELTA:X?` might return `1.28E+6`, indicating that the delta marker frequency is 1.28 MHz.

CALCulate:SPECtrum:MARKer<x>:DELTa:Y? (Query Only)

Returns the delta marker amplitude for the selected marker on the spectrum trace.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

Conditions Measurement views: Spectrum

Group Calculate commands

Syntax `CALCulate:SPECtrum:MARKer<x>:DELTa:Y?`

Related Commands CALCulate:SPECtrum:MARKer<x>:DELTa:X?

Arguments None

Returns `<NRf>` Delta marker amplitude for the selected marker.

Examples `CALCULATE:SPECTRUM:MARKER1:DELTA:Y?` might return `23.45`, indicating that the delta marker amplitude is 23.45 dB.

CALCulate:SPECtrum:MARKer<x>:MAXimum (No Query Form)

Moves the selected marker to the highest peak on the spectrum trace.

Conditions Measurement views: Spectrum

Group Calculate commands

Syntax `CALCulate:SPECtrum:MARKer<x>:MAXimum`

Arguments None

Examples `CALCULATE:SPECTRUM:MARKER1:MAXIMUM` moves Marker 1 (M1) to the highest peak on the trace.

CALCulate:SPECtrum:MARKer<x>:PEAK:HIGHer (No Query Form)

Moves the selected marker to the next peak higher in amplitude on the spectrum trace.

Conditions Measurement views: Spectrum

Group Calculate commands

Syntax `CALCulate:SPECtrum:MARKer<x>:PEAK:HIGHer`

Related Commands CALCulate:SPECtrum:MARKer<x>:PEAK:LOWer

Arguments None

Examples `CALCULATE:SPECTRUM:MARKER1:PEAK:HIGHER` moves Marker 1 (M1) to the next peak higher in amplitude on the trace.

CALCulate:SPECtrum:MARKer<x>:PEAK:LEFT (No Query Form)

Moves the selected marker to the next peak to the left on the spectrum trace.

Conditions Measurement views: Spectrum

Group Calculate commands

Syntax `CALCulate:SPECtrum:MARKer<x>:PEAK:LEFT`

Related Commands CALCulate:SPECtrum:MARKer<x>:PEAK:RIGHt

Arguments None

Examples `CALCULATE:SPECTRUM:MARKER1:PEAK:LEFT` moves Marker 1 (M1) to the next peak to the left on the trace.

CALCulate:SPECtrum:MARKer<x>:PEAK:LOWer (No Query Form)

Moves the selected marker to the next peak lower in amplitude on the spectrum trace.

Conditions Measurement views: Spectrum

Group Calculate commands

Syntax `CALCulate:SPECtrum:MARKer<x>:PEAK:LOWer`

Related Commands CALCulate:SPECtrum:MARKer<x>:PEAK:HIGHer

Arguments None

Examples `CALCULATE:SPECTRUM:MARKER1:PEAK:LOWER` moves Marker 1 (M1) to the next peak lower in amplitude on the trace.

CALCulate:SPECtrum:MARKer<x>:PEAK:RIGHt (No Query Form)

Moves the selected marker to the next peak to the right on the spectrum trace.

Conditions Measurement views: Spectrum

Group Calculate commands

Syntax `CALCulate:SPECtrum:MARKer<x>:PEAK:RIGHt`

Related Commands CALCulate:SPECtrum:MARKer<x>:PEAK:LEFT

Arguments None

Examples `CALCULATE:SPECTRUM:MARKER1:PEAK:RIGHT` moves Marker 1 (M1) to the next peak to the right on the trace.

CALCulate:SPECtrum:MARKer<x>[:SET]:CENTer (No Query Form)

Sets the center frequency to the marker frequency in the spectrum measurement.

Conditions Measurement views: Spectrum

Group Calculate commands

Syntax `CALCulate:SPECtrum:MARKer<x>[:SET]:CENTer`

Arguments None

Examples `CALCULATE:SPECTRUM:MARKER1:SET:CENTER` sets the center frequency to the marker frequency in the spectrum measurement.

CALCulate:SPECtrum:MARKer<x>:TRACe

Selects or queries the trace on which the specified marker is placed in the spectrum measurement.

Conditions

Measurement views: Spectrum

Group

Calculate commands

Syntax

```
CALCulate:SPECtrum:MARKer<x>:TRACe { TRACE1 | TRACE2 |
TRACE3 | TRACE4 | TRACE5 }
CALCulate:SPECtrum:MARKer<x>:TRACe?
```

Arguments

`TRACE1` places the specified marker on Trace 1.

`TRACE2` places the specified marker on Trace 2.

`TRACE3` places the specified marker on n Trace 3.

`TRACE4` places the specified marker on Trace 4 (math trace).

`TRACE5` places the specified marker on Trace 5 (spectrogram).

Trace 1 to 3 can be defined as Normal, Average, Max Hold or Min Hold using the TRACe<x>:SPECtrum:FUNCtion command.

Examples

`CALCULATE:SPECTRUM:MARKER1:TRACE TRACE1` places Marker 1 (M1) on Trace 1.

CALCulate:SPECtrum:MARKer<x>:X

Sets or queries the horizontal position of the selected marker on the spectrum trace.

Conditions

Measurement views: Spectrum

Group

Calculate commands

Syntax

```
CALCulate:SPECtrum:MARKer<x>:X <value>
CALCulate:SPECtrum:MARKer<x>:X?
```

Related Commands

CALCulate:SPECtrum:MARKer<x>:Y?

Arguments `<value>::=<NRf>` specifies the horizontal position of the marker.
Range: (center frequency) ± (span)/2.

Examples `CALCULATE:SPECTRUM:MARKER1:X 800MHz` places Marker 1 (M1) at 800 MHz on the spectrum trace.

CALCulate:SPECtrum:MARKer<x>:Y? (Query Only)

Queries the vertical position of the selected marker on the spectrum trace.

Conditions Measurement views: Spectrum

Group Calculate commands

Syntax `CALCulate:SPECtrum:MARKer<x>:Y?`

Related Commands CALCulate:SPECtrum:MARKer<x>:X

Arguments None

Examples `CALCULATE:SPECTRUM:MARKER1:Y?` might return `-34.28`, indicating Marker 1 (M1) is at -34.28 dBm.

CALCulate:SPURious:MARKer<x>:DELTa:X? (Query Only)

Returns the delta marker frequency for the selected marker on the spectrum trace.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

Conditions Measurement views: Spurious

Group Calculate commands

Syntax `CALCulate:SPURious:MARKer<x>:DELTa:X?`

Arguments None

Returns `<NRf>` Delta marker frequency for the selected marker.

Examples `CALCULATE:SPURIOUS:MARKER1:DELTA:X?` might return `1.28E+6`, indicating that the delta marker frequency is 1.28 MHz.

CALCulate:SPURious:MARKer<x>:DELTa:Y? (Query Only)

Returns the delta marker amplitude for the selected marker on the spectrum trace.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

Conditions Measurement views: Spurious

Group Calculate commands

Syntax `CALCulate:SPURious:MARKer<x>:DELTa:Y?`

Arguments None

Returns `<NRf>` Delta marker amplitude for the selected marker.

Examples `CALCULATE:SPURIOUS:MARKER1:DELTA:Y?` might return `23.45`, indicating that the delta marker amplitude is 23.45 dB.

CALCulate:SPURious:MARKer<x>:MAXimum (No Query Form)

Moves the selected marker to the highest peak on the spectrum trace.

Conditions Measurement views: Spurious

Group Calculate commands

Syntax `CALCulate:SPURious:MARKer<x>:MAXimum`

Arguments None

Examples `CALCULATE:SPURIOUS:MARKER1:MAXIMUM` moves Marker 1 (M1) to the highest peak on the trace.

CALCulate:SPURious:MARKer<x>:PEAK:HIGHer (No Query Form)

Moves the selected marker to the next peak higher in amplitude on the spectrum trace.

Conditions Measurement views: Spurious

Group Calculate commands

Syntax `CALCulate:SPURious:MARKer<x>:PEAK:HIGHer`

Arguments None

Examples `CALCULATE:SPURIOUS:MARKER1:PEAK:HIGHER` moves Marker 1 (M1) to the next peak higher in amplitude on the trace.

CALCulate:SPURious:MARKer<x>:PEAK:LEFT (No Query Form)

Moves the selected marker to the next peak to the left on the spectrum trace.

Conditions Measurement views: Spurious

Group Calculate commands

Syntax `CALCulate:SPURious:MARKer<x>:PEAK:LEFT`

Arguments None

Examples `CALCULATE:SPURIOUS:MARKER1:PEAK:LEFT` moves Marker 1 (M1) to the next peak to the left on the trace.

CALCulate:SPURious:MARKer<x>:PEAK:LOWer (No Query Form)

Moves the selected marker to the next peak lower in amplitude on the spectrum trace.

Conditions Measurement views: Spurious

Group Calculate commands

Syntax `CALCulate:SPURious:MARKer<x>:PEAK:LOWer`

Arguments None

Examples `CALCULATE:SPURIOUS:MARKER1:PEAK:LOWER` moves Marker 1 (M1) to the next peak lower in amplitude on the trace.

CALCulate:SPURious:MARKer<x>:PEAK:RIGHt (No Query Form)

Moves the selected marker to the next peak to the right on the spectrum trace.

Conditions Measurement views: Spurious

Group Calculate commands

Syntax `CALCulate:SPURious:MARKer<x>:PEAK:RIGHt`

Arguments None

Examples `CALCULATE:SPURIOUS:MARKER1:PEAK:RIGHT` moves Marker 1 (M1) to the next peak to the right on the trace.

CALCulate:SPURious:MARKer<x>[:SET]:CENTer (No Query Form)

Sets the center frequency to the marker frequency in the Spurious measurement.

Conditions Measurement views: Spurious

Group Calculate commands

Syntax `CALCulate:SPURious:MARKer<x>[:SET]:CENTer`

Arguments None

Examples `CALCULATE:SPURIOUS:MARKER1:SET:CENTER` sets the center frequency to the value at Marker 1.

CALCulate:SPURious:MARKer<x>:X

Sets or queries the horizontal position of the selected marker on the spectrum trace.

Conditions Measurement views: Spurious

Group Calculate commands

Syntax
```
CALCulate:SPURious:MARKer<x>:X <value>
CALCulate:SPURious:MARKer<x>:X?
```

Arguments `<value>::=<NRf>` specifies the horizontal position of the marker.
Range: (Start frequency) to (Stop frequency).

The start and stop frequencies are set using the DISPlay:SPURious:X[:SCALe]:STARt and DISPlay:SPURious:X[:SCALe]:STOP commands.

Examples `CALCULATE:SPURIOUS:MARKER1:X 800MHz` places Marker 1 (M1) at 800 MHz on the spectrum trace.

CALCulate:SPURious:MARKer<x>:Y? (Query Only)

Queries the vertical position of the selected marker on the spectrum trace.

Conditions Measurement views: Spurious

Group Calculate commands

Syntax `CALCulate:SPURious:MARKer<x>:Y?`

Arguments None

Examples `CALCULATE:SPURIOUS:MARKER1:Y?` might return `-34.28`, indicating Marker 1 (M1) is at -34.28 dBm.

CALCulate:TOVerview:MARKer<x>:DELTa:X? (Query Only)

Returns the delta marker time for the selected marker on the time overview trace.

Conditions Measurement views: Time overview

Group Calculate commands

Syntax `CALCulate:TOVerview:MARKer<x>:DELTa:X?`

Related Commands CALCulate:TOVerview:MARKer<x>:DELTa:Y?

Arguments None

Returns `<NRf>` Delta marker time for the selected marker.

Examples `CALCULATE:TOVERVIEW:MARKER1:DELTA:X?` might return `38.0E-9`, indicating that the delta marker time is 38.0 ns.

CALCulate:TOVerview:MARKer<x>:DELTa:Y? (Query Only)

Returns the delta marker amplitude for the selected marker on the time overview trace.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

Conditions Measurement views: Time overview

Group Calculate commands

Syntax `CALCulate:TOVerview:MARKer<x>:DELTa:Y?`

Related Commands CALCulate:TOVerview:MARKer<x>:DELTa:X?

Arguments None

Returns `<NRf>` Delta marker amplitude for the selected marker.

Examples `CALCULATE:TOVERVIEW:MARKER1:DELTA:Y?` might return `23.45`, indicating that the delta marker amplitude is 23.45 dB.

CALCulate:TOVerview:MARKer<x>:MAXimum (No Query Form)

Moves the selected marker to the highest peak on the time overview trace.

Conditions Measurement views: Time overview

Group Calculate commands

Syntax `CALCulate:TOVerview:MARKer<x>:MAXimum`

Arguments None

Examples `CALCULATE:TOVERVIEW:MARKER1:MAXIMUM` moves Marker 1 (M1) to the highest peak on the trace.

CALCulate:TOVerview:MARKer<x>:PEAK:HIGHer (No Query Form)

Moves the selected marker to the next peak higher in amplitude on the time overview trace.

Conditions Measurement views: Time overview

Group Calculate commands

Syntax `CALCulate:TOVerview:MARKer<x>:PEAK:HIGHer`

Related Commands CALCulate:TOVerview:MARKer<x>:PEAK:LOWer

Arguments None

Examples `CALCULATE:TOVERVIEW:MARKER1:PEAK:HIGHER` moves Marker 1 (M1) to the next peak higher in amplitude on the trace.

CALCulate:TOVerview:MARKer<x>:PEAK:LEFT (No Query Form)

Moves the selected marker to the next peak to the left on the time overview trace.

Conditions Measurement views: Time overview

Group Calculate commands

Syntax `CALCulate:TOVerview:MARKer<x>:PEAK:LEFT`

Related Commands CALCulate:TOVerview:MARKer<x>:PEAK:RIGHt

Arguments None

Examples `CALCULATE:TOVERVIEW:MARKER1:PEAK:LEFT` moves Marker 1 (M1) to the next peak to the left on the trace.

CALCulate:TOVerview:MARKer<x>:PEAK:LOWer (No Query Form)

Moves the selected marker to the next peak lower in amplitude on the time overview trace.

Conditions Measurement views: Time overview

Group Calculate commands

Syntax `CALCulate:TOVerview:MARKer<x>:PEAK:LOWer`

Related Commands CALCulate:TOVerview:MARKer<x>:PEAK:HIGHer

Arguments None

Examples `CALCULATE:TOVERVIEW:MARKER1:PEAK:LOWER` moves Marker 1 (M1) to the next peak lower in amplitude on the trace.

CALCulate:TOVerview:MARKer<x>:PEAK:RIGHt (No Query Form)

Moves the selected marker to the next peak to the right on the time overview trace.

Conditions Measurement views: Time overview

Group Calculate commands

Syntax
```
CALCulate:TOVerview:MARKer<x>:PEAK:RIGHt
```

Related Commands CALCulate:TOVerview:MARKer<x>:PEAK:LEFT

Arguments None

Examples `CALCULATE:TOVERVIEW:MARKER1:PEAK:RIGHT` moves Marker 1 (M1) to the next peak to the right on the trace.

CALCulate:TOVerview:MARKer<x>:X

Sets or queries the horizontal position of the selected marker on the time overview trace.

Conditions Measurement views: Time overview

Group Calculate commands

Syntax
```
CALCulate:TOVerview:MARKer<x>:X <value>
CALCulate:TOVerview:MARKer<x>:X?
```

Related Commands CALCulate:TOVerview:MARKer<x>:Y?

Arguments `<value>::=<NRf>` specifies the horizontal position of the marker.
Range: (center frequency) ± (span)/2.

Examples `CALCULATE:TOVERVIEW:MARKER1:X 800MHz` places Marker 1 (M1) at 800 MHz on the trace.

CALCulate:TOVerview:MARKer<x>:Y? (Query Only)

Queries the vertical position of the selected marker in the time overview.

Conditions Measurement views: Time overview

Group Calculate commands

Syntax `CALCulate:TOVerview:MARKer<x>:Y?`

Related Commands CALCulate:TOVerview:MARKer<x>:X

Arguments None

Examples `CALCULATE:TOVERVIEW:MARKER1:Y?` might return `-34.28`, indicating Marker 1 (M1) is at -34.28 dBm.

CALibration:ABORt (No Query Form)

Aborts any actions related to the alignments in progress.

Conditions Measurement views: All

Group Calibration commands

Syntax `CALibration:ABORt`

Arguments None

Examples `CALIBRATION:ABORT` aborts any actions related to the alignments in progress.

CALibration:AUTO

Selects or queries whether or not to run alignments automatically.

Conditions Measurement views: All

Group Calibration commands

Syntax
```
CALibration:AUTO { OFF | ON | 0 | 1 }
CALibration:AUTO?
```

Arguments `OFF` or `0` runs alignments on user request.
Use the *CAL command to perform alignments.

`ON` or `1` runs alignments as needed without user intervention.
You have to restart measurement if interrupted.

Examples `CALIBRATION:AUTO ON` runs alignments automatically as needed.

CALibration:CORRection:EXTernal:EDIT<x>:LABel

Sets or queries the name of the external loss table.

The parameter <x> = 1 to 3 represent the External Loss Table 1 to 3, respectively.

Conditions Measurement views: All

Group Calibration commands

Syntax
```
CALibration:CORRection:EXTernal:EDIT<x>:LABel <name>
CALibration:CORRection:EXTernal:EDIT<x>:LABel?
```

Arguments `<name>::=<string>` specifies the name of the external loss table.

Examples `CALIBRATION:CORRECTION:EXTERNAL:EDIT1:LABEL "Sample Table 1"`
names the External Loss Table 1 "Sample Table 1".

CALibration:CORRection:EXTernal:EDIT<x>:STATe

Determines whether to enable or disable the external loss table.

The parameter <x> = 1 to 3 represent the External Loss Table 1 to 3, respectively.

Conditions Measurement views: All

Group Calibration commands

Syntax

```
CALibration:CORRection:EXTernal:EDIT<x>:STATe { OFF | ON |
0 | 1 }
CALibration:CORRection:EXTernal:EDIT<x>:STATe?
```

Arguments `OFF` or `0` disables the external loss table.

`ON` or `1` enables the external loss table.

NOTE. *You can enable one or more tables at the same time.*

Examples `CALIBRATION:CORRECTION:EXTERNAL:EDIT3:STATE ON` enables the External Loss Table 3.

CALibration:CORRection:EXTernal:GAIN[:MAGNitude]

Sets or queries the external gain value. It can be enabled or disabled using the CALibration:CORRection:EXTernal:GAIN:STATe command.

Conditions Measurement views: All

Group Calibration commands

Syntax

```
CALibration:CORRection:EXTernal:GAIN[:MAGNitude] <value>
CALibration:CORRection:EXTernal:GAIN[:MAGNitude]?
```

Arguments `<value>::=<NRf>` specifies the external gain value. Range: -50 to +30 dB.

Examples `CALIBRATION:CORRECTION:EXTERNAL:GAIN:MAGNITUDE -10` sets the external gain to -10 dB.

CALibration:CORRection:EXTernal:GAIN:STATe

Determines whether to enable or disable the external gain value.

Conditions Measurement views: All

Group Calibration commands

Syntax
```
CALibration:CORRection:EXTernal:GAIN:STATe { OFF | ON | 0 |
1 }
CALibration:CORRection:EXTernal:GAIN:STATe?
```

Related Commands CALibration:CORRection:EXTernal:GAIN[:MAGNitude]

Arguments `OFF` or `0` disables the external gain value.

`ON` or `1` enables the external gain value.

Examples `CALIBRATION:CORRECTION:EXTERNAL:GAIN:STATE ON` enables the external gain value.

CALibration:CORRection:EXTernal:PROBe:CONNect? (Query Only)

Queries whether the external probe is connected to the analyzer or not.

Conditions Measurement views: All

Group Calibration commands

Syntax
```
CALibration:CORRection:EXTernal:PROBe:CONNect?
```

Arguments None

Returns `{ 0 | 1 }`

`0` indicates that the external probe is not connected to the analyzer.

`1` indicates that the external probe is connected to the analyzer.

Examples `CALIBRATION:CORRECTION:EXTERNAL:PROBE:CONNECT?` might return `ON`, indicating that the external probe is connected to the analyzer.

CALibration:CORRection:EXTernal:PROBe[:MAGNitude]? (Query Only)

Queries the external probe attenuation.

Conditions Measurement views: All

Group Calibration commands

Syntax `CALibration:CORRection:EXTernal:PROBe[:MAGNitude]?`

Arguments None

Returns `<attenuation>::=<NRf>` The probe attenuation value in dB.

Examples `CALIBRATION:CORRECTION:EXTERNAL:PROBE:MAGNITUDE?` might return `10`, indicating that the probe attenuation is 10 dB.

CALibration:CORRection:EXTernal:PROBe:STATe

Determines whether or not to correct data for the external probe attenuation.

Conditions Measurement views: All

Group Calibration commands

Syntax

```
CALibration:CORRection:EXTernal:PROBe:STATe { OFF | ON | 0 |
1 }
CALibration:CORRection:EXTernal:PROBe:STATe?
```

Arguments `OFF` or `0` does not correct data for the external probe attenuation.

`ON` or `1` corrects data for the external probe attenuation.

Examples `CALIBRATION:CORRECTION:EXTERNAL:PROBE:STATE ON` corrects data for the external probe attenuation.

*CLS (No Query Form)

Clears the analyzer status data structures. Refer to Section 3, *Status and Events*, for the register information.

The *CLS command clears the following

- the Event Queue
- the Standard Event Status Register (SESR)
- the Status Byte Register (except the MAV bit; see below)

If the *CLS command immediately follows an <EOI>, the Output Queue and MAV bit (Status Byte Register bit 4) are also cleared. MAV indicates information is in the output queue. The device clear (DCL) GPIB control message will clear the output queue and thus MAV. *CLS does not clear the output queue or MAV. (A complete discussion of these registers and bits, and of event handling in general is described in the *Status and Events* section)

*CLS can suppress a Service Request that is to be generated by an *OPC. This will happen if a hardcopy output or single sequence acquisition operation is still being processed when the *CLS command is executed.

Conditions Measurement views: All

Group IEEE common commands

Syntax `*CLS`

Related Commands *ESE, *ESR?, *SRE, *STB?

Arguments None

Examples `*CLS` clears the analyzer status data structures.

DISPlay:ACPower:MARKer:SHOW:STATe

Determines whether to show or hide the readout for the selected marker in the Channel power and ACPR view.

Conditions Measurement views: Channel power and ACPR

Group Display commands

Syntax

```
DISPlay:ACPower:MARKer:SHOW:STATe { OFF | ON | 0 | 1 }
DISPlay:ACPower:MARKer:SHOW:STATe?
```

Arguments OFF or 0 hides the readout for the selected marker in the graph.

ON or 1 shows the readout for the selected marker in the graph.

Examples DISPLAY:ACPOWER:MARKER:SHOW:STATE ON shows the readout for the selected marker in the graph.

DISPlay:ACPower:PLEVel:SHOW:STATe

Determines whether to show or hide the power levels in the Channel power and ACPR view.

Conditions Measurement views: Channel power and ACPR

Group Display commands

Syntax

```
DISPlay:ACPower:PLEVel:SHOW:STATe { OFF | ON | 0 | 1 }
DISPlay:ACPower:PLEVel:SHOW:STATe?
```

Arguments OFF or 0 hides the power levels in the graph.

ON or 1 shows the power levels in the graph.

Examples DISPLAY:ACPOWER:PLEVEL:SHOW:STATE ON shows the power levels in the graph.

DISPlay:ACPower:RESet:SCALe (No Query Form)

Resets the horizontal and vertical scale to the default values described below in the Channel power and ACPR view.

Vertical offset = Reference level,
Vertical scale = 100 dB,
Horizontal offset = Center frequency, and
Horizontal scale = Default span

Conditions Measurement views: Channel power and ACPR

Group Display commands

Syntax `DISPlay:ACPower:RESet:SCALe`

Arguments None

Examples `DISPLAY:ACPOWER:RESET:SCALE` resets the horizontal and vertical scale to the default values.

DISPlay:ACPower:WINDow:TRACe:GRATicule:GRID:STATe

Determines whether to show or hide the graticule grid on the screen.

Conditions Measurement views: Channel power and ACPR

Group Display commands

Syntax

```
DISPlay:ACPower:WINDow:TRACe:GRATicule:GRID:STATe { OFF |
ON | 0 | 1 }
DISPlay:ACPower:WINDow:TRACe:GRATicule:GRID:STATe?
```

Arguments `OFF` or `0` hides the graticule grid.

`ON` or `1` shows the graticule grid.

Examples `DISPLAY:ACPOWER:WINDOW:TRACE:GRATICULE:GRID:STATE ON` shows the graticule grid on the screen.

DISPlay:ACPower:X[:SCALe]

Sets or queries the horizontal range of the Channel power and ACPR graph.

Conditions Measurement views: Channel power and ACPR

Group Display commands

Syntax

```
DISPlay:ACPower:X[:SCALe] <value>
DISPlay:ACPower:X[:SCALe]?
```

Related Commands DISPlay:ACPower:X[:SCALe]:OFFSet

Arguments `<value>::=<NRf>` specifies the horizontal range.
Range: 10 Hz to 6.2 GHz (RSA6106A) / 14 GHz (RSA6114A).

Examples `DISPLAY:ACPOWER:X:SCALE 10MHz` sets the horizontal range to 10 MHz.

DISPlay:ACPower:X[:SCALe]:AUTO (No Query Form)

Rescales the horizontal axis automatically to fit the waveform to the screen in the Channel power and ACPR view.

Conditions Measurement views: Channel power and ACPR

Group Display commands

Syntax

```
DISPlay:ACPower:X[:SCALe]:AUTO
```

Arguments None

Examples `DISPLAY:ACPOWER:X:SCALE:AUTO` rescales the horizontal scale automatically to fit the waveform to the screen.

DISPlay:ACPower:X[:SCALe]:OFFSet

Sets or queries the minimum horizontal value (left edge) of the Channel power and ACPR graph.

Conditions Measurement views: Channel power and ACPR

Group Display commands

Syntax

```
DISPlay:ACPower:X[:SCALe]:OFFSet <value>
DISPlay:ACPower:X[:SCALe]:OFFSet?
```

Related Commands DISPlay:ACPower:X[:SCALe]

Arguments `<value>::=<NRf>` specifies the minimum horizontal value.
Range: [(center frequency) - (X scale) × 0.9] to [(center frequency) + (X scale) × 0.9]

Examples `DISPlay:ACPower:X:SCALe:OFFSet 1.45GHz` sets the minimum horizontal value to 1.45 GHz in the Channel power and ACPR graph.

DISPlay:ACPower:Y[:SCALe]

Sets or queries the vertical range of the channel power and ACPR graph.

Conditions Measurement views: Channel power and ACPR

Group Display commands

Syntax

```
DISPlay:ACPower:Y[:SCALe] <value>
DISPlay:ACPower:Y[:SCALe]?
```

Related Commands DISPlay:ACPower:Y[:SCALe]:OFFSet

Arguments `<value>::=<NRf>` specifies the vertical range. Range: 0.1 to 200 dB.

Examples `DISPLAY:ACPOWER:Y:SCALE 100` sets the vertical range to 100 dB in the Channel power and ACPR graph.

DISPlay:ACPower:Y[:SCALe]:AUTO (No Query Form)

Rescales the vertical axis automatically to fit the waveform to the screen in the Channel power and ACPR view.

Conditions Measurement views: Channel power and ACPR

Group Display commands

Syntax

```
DISPlay:ACPower:Y[:SCALe]:AUTO
```

Arguments None

Examples `DISPLAY:ACPOWER:Y:SCALE:AUTO` rescales the vertical scale automatically to fit the waveform to the screen.

DISPlay:ACPower:Y[:SCALe]:OFFSet

Sets or queries the vertical offset (the value at the top edge of the vertical axis) in the Channel power and ACPR graph.

Conditions Measurement views: Channel power and ACPR

Group Display commands

Syntax
```
DISPlay:ACPower:Y[:SCALe]:OFFSet <value>
DISPlay:ACPower:Y[:SCALe]:OFFSet?
```

Related Commands DISPlay:ACPower:Y[:SCALe]

Arguments `<value>::=<NRf>` specifies the vertical offset. Range: -170 to +50 dBm.

Examples `DISPLAY:ACPOWER:Y:SCALE:OFFSET -12.5` sets the vertical offset to -12.5 dBm in the Channel power and ACPR graph.

DISPlay:AVTime:MARKer:SHOW:STATe

Determines whether to show or hide the readout for the selected marker on the screen in the Amplitude versus Time measurement.

Conditions Measurement views: Amplitude versus Time

Group Display commands

Syntax
```
DISPlay:AVTime:MARKer:SHOW:STATe { OFF | ON | 0 | 1 }
DISPlay:AVTime:MARKer:SHOW:STATe?
```

Arguments

`OFF` or `0` hides the readout for the selected marker on screen.

`ON` or `1` shows the readout for the selected marker on screen.

Examples

`DISPLAY:AVTIME:MARKER:SHOW:STATE ON` shows the readout for the selected marker on screen.

DISPlay:AVTime:RESet (No Query Form)

Resets the horizontal and vertical scale to the default values described below in the Amplitude versus Time view.

Vertical offset = Reference level,
Vertical scale = 100 dB,
Horizontal offset = Analysis offset, and
Horizontal scale = Analysis length

Conditions

Measurement views: Amplitude versus Time

Group

Display commands

Syntax

```
DISPlay:AVTime:RESet
```

Arguments

None

Examples

`DISPLAY:AVTIME:RESET` resets the horizontal and vertical scale to the default values.

DISPlay:AVTime:TRIGger:LEVel:STATe

Determines whether to show or hide the power trigger level line on the screen in the Amplitude versus Time measurement.

Conditions

Measurement views: Amplitude versus Time

Group

Display commands

Syntax

```
DISPlay:AVTime:TRIGger:LEVel:STATe { OFF | ON | 0 | 1 }
DISPlay:AVTime:TRIGger:LEVel:STATe?
```

Arguments

OFF or 0 hides the power trigger level line.

ON or 1 shows the power trigger level line.

Examples

DISPLAY:AVTIME:TRIGGER:LEVEL:STATE ON shows the power trigger level line on the screen.

DISPlay:AVTime:WINDow:TRACe:GRATicule:GRID:STATe

Determines whether to show or hide the graticule grid on the screen.

Conditions

Measurement views: Amplitude versus Time

Group

Display commands

Syntax

```
DISPlay:AVTime:WINDow:TRACe:GRATicule:GRID:STATe { OFF | ON
| 0 | 1 }
DISPlay:AVTime:WINDow:TRACe:GRATicule:GRID:STATe?
```

Arguments

OFF or 0 hides the graticule grid.

ON or 1 shows the graticule grid.

Examples

DISPLAY:AVTIME:WINDOW:TRACE:GRATICULE:GRID:STATE ON shows the graticule grid on the screen in the Amplitude versus Time view.

DISPlay:AVTime:X:RSCale (No Query Form)

Rescales the horizontal axis automatically to fit the waveform to the screen in the Amplitude versus Time display.

Conditions

Measurement views: Amplitude versus Time

Group

Display commands

Syntax

```
DISPlay:AVTime:X:RSCale
```

Arguments

None

Examples `DISPLAY:AVTIME:X:RSCALE` rescales the horizontal scale automatically to fit the waveform to the screen.

DISPlay:AVTime:X[:SCALe]:AUTO (No Query Form)

Sets the horizontal scale automatically to fit the waveform to the screen in the Amplitude versus Time view. Executing this command sets DISPlay:AVTime:X[:SCALe]:AUTO:STATe ON.

Conditions Measurement views: Amplitude versus Time

Group Display commands

Syntax `DISPlay:AVTime:X[:SCALe]:AUTO`

Related Commands DISPlay:AVTime:X[:SCALe]:AUTO:STATe

Arguments None

Examples `DISPlay:AVTime:X:SCALe:AUTO` sets the horizontal scale automatically to fit the waveform to the screen.

DISPlay:AVTime:X[:SCALe]:AUTO:STATe

Determines whether to set the horizontal scale automatically or manually.

Conditions Measurement views: Amplitude versus Time

Group Display commands

Syntax

```
DISPlay:AVTime:X[:SCALe]:AUTO:STATe { OFF | ON | 0 | 1 }
DISPlay:AVTime:X[:SCALe]:AUTO:STATe?
```

Arguments `OFF` or `0` specifies that the horizontal scale is set manually. To set it, use the DISPlay:AVTime:X[:SCALe]:FULL and DISPlay:AVTime:X[:SCALe]:OFFSet commands.

`ON` or `1` specifies that the horizontal scale is set automatically.

Examples `DISPLAY:AVTIME:X:SCALE:AUTO:STATE ON` specifies that the horizontal scale is set automatically.

DISPlay:AVTime:X[:SCALe]:FULL

Sets or queries the horizontal scale (full-scale time) of the Amplitude versus Time graph. Programming a specified scale sets DISPlay:AVTime:X[:SCALe] AUTO:STATe OFF.

Conditions Measurement views: Amplitude versus Time

Group Display commands

Syntax

```
DISPlay:AVTime:X[:SCALe]:FULL <value>
DISPlay:AVTime:X[:SCALe]:FULL?
```

Related Commands DISPlay:AVTime:X[:SCALe]:AUTO:STATe, DISPlay:AVTime:X[:SCALe]:OFFSet

Arguments `<value>::=<NRf>` specifies the horizontal scale in full-scale time. Use the DISPlay:AVTime:X[:SCALe]:MAXimum? and DISPlay:AVTime:X[:SCALe]:MINimum? queries to get the upper and lower limits of the setting range.

Examples `DISPLAY:AVTIME:X:SCALE:FULL 25.6us` sets the horizontal scale to 25.6 µs.

DISPlay:AVTime:X[:SCALe]:MAXimum? (Query Only)

Queries the upper limit of the horizontal scale setting range.

Conditions Measurement views: Amplitude versus Time

Group Display commands

Syntax

```
DISPlay:AVTime:X[:SCALe]:MAXimum?
```

Related Commands DISPlay:AVTime:X[:SCALe]:FULL

Arguments None

Returns `<NRf>` The upper limit of the horizontal scale setting range.

Examples `DISPLAY:AVTIME:X:SCALE:MAXIMUM?` might return `18.135E-3`, indicating that the upper limit of the horizontal scale setting range is 18.135 ms.

DISPlay:AVTime:X[:SCALe]:MINimum? (Query Only)

Queries the lower limit of the horizontal scale setting range.

Conditions Measurement views: Amplitude versus Time

Group Display commands

Syntax `DISPlay:AVTime:X[:SCALe]:MINimum?`

Related Commands DISPlay:AVTime:X[:SCALe]:FULL

Arguments None

Returns `<NRf>` The lower limit of the horizontal scale setting range.

Examples `DISPLAY:AVTIME:X:SCALE:MINIMUM?` might return `10.0E-9`, indicating that the lower limit of the horizontal scale setting range is 10.0 ns.

DISPlay:AVTime:X[:SCALe]:OFFSet

Sets or queries the minimum horizontal value (left edge) of the Amplitude versus Time graph. Programming a specified offset sets DISPlay:AVTime:X[:SCALe] AUTO:STATe OFF.

Conditions Measurement views: Amplitude versus Time

Group Display commands

Syntax
```
DISPlay:AVTime:X[:SCALe]:OFFSet <value>
DISPlay:AVTime:X[:SCALe]:OFFSet?
```

Related Commands DISPlay:AVTime:X[:SCALe]:AUTO:STATe, DISPlay:AVTime:X[:SCALe]:FULL

Arguments `<value>::=<NRf>` specifies the minimum horizontal value. Use the DISPlay:AVTime:X[:SCALe]:OFFSet:MAXimum? and DISPlay:AVTime:X[:SCALe]:OFFSet:MINimum? queries to get the upper and lower limits of the setting range.

Examples `DISPLAY:AVTIME:X:SCALE:OFFSET 800ns` sets the minimum horizontal value to 800 ns in the Amplitude versus Time graph.

DISPlay:AVTime:X[:SCALe]:OFFSet:MAXimum? (Query Only)

Queries the upper limit of the horizontal offset setting range.

Conditions Measurement views: Amplitude versus Time

Group Display commands

Syntax `DISPlay:AVTime:X[:SCALe]:OFFSet:MAXimum?`

Related Commands DISPlay:AVTime:X[:SCALe]:OFFSet

Arguments None

Returns `<NRf>` The upper limit of the horizontal offset setting range.

Examples `DISPLAY:AVTIME:X:SCALE:OFFSET:MAXIMUM?` might return `-1.812E-3`, indicating that the upper limit of the horizontal offset setting range is -1.812 ms.

DISPlay:AVTime:X[:SCALe]:OFFSet:MINimum? (Query Only)

Queries the lower limit of the horizontal offset setting range.

Conditions Measurement views: Amplitude versus Time

Group Display commands

Syntax `DISPlay:AVTime:X[:SCALe]:OFFSet:MINimum?`

Related Commands DISPlay:AVTime:X[:SCALe]:OFFSet

Arguments None

Returns `<NRf>` The lower limit of the horizontal offset setting range.

Examples `DISPLAY:AVTIME:X:SCALE:OFFSET:MINIMUM?` might return `-16.28E-3`, indicating that the lower limit of the horizontal offset setting range is -16.28 ms.

DISPlay:AVTime:Y:RSCale (No Query Form)

Rescales the vertical axis automatically to fit the waveform to the screen in the Amplitude versus Time display.

Conditions Measurement views: Amplitude versus Time

Group Display commands

Syntax `DISPlay:AVTime:Y:RSCale`

Arguments None

Examples `DISPLAY:AVTIME:Y:RSCALE` rescales the vertical scale automatically to fit the waveform to the screen.

DISPlay:AVTime:Y[:SCALe]:FULL

Sets or queries the vertical range of the Amplitude versus Time graph.

Conditions Measurement views: Amplitude versus Time

Group Display commands

Syntax

```
DISPlay:AVTime:Y[:SCALe]:FULL <value>
DISPlay:AVTime:Y[:SCALe]:FULL?
```

Related Commands DISPlay:AVTime:Y[:SCALe]:OFFSet

Arguments `<value>::=<NRf>` specifies the vertical range. Range: 0.1 to 200 dB.

Examples `DISPLAY:AVTIME:Y:SCALE:FULL 100` sets the vertical range to 100 dB in the Amplitude versus Time graph.

DISPlay:AVTime:Y[:SCALe]:OFFSet

Sets or queries the vertical offset (the value at the top edge of the vertical axis) in the Amplitude versus Time graph.

Conditions Measurement views: Amplitude versus Time

Group Display commands

Syntax

```
DISPlay:AVTime:Y[:SCALe]:OFFSet <value>
DISPlay:AVTime:Y[:SCALe]:OFFSet?
```

Related Commands DISPlay:AVTime:Y[:SCALe]:FULL

Arguments `<value>::=<NRf>` specifies the vertical offset. Range: -170 to +50 dBm.

Examples `DISPLAY:AVTIME:Y:SCALE:OFFSET -12.5` sets the vertical offset to -12.5 dBm in the Amplitude versus Time graph.

DISPlay:CONSte:WINDow:TRACe:GRATicule:GRID:STATe

Determines whether to show or hide the graticule grid on the screen.

Conditions Measurement views: Constellation

Group Display commands

Syntax

```
DISPlay:CONSte:WINDow:TRACe:GRATicule:GRID:STATe { OFF | ON
| 0 | 1 }
DISPlay:CONSte:WINDow:TRACe:GRATicule:GRID:STATe?
```

Arguments `OFF` or `0` hides the graticule grid.

`ON` or `1` shows the graticule grid.

Examples `DISPLAY:CONSTE:WINDOW:TRACE:GRATICULE:GRID:STATE ON` shows the graticule grid on the screen.

DISPlay:DDEMod:MEASview:DELete (No Query Form)

Deletes the measurement view in the general purpose digital modulation measurements.

Conditions Measurement views: General purpose digital modulation

Group Display commands

Syntax

```
DISPlay:DDEMod:MEASview:DELete { CONSte | EVM | MERRor |
PERRor | SIGNalqual | STABle }
```

Arguments The following table lists the arguments. The arguments are the string type.

Table 2-28: Modulation measurement views

Argument	View
CONSte	Constellation
EVM	EVM (Error Vector Magnitude) versus Time
MERRor	Magnitude error versus Time
PERRor	Phase error versus Time
SIGNalqual	Signal quality
STABle	Symbol table

If you attempt to delete a view that is not displayed on screen, the error (-200, "Execution error; Measurement not running" will be returned.

Examples `DISPLAY:DDEMOD:MEASVIEW:DELETE CONSte` deletes the constellation view.

DISPlay:DDEMod:MEASview:NEW (No Query Form)

Displays a new measurement view in the general purpose digital modulation measurements.

Conditions Measurement views: General purpose digital modulation

Group Display commands

Syntax

```
DISPlay:DDEMod:MEASview:NEW { CONSte | EVM | MERRor | PERRor
| SIGNalqual | STABle }
```

Arguments (See Table 2-28 on page 2-188.) If you attempt to open a view that is currently displayed on screen, the error (-200, "Execution error; Measurement is already running") will be returned.

Examples `DISPLAY:DDEMOD:MEASVIEW:NEW CONSte` creates the constellation view.

DISPlay:DDEMod:MEASview:SELect

Selects a measurement view in the general purpose digital modulation measurements on the screen. The query command returns the currently selected view.

Selecting a measurement optimizes it. Other measurements may be optimized as a side effect. Refer to the DISPlay:WINDow:OPTimized:MEASurement? query.

Conditions Measurement views: General purpose digital modulation

Group Display commands

Syntax

```
DISPlay:DDEMod:MEASview:SELect { CONSte | EVM | MERRor |
PERRor | SIGNalqual | STABle }
DISPlay:DDEMod:MEASview:SELect?
```

Arguments (See Table 2-28 on page 2-188.) If you attempt to select a view that is not displayed on screen, the error (-200, "Execution error; Measurement not running") will be returned.

Examples `DISPLAY:DDEMOD:MEASVIEW:SELECT CONSte` selects the constellation view.

DISPlay:DDEMod:RADix

Selects or queries the base of symbols. This command is effective in the symbol table.

Conditions Measurement views: Symbol table

Group Display commands

Syntax

```
DISPlay:DDEMod:RADix { BINary | HEXadecimal }
DISPlay:DDEMod:RADix?
```

Arguments `BINary` selects binary notation.

`HEXadecimal` selects hexadecimal notation.

Examples `DISPLAY:DDEMOD:RADIX BINary` selects binary notation for the symbol table.

DISPlay:DDEMod:X[:SCALe]

Sets or queries the horizontal scale (full-scale time) for the time measurements in the general purpose digital modulation analysis. Programming a specified scale sets DISPlay:DDEMod:X[:SCALe]:AUTO:STATe OFF.

Conditions Measurement views: EVM versus Time, Magnitude error versus Time, Phase error versus Time

Group Display commands

Syntax

```
DISPlay:DDEMod:X[:SCALe] <value>
DISPlay:DDEMod:X[:SCALe]?
```

Related Commands DISPlay:DDEMod:X[:SCALe]:AUTO:STATe, DISPlay:DDEMod:X[:SCALe]:OFFSet

Arguments `<value>::=<NRf>` specifies the horizontal scale in full-scale time. Use the DISPlay:DDEMod:X[:SCALe]:MAXimum? and DISPlay:DDEMod:X[:SCALe]:MINimum? queries to get the upper and lower limits of the setting range. The unit can be changed by the [SENSe]:DDEMod:TIMe:UNITs command.

Examples `DISPLAY:DDEMOD:X:SCALE 1.5us` sets the horizontal scale to 1.5 µs.

DISPlay:DDEMod:X[:SCALe]:AUTO (No Query Form)

Sets the horizontal scale automatically to fit the waveform to the screen in the general purpose digital modulation analysis. Executing this command sets DISPlay:DDEMod:X[:SCALe]:AUTO:STATe ON.

Conditions Measurement views: EVM versus Time, Magnitude error versus Time, Phase error versus Time

Group Display commands

Syntax `DISPlay:DDEMod:X[:SCALe]:AUTO`

Related Commands DISPlay:DDEMod:X[:SCALe]:AUTO:STATe

Arguments None

Examples `DISPLAY:DDEMOD:X:SCALE:AUTO` sets the horizontal scale automatically to fit the waveform to the screen.

DISPlay:DDEMod:X[:SCALe]:AUTO:STATe

Determines whether to set the horizontal scale automatically or manually.

Conditions Measurement views: EVM versus Time, Magnitude error versus Time, Phase error versus Time

Group Display commands

Syntax

```
DISPlay:DDEMod:X[:SCALe]:AUTO:STATe { OFF | ON | 0 | 1 }
DISPlay:DDEMod:X[:SCALe]:AUTO:STATe?
```

Arguments `OFF` or `0` specifies that the horizontal scale is set manually. To set it, use the DISPlay:DDEMod:X[:SCALe] and DISPlay:DDEMod:X[:SCALe]:OFFSet commands.

`ON` or `1` specifies that the horizontal scale is set automatically.

Examples `DISPLAY:DDEMOD:X:SCALE:AUTO:STATE ON` specifies that the horizontal scale is set automatically.

DISPlay:DDEMod:X[:SCALe]:MAXimum? (Query Only)

Queries the upper limit of the horizontal scale setting range.

Conditions Measurement views: EVM versus Time, Magnitude error versus Time, Phase error versus Time

Group Display commands

Syntax `DISPlay:DDEMod:X[:SCALe]:MAXimum?`

Arguments None

Returns `<NRf>` The upper limit of the horizontal scale setting range.
The unit can be changed by the [SENSe]:DDEMod:TIMe:UNITs command.

Examples `DISPLAY:DDEMOD:X:SCALE:MAXIMUM?` might return `18.135E-3`, indicating that the upper limit of the horizontal scale setting range is 18.135 ms.

DISPlay:DDEMod:X[:SCALe]:MINimum? (Query Only)

Queries the lower limit of the horizontal scale setting range.

Conditions Measurement views: EVM versus Time, Magnitude error versus Time, Phase error versus Time

Group Display commands

Syntax `DISPlay:DDEMod:X[:SCALe]:MINimum?`

Arguments None

Returns `<NRf>` The lower limit of the horizontal scale setting range.
The unit can be changed by the [SENSe]:DDEMod:TIMe:UNITs command.

Examples `DISPLAY:DDEMOD:X:SCALE:MINIMUM` might return `10.0E-9`, indicating that the lower limit of the horizontal scale setting range is 10.0 ns.

DISPlay:DDEMod:X[:SCALe]:OFFSet

Sets or queries the minimum horizontal value (left edge) for the time measurements in the general purpose digital modulation analysis. Programming a specified offset sets DISPlay:DDEMod:X[:SCALe]:AUTO:STATe OFF.

Conditions Measurement views: EVM versus Time, Magnitude error versus Time, Phase error versus Time

Group Display commands

Syntax

```
DISPlay:DDEMod:X[:SCALe]:OFFSet <value>
DISPlay:DDEMod:X[:SCALe]:OFFSet?
```

Related Commands DISPlay:DDEMod:X[:SCALe]:AUTO:STATe, DISPlay:DDEMod:X[:SCALe]

Arguments `<value>::=<NRf>` specifies the minimum horizontal value.

Use the DISPlay:DDEMod:X[:SCALe]:OFFSet:MAXimum? and DISPlay:DDEMod:X[:SCALe]:OFFSet:MINimum? queries to get the upper and lower limits of the setting range.

The unit can be changed by the [SENSe]:DDEMod:TIMe:UNITs command.

Examples `DISPLAY:DDEMOD:X:SCALE:OFFSET 20.075us` sets the minimum horizontal value to 20.075 µs.

DISPlay:DDEMod:X[:SCALe]:OFFSet:MAXimum? (Query Only)

Queries the upper limit of the horizontal offset setting range.

Conditions Measurement views: EVM versus Time, Magnitude error versus Time, Phase error versus Time

Group Display commands

Syntax `DISPlay:DDEMod:X[:SCALe]:OFFSet:MAXimum?`

Arguments None

Returns `<NRf>` The upper limit of the horizontal offset setting range. The unit can be changed by the [SENSe]:DDEMod:TIMe:UNITs command.

Examples `DISPLAY:DDEMOD:X:SCALE:OFFSET:MAXIMUM?` might return `-1.812E-3`, indicating that the upper limit of the horizontal offset setting range is -1.812 ms.

DISPlay:DDEMod:X[:SCALe]:OFFSet:MINimum? (Query Only)

Queries the lower limit of the horizontal offset setting range.

Conditions Measurement views: EVM versus Time, Magnitude error versus Time, Phase error versus Time

Group Display commands

Syntax `DISPlay:DDEMod:X[:SCALe]:OFFSet:MINimum?`

Arguments None

Returns `<NRf>` The lower limit of the horizontal offset setting range. The unit can be changed by the [SENSe]:DDEMod:TIMe:UNITs command.

Examples `DISPLAY:DDEMOD:X:SCALE:OFFSET:MINIMUM?` might return `-16.28E-3`, indicating that the lower limit of the horizontal offset setting range is -16.28 ms.

DISPlay:DDEMod:X[:SCALe]:RESet (No Query Form)

Presets the horizontal scale to the default value for the time measurements in the general purpose digital modulation analysis.

Conditions Measurement views: EVM versus Time, Magnitude error versus Time, Phase error versus Time

Group Display commands

Syntax `DISPlay:DDEMod:X[:SCALe]:RESet`

Arguments None

Examples `DISPLAY:DDEMOD:X:SCALE:RESET` presets the horizontal scale to the default value.

DISPlay:DPSA:WINDow:TRACe:GRATicule:GRID:STATe

Determines whether to show or hide the graticule grid on the screen.

Conditions Measurement views: DPX spectrum

Group Display commands

Syntax

```
DISPlay:DPSA:WINDow:TRACe:GRATicule:GRID:STATe { OFF | ON
| 0 | 1 }
DISPlay:DPSA:WINDow:TRACe:GRATicule:GRID:STATe?
```

Arguments `OFF` or `0` hides the graticule grid.

`ON` or `1` shows the graticule grid.

Examples `DISPLAY:DPSA:WINDOW:TRACE:GRATICULE:GRID:STATE ON` shows the graticule grid on the screen in the DPX spectrum view.

DISPlay:DPSA:Y[:SCALe]:PDIVision

Sets or queries the vertical scale (per division) in the DPX spectrum view.

Conditions Measurement views: DPX spectrum

Group Display commands

Syntax
```
DISPlay:DPSA:Y[:SCALe]:PDIVision <value>
DISPlay:DPSA:Y[:SCALe]:PDIVision?
```

Related Commands [SENSe]:POWer:UNITs

Arguments `<value>::=<NRf>` specifies the vertical scale (per division).
Range: 2 to 20 dB/div.

Examples `SENSE:DPSA:Y:SCALE:PDIVISION 0.5` sets the vertical scale to 0.5 dB/div.

DISPlay:EVM:Y[:SCALe]

Sets or queries the vertical range of the EVM versus Time graph.

Conditions Measurement views: EVM versus Time

Group Display commands

Syntax
```
DISPlay:EVM:Y[:SCALe] <value>
DISPlay:EVM:Y[:SCALe]?
```

Related Commands DISPlay:EVM:Y[:SCALe]:OFFSet

Arguments `<value>::=<NRf>` specifies the vertical range. Range: 1 to 100%.

Examples `DISPLAY:EVM:Y:SCALE 50` sets the vertical range to 50% in the EVM versus Time graph.

DISPlay:EVM:Y[:SCALe]:AUTO (No Query Form)

Sets the vertical scale automatically to fit the waveform to the screen in the EVM versus Time display.

Conditions Measurement views: EVM versus Time

Group Display commands

Syntax DISPlay:EVM:Y[:SCALe]:AUTO

Arguments None

Examples DISPLAY:EVM:Y:SCALE:AUTO sets the vertical scale automatically to fit the waveform to the screen.

DISPlay:EVM:Y[:SCALe]:OFFSet

Sets or queries the minimum vertical value (bottom edge) of the EVM versus Time graph.

Conditions Measurement views: EVM versus Time

Group Display commands

Syntax
```
DISPlay:EVM:Y[:SCALe]:OFFSet <value>
DISPlay:EVM:Y[:SCALe]:OFFSet?
```

Related Commands DISPlay:EVM:Y[:SCALe]

Arguments <value>::=<NRf> specifies the minimum vertical value. Range: -100 to 100%.

Examples DISPLAY:EVM:Y:SCALE:OFFSET -9.5 sets the minimum vertical value to -9.5% in the EVM versus Time graph.

DISPlay:FVTime:WINDow:TRACe:GRATicule:GRID:STATe

Determines whether to show or hide the graticule grid on the screen.

Conditions Measurement views: Frequency versus Time

Group Display commands

Syntax
```
DISPlay:FVTime:WINDow:TRACe:GRATicule:GRID:STATe { OFF | ON
| 0 | 1 }
DISPlay:FVTime:WINDow:TRACe:GRATicule:GRID:STATe?
```

Arguments

OFF or 0 hides the graticule grid.

ON or 1 shows the graticule grid.

Examples

DISPLAY:FVTIME:WINDOW:TRACE:GRATICULE:GRID:STATE ON shows the graticule grid on the Frequency versus Time view.

DISPlay:FVTime:X[:SCALe]

Sets or queries the horizontal scale (full-scale time) of the Frequency versus Time graph. Programming a specified scale sets DISPlay:FVTime:X[:SCALe] AUTO:STATe OFF.

Conditions

Measurement views: Frequency versus Time

Group

Display commands

Syntax

```
DISPlay:FVTime:X[:SCALe] <value>
DISPlay:FVTime:X[:SCALe]?
```

Related Commands

DISPlay:FVTime:X[:SCALe]:AUTO:STATe, DISPlay:FVTime:X[:SCALe]:OFFSet

Arguments

<value>::={ <NRf> | MAXimum | MINimum } specifies the horizontal scale in full-scale time. MAXimum and MINimum represent the upper and lower limits of the setting range, respectively.

Use the DISPlay:FVTime:X[:SCALe]:MAXimum? and DISPlay:FVTime:X[:SCALe]:MINimum? queries to get the upper and lower limit values of the setting range.

Examples

DISPLAY:FVTIME:X:SCALE 25.6us sets the horizontal scale to 25.6 µs.

DISPlay:FVTime:X[:SCALe]:AUTO (No Query Form)

Sets the horizontal scale automatically to fit the waveform to the screen in the Frequency versus Time view. Executing this command sets DISPlay:FVTime X[:SCALe]:AUTO:STATe ON.

Conditions

Measurement views: Frequency versus Time

Group Display commands

Syntax `DISPlay:FVTime:X[:SCALe]:AUTO`

Related Commands DISPlay:FVTime:X[:SCALe]:AUTO:STATe

Arguments None

Examples `DISPLAY:FVTIME:X:SCALE:AUTO` sets the horizontal scale automatically to fit the waveform to the screen.

DISPlay:FVTime:X[:SCALe]:AUTO:STATe

Determines whether to set the horizontal scale automatically or manually.

Conditions Measurement views: Frequency versus Time

Group Display commands

Syntax

```
DISPlay:FVTime:X[:SCALe]:AUTO:STATe { OFF | ON | 0 | 1 }
DISPlay:FVTime:X[:SCALe]:AUTO:STATe?
```

Arguments `OFF` or `0` specifies that the horizontal scale is set manually. To set it, use the DISPlay:FVTime:X[:SCALe] and DISPlay:FVTime:X[:SCALe]:OFFSet commands.

`ON` or `1` specifies that the horizontal scale is set automatically.

Examples `DISPLAY:FVTIME:X:SCALE:AUTO:STATE ON` specifies that the horizontal scale is set automatically.

DISPlay:FVTime:X[:SCALe]:MAXimum? (Query Only)

Queries the upper limit of the horizontal scale setting range.

Conditions Measurement views: Frequency versus Time

Group Display commands

Syntax DISPlay:FVTime:X[:SCALe]:MAXimum?

Related Commands DISPlay:FVTime:X[:SCALe]

Arguments None

Returns <NRf> The upper limit of the horizontal scale setting range.

Examples DISPLAY:FVTIME:X:SCALE:MAXIMUM? might return 18.135E-3, indicating that the upper limit of the horizontal scale setting range is 18.135 ms.

DISPlay:FVTime:X[:SCALe]:MINimum? (Query Only)

Queries the lower limit of the horizontal scale setting range.

Conditions Measurement views: Frequency versus Time

Group Display commands

Syntax DISPlay:FVTime:X[:SCALe]:MINimum?

Related Commands DISPlay:FVTime:X[:SCALe]

Arguments None

Returns <NRf> The lower limit of the horizontal scale setting range.

Examples DISPLAY:FVTIME:X:SCALE:MINIMUM? might return 10.0E-9, indicating that the lower limit of the horizontal scale setting range is 10.0 ns.

DISPlay:FVTime:X[:SCALe]:OFFSet

Sets or queries the minimum horizontal value (left edge) of the Frequency versus Time graph. Programming a specified offset sets DISPlay:FVTime:X[:SCALe] AUTO:STATe OFF.

Conditions Measurement views: Frequency versus Time

Group Display commands

Syntax

```
DISPlay:FVTime:X[:SCALe]:OFFSet <value>
DISPlay:FVTime:X[:SCALe]:OFFSet?
```

Related Commands DISPlay:FVTime:X[:SCALe]:AUTO:STATe, DISPlay:FVTime:X[:SCALe]

Arguments `<value>::={ <NRf> | MAXimum | MINimum }` specifies the horizontal offset. `MAXimum` and `MINimum` represent the upper and lower limits of the setting range, respectively.

Use the DISPlay:FVTime:X[:SCALe]:OFFSet:MAXimum? and DISPlay:FVTime:X[:SCALe]:OFFSet:MINimum? queries to get the upper and lower limit values of the setting range.

Examples `DISPLAY:FVTIME:X:SCALE:OFFSET 800ns` sets the minimum horizontal value to 800 ns in the Frequency versus Time graph.

DISPlay:FVTime:X[:SCALe]:OFFSet:MAXimum? (Query Only)

Queries the upper limit of the horizontal offset setting range.

Conditions Measurement views: Frequency versus Time

Group Display commands

Syntax

```
DISPlay:FVTime:X[:SCALe]:OFFSet:MAXimum?
```

Related Commands DISPlay:FVTime:X[:SCALe]:OFFSet

Arguments None

Returns `<NRf>` The upper limit of the horizontal offset setting range.

Examples `DISPLAY:FVTIME:X:SCALE:OFFSET:MAXIMUM?` might return `-1.812E-3`, indicating that the upper limit of the horizontal offset setting range is -1.812 ms.

DISPlay:FVTime:X[:SCALe]:OFFSet:MINimum? (Query Only)

Queries the lower limit of the horizontal offset setting range.

Conditions Measurement views: Frequency versus Time

Group Display commands

Syntax `DISPlay:FVTime:X[:SCALe]:OFFSet:MINimum?`

Related Commands DISPlay:FVTime:X[:SCALe]:OFFSet

Arguments None

Returns `<NRf>` The lower limit of the horizontal offset setting range.

Examples `DISPLAY:FVTIME:X:SCALE:OFFSET:MINIMUM?` might return `-16.28E-3`, indicating that the lower limit of the horizontal offset setting range is -16.28 ms.

DISPlay:FVTime:Y[:SCALe]

Sets or queries the vertical range of the Frequency versus Time graph.

Conditions Measurement views: Frequency versus Time

Group Display commands

Syntax

```
DISPlay:FVTime:Y[:SCALe] <value>
DISPlay:FVTime:Y[:SCALe]?
```

Related Commands DISPlay:FVTime:Y[:SCALe]:OFFSet

Arguments `<value>::=<NRf>` specifies the vertical range. Range: 10 Hz to 120 MHz.

Examples `DISPlay:FVTime:Y:SCALe 30MHz` sets the vertical range to 30 MHz in the Frequency versus Time graph.

DISPlay:FVTime:Y[:SCALe]:AUTO (No Query Form)

Sets the vertical scale automatically to fit the waveform to the screen in the Frequency versus Time view.

Conditions Measurement views: Frequency versus Time

Group Display commands

Syntax

```
DISPlay:FVTime:Y[:SCALe]:AUTO
```

Arguments None

Examples `DISPLAY:FVTIME:Y:SCALE:AUTO` sets the vertical scale automatically to fit the waveform to the screen.

DISPlay:FVTime:Y[:SCALe]:OFFSet

Sets or queries the vertical offset (the value at the center of the vertical axis) in the Frequency versus Time graph.

Conditions Measurement views: Frequency versus Time

Group Display commands

Syntax

```
DISPlay:FVTime:Y[:SCALe]:OFFSet <value>
DISPlay:FVTime:Y[:SCALe]:OFFSet?
```

Related Commands DISPlay:FVTime:Y[:SCALe]

Arguments `<value>::=<NRf>` specifies the vertical offset. Range: -60 MHz to +60 MHz.

Examples `DISPLAY:FVTIME:Y:SCALE:OFFSET -14.5MHz` sets the vertical offset to -14.5 MHz in the Frequency versus Time graph.

DISPlay:GENeral:MEASview:DELete (No Query Form)

Deletes a measurement view in the general signal viewing.

Conditions Measurement views: General signal viewing

Group Display commands

Syntax

```
DISPlay:GENeral:MEASview:DELete { SPECtrum | DPSA | AVTime|
FVTime | PHVTime | IQVTime | SGRam | TOVerview }
```

Arguments The following table shows the arguments. The arguments are the string type.

Table 2-29: General signal viewing views

Argument	View
SPECtrum	Spectrum
DPSA	DPX (Digital Phosphor) spectrum
AVTime	Amplitude versus Time
FVTime	Frequency versus Time
PHVTime	Phase versus Time
IQVTime	IQ versus Time
SGRam	Spectrogram
TOVerview	Time overview

If you attempt to delete a view that is not displayed on screen, the error (-200, "Execution error; Measurement not running") will be returned.

Examples `DISPLAY:GENERAL:MEASVIEW:DELETE DPSA` deletes the DPX spectrum view.

DISPlay:GENeral:MEASview:NEW (No Query Form)

Displays a new measurement view in the general signal viewing.

Conditions Measurement views: General signal viewing

Group Display commands

Syntax

```
DISPlay:GENeral:MEASview:NEW { SPECtrum | DPSA | AVTime|
FVTime | PHVTime | IQVTime | SGRam | TOVerview }
```

Arguments (See Table 2-29 on page 2-204.) If you attempt to open a view that is currently displayed on screen, the error (-200, "Execution error; Measurement is already running") will be returned.

Examples `DISPLAY:GENERAL:MEASVIEW:NEW DPSA` creates the DPX spectrum view.

DISPlay:GENeral:MEASview:SELect

Selects a measurement view in the general signal viewing on the screen. The query command returns the currently selected view.

Selecting a measurement optimizes it. Other measurements may be optimized as a side effect. Refer to the DISPlay:WINDow:OPTimized:MEASurement? query.

Conditions Measurement views: General signal viewing

Group Display commands

Syntax

```
DISPlay:GENeral:MEASview:SELect { SPECtrum | DPSA | AVTime|
FVTime | PHVTime | IQVTime | SGRam | TOVerview }
DISPlay:GENeral:MEASview:SELect?
```

Arguments (See Table 2-29 on page 2-204.) If you attempt to select a view that is not displayed on screen, the error (-200, "Execution error; Measurement not running") will be returned.

The time overview (TOVerview) cannot be selected as the primary measurement. If you select it, the error (-200, "Execution error; Time Overview cannot be Primary measurement") will be returned. If you use the DISPlay:GENeral MEASview:SELect? query with the time overview as the only measurement active, the error (-200, "Execution error; Analysis selected is not running") will be returned.

Examples `DISPLAY:GENERAL:MEASVIEW:SELECT DPSA` selects the DPX spectrum view.

DISPlay:GPRF:MEASview:DELete (No Query Form)

Deletes a selected measurement view in the RF measurements.

Conditions Measurement views: RF measurements

Group Display commands

Syntax

```
DISPlay:GPRF:MEASview:DELete { CCDF | ACPower | MCPower |
OBW | PNOise | SPURious }
```

Arguments

`CCDF` deletes the CCDF view.

`ACPower` deletes the Channel power and ACPR view.

`MCPower` deletes the MCPR view.

`OBW` deletes the Occupied Bandwidth view.

`PNOise` deletes the Phase Noise view (Option 11 only).

`SPURious` deletes the Spurious view.

If you attempt to delete a view that is not displayed on screen, the error (-200, "Execution error; Measurement not running") will be returned.

Examples

`DISPLAY:GPRF:MEASVIEW:DELETE ACPower` deletes the Channel power and ACPR view.

DISPlay:GPRF:MEASview:NEW (No Query Form)

Displays a new measurement view in the RF measurements.

Conditions Measurement views: RF measurements

Group Display commands

Syntax

```
DISPlay:GPRF:MEASview:NEW { CCDF | ACPower | MCPower | OBW
| PNOise | SPURious }
```

Arguments

`CCDF` opens the CCDF view.

`ACPower` opens the Channel power and ACPR view.

`MCPower` opens the MCPR view.

`OBW` opens the Occupied Bandwidth view.

`PNOise` opens the Phase Noise view (Option 11 only).

`SPURious` opens the Spurious view.

If you attempt to open a view that is currently displayed on screen, the error (-200, "Execution error; Measurement is already running") will be returned.

Examples `DISPLAY:GPRF:MEASVIEW:NEW ACPower` creates the Channel power and ACPR view.

DISPlay:GPRF:MEASview:SELect

Selects a measurement view in the RF measurements on the screen. The query command returns the currently selected view.

Selecting a measurement optimizes it. Other measurements may be optimized as a side effect. Refer to the DISPlay:WINDow:OPTimized:MEASurement? query.

Conditions Measurement views: RF measurements

Group Display commands

Syntax

```
DISPlay:GPRF:MEASview:SELect { CCDF | ACPower | MCPower |
OBW | PNOise | SPURious }
DISPlay:GPRF:MEASview:SELect?
```

Arguments

`CCDF` selects the CCDF view.

`ACPower` selects the Channel power and ACPR view.

`MCPower` selects the MCPR view.

`OBW` selects the Occupied Bandwidth view.

`PNOise` selects the Phase Noise view (Option 11 only).

`SPURious` selects the Spurious view.

If you attempt to select a view that is not displayed on screen, the error (-200, "Execution error; Measurement not running") will be returned.

Examples `DISPLAY:GPRF:MEASVIEW:SELECT ACPower` selects the Channel power and ACPR view.

DISPlay:IQVTime:WINDow:TRACe:GRATicule:GRID:STATe

Determines whether to show or hide the graticule grid on the screen.

Conditions Measurement views: RF I&Q versus Time

Group Display commands

Syntax
```
DISPlay:IQVTime:WINDow:TRACe:GRATicule:GRID:STATe { OFF |
ON | 0 | 1 }
DISPlay:IQVTime:WINDow:TRACe:GRATicule:GRID:STATe?
```

Arguments `OFF` or `0` hides the graticule grid.

`ON` or `1` shows the graticule grid.

Examples `DISPLAY:IQVTIME:WINDOW:TRACE:GRATICULE:GRID:STATE ON` shows the graticule grid on the RF I&Q versus Time view.

DISPlay:IQVTime:X[:SCALe]

Sets or queries the horizontal scale (full-scale time) of the RF I&Q versus Time graph. Programming a specified scale sets DISPlay:IQVTime:X[:SCALe] AUTO:STATe OFF.

Conditions Measurement views: RF I&Q versus Time

Group Display commands

Syntax
```
DISPlay:IQVTime:X[:SCALe] <value>
DISPlay:IQVTime:X[:SCALe]?
```

Related Commands DISPlay:IQVTime:X[:SCALe]:AUTO:STATe, DISPlay:IQVTime:X[:SCALe]:OFFSet

Arguments `<value>::={ <NRf> | MAXimum | MINimum }` specifies the horizontal scale in full-scale time. `MAXimum` and `MINimum` represent the upper and lower limits of the setting range, respectively. Use the DISPlay:IQVTime:X[:SCALe]:MAXimum? and DISPlay:IQVTime:X[:SCALe]:MINimum? queries to get the upper and lower limit values of the setting range.

Examples `DISPLAY:IQVTIME:X:SCALE 100us` sets the horizontal scale to 100 µs.

DISPlay:IQVTime:X[:SCALe]:AUTO (No Query Form)

Sets the horizontal scale automatically to fit the waveform to the screen in the RF I&Q versus Time view. Executing this command sets DISPlay:IQVTime X[:SCALe]:AUTO:STATe ON.

Conditions Measurement views: RF I&Q versus Time

Group Display commands

Syntax `DISPlay:IQVTime:X[:SCALe]:AUTO`

Related Commands DISPlay:IQVTime:X[:SCALe], DISPlay:IQVTime:X[:SCALe]:AUTO:STATe

Arguments None

Examples `DISPLAY:IQVTIME:X:SCALE:AUTO` sets the horizontal scale automatically to fit the waveform to the screen.

DISPlay:IQVTime:X[:SCALe]:AUTO:STATe

Determines whether to set the horizontal scale automatically or manually.

Conditions Measurement views: RF I&Q versus Time

Group Display commands

Syntax
```
DISPlay:IQVTime:X[:SCALe]:AUTO:STATe { OFF | ON | 0 | 1 }
DISPlay:IQVTime:X[:SCALe]:AUTO:STATe?
```

Arguments `OFF` or `0` specifies that the horizontal scale is set manually. To set it, use the DISPlay:IQVTime:X[:SCALe] and DISPlay:IQVTime:X[:SCALe]:OFFSet commands.

`ON` or `1` specifies that the horizontal scale is set automatically.

Examples `DISPLAY:IQVTIME:X:SCALE:AUTO:STATE ON` specifies that the horizontal scale is set automatically.

DISPlay:IQVTime:X[:SCALe]:MAXimum? (Query Only)

Queries the upper limit of the horizontal scale setting range.

Conditions Measurement views: RF I&Q versus Time

Group Display commands

Syntax `DISPlay:IQVTime:X[:SCALe]:MAXimum?`

Related Commands DISPlay:IQVTime:X[:SCALe]

Arguments None

Returns `<NRf>` The upper limit of the horizontal scale setting range.

Examples `DISPLAY:IQVTIME:X:SCALE:MAXIMUM?` might return `18.135E-3`, indicating that the upper limit of the horizontal scale setting range is 18.135 ms.

DISPlay:IQVTime:X[:SCALe]:MINimum? (Query Only)

Queries the lower limit of the horizontal scale setting range.

Conditions Measurement views: RF I&Q versus Time

Group Display commands

Syntax `DISPlay:IQVTime:X[:SCALe]:MINimum?`

Related Commands DISPlay:IQVTime:X[:SCALe]

Arguments None

Returns `<NRf>` The lower limit of the horizontal scale setting range.

Examples `DISPLAY:IQVTIME:X:SCALE:MINIMUM?` might return `10.0E-9`, indicating that the lower limit of the horizontal scale setting range is 10.0 ns.

DISPlay:IQVTime:X[:SCALe]:OFFSet

Sets or queries the minimum horizontal value (left edge) of the RF I&Q versus Time graph. Programming a specified offset sets DISPlay:IQVTime:X[:SCALe] AUTO:STATe OFF.

Conditions Measurement views: RF I&Q versus Time

Group Display commands

Syntax

```
DISPlay:IQVTime:X[:SCALe]:OFFSet <value>
DISPlay:IQVTime:X[:SCALe]:OFFSet?
```

Related Commands DISPlay:IQVTime:X[:SCALe]:AUTO:STATe, DISPlay:IQVTime:X[:SCALe]

Arguments `<value>::={ <NRf> | MAXimum | MINimum }` specifies the horizontal offset. `MAXimum` and `MINimum` represent the upper and lower limits of the setting range, respectively.

Use the DISPlay:IQVTime:X[:SCALe]:OFFSet:MAXimum? and DISPlay:IQVTime:X[:SCALe]:OFFSet:MINimum? queries to get the upper and lower limit values of the setting range.

Examples `DISPLAY:IQVTIME:X:SCALE:OFFSET 800ns` sets the minimum horizontal value to 800 ns in the RF I&Q versus Time graph.

DISPlay:IQVTime:X[:SCALe]:OFFSet:MAXimum? (Query Only)

Queries the upper limit of the horizontal offset setting range.

Conditions Measurement views: RF I&Q versus Time

Group Display commands

Syntax

```
DISPlay:IQVTime:X[:SCALe]:OFFSet:MAXimum?
```

Related Commands DISPlay:IQVTime:X[:SCALe]:OFFSet

Arguments None

Returns `<NRf>` The upper limit of the horizontal offset setting range.

Examples `DISPLAY:IQVTIME:X:SCALE:OFFSET:MAXIMUM?` might return `-1.812E-3`, indicating that the upper limit of the horizontal offset setting range is -1.812 ms.

DISPlay:IQVTime:X[:SCALe]:OFFSet:MINimum? (Query Only)

Queries the lower limit of the horizontal offset setting range.

Conditions Measurement views: RF I&Q versus Time

Group Display commands

Syntax `DISPlay:IQVTime:X[:SCALe]:OFFSet:MINimum?`

Related Commands DISPlay:IQVTime:X[:SCALe]:OFFSet

Arguments None

Returns `<NRf>` The lower limit of the horizontal offset setting range.

Examples `DISPLAY:IQVTIME:X:SCALE:OFFSET:MINIMUM?` might return `-16.28E-3`, indicating that the lower limit of the horizontal offset setting range is -16.28 ms.

DISPlay:IQVTime:Y[:SCALe]

Sets or queries the vertical range of the RF I&Q versus Time graph.

Conditions Measurement views: RF I&Q versus Time

Group Display commands

Syntax
```
DISPlay:IQVTime:Y[:SCALe] <value>
DISPlay:IQVTime:Y[:SCALe]?
```

Related Commands DISPlay:IQVTime:Y[:SCALe]:OFFSet

Arguments `<value>::=<NRf>` specifies the vertical range. Range: 1µ to 10 V.

Examples `DISPLAY:IQVTIME:Y:SCALE 1.5` sets the vertical range to 1.5 V in the RF I&Q versus Time graph.

DISPlay:IQVTime:Y[:SCALe]:AUTO (No Query Form)

Sets the vertical scale automatically to fit the waveform to the screen in the RF I&Q versus Time view.

Conditions Measurement views: RF I&Q versus Time

Group Display commands

Syntax
```
DISPlay:IQVTime:Y[:SCALe]:AUTO
```

Arguments None

Examples `DISPLAY:IQVTIME:Y:SCALE:AUTO` sets the vertical scale automatically to fit the waveform to the screen.

DISPlay:IQVTime:Y[:SCALe]:OFFSet

Sets or queries the vertical offset (the value at the center of the vertical axis) in the RF I&Q versus Time graph.

Conditions Measurement views: RF I&Q versus Time

Group Display commands

Syntax
```
DISPlay:IQVTime:Y[:SCALe]:OFFSet <value>
DISPlay:IQVTime:Y[:SCALe]:OFFSet?
```

Related Commands DISPlay:IQVTime:Y[:SCALe]

Arguments `<value>::=<NRf>` specifies the vertical offset. Range: -5 to +5 V.

Examples `DISPLAY:IQVTIME:Y:SCALE:OFFSET -82.75mV` sets the vertical offset to -82.75 mV in the RF I&Q versus Time graph.

DISPlay:IQVTime:Y[:SCALe]:RESCale (No Query Form)

Rescales the vertical axis automatically to fit the waveform to the screen.

Conditions Measurement views: RF I&Q versus Time

Group Display commands

Syntax `DISPlay:IQVTime:Y[:SCALe]:RESCale`

Arguments None

Examples `DISPLAY:IQVTIME:Y:SCALE:RESCALE` rescales the vertical axis automatically to fit the waveform to the screen.

DISPlay:MCPower:MARKer:SHOW:STATe

Determines whether to show or hide the readout for the selected marker in the MCPR view.

Conditions Measurement views: MCPR

Group Display commands

Syntax
```
DISPlay:MCPower:MARKer:SHOW:STATe { OFF | ON | 0 | 1 }
DISPlay:MCPower:MARKer:SHOW:STATe?
```

Arguments `OFF` or `0` hides the readout for the selected marker in the graph.

`ON` or `1` shows the readout for the selected marker in the graph.

Examples `DISPLAY:MCPOWER:MARKER:SHOW:STATE ON` shows the readout for the selected marker in the graph.

DISPlay:MCPower:PLEVel:SHOW:STATe

Determines whether to show or hide the power levels in the MCPR view.

Conditions Measurement views: MCPR

Group Display commands

Syntax

```
DISPlay:MCPower:PLEVel:SHOW:STATe { OFF | ON | 0 | 1 }
DISPlay:MCPower:PLEVel:SHOW:STATe?
```

Arguments `OFF` or `0` hides the power levels in the graph.

`ON` or `1` shows the power levels in the graph.

Examples `DISPLAY:MCPOWER:PLEVEL:SHOW:STATE ON` shows the power levels in the graph.

DISPlay:MCPower:RESet:SCALe (No Query Form)

Resets the horizontal and vertical scale to the default values described below in the MCPR view.

Vertical offset = Reference level,
Vertical scale = 100 dB,
Horizontal offset = Center frequency, and
Horizontal scale = Default span

Conditions Measurement views: MCPR

Group Display commands

Syntax

```
DISPlay:MCPower:RESet:SCALe
```

Arguments None

Examples `DISPLAY:MCPOWER:RESET:SCALE` resets the horizontal and vertical scale to the default values.

DISPlay:MCPower:WINDow:TRACe:GRATicule:GRID:STATe

Determines whether to show or hide the graticule grid on the screen.

Conditions Measurement views: MCPR

Group Display commands

Syntax

```
DISPlay:MCPower:WINDow:TRACe:GRATicule:GRID:STATe { OFF |
ON | 0 | 1 }
DISPlay:MCPower:WINDow:TRACe:GRATicule:GRID:STATe?
```

Arguments `OFF` or `0` hides the graticule grid.

`ON` or `1` shows the graticule grid.

Examples `DISPLAY:MCPOWER:WINDOW:TRACE:GRATICULE:GRID:STATE ON` shows the graticule grid on the screen.

DISPlay:MCPower:X[:SCALe]

Sets or queries the horizontal range of the MCPR graph.

Conditions Measurement views: MCPR

Group Display commands

Syntax

```
DISPlay:MCPower:X[:SCALe] <value>
DISPlay:MCPower:X[:SCALe]?
```

Related Commands DISPlay:MCPower:X[:SCALe]:OFFSet

Arguments `<value>::=<NRf>` specifies the horizontal range.
Range: 10 Hz to 6.2 GHz (RSA6106A) / 14 GHz (RSA6114A).

Examples DISPLAY:MCPOWER:X:SCALE 10MHz sets the horizontal range to 10 MHz.

DISPlay:MCPower:X[:SCALe]:AUTO (No Query Form)

Rescales the horizontal axis automatically to fit the waveform to the screen in the MCPR view.

Conditions Measurement views: MCPR

Group Display commands

Syntax DISPlay:MCPower:X[:SCALe]:AUTO

Arguments None

Examples DISPLAY:MCPOWER:X:SCALE:AUTO rescales the horizontal scale automatically to fit the waveform to the screen.

DISPlay:MCPower:X[:SCALe]:OFFSet

Sets or queries the minimum horizontal value (left edge) of the MCPR graph.

Conditions Measurement views: MCPR

Group Display commands

Syntax

```
DISPlay:MCPower:X[:SCALe]:OFFSet <value>
DISPlay:MCPower:X[:SCALe]:OFFSet?
```

Related Commands DISPlay:MCPower:X[:SCALe]

Arguments <value>::=<NRf> specifies the minimum horizontal value.
Range: [(center frequency) - (X scale) × 0.9] to [(center frequency) + (X scale) × 0.9]

Examples DISPLAY:MCPOWER:X:SCALE:OFFSET 1.45GHz sets the minimum horizontal value to 1.45 GHz in the MCPR graph.

DISPlay:MCPower:Y[:SCALe]

Sets or queries the vertical range of the MCPR graph.

Conditions Measurement views: MCPR

Group Display commands

Syntax

```
DISPlay:MCPower:Y[:SCALe] <value>
DISPlay:MCPower:Y[:SCALe]?
```

Related Commands DISPlay:MCPower:Y[:SCALe]:OFFSet

Arguments `<value>::=<NRf>` specifies the vertical range. Range: 0.1 to 200 dB.

Examples `DISPLAY:MCPOWER:Y:SCALE 100` sets the vertical range to 100 dB in the MCPR graph.

DISPlay:MCPower:Y[:SCALe]:AUTO (No Query Form)

Rescales the vertical axis automatically to fit the waveform to the screen in the MCPR view.

Conditions Measurement views: MCPR

Group Display commands

Syntax

```
DISPlay:MCPower:Y[:SCALe]:AUTO
```

Arguments None

Examples `DISPLAY:MCPOWER:Y:SCALE:AUTO` rescales the vertical scale automatically to fit the waveform to the screen.

DISPlay:MCPower:Y[:SCALe]:OFFSet

Sets or queries the vertical offset (the value at the top edge of the vertical axis) in the MCPR graph.

Conditions Measurement views: MCPR

Group Display commands

Syntax

```
DISPlay:MCPower:Y[:SCALe]:OFFSet <value>
DISPlay:MCPower:Y[:SCALe]:OFFSet?
```

Related Commands DISPlay:MCPower:Y[:SCALe]

Arguments `<value>::=<NRf>` specifies the vertical offset. Range: -170 to +50 dBm.

Examples `DISPLAY:MCPOWER:Y:SCALE:OFFSET -12.5` sets the vertical offset to -12.5 dBm in the MCPR graph.

DISPlay:MERRor:Y[:SCALe]

Sets or queries the vertical range of the Magnitude error versus Time graph.

Conditions Measurement views: Magnitude error versus Time

Group Display commands

Syntax

```
DISPlay:MERRor:Y[:SCALe] <value>
DISPlay:MERRor:Y[:SCALe]?
```

Related Commands DISPlay:MERRor:Y[:SCALe]:OFFSet

Arguments `<value>::=<NRf>` specifies the vertical range. Range: 1 to 100%.

Examples `DISPLAY:MERROR:Y:SCALE 50` sets the vertical range to 50% in the Magnitude error versus Time graph.

DISPlay:MERRor:Y[:SCALe]:AUTO (No Query Form)

Sets the vertical scale automatically to fit the waveform to the screen in the Magnitude error versus Time view.

Conditions Measurement views: Magnitude error versus Time

Group Display commands

Syntax

```
DISPlay:MERRor:Y[:SCALe]:AUTO
```

Arguments None

Examples `DISPLAY:MERROR:Y:SCALE:AUTO` sets the vertical scale automatically to fit the waveform to the screen.

DISPlay:MERRor:Y[:SCALe]:OFFSet

Sets or queries the minimum vertical value (bottom edge) of the Magnitude error versus Time graph.

Conditions Measurement views: Magnitude error versus Time

Group Display commands

Syntax

```
DISPlay:MERRor:Y[:SCALe]:OFFSet <value>
DISPlay:MERRor:Y[:SCALe]:OFFSet?
```

Related Commands DISPlay:MERRor:Y[:SCALe]

Arguments `<value>::=<NRf>` specifies the minimum vertical value. Range: -100 to +100%.

Examples `DISPLAY:MERROR:Y:SCALE:OFFSET -9.5` sets the minimum vertical value to -9.5% in the Magnitude error versus Time graph.

DISPlay:OBWidth:MARKer:SHOW:STATe

Determines whether to show or hide the readout for the selected marker in the Occupied Bandwidth view.

Conditions Measurement views: Occupied Bandwidth

Group Display commands

Syntax

```
DISPlay:OBWidth:MARKer:SHOW:STATe { OFF | ON | 0 | 1 }
DISPlay:OBWidth:MARKer:SHOW:STATe?
```

Arguments `OFF` or `0` hides the readout for the selected marker in the view.

`ON` or `1` shows the readout for the selected marker in the view.

Examples `DISPLAY:OBWIDTH:MARKER:SHOW:STATE ON` shows the readout for the selected marker in the view.

DISPlay:OBWidth:RESet:SCALe (No Query Form)

Resets the horizontal and vertical scale to the default values described below in the Occupied Bandwidth view.

Vertical offset = Reference level,
Vertical scale = 100 dB,
Horizontal offset = Center frequency, and
Horizontal scale = Default span

Conditions Measurement views: Occupied Bandwidth

Group Display commands

Syntax

```
DISPlay:OBWidth:RESet:SCALe
```

Arguments None

Examples `DISPLAY:OBWIDTH:RESET:SCALE` resets the horizontal and vertical scale to the default values.

DISPlay:OBWidth:SELected:BANDwidth

Selects or queries the bandwidth (OBW or x dB BW) to measure in the Occupied Bandwidth view.

Conditions Measurement views: Occupied Bandwidth

Group Display commands

Syntax

```
DISPlay:OBWidth:SELected:BANDwidth { OBWidth | XDBbandwidth
}
DISPlay:OBWidth:SELected:BANDwidth?
```

Arguments

`OBWidth` selects the occupied bandwidth to measure.

`XDBbandwidth` selects the x dB bandwidth to measure.

Examples

`DISPLAY:OBWIDTH:SELECTED:BANDWIDTH OBWidth` selects the occupied bandwidth to measure.

DISPlay:OBWidth:WINDow:TRACe:GRATicule:GRID:STATe

Determines whether to show or hide the graticule grid on the screen.

Conditions Measurement views: Occupied Bandwidth

Group Display commands

Syntax

```
DISPlay:OBWidth:WINDow:TRACe:GRATicule:GRID:STATe { OFF |
ON | 0 | 1 }
DISPlay:OBWidth:WINDow:TRACe:GRATicule:GRID:STATe?
```

Arguments

`OFF` or `0` hides the graticule grid.

`ON` or `1` shows the graticule grid.

Examples

`DISPLAY:OBWIDTH:WINDOW:TRACE:GRATICULE:GRID:STATE ON` shows the graticule grid on the screen.

DISPlay:OBWidth:X[:SCALe]

Sets or queries the horizontal range of the Occupied Bandwidth view.

Conditions Measurement views: Occupied Bandwidth

Group Display commands

Syntax
```
DISPlay:OBWidth:X[:SCALe] <value>
DISPlay:OBWidth:X[:SCALe]?
```

Related Commands DISPlay:OBWidth:X[:SCALe]:OFFSet

Arguments `<value>::=<NRf>` specifies the horizontal range.
Range: 10 Hz to 6.2 GHz (RSA6106A) / 14 GHz (RSA6114A).

Examples `DISPLAY:OBWIDTH:X:SCALE 10MHz` sets the horizontal range to 10 MHz.

DISPlay:OBWidth:X[:SCALe]:AUTO (No Query Form)

Rescales the horizontal axis automatically to fit the waveform to the screen in the Occupied Bandwidth view.

Conditions Measurement views: Occupied Bandwidth

Group Display commands

Syntax
```
DISPlay:OBWidth:X[:SCALe]:AUTO
```

Arguments None

Examples `DISPLAY:OBWIDTH:X:SCALE:AUTO` rescales the horizontal scale automatically to fit the waveform to the screen.

DISPlay:OBWidth:X[:SCALe]:OFFSet

Sets or queries the minimum horizontal value (left edge) of the Occupied Bandwidth view.

Conditions Measurement views: Occupied Bandwidth

Group Display commands

Syntax

```
DISPlay:OBWidth:X[:SCALe]:OFFSet <value>
DISPlay:OBWidth:X[:SCALe]:OFFSet?
```

Related Commands DISPlay:OBWidth:X[:SCALe]

Arguments `<value>::=<NRf>` specifies the minimum horizontal value.
Range: [(center frequency) - (X scale) × 0.9] to [(center frequency) + (X scale) × 0.9]

Examples `DISPLAY:OBWIDTH:X:SCALE:OFFSET 1.45GHz` sets the minimum horizontal value to 1.45 GHz in the Occupied Bandwidth view.

DISPlay:OBWidth:Y[:SCALe]

Sets or queries the vertical range of the Occupied Bandwidth view.

Conditions Measurement views: Occupied Bandwidth

Group Display commands

Syntax

```
DISPlay:OBWidth:Y[:SCALe] <value>
DISPlay:OBWidth:Y[:SCALe]?
```

Related Commands DISPlay:OBWidth:Y[:SCALe]:OFFSet

Arguments `<value>::=<NRf>` specifies the vertical range. Range: 0.1 to 200 dB.

Examples `DISPLAY:OBWIDTH:Y:SCALE 100` sets the vertical range to 100 dB in the Occupied Bandwidth view.

DISPlay:OBWidth:Y[:SCALe]:AUTO (No Query Form)

Rescales the vertical axis automatically to fit the waveform to the screen in the Occupied Bandwidth view.

Conditions Measurement views: Occupied Bandwidth

Group Display commands

Syntax

```
DISPlay:OBWidth:Y[:SCALe]:AUTO
```

Arguments None

Examples `DISPLAY:OBWIDTH:Y:SCALE:AUTO` rescales the vertical scale automatically to fit the waveform to the screen.

DISPlay:OBWidth:Y[:SCALe]:OFFSet

Sets or queries the vertical offset (the value at the top edge of the vertical axis) in the Occupied Bandwidth view.

Conditions Measurement views: Occupied Bandwidth

Group Display commands

Syntax

```
DISPlay:OBWidth:Y[:SCALe]:OFFSet <value>
DISPlay:OBWidth:Y[:SCALe]:OFFSet?
```

Related Commands DISPlay:OBWidth:Y[:SCALe]

Arguments `<value>::=<NRf>` specifies the vertical offset. Range: -170 to +50 dBm.

Examples `DISPLAY:OBWIDTH:Y:SCALE:OFFSET -12.5` sets the vertical offset to -12.5 dBm in the Occupied Bandwidth view.

DISPlay:PERRor:Y[:SCALe]

Sets or queries the vertical range of the Phase error versus Time graph.

Conditions Measurement views: Phase error versus Time

Group Display commands

Syntax

```
DISPlay:PERRor:Y[:SCALe] <value>
DISPlay:PERRor:Y[:SCALe]?
```

Related Commands DISPlay:PERRor:Y[:SCALe]:OFFSet

Arguments `<value>::=<NRf>` specifies the vertical range. Range: 1 to 360°.

Examples `DISPLAY:PERROR:Y:SCALE 30` sets the vertical range to 30° in the Phase error versus Time graph.

DISPlay:PERRor:Y[:SCALe]:AUTO (No Query Form)

Sets the vertical scale automatically to fit the waveform to the screen in the Phase error versus Time view.

Conditions Measurement views: Phase error versus Time

Group Display commands

Syntax

```
DISPlay:PERRor:Y[:SCALe]:AUTO
```

Arguments None

Examples `DISPLAY:PERROR:Y:SCALE:AUTO` sets the vertical scale automatically to fit the waveform to the screen.

DISPlay:PERRor:Y[:SCALe]:OFFSet

Sets or queries the minimum vertical value (bottom edge) of the Phase error versus Time graph.

Conditions Measurement views: Phase error versus Time

Group Display commands

Syntax

```
DISPlay:PERRor:Y[:SCALe]:OFFSet <value>
DISPlay:PERRor:Y[:SCALe]:OFFSet?
```

Related Commands DISPlay:PERRor:Y[:SCALe]

Arguments `<value>::=<NRf>` specifies the minimum vertical value. Range: -360 to +360°.

Examples `DISPLAY:PERROR:Y:SCALE:OFFSET -14.5` sets the minimum vertical value to -14.5° in the Phase Error versus Time graph.

DISPlay:PHVTime:WINDow:TRACe:GRATicule:GRID:STATe

Determines whether to show or hide the graticule grid on the screen.

Conditions Measurement views: Phase versus Time

Group Display commands

Syntax

```
DISPlay:PHVTime:WINDow:TRACe:GRATicule:GRID:STATe { OFF |
ON | 0 | 1 }
DISPlay:PHVTime:WINDow:TRACe:GRATicule:GRID:STATe?
```

Arguments `OFF` or `0` hides the graticule grid.

`ON` or `1` shows the graticule grid.

Examples `DISPLAY:PHVTIME:WINDOW:TRACE:GRATICULE:GRID:STATE ON` shows the graticule grid on the Frequency versus Time view.

DISPlay:PHVTime:X[:SCALe]

Sets or queries the horizontal scale (full-scale time) of the Phase versus Time graph. Programming a specified scale sets DISPlay:PHVTime:X[:SCALe] AUTO:STATe OFF.

Conditions Measurement views: Phase versus Time

Group Display commands

Syntax

```
DISPlay:PHVTime:X[:SCALe] <value>
DISPlay:PHVTime:X[:SCALe]?
```

Related Commands DISPlay:PHVTime:X[:SCALe]:AUTO:STATe, DISPlay:PHVTime:X[:SCALe]:OFFSet

Arguments `<value>::={ <NRf> | MAXimum | MINimum }` specifies the horizontal scale in full-scale time. `MAXimum` and `MINimum` represent the upper and lower limits of the setting range, respectively.

Use the DISPlay:PHVTime:X[:SCALe]:MAXimum? and DISPlay:PHVTime:X[:SCALe]:OFFSet:MINimum? queries to get the upper and lower limit values of the setting range.

Examples `DISPLAY:PHVTIME:X:SCALE 1.5ms` sets the horizontal scale to 1.5 ms.

DISPlay:PHVTime:X[:SCALe]:AUTO (No Query Form)

Sets the horizontal scale automatically to fit the waveform to the screen in the Phase versus Time view. Executing this command sets DISPlay:PHVTime:X[SCALe]:AUTO:STATe ON.

Conditions Measurement views: Phase versus Time

Group Display commands

Syntax

```
DISPlay:PHVTime:X[:SCALe]:AUTO
```

Related Commands DISPlay:PHVTime:X[:SCALe]:AUTO:STATe

Arguments None

Examples `DISPLAY:PHVTIME:X:SCALE:AUTO` sets the horizontal scale automatically to fit the waveform to the screen.

DISPlay:PHVTime:X[:SCALe]:AUTO:STATe

Determines whether to set the horizontal scale automatically or manually.

Conditions Measurement views: Phase versus Time

Group Display commands

Syntax
```
DISPlay:PHVTime:X[:SCALe]:AUTO:STATe { OFF | ON | 0 | 1 }
DISPlay:PHVTime:X[:SCALe]:AUTO:STATe?
```

Arguments `OFF` or `0` specifies that the horizontal scale is set manually. To set it, use the DISPlay:PHVTime:X[:SCALe] and DISPlay:PHVTime:X[:SCALe]:OFFSet commands.

`ON` or `1` specifies that the horizontal scale is set automatically.

Examples `DISPLAY:PHVTIME:X:SCALE:AUTO:STATE ON` specifies that the horizontal scale is set automatically.

DISPlay:PHVTime:X[:SCALe]:MAXimum? (Query Only)

Queries the upper limit of the horizontal scale setting range.

Conditions Measurement views: Phase versus Time

Group Display commands

Syntax
```
DISPlay:PHVTime:X[:SCALe]:MAXimum?
```

Related Commands DISPlay:PHVTime:X[:SCALe]

Arguments None

Returns `<NRf>` The upper limit of the horizontal scale setting range.

Examples `DISPLAY:PHVTIME:X:SCALE:MAXIMUM?` might return `18.135E-3`, indicating that the upper limit of the horizontal scale setting range is 18.135 ms.

DISPlay:PHVTime:X[:SCALe]:MINimum? (Query Only)

Queries the lower limit of the horizontal scale setting range.

Conditions Measurement views: Phase versus Time

Group Display commands

Syntax `DISPlay:PHVTime:X[:SCALe]:MINimum?`

Related Commands DISPlay:PHVTime:X[:SCALe]

Arguments None

Returns `<NRf>` The lower limit of the horizontal scale setting range.

Examples `DISPLAY:PHVTIME:X:SCALE:MINIMUM?` might return `10.0E-9`, indicating that the lower limit of the horizontal scale setting range is 10.0 ns.

DISPlay:PHVTime:X[:SCALe]:OFFSet

Sets or queries the minimum horizontal value (left edge) of the Phase versus Time graph. Programming a specified offset sets DISPlay:PHVTime:X [SCALe]:AUTO:STATe OFF.

Conditions Measurement views: Phase versus Time

Group Display commands

Syntax
```
DISPlay:PHVTime:X[:SCALe]:OFFSet
DISPlay:PHVTime:X[:SCALe]:OFFSet?
```

Related Commands DISPlay:PHVTime:X[:SCALe]:AUTO:STATe, DISPlay:PHVTime:X[:SCALe],

Arguments `<value>::={ <NRf> | MAXimum | MINimum }` specifies the horizontal offset. `MAXimum` and `MINimum` represent the upper and lower limits of the setting range, respectively.

Use the DISPlay:PHVTime:X[:SCALe]:OFFSet:MAXimum? and DISPlay:PHVTime:X[:SCALe]:OFFSet:MINimum? queries to get the upper and lower limit values of the setting range.

Examples `DISPLAY:PHVTIME:X:SCALE:OFFSET 800ns` sets the minimum horizontal value to 800 ns in the Phase versus Time graph.

DISPlay:PHVTime:X[:SCALe]:OFFSet:MAXimum? (Query Only)

Queries the upper limit of the horizontal offset setting range.

Conditions Measurement views: Phase versus Time

Group Display commands

Syntax
```
DISPlay:PHVTime:X[:SCALe]:OFFSet:MAXimum?
```

Related Commands DISPlay:PHVTime:X[:SCALe]:OFFSet

Arguments None

Returns `<NRf>` The upper limit of the horizontal offset setting range.

Examples `DISPLAY:PHVTIME:X:SCALE:OFFSET:MAXIMUM?` might return `-1.812E-3`, indicating that the upper limit of the horizontal offset setting range is -1.812 ms.

DISPlay:PHVTime:X[:SCALe]:OFFSet:MINimum? (Query Only)

Queries the lower limit of the horizontal offset setting range.

Conditions Measurement views: Phase versus Time

Group Display commands

Syntax `DISPlay:PHVTime:X[:SCALe]:OFFSet:MINimum?`

Related Commands DISPlay:PHVTime:X[:SCALe]:OFFSet

Arguments None

Returns `<NRf>` The lower limit of the horizontal offset setting range.

Examples `DISPLAY:PHVTIME:X:SCALE:OFFSET:MINIMUM?` might return `-16.28E-3`, indicating that the lower limit of the horizontal offset setting range is -16.28 ms.

DISPlay:PHVTime:Y[:SCALe]

Sets or queries the vertical range of the Phase versus Time graph.

Conditions Measurement views: Phase versus Time

Group Display commands

Syntax

```
DISPlay:PHVTime:Y[:SCALe] <value>
DISPlay:PHVTime:Y[:SCALe]?
```

Related Commands DISPlay:PHVTime:Y[:SCALe]:OFFSet

Arguments `<value>::=<NRf>` specifies the vertical range. Range: 1 to 1T°.

Examples `DISPLAY:PHVTIME:Y:SCALE 180` sets the vertical range to 180° in the Phase versus Time graph.

DISPlay:PHVTime:Y[:SCALe]:AUTO (No Query Form)

Sets the vertical scale automatically to fit the waveform to the screen in the Phase versus Time view.

Conditions Measurement views: Phase versus Time

Group Display commands

Syntax

```
DISPlay:PHVTime:Y[:SCALe]:AUTO
```

Arguments None

Examples `DISPLAY:PHVTIME:Y:SCALE:AUTO` sets the vertical scale automatically to fit the waveform to the screen.

DISPlay:PHVTime:Y[:SCALe]:AXIS

Selects or queries the vertical axis representation.

Conditions Measurement views: Phase versus Time

Group Display commands

Syntax

```
DISPlay:PHVTime:Y[:SCALe]:AXIS { MODulopi | CONTinuous }
DISPlay:PHVTime:Y[:SCALe]:AXIS?
```

Arguments `MODulopi` (modulo π) shows the phase constrained within ±180° along the vertical axis.

`CONTinuous` shows the phase as continuous quantity along the vertical axis.

Examples `DISPLAY:PHVTIME:Y:SCALE:AXIS MODulopi` selects modulo π representation for the vertical axis.

DISPlay:PHVTime:Y[:SCALe]:AXIS:REFerence

Selects or queries which time point in the analysis period to use as the zero-phase-value reference.

Conditions Measurement views: Phase versus Time

Group Display commands

Syntax
```
DISPlay:PHVTime:Y[:SCALe]:AXIS:REFerence <value>
DISPlay:PHVTime:Y[:SCALe]:AXIS:REFerence?
```

Arguments `<value>::=<NRf>` specifies the phase reference time.

Examples `DISPLAY:PHVTIME:Y:SCALE:AXIS:REFERENCE 1.5us` sets the phase reference time to 1.5 μs.

DISPlay:PHVTime:Y[:SCALe]:OFFSet

Sets or queries the vertical offset (the value at the center of the vertical axis) in the Phase versus Time graph.

Conditions Measurement views: Phase versus Time

Group Display commands

Syntax
```
DISPlay:PHVTime:Y[:SCALe]:OFFSet <value>
DISPlay:PHVTime:Y[:SCALe]:OFFSet?
```

Related Commands DISPlay:PHVTime:Y[:SCALe]

Arguments `<value>::=<NRf>` specifies the vertical offset. Range: -0.5T to +0.5T°.

Examples `DISPLAY:PHVTIME:Y:SCALE:OFFSET -158.5` sets the vertical offset to -158.5° in the Phase versus Time graph.

DISPlay:PHVTime:Y[:SCALe]:RESCale (No Query Form)

Rescales the vertical axis automatically to fit the Phase versus Time waveform to the screen.

Conditions Measurement views: Phase versus Time

Group Display commands

Syntax `DISPlay:PHVTime:Y[:SCALe]:RESCale`

Arguments None

Examples `DISPLAY:PHVTIME:Y:SCALE:RESCALE` rescales the vertical axis automatically to fit the Phase versus Time waveform to the screen.

DISPlay:PNOise:MARKer:SHOW:STATe

Determines whether to show or hide the readout for the selected marker in the phase noise view.

Conditions Measurement views: Phase noise

Group Display commands

Syntax

```
DISPlay:PNOise:MARKer:SHOW:STATe { OFF | ON | 0 | 1 }
DISPlay:PNOise:MARKer:SHOW:STATe?
```

Arguments `OFF` or `0` hides the readout for the selected marker in the graph.

`ON` or `1` shows the readout for the selected marker in the graph.

Examples `DISPLAY:PNOISE:MARKER:SHOW:STATE ON` shows the readout for the selected marker in the graph.

DISPlay:PNOise:RESet:SCALe (No Query Form)

Resets the horizontal and vertical scale to the default values described below in the phase noise view.

Vertical offset = -50 dBc/Hz,
Vertical scale = 100 dB,
Horizontal start = 10 Hz, and
Horizontal stop = 1 GHz

Conditions Measurement views: Phase noise

Group Display commands

Syntax

```
DISPlay:PNOise:RESet:SCALe
```

Arguments None

Examples `DISPLAY:PNOISE:RESET:SCALE` resets the horizontal and vertical scale to the default values.

DISPlay:PNOise:WINDow:TRACe:GRATicule:GRID:STATe

Determines whether to show or hide the graticule grid on the screen.

Conditions Measurement views: Phase noise

Group Display commands

Syntax

```
DISPlay:PNOise:WINDow:TRACe:GRATicule:GRID:STATe { OFF | ON
| 0 | 1 }
DISPlay:PNOise:WINDow:TRACe:GRATicule:GRID:STATe?
```

Arguments

`OFF` or `0` hides the graticule grid.

`ON` or `1` shows the graticule grid.

Examples `DISPLAY:PNOISE:WINDOW:TRACE:GRATICULE:GRID:STATE ON` shows the graticule grid on the screen.

DISPlay:PNOise:X[:SCALe]:AUTO (No Query Form)

Rescales the horizontal axis automatically to fit the waveform to the screen in the phase noise view.

Conditions Measurement views: Phase noise

Group Display commands

Syntax `DISPlay:PNOise:X[:SCALe]:AUTO`

Arguments None

Examples `DISPLAY:PNOISE:X:SCALE:AUTO` rescales the horizontal scale automatically to fit the waveform to the screen.

DISPlay:PNOise:X[:SCALe]:STARt

Sets or queries the start frequency (left edge) of the phase noise graph.

Conditions Measurement views: Phase noise

Group Display commands

Syntax

```
DISPlay:PNOise:X[:SCALe]:STARt <value>
DISPlay:PNOise:X[:SCALe]:STARt?
```

Arguments `<value>::=<NRf>` specifies the start frequency.
Range: 10 mHz to 100 MHz.

Note that (start frequency) = $10^4 \times$ (stop frequency).

Examples `DISPLAY:PNOISE:X:SCALE:START 10Hz` sets the start frequency to 10 Hz in the phase noise graph.

DISPlay:PNOise:X[:SCALe]:STOP

Sets or queries the stop frequency (right edge) of the phase noise graph.

Conditions Measurement views: Phase noise

Group Display commands

Syntax

```
DISPlay:PNOise:X[:SCALe]:STOP <value>
DISPlay:PNOise:X[:SCALe]:STOP?
```

Arguments <value>::=<NRf> specifies the stop frequency.
Range: 100 Hz to 1 THz.

Note that (start frequency) = $10^4 \times$ (stop frequency).

Examples DISPLAY:PNOISE:X:SCALE:STOP 2GHz sets the stop frequency to 2 GHz in the phase noise graph.

DISPlay:PNOise:Y[:SCALe]

Sets or queries the vertical range of the phase noise graph.

Conditions Measurement views: Phase noise

Group Display commands

Syntax

```
DISPlay:PNOise:Y[:SCALe] <value>
DISPlay:PNOise:Y[:SCALe]?
```

Arguments <value>::=<NRf> specifies the vertical range. Range: 0.1 to 200 dB.

Examples DISPLAY:PNOISE:Y:SCALE 100 sets the vertical range to 100 dB for the phase noise graph.

DISPlay:PNOise:Y[:SCALe]:AUTO (No Query Form)

Rescales the vertical axis automatically to fit the waveform to the screen in the phase noise view.

Conditions Measurement views: Phase noise

Group Display commands

Syntax

```
DISPlay:PNOise:Y[:SCALe]:AUTO
```

Arguments None

Examples `DISPLAY:PNOISE:Y:SCALE:AUTO` rescales the vertical scale automatically to fit the waveform to the screen.

DISPlay:PNOise:Y[:SCALe]:OFFSet

Sets or queries the vertical offset (the value at the top edge of the vertical axis) of the phase noise graph.

Conditions Measurement views: Phase noise

Group Display commands

Syntax

```
DISPlay:PNOise:Y[:SCALe]:OFFSet <value>
DISPlay:PNOise:Y[:SCALe]:OFFSet?
```

Arguments `<value>::=<NRf>` specifies the vertical offset. Range: -200 to +20 dBc/Hz.

Examples `DISPLAY:PNOISE:Y:SCALE:OFFSET -12.5` sets the vertical offset to -12.5 dBc/Hz for the phase noise graph.

DISPlay:PNOise:Y[:SCALe]:PDIVision

Sets or queries the vertical scale (per division) of the phase noise graph.

Conditions Measurement views: Phase noise

Group Display commands

Syntax

```
DISPlay:PNOise:Y[:SCALe]:PDIVision <value>
DISPlay:PNOise:Y[:SCALe]:PDIVision?
```

Arguments `<value>::=<NRf>` specifies the vertical scale (per division).
Range: 0.01 to 20 dB/div.

Examples `DISPLAY:PNOISE:Y:SCALE:PDIVISION 5` sets the vertical scale to 5 dB/div.

DISPlay:PULSe:MEASview:DELete (No Query Form)

Deletes the measurement view in the pulsed RF measurements.

Conditions Measurement views: Pulsed RF measurements

Group Display commands

Syntax

```
DISPlay:PULSe:MEASview:DELete { RESult | TRACe | STATistics
}
```

Arguments `RESult` deletes the pulse table view.

`TRACe` deletes the pulse trace view.

`STATistics` deletes the pulse statistics view.

If you attempt to delete a view that is not displayed on screen, the error (-200, "Execution error; Measurement not running") will be returned.

Examples `DISPLAY:PULSE:MEASVIEW:DELETE TRACe` deletes the pulse trace view.

DISPlay:PULSe:MEASview:NEW (No Query Form)

Displays a new measurement view in the pulsed RF measurements.

Conditions Measurement views: Pulsed RF measurements

Group Display commands

Syntax

```
DISPlay:PULSe:MEASview:NEW { RESult | TRACe | STATistics }
```

Arguments

`RESult` opens the pulse table view.

`TRACe` opens the pulse trace view.

`STATistics` opens the pulse statistics view.

If you attempt to open a view that is currently displayed on screen, the error (-200, "Execution error; Measurement is already running") will be returned.

Examples

`DISPLAY:PULSE:MEASVIEW:NEW STATistics` creates the pulse statistics view.

DISPlay:PULSe:MEASview:SELect

Selects a measurement view in the pulsed RF measurements on the screen. The query command returns the currently selected view.

Conditions

Measurement views: Pulsed RF measurements

Group

Display commands

Syntax

```
DISPlay:PULSe:MEASview:SELect { RESult | TRACe | STATistics
}
DISPlay:PULSe:MEASview:SELect?
```

Arguments

`RESult` selects the pulse table view.

`TRACe` selects the pulse trace view.

`STATistics` selects the pulse statistics view.

If you attempt to select a view that is not displayed on screen, the error (-200, "Execution error; Measurement not running") will be returned.

Examples

`DISPLAY:PULSE:MEASVIEW:SELECT TRACe` selects the pulse trace view.

DISPlay:PULSe:RESult:ATX

Determines whether or not to show the average transmitted power measurement result in the pulse table.

Conditions

Measurement views: Pulse table

Group Display commands

Syntax
```
DISPlay:PULSe:RESult:ATX { OFF | ON | 0 | 1 }
DISPlay:PULSe:RESult:ATX?
```

Arguments `OFF` or `0` does not show the average transmitted power measurement result.

`ON` or `1` shows the average transmitted power measurement result in the pulse table.

Examples `DISPLAY:PULSE:RESULT:ATX ON` shows the average transmitted power measurement result in the pulse table.

DISPlay:PULSe:RESult:AVERage

Determines whether or not to show the average on power measurement result in the pulse table.

Conditions Measurement views: Pulse table

Group Display commands

Syntax
```
DISPlay:PULSe:RESult:AVERage { OFF | ON | 0 | 1 }
DISPlay:PULSe:RESult:AVERage?
```

Arguments `OFF` or `0` does not show the average on power measurement result.

`ON` or `1` shows the average on power measurement result in the results table.

Examples `DISPLAY:PULSE:RESULT:AVERAGE ON` shows the average on power measurement result in the pulse table.

DISPlay:PULSe:RESult:DROop

Determines whether or not to show the droop measurement result in the pulse table.

Conditions Measurement views: Pulse table

Group Display commands

Syntax

```
DISPlay:PULSe:RESult:DROop { OFF | ON | 0 | 1 }
DISPlay:PULSe:RESult:DROop?
```

Arguments

`OFF` or `0` does not show the droop measurement result.

`ON` or `1` shows the droop measurement result in the pulse table.

Examples

`DISPLAY:PULSE:RESULT:DROOP ON` shows the droop measurement result in the pulse table.

DISPlay:PULSe:RESult:DUTPct

Determines whether or not to show the duty factor (%) measurement result in the pulse table.

Conditions

Measurement views: Pulse table

Group

Display commands

Syntax

```
DISPlay:PULSe:RESult:DUTPct { OFF | ON | 0 | 1 }
DISPlay:PULSe:RESult:DUTPct?
```

Arguments

`OFF` or `0` does not show the duty factor measurement result.

`ON` or `1` shows the duty factor measurement result in the pulse table.

Examples

`DISPLAY:PULSE:RESULT:DUTPCT ON` shows the duty factor (%) measurement result in the pulse table.

DISPlay:PULSe:RESult:DUTRatio

Determines whether or not to show the duty factor (ratio) measurement result in the pulse table.

Conditions

Measurement views: Pulse table

Group

Display commands

Syntax
```
DISPlay:PULSe:RESult:DUTRatio { OFF | ON | 0 | 1 }
DISPlay:PULSe:RESult:DUTRatio?
```

Arguments

`OFF` or `0` does not show the duty factor measurement result.

`ON` or `1` shows the duty factor measurement result in the pulse table.

Examples

`DISPLAY:PULSE:RESULT:DUTRATIO ON` shows the duty factor (ratio) measurement result in the pulse table.

DISPlay:PULSe:RESult:FALL

Determines whether or not to show the fall time measurement result in the pulse table.

Conditions

Measurement views: Pulse table

Group

Display commands

Syntax
```
DISPlay:PULSe:RESult:FALL { OFF | ON | 0 | 1 }
DISPlay:PULSe:RESult:FALL?
```

Arguments

`OFF` or `0` does not show the fall time measurement result.

`ON` or `1` shows the fall time measurement result in the pulse table.

Examples

`DISPLAY:PULSE:RESULT:FALL ON` shows the fall time measurement result in the pulse table.

DISPlay:PULSe:RESult:FRDeviation

Determines whether or not to show the frequency deviation measurement result in the pulse table.

Conditions

Measurement views: Pulse table

Group

Display commands

Syntax
```
DISPlay:PULSe:RESult:FRDeviation { OFF | ON | 0 | 1 }
DISPlay:PULSe:RESult:FRDeviation?
```

Arguments
`OFF` or `0` does not show the frequency deviation measurement result.

`ON` or `1` shows the frequency deviation measurement result in the pulse table.

Examples
`DISPLAY:PULSE:RESULT:FRDEVIATION ON` shows the frequency deviation measurement result in the pulse table.

DISPlay:PULSe:RESult:MFReqerror

Determines whether or not to show the maximum frequency error measurement result in the pulse table.

Conditions
Measurement views: Pulse table

Group
Display commands

Syntax
```
DISPlay:PULSe:RESult:MFReqerror { OFF | ON | 0 | 1 }
DISPlay:PULSe:RESult:MFReqerror?
```

Arguments
`OFF` or `0` does not show the maximum frequency error measurement result.

`ON` or `1` shows the maximum frequency error measurement result in the pulse table.

Examples
`DISPLAY:PULSE:RESULT:MFREQERROR ON` shows the maximum frequency error measurement result in the pulse table.

DISPlay:PULSe:RESult:MPHerror

Determines whether or not to show the maximum phase error measurement result in the pulse table.

Conditions
Measurement views: Pulse table

Group
Display commands

Syntax
```
DISPlay:PULSe:RESult:MPHerror { OFF | ON | 0 | 1 }
DISPlay:PULSe:RESult:MPHerror?
```

Arguments
`OFF` or `0` does not show the maximum phase error measurement result.

`ON` or `1` shows the maximum phase error measurement result in the pulse table.

Examples
`DISPLAY:PULSE:RESULT:MPHERROR ON` shows the maximum phase error measurement result in the pulse table.

DISPlay:PULSe:RESult:PHDeviation

Determines whether or not to show the phase deviation measurement result in the pulse table.

Conditions
Measurement views: Pulse table

Group
Display commands

Syntax
```
DISPlay:PULSe:RESult:PHDeviation { OFF | ON | 0 | 1 }
DISPlay:PULSe:RESult:PHDeviation?
```

Arguments
`OFF` or `0` does not show the phase deviation measurement result.

`ON` or `1` shows the phase deviation measurement result in the pulse table.

Examples
`DISPLAY:PULSE:RESULT:PHDEVIATION ON` shows the phase deviation measurement result in the pulse table.

DISPlay:PULSe:RESult:PPFRequency

Determines whether or not to show the pulse-pulse carrier frequency measurement result in the pulse table.

Conditions
Measurement views: Pulse table

Group
Display commands

Syntax

```
DISPlay:PULSe:RESult:PPFRequency { OFF | ON | 0 | 1 }
DISPlay:PULSe:RESult:PPFRequency?
```

Arguments

`OFF` or `0` does not show the pulse-pulse carrier frequency measurement result.

`ON` or `1` shows the pulse-pulse carrier frequency measurement result in the pulse table.

Examples

`DISPLAY:PULSE:RESULT:PPFREQUENCY ON` shows the pulse-pulse carrier frequency measurement result in the pulse table.

DISPlay:PULSe:RESult:PPOWer

Determines whether or not to show the peak power measurement result in the pulse table.

Conditions

Measurement views: Pulse table

Group

Display commands

Syntax

```
DISPlay:PULSe:RESult:PPOWer { OFF | ON | 0 | 1 }
DISPlay:PULSe:RESult:PPOWer?
```

Arguments

`OFF` or `0` does not show the peak power measurement result.

`ON` or `1` shows the peak power measurement result in the pulse table.

Examples

`DISPLAY:PULSE:RESULT:PPOWER ON` shows the peak power measurement result in the pulse table.

DISPlay:PULSe:RESult:PPPHase

Determines whether or not to show the pulse-pulse carrier phase measurement result in the pulse table.

Conditions

Measurement views: Pulse table

Group

Display commands

Syntax

```
DISPlay:PULSe:RESult:PPPHase { OFF | ON | 0 | 1 }
DISPlay:PULSe:RESult:PPPHase?
```

Arguments

`OFF` or `0` does not show the pulse-pulse carrier phase measurement result.

`ON` or `1` shows the pulse-pulse carrier phase measurement result in the pulse table.

Examples

`DISPLAY:PULSE:RESULT:PPPHASE ON` shows the pulse-pulse carrier phase measurement result in the pulse table.

DISPlay:PULSe:RESult:RINTerval

Determines whether or not to show the repetition interval measurement result in the pulse table.

Conditions

Measurement views: Pulse table

Group

Display commands

Syntax

```
DISPlay:PULSe:RESult:RINTerval { OFF | ON | 0 | 1 }
DISPlay:PULSe:RESult:RINTerval?
```

Arguments

`OFF` or `0` does not show the repetition interval measurement result.

`ON` or `1` shows the repetition interval measurement result in the results table.

Examples

`DISPLAY:PULSE:RESULT:RINTERVAL ON` shows the repetition interval measurement result in the pulse table.

DISPlay:PULSe:RESult:RIPPle

Determines whether or not to show the ripple measurement result in the pulse table.

Conditions

Measurement views: Pulse table

Group

Display commands

Syntax
```
DISPlay:PULSe:RESult:RIPPle { OFF | ON | 0 | 1 }
DISPlay:PULSe:RESult:RIPPle?
```

Arguments `OFF` or `0` does not show the ripple measurement result.

`ON` or `1` shows the ripple measurement result in the pulse table.

Examples `DISPLAY:PULSE:RESULT:RIPPLE ON` shows the ripple measurement result in the pulse table.

DISPlay:PULSe:RESult:RISE

Determines whether or not to show the rise time measurement result in the pulse table.

Conditions Measurement views: Pulse table

Group Display commands

Syntax
```
DISPlay:PULSe:RESult:RISE { OFF | ON | 0 | 1 }
DISPlay:PULSe:RESult:RISE?
```

Arguments `OFF` or `0` does not show the rise time measurement result.

`ON` or `1` shows the rise time measurement result in the pulse table.

Examples `DISPLAY:PULSE:RESULT:RISE ON` shows the rise time measurement result in the pulse table.

DISPlay:PULSe:RESult:RMSFreqerror

Determines whether or not to show the RMS frequency error measurement result in the pulse table.

Conditions Measurement views: Pulse table

Group Display commands

Syntax

```
DISPlay:PULSe:RESult:RMSFreqerror { OFF | ON | 0 | 1 }
DISPlay:PULSe:RESult:RMSFreqerror?
```

Arguments

`OFF` or `0` does not show the RMS frequency error measurement result.

`ON` or `1` shows the RMS frequency error measurement result in the pulse table.

Examples

`DISPLAY:PULSE:RESULT:RMSFREQERROR ON` shows the RMS frequency error measurement result in the pulse table.

DISPlay:PULSe:RESult:RMSPherror

Determines whether or not to show the RMS phase error measurement result in the pulse table.

Conditions

Measurement views: Pulse table

Group

Display commands

Syntax

```
DISPlay:PULSe:RESult:RMSPherror { OFF | ON | 0 | 1 }
DISPlay:PULSe:RESult:RMSPherror?
```

Arguments

`OFF` or `0` does not show the RMS phase error measurement result.

`ON` or `1` shows the RMS phase error measurement result in the pulse table.

Examples

`DISPLAY:PULSE:RESULT:RMSPHERROR ON` shows the RMS phase error measurement result in the pulse table.

DISPlay:PULSe:RESult:RRATe

Determines whether or not to show the repetition rate measurement result in the pulse table.

Conditions

Measurement views: Pulse table

Group

Display commands

Syntax

```
DISPlay:PULSe:RESult:RRATe { OFF | ON | 0 | 1 }
DISPlay:PULSe:RESult:RRATe?
```

Arguments

OFF or 0 does not show the repetition rate measurement result.

ON or 1 shows the repetition rate measurement result in the pulse table.

Examples

DISPLAY:PULSE:RESULT:RRATE ON shows the repetition rate measurement result in the pulse table.

DISPlay:PULSe:RESult:TIME

Determines whether or not to show the time measurement result in the pulse table.

Conditions

Measurement views: Pulse table

Group

Display commands

Syntax

```
DISPlay:PULSe:RESult:TIME { OFF | ON | 0 | 1 }
DISPlay:PULSe:RESult:TIME?
```

Arguments

OFF or 0 does not show the time measurement result.

ON or 1 shows the time measurement result in the pulse table.

Examples

DISPLAY:PULSE:RESULT:TIME ON shows the time measurement result in the pulse table.

DISPlay:PULSe:RESult:WIDTh

Determines whether or not to show the pulse width measurement result in the pulse table.

Conditions

Measurement views: Pulse table

Group

Display commands

Syntax

```
DISPlay:PULSe:RESult:WIDTh { OFF | ON | 0 | 1 }
DISPlay:PULSe:RESult:WIDTh?
```

Arguments

`OFF` or `0` does not show the pulse width measurement result.

`ON` or `1` shows the pulse width measurement result in the pulse table.

Examples

`DISPLAY:PULSE:RESULT:WIDTH ON` shows the pulse width measurement result in the pulse table.

DISPlay:PULSe:SELect:NUMBer

Selects or queries a pulse to measure. For the selected pulse, the statistics view indicates the measurement result while the table view highlights it, and the trace view displays the waveform.

Conditions

Measurement views: Pulse statistics, Pulse table, Pulse trace

Group

Display commands

Syntax

```
DISPlay:PULSe:SELect:NUMBer <number>
DISPlay:PULSe:SELect:NUMBer?
```

Arguments

`<number>::=<NR1>` specifies the number of pulse to measure.
Range: -(the number of acquired pulses before the time reference) to +(the number of acquired pulses after the time reference).

Zero (0) represents the pulse at the analysis time reference specified using the [SENSe]:ANALysis:REFerence command. The number of acquired pulses depends on the analysis range.

Examples

`DISPLAY:WINDOW:SELECT:NUMBER -28` measures the pulse #-28.

DISPlay:PULSe:SELect:RESult

Selects or queries which result is shown in the pulse trace and statistics views.

Conditions

Measurement views: Pulse statistics, Pulse trace

Group

Display commands

Syntax

```
DISPlay:PULSe:SELect:RESult { AVERage | PPOWer | ATX |
WIDTh| RISE | FALL | RINTerval | RRATe | DUTPct | DUTRatio
| RIPPle| DROop | PPPHase | PPFRequency | RMSFreqerror
| MFReqerror| RMSPherror | MPHerror | FRDeviation |
PHDeviation }
DISPlay:PULSe:SELect:RESult?
```

Arguments

The following table lists the arguments.

Table 2-30: Pulse results

Argument	Result
AVERage	Average on power
PPOWer	Peak power
ATX	Average transmitted power
WIDTh	Pulse width
RISE	Rise time
FALL	Fall time
RINTerval	Repetition interval
RRATe	Repetition rate
DUTPct	Duty factor (%)
DUTRatio	Duty factor (ratio)
RIPPle	Ripple
DROop	Droop
PPPHase	Pulse-pulse carrier phase
PPFRequency	Pulse-pulse carrier frequency
RMSFreqerror	RMS frequency error
MFReqerror	Maximum frequency error
RMSPherror	RMS phase error
MPHerror	Maximum phase error
FRDeviation	Frequency deviation
PHDeviation	Phase deviation

Examples

`DISPLAY:PULSE:SELECT:RESULT AVERage` shows the average on power result in the pulse trace and statistics views.

DISPlay:PULSe:STATistics:MARKer:SHOW:STATe

Determines whether to show or hide the marker readout in the statistics graph. This command is valid when DISPlay:PULSe:STATistics:PLOT is set to FFT.

Conditions Measurement views: Pulse statistics

Group Display commands

Syntax

```
DISPlay:PULSe:STATistics:MARKer:SHOW:STATe { OFF | ON | 0 |
1 }
DISPlay:PULSe:STATistics:MARKer:SHOW:STATe?
```

Arguments `OFF` or `0` hides the marker readout.

`ON` or `1` shows the marker readout.

Examples `DISPLAY:PULSE:STATISTICS:MARKER:SHOW:STATE ON` shows the marker readout in the statistics graph.

DISPlay:PULSe:STATistics:PLOT

Selects or queries how to show the statistics graph.

Conditions Measurement views: Pulse statistics

Group Display commands

Syntax

```
DISPlay:PULSe:STATistics:PLOT { TREND | FFT }
DISPlay:PULSe:STATistics:PLOT?
```

Arguments `TREND` shows the statistics result along with the pulse number.

`FFT` shows the statistics result transformed into the frequency domain by FFT.

Examples `DISPLAY:PULSE:STATISTICS:PLOT TREND` shows the statistics result along with the pulse number.

DISPlay:PULSe:STATistics:WINDow:TRACe:GRATicule:GRID:STATe

Determines whether to show or hide the graticule grid in the statistics view.

Conditions Measurement views: Pulse statistics

Group Display commands

Syntax

```
DISPlay:PULSe:STATistics:WINDow:TRACe:GRATicule:GRID:STATe {
OFF | ON | 0 | 1 }
DISPlay:PULSe:STATistics:WINDow:TRACe:GRATicule:GRID:STATe?
```

Arguments `OFF` or `0` hides the graticule grid.

`ON` or `1` shows the graticule grid.

Examples `DISPLAY:PULSE:STATISTICS:WINDOW:TRACE:GRATICULE:GRID:STATE ON` shows the graticule grid on the statistics view.

DISPlay:PULSe:STATistics:X:RSCale (No Query Form)

Rescales the horizontal axis to fit the waveform to the screen in the statistics graph.

Conditions Measurement views: Pulse statistics

Group Display commands

Syntax `DISPlay:PULSe:STATistics:X:RSCale`

Arguments None

Examples `DISPLAY:PULSE:STATISTICS:X:RSCALE` rescales the horizontal axis of the statistics graph.

DISPlay:PULSe:STATistics:X[:SCALe]:NUMBer

Sets or queries the horizontal full scale in the statistics graph.

Conditions Measurement views: Pulse statistics

Group Display commands

Syntax

```
DISPlay:PULSe:STATistics:X[:SCALe]:NUMBer <value>
DISPlay:PULSe:STATistics:X[:SCALe]:NUMBer?
```

Related Commands DISPlay:PULSe:STATistics:X[:SCALe]:OFFSet

Arguments `<value>::=<NRf>` specifies the horizontal full scale. The setting range depends on the DISPlay:PULSe:STATistics:PLOT command parameters as shown in the table below.

DISPlay:PULSe:STATistics:PLOT	Setting range
TRENd	1 to 1000
FFT	1 Hz to 120 MHz

Examples `DISPlay:PULSe:STATistics:X:SCALe:NUMBer 50` sets the horizontal full scale to 50 pulses when the plot is trend.

DISPlay:PULSe:STATistics:X[:SCALe]:OFFSet

Selects or queries the minimum horizontal value (the first pulse to show) in the statistics graph.

Conditions Measurement views: Pulse statistics

Group Display commands

Syntax

```
DISPlay:PULSe:STATistics:X[:SCALe]:OFFSet <value>
DISPlay:PULSe:STATistics:X[:SCALe]:OFFSet?
```

Arguments `<value>::=<NRf>` specifies the number of the first pulse.
Range: -(X - X/10) to +(X - X/10)
where X is the horizontal scale set by the DISPlay:PULSe:STATistics:X[:SCALe]:NUMBer command.

Examples `DISPLAY:PULSE:STATISTICS:X:SCALE:OFFSET 120` sets the first pulse number to #120.

DISPlay:PULSe:STATistics:Y:RSCale (No Query Form)

Rescales the vertical axis to fit the waveform to the screen in the statistics graph.

Conditions Measurement views: Pulse statistics

Group Display commands

Syntax `DISPlay:PULSe:STATistics:Y:RSCale`

Arguments None

Examples `DISPLAY:PULSE:STATISTICS:Y:RSCALE` rescales the vertical axis of the statistics graph.

DISPlay:PULSe:STATistics:Y[:SCALe]:FULL

Sets or queries the vertical full scale in the statistics graph.

Conditions Measurement views: Pulse statistics

Group Display commands

Syntax

```
DISPlay:PULSe:STATistics:Y[:SCALe]:FULL <value>
DISPlay:PULSe:STATistics:Y[:SCALe]:FULL?
```

Related Commands DISPlay:PULSe:STATistics:X[:SCALe]:OFFSet

Arguments `<value>::=<NRf>` specifies the vertical full scale. The setting range depends on the DISPlay:PULSe:STATistics:PLOT command parameters and the measurement items as shown in the following table.

DISPlay:PULSe:STATistics:PLOT	Measurement item	Setting range
TRENd	Average on power, Peak power, Average transmitted power	0.1 to 200 dB
	Pulse width, Rise time, Fall time, Repetition interval	1 n to 5 Ms
	Repetition rate	100 m to 100 MHz
	Duty factor, Ripple	1 to 100%
	Droop	1 to 200%
	Pulse-pulse carrier phase	1 to 360°
FFT	All	10 to 200 dB

Examples

`DISPLAY:PULSE:STATISTICS:Y:SCALE:FULL 100` sets the vertical full scale to 100 dB.

DISPlay:PULSe:STATistics:Y[:SCALe]:OFFSet

Sets or queries the vertical offset in the statistics graph.

Conditions

Measurement views: Pulse statistics

Group

Display commands

Syntax

```
DISPlay:PULSe:STATistics:Y[:SCALe]:OFFSet <value>
DISPlay:PULSe:STATistics:Y[:SCALe]:OFFSet?
```

Arguments

`<value>::=<NRf>` specifies the vertical offset. The vertical offset is the value at the top or the bottom edge of the graph depending on measurement items. The setting range depends on the DISPlay:PULSe:STATistics:PLOT command parameters and the measurement items as shown in the table below.

DISPlay:PULSe :STATistics:PLOT	Measurement item	Setting range	Offset position [1]
TRENd	Average on power, Peak power, Average transmitted power	-170 to +50 dBm	Top (Bottom for the unit of Volts or Watts)
	Pulse width, Rise time, Fall time, Repetition interval	0 to 5 Ms	Bottom
	Repetition rate	0 to 100 MHz	Bottom

DISPlay:PULSe :STATistics:PLOT	Measurement item	Setting range	Offset position [1]
	Duty factor, Ripple	0 to 100%	Bottom
	Droop	0 to +100%	Bottom
	Pulse-pulse carrier phase	-180 to +180°	Bottom
FFT	All	-400 to +100 dB	Top

1 ndicates whether the offset is the value at the top or the bottom edge of the graph.

Examples `DISPLAY:PULSE:STATISTICS:Y:SCALE:OFFSET 24.8` sets the maximum vertical value to 24.8 dBm in the statistics graph.

DISPlay:PULSe:STATistics:Y[:SCALe]:STOP? (Query Only)

Queries the minimum vertical value (bottom edge) in the statistics graph.

Conditions Measurement views: Pulse statistics

Group Display commands

Syntax `DISPlay:PULSe:STATistics:Y[:SCALe]:STOP?`

Related Commands DISPlay:PULSe:STATistics:X[:SCALe]:OFFSet

Arguments None

Returns `<y_stop>::=<NRf>` is the minimum vertical value (bottom edge).

Examples `DISPLAY:PULSE:STATISTICS:Y:SCALE:STOP?` might return `-150.0`, indicating that the minimum vertical value is -150 dBm in the pulse statistics graph.

DISPlay:PULSe:TRACe:MARKer:SHOW:STATe

Determines whether to show or hide the marker readout in the pulse trace view.

Conditions Measurement views: Pulse trace

Group Display commands

Syntax
```
DISPlay:PULSe:TRACe:MARKer:SHOW:STATe { OFF | ON | 0 | 1 }
DISPlay:PULSe:TRACe:MARKer:SHOW:STATe?
```

Arguments `OFF` or `0` hides the marker readout.

`ON` or `1` shows the marker readout.

Examples `DISPLAY:PULSE:TRACE:MARKER:SHOW:STATE ON` shows the marker readout on the pulse trace view.

DISPlay:PULSe:TRACe:POINt:SHOW

Determines whether to show or hide the measurement points and lines in the pulse trace view.

Conditions Measurement views: Pulse trace

Group Display commands

Syntax
```
DISPlay:PULSe:TRACe:POINt:SHOW { OFF | ON | 0 | 1 }
DISPlay:PULSe:TRACe:POINt:SHOW?
```

Arguments `OFF` or `0` hides the measurement points and lines.

`ON` or `1` shows the measurement points and lines.

Examples `DISPLAY:PULSE:TRACE:POINT:SHOW ON` shows the measurement points and lines in the pulse trace view.

DISPlay:PULSe:TRACe:WINDow:TRACe:GRATicule:GRID:STATe

Determines whether to show or hide the graticule grid in the pulse trace view.

Conditions Measurement views: Pulse trace

Group Display commands

Syntax
```
DISPlay:PULSe:TRACe:WINDOw:TRACe:GRATicule:GRID:STATe { OFF
| ON | 0 | 1 }
DISPlay:PULSe:TRACe:WINDOw:TRACe:GRATicule:GRID:STATe?
```

Arguments

`OFF` or `0` hides the graticule grid.

`ON` or `1` shows the graticule grid.

Examples

`DISPLAY:PULSE:TRACE:WINDOW:TRACE:GRATICULE:GRID:STATE ON` shows the graticule grid on the pulse trace view.

DISPlay:PULSe:TRACe:X:RSCale (No Query Form)

Rescales the horizontal axis to fit the waveform to the screen in the pulse trace view.

Conditions

Measurement views: Pulse trace

Group

Display commands

Syntax
```
DISPlay:PULSe:TRACe:X:RSCale
```

Arguments

None

Examples

`DISPLAY:PULSE:TRACE:X:RSCALE` rescales the horizontal axis in the pulse trace view.

DISPlay:PULSe:TRACe:X[:SCALe]

Sets or queries the horizontal full scale in the pulse trace view.

Conditions

Measurement views: Pulse trace

Group

Display commands

Syntax
```
DISPlay:PULSe:TRACe:X[:SCALe] <value>
DISPlay:PULSe:TRACe:X[:SCALe]?
```

Arguments

`<value>::=<NRf>` specifies the horizontal full scale.
Range: 10 ns to acquisition memory capacity.

Examples

`DISPLAY:PULSE:TRACE:X:SCALE 5.5E-6` sets the horizontal scale to 5.5 μ s.

DISPlay:PULSe:TRACe:X[:SCALe]:FULL

Selects or queries the full-scale reference for the horizontal rescale.

Conditions

Measurement views: Pulse trace

Group

Display commands

Syntax

```
DISPlay:PULSe:TRACe:X[:SCALe]:FULL { SELected | MAXimum }
DISPlay:PULSe:TRACe:X[:SCALe]:FULL?
```

Arguments

`SELected` uses the selected pulse for the full-scale reference.

`MAXimum` uses the maximum pulse for the full-scale reference.

Examples

`DISPLAY:PULSE:TRACE:X:SCALE:FULL SELected` uses the selected pulse for the full-scale reference.

DISPlay:PULSe:TRACe:X[:SCALe]:OFFSet

Sets or queries the minimum horizontal value (left edge) in the pulse trace view.

Conditions

Measurement views: Pulse trace

Group

Display commands

Syntax

```
DISPlay:PULSe:TRACe:X[:SCALe]:OFFSet <value>
DISPlay:PULSe:TRACe:X[:SCALe]:OFFSet?
```

Related Commands

DISPlay:PULSe:TRACe:X[:SCALe]:PDIVision

Arguments

<value>::=<NRf> specifies the minimum horizontal value.
Range: [(analysis offset) - (X scale) × 0.9] to [(analysis offset) + (analysis length) - (X scale) × 0.1]

Examples

DISPLAY:PULSE:TRACE:X:SCALE:OFFSET 937.5E-9 sets the minimum horizontal value to 937.5 ns.

DISPlay:PULSe:TRACe:X[:SCALe]:PDIVision

Sets or queries the horizontal full scale in the pulse trace view.

Conditions

Measurement views: Pulse trace

Group

Display commands

Syntax

```
DISPlay:PULSe:TRACe:X[:SCALe]:PDIVision <value>
DISPlay:PULSe:TRACe:X[:SCALe]:PDIVision?
```

Arguments

<value>::=<NRf> specifies the horizontal full scale.
Range: 10 ns to acquisition memory capacity.

Examples

DISPLAY:PULSE:TRACE:X:SCALE:PDIVISION 5.5E-6 sets the horizontal scale to 5.5 μs.

DISPlay:PULSe:TRACe:Y:RSCale (No Query Form)

Rescales the vertical axis to fit the waveform to the screen in the pulse trace view.

Conditions

Measurement views: Pulse trace

Group

Display commands

Syntax

```
DISPlay:PULSe:TRACe:Y:RSCale
```

Arguments

None

Examples `DISPLAY:PULSE:TRACE:Y:RSCALE` rescales the vertical axis in the pulse trace view.

DISPlay:PULSe:TRACe:Y[:SCALe]:FULL

Sets or queries the vertical full scale in the pulse trace view.

Conditions Measurement views: Pulse trace

Group Display commands

Syntax

```
DISPlay:PULSe:TRACe:Y[:SCALe]:FULL <value>
DISPlay:PULSe:TRACe:Y[:SCALe]:FULL?
```

Related Commands DISPlay:PULSe:TRACe:Y[:SCALe]:OFFSet

Arguments `<value>::=<NRf>` specifies the vertical full scale.
Range: 0.1 to 200 dB.
The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples `DISPLAY:PULSE:TRACE:Y:SCALE:FULL 100` sets the vertical full scale to 100 dB.

DISPlay:PULSe:TRACe:Y[:SCALe]:OFFSet

Sets or queries the vertical offset (the value at the top edge of the vertical axis) in the pulse trace view.

Conditions Measurement views: Pulse trace

Group Display commands

Syntax

```
DISPlay:PULSe:TRACe:Y[:SCALe]:OFFSet <value>
DISPlay:PULSe:TRACe:Y[:SCALe]:OFFSet?
```

Related Commands DISPlay:PULSe:TRACe:Y[:SCALe]:STOP?

Arguments <value>::=<NRf> specifies the vertical offset. Range: -170 to +50 dBm.

Examples DISPLAY:PULSE:TRACE:Y:SCALE:OFFSET 23.5 sets the vertical offset to 23.5 dBm.

DISPlay:PULSe:TRACe:Y[:SCALe]:STOP? (Query Only)

Queries the minimum vertical value (bottom edge) in the pulse trace view.

Conditions Measurement views: Pulse trace

Group Display commands

Syntax `DISPlay:PULSe:TRACe:Y[:SCALe]:STOP?`

Related Commands DISPlay:PULSe:TRACe:Y[:SCALe]:OFFSet

Arguments None

Returns <y_stop>::=<NRf> is the minimum vertical value (bottom edge).

Examples DISPLAY:PULSE:TRACE:Y:SCALE:STOP? might return -150.0, indicating that the minimum vertical value is -150 dBm in the pulse trace view.

DISPlay:SGRam:FREQuency:AUTO (No Query Form)

Rescales the horizontal axis automatically to fit the waveform to the screen in the spectrogram view.

Conditions Measurement views: Spectrogram

Group Display commands

Syntax `DISPlay:SGRam:FREQuency:AUTO`

Arguments None

Examples `DISPLAY:SGRAM:FREQUENCY:AUTO` rescales the horizontal scale automatically to fit the waveform to the screen.

DISPlay:SGRam:FREQuency:OFFSet

Sets or queries the frequency offset (the value at the center of the horizontal axis) in the spectrogram.

Conditions Measurement views: Spectrogram

Group Display commands

Syntax

```
DISPlay:SGRam:FREQuency:OFFSet <value>
DISPlay:SGRam:FREQuency:OFFSet?
```

Related Commands DISPlay:SPECtrum:FREQuency[:SCAle]

Arguments `<value>::=<NRf>` specifies the frequency offset.
Range: [(center frequency) - (X scale) × 0.9] to [(center frequency) + (X scale) × 0.9]

Examples `DISPLAY:SGRAM:FREQUENCY:OFFSET 1.45GHz` sets the frequency offset to 1.45 GHz in the spectrogram.

DISPlay:SGRam:FREQuency:SCAle

Sets or queries the horizontal range of the spectrogram.

Conditions Measurement views: Spectrogram

Group Display commands

Syntax

```
DISPlay:SGRam:FREQuency:SCAle <value>
DISPlay:SGRam:FREQuency:SCAle?
```

Related Commands DISPlay:SPECtrum:FREQuency:OFFSet

Arguments

<value>::=<NRf> specifies the horizontal range.
Range: 10 Hz to 6.2 GHz (RSA6106A) / 14 GHz (RSA6114A).

Examples

`DISPLAY:SGRAM:FREQUENCY:SCALE 10MHz` sets the horizontal range to 10 MHz.

DISPlay:SGRam:TIME:AUTO (No Query Form)

Rescales the vertical axis automatically to fit the graph to the screen in the spectrogram view.

Conditions

Measurement views: Spectrogram

Group

Display commands

Syntax

```
DISPlay:SGRam:TIME:AUTO
```

Arguments

None

Examples

`DISPLAY:SGRAM:TIME:AUTO` rescales the vertical scale automatically to fit the waveform to the screen.

DISPlay:SGRam:TIME:OFFSet

Sets or queries the vertical axis (time) offset (bottom line number) in the spectrogram.

Conditions

Measurement views: Spectrogram

Group

Display commands

Syntax

```
DISPlay:SGRam:TIME:OFFSet <value>
DISPlay:SGRam:TIME:OFFSet?
```

Related Commands

DISPlay:SGRam:TIME:SCAle

Arguments
`<value>::=<NRf>` specifies the time offset.
Range: Line #0 to 125000. Zero (0) represents the latest line.

Examples
`DISPLAY:SGRAM:TIME:OFFSET 15` sets the time offset to Line #15.

DISPlay:SGRam:TIME:OVERlap

Determines whether or not to allow overlap between adjacent FFT frames on the time axis in the spectrogram.

Conditions
Measurement views: Spectrogram

Group
Display commands

Syntax
```
DISPlay:SGRam:TIME:OVERlap { OFF | ON | 0 | 1 }
DISPlay:SGRam:TIME:OVERlap?
```

Arguments
`OFF` or `0` inhibits overlap between adjacent FFT frames on the time axis.

`ON` or `1` allows overlap between adjacent FFT frames on the time axis.

Examples
`DISPLAY:SGRAM:TIME:OVERLAP ON` allows overlap between adjacent FFT frames on the time axis in the spectrogram.

DISPlay:SGRam:TIME:SCAle

Sets or queries the vertical scale (the amount of time in each line) in the spectrogram. The vertical axis is composed of successive spectral displays. The new spectra can be added at a timed rate specified by this command. For example, if you set the scale to -5, one line is displayed every 5 spectra.

Conditions
Measurement views: Spectrogram

Group
Display commands

Syntax
```
DISPlay:SGRam:TIME:SCAle <value>
DISPlay:SGRam:TIME:SCAle?
```

Related Commands
DISPlay:SGRam:TIME:OFFSet

Arguments

<value>::=<NR1> specifies the vertical scale.
Range: -1023 to 0. Zero (0) displays every spectrum.

Examples

`DISPLAY:SGRAM:TIME:SCALE -5` displays one line every 5 spectra in the spectrogram.

DISPlay:SPECtrum:MARKer:NOISe:MODE

Determines whether to enable or disable the marker noise mode in the spectrum view. In this mode, the marker readout indicates amplitude in dBm/Hz. It is valid for all markers except for the reference marker.

NOTE. *To use the marker noise mode, select dBm as the power unit by the [SENSe]:POWer:UNITs command.*

Conditions

Measurement views: Spectrum

Group

Display commands

Syntax

```
DISPlay:SPECtrum:MARKer:NOISe:MODE { OFF | ON | 0 | 1 }
DISPlay:SPECtrum:MARKer:NOISe:MODE?
```

Related Commands

[SENSe]:POWer:UNITs

Arguments

`OFF` or `0` disables the marker noise mode.

`ON` or `1` enables the marker noise mode.

Examples

`DISPLAY:SPECTRUM:MARKER:NOISE:MODE ON` enables the marker noise mode.

DISPlay:SPECtrum:FREQuency:AUTO (No Query Form)

Rescales the horizontal axis automatically to fit the waveform to the screen in the spectrum view.

Conditions

Measurement views: Spectrum

Group

Display commands

Syntax `DISPlay:SPECtrum:FREQuency:AUTO`

Arguments None

Examples `DISPLAY:SPECTRUM:FREQUENCY:AUTO` rescales the horizontal scale automatically to fit the waveform to the screen.

DISPlay:SPECtrum:FREQuency:OFFSet

Sets or queries the frequency offset (the value at the center of the horizontal axis) in the spectrum graph.

Conditions Measurement views: Spectrum

Group Display commands

Syntax
```
DISPlay:SPECtrum:FREQuency:OFFSet <value>
DISPlay:SPECtrum:FREQuency:OFFSet?
```

Related Commands DISPlay:SPECtrum:FREQuency[:SCAle]

Arguments `<value>::=<NRf>` specifies the frequency offset.
Range: [(center frequency) - (X scale) × 0.9] to [(center frequency) + (X scale) × 0.9]

Examples `DISPLAY:SPECTRUM:FREQUENCY:OFFSET 1.45GHz` sets the frequency offset to 1.45 GHz in the spectrum.

DISPlay:SPECtrum:FREQuency[:SCAle]

Sets or queries the horizontal range of the spectrum graph.

Conditions Measurement views: Spectrum

Group Display commands

Syntax
```
DISPlay:SPECtrum:FREQuency[:SCAle] <value>
DISPlay:SPECtrum:FREQuency[:SCAle]?
```

Related Commands DISPlay:SPECtrum:FREQuency:OFFSet

Arguments `<value>::=<NRf>` specifies the horizontal range.
Range: 10 Hz to 6.2 GHz (RSA6106A) / 14 GHz (RSA6114A).

Examples `DISPLAY:SPECTRUM:FREQUENCY:SCALE 10MHz` sets the horizontal range to 10 MHz.

DISPlay:SPECtrum:SCALe:LOG:STATe

Determines whether or not to set the horizontal axis logarithmic in the Spectrum view.

Conditions Measurement views: Spectrum

Group Display commands

Syntax
```
DISPlay:SPECtrum:SCALe:LOG:STATe { OFF | ON | 0 | 1 }
DISPlay:SPECtrum:SCALe:LOG:STATe?
```

Related Commands DISPlay:SPECtrum:X:LABel

Arguments `OFF` or `0` sets the horizontal axis linear (default).

`ON` or `1` sets the horizontal axis logarithmic.

Executing DISPlay:SPECtrum:SCALe:LOG:STATe ON sets DISPlay:SPECtrum:X:LABel SSFReq.

Examples `DISPLAY:SPECTRUM:SCALE:LOG:STATE ON` sets the horizontal axis logarithmic in the Spectrum view.

DISPlay:SPECtrum:WINDow:TRACe:GRATicule:GRID:STATe

Determines whether to show or hide the graticule grid on the screen.

Conditions Measurement views: Spectrum

Group Display commands

Syntax
```
DISPlay:SPECtrum:WINDow:TRACe:GRATicule:GRID:STATe { OFF |
ON | 0 | 1 }
DISPlay:SPECtrum:WINDow:TRACe:GRATicule:GRID:STATe?
```

Arguments OFF or 0 hides the graticule grid.

ON or 1 shows the graticule grid.

Examples DISPLAY:SPECTRUM:WINDOW:TRACE:GRATICULE:GRID:STATE ON shows the graticule grid on the screen in the spectrum measurement.

DISPlay:SPECtrum:WINDow:TRACe:LEGend:STATe

Determines whether to show or hide the trace legend in the Spectrum view. The legend indicates the trace detection and function on the screen for each displayed spectrum trace.

Conditions Measurement views: Spectrum

Group Display commands

Syntax
```
DISPlay:SPECtrum:WINDow:TRACe:LEGend:STATe { OFF | ON | 0 |
1 }
DISPlay:SPECtrum:WINDow:TRACe:LEGend:STATe?
```

Arguments OFF or 0 hides the trace legend.

ON or 1 shows the trace legend.

Examples DISPLAY:SPECTRUM:WINDOW:TRACE:LEGEND:STATE ON shows the trace legend on the screen in the spectrum measurement.

DISPlay:SPECtrum:X:LABel

Selects or queries the labels for the horizontal axis in the Spectrum view. The labels are indicated right under the spectrum graph on the screen.

Conditions Measurement views: Spectrum

Group Display commands

Syntax
```
DISPlay:SPECtrum:X:LABel { SSFReq | CFSPan }
DISPlay:SPECtrum:X:LABel?
```

Arguments SSFReq sets the labels to the start and stop frequencies.

CFSPan sets the labels to the center frequency and span.

Examples DISPLAY:SPECTRUM:X:LABEL SSFReq sets the labels to the start and stop frequencies for the horizontal azis in the Spectrum view.

DISPlay:SPECtrum:Y[:SCALe]

Sets or queries the vertical range of the spectrum graph.

Conditions Measurement views: Spectrum

Group Display commands

Syntax
```
DISPlay:SPECtrum:Y[:SCALe] <value>
DISPlay:SPECtrum:Y[:SCALe]?
```

Arguments <value>::=<NRf> specifies the vertical range. Range: 0.1 to 200 dB. The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples DISPLAY:SPECTRUM:Y:SCALE 100 sets the vertical range to 100 dB in the Spectrum view.

DISPlay:SPECtrum:Y[:SCALe]:AUTO (No Query Form)

Rescales the vertical axis automatically to fit the waveform to the screen in the Spectrum view.

Conditions Measurement views: Spectrum

Group Display commands

Syntax `DISPlay:SPECtrum:Y[:SCALe]:AUTO`

Arguments None

Examples `DISPLAY:SPECTRUM:Y:SCALE:AUTO` rescales the vertical scale automatically to fit the waveform to the screen.

DISPlay:SPECtrum:Y[:SCALe]:OFFSet

Sets or queries the vertical offset (the value at the top edge of the vertical axis) of the spectrum graph.

Conditions Measurement views: Spectrum

Group Display commands

Syntax
```
DISPlay:SPECtrum:Y[:SCALe]:OFFSet <value>
DISPlay:SPECtrum:Y[:SCALe]:OFFSet?
```

Arguments `<value>::=<NRf>` specifies the vertical offset. Range: -270 to +150 dBm. The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples `DISPLAY:SPECTRUM:Y:SCALE:OFFSET -12.5` sets the vertical offset to -12.5 dBm.

DISPlay:SPECtrum:Y[:SCALe]:PDIVision

Sets or queries the vertical scale (per division) of the spectrum graph.

Conditions Measurement views: Spectrum

Group Display commands

Syntax
```
DISPlay:SPECtrum:Y[:SCALe]:PDIVision <value>
DISPlay:SPECtrum:Y[:SCALe]:PDIVision?
```

Related Commands [SENSe]:POWer:UNITs

Arguments `<value>::=<NRf>` specifies the vertical scale (per division).
Range: 0.01 to 20 dB/div.

Examples `SENSE:SPECTRUM:Y:SCALE:PDIVISION 0.5` sets the vertical scale to 0.5 dB/div.

DISPlay:SPECtrum:Y[:SCALe]:RESet (No Query Form)

Resets the vertical scale of the spectrum graph to the default values:
Vertical offset = Reference level and Vertical scale = 100 dB

Conditions Measurement views: Spectrum

Group Display commands

Syntax `DISPlay:SPECtrum:Y[:SCALe]:RESet`

Arguments None

Examples `DISPLAY:SPECTRUM:Y:SCALE:RESET` resets the vertical scale to the default values in the Spectrum view.

DISPlay:SPURious:MARKer:SHOW:STATe

Determines whether to show or hide the readout for the selected marker in the Spurious view.

Conditions Measurement views: Spurious

Group Display commands

Syntax

```
DISPlay:SPURious:MARKer:SHOW:STATe { OFF | ON | 0 | 1 }
DISPlay:SPURious:MARKer:SHOW:STATe?
```

Arguments

`OFF` or `0` hides the readout for the selected marker in the graph.

`ON` or `1` shows the readout for the selected marker in the graph.

Examples

`DISPLAY:SPURIOUS:MARKER:SHOW:STATE ON` shows the readout for the selected marker in the graph.

DISPlay:SPURious:RESet:SCALe (No Query Form)

Resets the horizontal and vertical scale to the default values described below in the Spurious view.

Vertical offset = 0 dBm,
Vertical scale = 100 dB,
Horizontal offset = Center frequency, and
Horizontal scale = Default span

Conditions

Measurement views: Spurious

Group

Display commands

Syntax

```
DISPlay:SPURious:RESet:SCALe
```

Arguments

None

Examples

`DISPLAY:SPURIOUS:RESET:SCALE` resets the horizontal and vertical scale to the default values.

DISPlay:SPURious:SCALe:LOG:STATe

Determines whether or not to set the horizontal axis logarithmic in the Spurious view.

Conditions

Measurement views: Spurious

Group

Display commands

Syntax

```
DISPlay:SPURious:SCALe:LOG:STATe { OFF | ON | 0 | 1 }
DISPlay:SPURious:SCALe:LOG:STATe?
```

Arguments

OFF or 0 sets the horizontal axis linear (default).

ON or 1 sets the horizontal axis logarithmic.

Examples

DISPLAY:SPURIOUS:SCALE:LOG:STATE ON sets the horizontal axis logarithmic in the Spurious view.

DISPlay:SPURious:SELect:NUMBer

Selects or queries the spurious number in the Spurious view.

Conditions

Measurement views: Spurious

Group

Display commands

Syntax

```
DISPlay:SPURious:SELect:NUMBer <number>
DISPlay:SPURious:SELect:NUMBer?
```

Arguments

<number>::=<NR1> specifies the spurious number.
Range: 1 to the number of spurious signals.

Use the FETCh:SPURious:COUNt? or READ:SPURious:COUNt? query to get the number of spurious signals.

Examples

DISPLAY:SPURIOUS:SELECT:NUMBER 7 selects the spurious #7.

DISPlay:SPURious:SHOW:LIMit

Selects or queries how to display the limits.

Conditions

Measurement views: Spurious

Group

Display commands

Syntax

```
DISPlay:SPURious:SHOW:LIMit { SHADed | LINE | OFF }
DISPlay:SPURious:SHOW:LIMit?
```

Arguments

SHADed displays the limits with shade.

LINE displays the limits with line only.

OFF hides the limits.

Examples

DISPLAY:SPURIOUS:SHOW:LIMIT LINE displays the limits with line only.

DISPlay:SPURious:WINDow:TRACe:GRATicule:GRID:STATe

Determines whether to show or hide the graticule grid on the screen.

Conditions

Measurement views: Spurious

Group

Display commands

Syntax

```
DISPlay:SPURious:WINDow:TRACe:GRATicule:GRID:STATe { OFF |
ON | 0 | 1 }
DISPlay:SPURious:WINDow:TRACe:GRATicule:GRID:STATe?
```

Arguments

OFF or 0 hides the graticule grid.

ON or 1 shows the graticule grid.

Examples

DISPLAY:SPURIOUS:WINDOW:TRACE:GRATICULE:GRID:STATE ON shows the graticule grid on the screen.

DISPlay:SPURious:X[:SCALe]:AUTO (No Query Form)

Rescales the horizontal axis automatically to fit the waveform to the screen in the Spurious view.

Conditions

Measurement views: Spurious

Group

Display commands

Syntax

```
DISPlay:SPURious:X[:SCALe]:AUTO
```

Arguments

None

Examples `DISPLAY:SPURIOUS:X:SCALE:AUTO` rescales the horizontal scale automatically to fit the waveform to the screen.

DISPlay:SPURious:X[:SCALe]:STARt

Sets or queries the minimum horizontal value (left edge) of the spectrum graph in the Spurious view.

Conditions Measurement views: Spurious

Group Display commands

Syntax
```
DISPlay:SPURious:X[:SCALe]:STARt <value>
DISPlay:SPURious:X[:SCALe]:STARt?
```

Arguments `<value>::=<NRf>` specifies the minimum horizontal value.
Range: 0 Hz to 6.2 GHz (RSA6106A) / 14 GHz (RSA6114A).

Examples `DISPLAY:SPURIOUS:X:SCALE:START 1.61GHz` sets the minimum horizontal value to 1.61 GHz in the spectrum graph.

DISPlay:SPURious:X[:SCALe]:STOP

Sets or queries the maximum horizontal value (right edge) of the spectrum graph in the Spurious view.

Conditions Measurement views: Spurious

Group Display commands

Syntax
```
DISPlay:SPURious:X[:SCALe]:STOP <value>
DISPlay:SPURious:X[:SCALe]:STOP?
```

Arguments `<value>::=<NRf>` specifies the minimum horizontal value.
Range: 0 Hz to 6.2 GHz (RSA6106A) / 14 GHz (RSA6114A).

Examples `DISPLAY:SPURIOUS:X:SCALE:STOP 2.16GHz` sets the maximum horizontal value to 2.16 GHz in the spectrum graph.

DISPlay:SPURious:Y[:SCALe]

Sets or queries the vertical range of the spectrum graph in the Spurious view.

Conditions Measurement views: Spurious

Group Display commands

Syntax

```
DISPlay:SPURious:Y[:SCALe] <value>
DISPlay:SPURious:Y[:SCALe]?
```

Arguments `<value>::=<NRf>` specifies the vertical range. Range: 0.1 to 200 dB.
The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples `DISPLAY:SPURIOUS:Y:SCALE 100` sets the vertical range to 100 dB in the Spurious view.

DISPlay:SPURious:Y[:SCALe]:AUTO (No Query Form)

Rescales the vertical axis automatically to fit the waveform to the screen in the Spurious view.

Conditions Measurement views: Spurious

Group Display commands

Syntax

```
DISPlay:SPURious:Y[:SCALe]:AUTO
```

Arguments None

Examples `DISPLAY:SPURIOUS:Y:SCALE:AUTO` rescales the vertical scale automatically to fit the waveform to the screen.

DISPlay:SPURious:Y[:SCALe]:OFFSet

Sets or queries the vertical offset (the value at the top edge of the vertical axis) of the spectrum graph in the Spurious view.

Conditions Measurement views: Spurious

Group Display commands

Syntax
```
DISPlay:SPURious:Y[:SCALe]:OFFSet <value>
DISPlay:SPURious:Y[:SCALe]:OFFSet?
```

Arguments `<value>::=<NRf>` specifies the vertical offset. Range: -270 to +150 dBm. The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples `DISPLAY:SPURIOUS:Y:SCALE:OFFSET -12.5` sets the vertical offset to -12.5 dBm in the spectrum graph.

DISPlay:TOVerview:WINDow:TRACe:GRATicule:GRID:STATe

Determines whether to show or hide the graticule grid on the screen.

Conditions Measurement views: Time overview

Group Display commands

Syntax
```
DISPlay:TOVerview:WINDow:TRACe:GRATicule:GRID:STATe { OFF |
ON | 0 | 1 }
DISPlay:TOVerview:WINDow:TRACe:GRATicule:GRID:STATe?
```

Arguments `OFF` or `0` hides the graticule grid.

`ON` or `1` shows the graticule grid.

Examples `DISPLAY:TOVERVIEW:WINDOW:TRACE:GRATICULE:GRID:STATE ON` shows the graticule grid on the screen in the time overview.

DISPlay:TOVerview:X[:SCALe]

Sets or queries the horizontal scale (full-scale time) of the time overview.

Conditions Measurement views: Time overview

Group Display commands

Syntax

```
DISPlay:TOVerview:X[:SCALe] <value>
DISPlay:TOVerview:X[:SCALe]?
```

Related Commands DISPlay:TOVerview:X[:SCALe]:OFFSet

Arguments `<value>::=<NRf>` specifies the horizontal scale in full-scale time.
Range: 10 ns to the acquisition memory capacity.

You can see the acquisition memory capacity using the [SENSe]:ACQuisition:MEMory:CAPacity[:TIME]? query.

Examples `DISPLAY:TOVERVIEW:X:SCALE 12.5us` sets the horizontal scale to 12.5 μs.

DISPlay:TOVerview:X[:SCALe]:AUTO (No Query Form)

Sets the horizontal scale and offset automatically to fit the waveform to the screen in the time overview.

Conditions Measurement views: Time overview

Group Display commands

Syntax

```
DISPlay:TOVerview:X[:SCALe]:AUTO
```

Arguments None

Examples `DISPLAY:TOVERVIEW:X:SCALE:AUTO` sets the horizontal scale and offset automatically to fit the waveform to the screen.

DISPlay:TOVerview:X[:SCALe]:OFFSet

Sets or queries the minimum horizontal value (left edge) of the time overview.

Conditions Measurement views: Time overview

Group Display commands

Syntax
```
DISPlay:TOVerview:X[:SCALe]:OFFSet <value>
DISPlay:TOVerview:X[:SCALe]:OFFSet?
```

Related Commands DISPlay:TOVerview:X[:SCALe]

Arguments `<value>::=<NRf>` specifies the minimum horizontal value.
Range: [(analysis offset) - (X scale) × 0.9] to [(analysis offset) + (analysis length) - (X scale) × 0.1]

Examples `DISPLAY:TOVERVIEW:X:SCALE:OFFSET 800ns` sets the minimum horizontal value to 800 ns in the time overview.

DISPlay:TOVerview:Y[:SCALe]

Sets or queries the vertical range of the time overview.

Conditions Measurement views: Time overview

Group Display commands

Syntax
```
DISPlay:TOVerview:Y[:SCALe] <value>
DISPlay:TOVerview:Y[:SCALe]?
```

Related Commands DISPlay:TOVerview:Y[:SCALe]:OFFSet

Arguments `<value>::=<NRf>` specifies the vertical range. Range: 0.1 to 200 dB.

Examples `DISPLAY:TOVERVIEW:Y:SCALE 50` sets the vertical range to 50 dBm in the time overview.

DISPlay:TOVerview:Y[:SCALe]:AUTO (No Query Form)

Sets the vertical scale and offset automatically to fit the waveform to the screen in the time overview.

Conditions Measurement views: Time overview

Group Display commands

Syntax
```
DISPlay:TOVerview:Y[:SCALe]:AUTO
```

Arguments None

Examples `DISPLAY:TOVERVIEW:Y:SCALE:AUTO` sets the vertical scale and offset automatically to fit the waveform to the screen.

DISPlay:TOVerview:Y[:SCALe]:OFFSet

Sets or queries the vertical offset (the value at the top edge of the vertical axis) in the time overview.

Conditions Measurement views: Time overview

Group Display commands

Syntax
```
DISPlay:TOVerview:Y[:SCALe]:OFFSet <value>
DISPlay:TOVerview:Y[:SCALe]:OFFSet?
```

Related Commands DISPlay:TOVerview:Y[:SCALe]

Arguments `<value>::=<NRf>` specifies the vertical offset. Range: -170 to +50 dBm.

Examples `DISPLAY:TOVERVIEW:Y:SCALE:OFFSET -80` sets the vertical offset to -80 dBm in the time overview.

DISPlay:TOVerview:Y[:SCALe]:RESCale (No Query Form)

Sets the vertical scale automatically to fit the waveform to the screen in the time overview.

Conditions Measurement views: Time overview

Group Display commands

Syntax `DISPlay:TOVerview:Y[:SCALe]:RESCale`

Arguments None

Examples `DISPLAY:TOVERVIEW:Y:SCALE:RESCALE` sets the vertical scale automatically to fit the waveform to the screen.

DISPlay:WINDow:ACTive:MEASurement? (Query Only)

Queries the active measurement views.

Conditions Measurement views: All

Group Display commands

Syntax `DISPlay:WINDow:ACTive:MEASurement?`

Arguments None

Returns `<view1>,<view2>,...,<view(n)>`

Where
`<view(n)>::=<string>` is the view name as shown in the following table.

Table 2-31: Measurement view mnemonic

Return value	Measurement view	Display group
“SPEC	Spectrum	General signal viewing
“DPSA”	DPX (Digital Phosphor) spectrum	
“MAGVT”	Amplitude versus Time	
“FVT”	Frequency versus Time	
“PHVT”	Phase versus Time	
“IQVT”	RF I&Q versus Time	
“SGRam”	Spectrogram	
“TOV”	Time overview	
“CONS”	Constellation	General purpose digital modulation
“EVM”	EVM versus Time	
“MERR”	Magnitude error versus Time	
“PERR”	Phase error versus Time	
“SIGN”	Signal quality	
“STAB”	Symbol table	
“CCDF”	CCDF	RF measurements
“ACP”	Channel power and ACPR	
“MCP”	MCPR (Multiple Carrier Power Ratio)	
“OBW”	Occupied bandwidth	
“PNO”	Phase noise	
“SPUR”	Spurious	
“STAT”	Pulse statistics	Pulsed RF
“RES”	Pulse table (results table)	
“TRAC”	Pulse trace	

Examples `DISPLAY:WINDOW:ACTIVE:MEASUREMENT?` might return `"SPEC","TRAC"`, indicating that the views of spectrum and pulse trace are displayed on the screen.

DISPlay:WINDow:COLor:SCHeme

Selects or queries the color scheme for displaying traces and background on the screen.

Conditions Measurement views: All

Group Display commands

Syntax
```
DISPlay:WINDow:COLor:SCHeme { THUNderstorm | BLIZzard |
CLASsic }
DISPlay:WINDow:COLor:SCHeme?
```

Arguments `THUNderstorm` displays the backgound in dark blue.

`BLIZzard` displays the backgound in white. It saves ink when printing the screen image.

`CLASsic` displays the backgound in black (default).

Examples `DISPLAY:WINDOW:COLOR:SCHEME BLIZzard` displays the backgound in white.

DISPlay:WINDow:OPTimized:MEASurement? (Query Only)

Queries the measurement views that are optimized. "Optimized" means that there is a perfect match between the view's settings and the actual acquisition parameters to meet the specifications. When multiple measurements are running at one time, the measurements can have different requirements for setting the acquisition hardware. You can make a measurement optimized by selecting it using the following commands:

- DISPlay:GENeral:MEASview:SELect for the general signal viewing
- DISPlay:DDEMod:MEASview:SELect for the digital modulation views
- DISPlay:GPRF:MEASview:SELect for the RF measurement views
- DISPlay:PULSe:MEASview:SELect for the pulsed RF measurement views

Conditions Measurement views: All

Group Display commands

Syntax `DISPlay:WINDow:OPTimized:MEASurement?`

Arguments None

Returns `<view1>,<view2>,...,<view(n)>`

Where

`<view(n)>::=<string>` is the view name as shown in the table. (See Table 2-31 on page 2-286.)

Examples

`DISPLAY:WINDOW:OPTIMIZED:MEASUREMENT?` might return `"SPEC","MCP"`, indicating that the views of spectrum and MCPR are optimized.

*ESE

Sets or queries the bits in the Event Status Enable Register (ESER). The ESER prevents events from being reported to the Status Byte Register (STB). Refer to Section 3, *Status and Events*, for the register information.

Conditions

Measurement views: All

Group

IEEE common commands

Syntax

```
*ESE <value>
*ESE?
```

Related Commands

*CLS, *ESR?, *SRE, *STB?

Arguments

`<value>::=<NR1>` is a value in the range from 0 through 255.
The binary bits of the ESER are set according to this value.

Examples

`*ESE 145` sets the ESER to binary 10010001, which enables the PON, EXE, and OPC bits.

`*ESE?` might return the string `*ESE 184`, showing that the ESER contains the binary value 10111000.

*ESR? (Query Only)

Returns the contents of the Standard Event Status Register (SESR). *ESR? also clears the SESR (since reading the SESR clears it). Refer to Section 3, *Status and Events*, for the register information.

Conditions

Measurement views: All

Group

IEEE common commands

Syntax *ESR?

Related Commands *CLS, *ESE, *SRE, *STB?

Arguments None

Returns <NR1> representing the contents of the SESR by a 0 to 255 decimal number.

Examples *ESR? might return the value 213, showing that the SESR contains binary 11010101.

FETCh:ACPower? (Query Only)

Returns the Channel power and ACPR measurement results for all available channels.

Conditions Measurement views: Channel power and ACPR

Group Fetch commands

Syntax FETCh:ACPower?

Arguments None

Returns <chan_power>,<acpr_lower(1)>,<acpr_upper(1)>,
<acpr_lower(2)>,<acpr_upper(2)>,...
<acpr_lower(n)>,<acpr_upper(n)>

Where
<chan_power> is the average power of the main channel as the power reference in dBm. The unit can be changed by the [SENSe]:POWer:UNITs command.
<acpr_lower(n)> is the ACPR for the lower channel #n in dB.
<acpr_upper(n)> is the ACPR for the upper channel #n in dB.

The number of n depends on the setting of the [SENSe]:ACPower:CHANnel:PAIRs command.

Examples FETCH:ACPOWER? might return
4.227,-28.420,-23.847,-22.316,-29.225,
indicating

(average power of the main channel) = 4.227 dBm,
(ACPR for the lower channel 1) = -28.420 dB,
(ACPR for the upper channel 1) = -23.847 dB,
(ACPR for the lower channel 2) = -22.316 dB, and
(ACPR for the upper channel 2) = -29.225 dB.

FETCh:ACPower:CHANnel:POWer? (Query Only)

Returns the average power of the main channel (power reference) in the Channel power and ACPR measurement.

Conditions Measurement views: Channel power and ACPR

Group Fetch commands

Syntax `FETCh:ACPower:CHANnel:POWer?`

Arguments None

Returns `<chan_power>::=<NRf>` is the average power of the main channel in dBm. The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples `FETCH:ACPOWER:CHANNEL:POWER?` might return `4.227`, indicating that the average power of the main channel is 4.227 dBm.

FETCh:ACPower:SPECtrum? (Query Only)

Returns spectrum trace data of the Channel power and ACPR measurement.

Conditions Measurement views: Channel power and ACPR

Group Fetch commands

Syntax `FETCh:ACPower:SPECtrum?`

Arguments None

Returns `#<num_digit><num_byte><data(1)><data(2)>...<data(n)>`

Where
`<num_digit>` is the number of digits in <num_byte>.
`<num_byte>` is the number of bytes of data that follow.
`<data(n)>` is the amplitude in dBm for the point #n,
4-byte little endian floating-point format specified in IEEE 488.2.
The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples `FETCH:ACPOWER:SPECTRUM?` might return `#43204xxxx...` (3204-byte data) for the spectrum trace data of the Channel power and ACPR measurement.

FETCh:AVTime:AVERage? (Query Only)

Returns the RMS (root-mean-square) value for the selected trace in the Amplitude versus Time measurement. Select the trace using the TRACe<x>:AVTime:SELect command.

Conditions Measurement views: Amplitude versus Time

Group Fetch commands

Syntax `FETCh:AVTime:AVERage?`

Arguments None

Returns `<avg>::=<NRf>` is the RMS amplitude in dBm.
The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples `FETCH:AVTIME:AVERAGE?` might return `-2.53`, indicating the RMS amplitude is -2.53 dBm.

FETCh:AVTime:{FIRSt|SECond|THIRd|FOURth}? (Query Only)

Returns the trace data in the Amplitude versus Time measurement.

The mnemonics FIRSt, SECond, THIRd, and FOURth represent Trace 1, Trace 2, Trace 3, and Math trace, respectively. The traces can be specified by the TRACe<x>:AVTime command subgroup.

Conditions Measurement views: Amplitude versus Time

Group Fetch commands

Syntax `FETCh:AVTime:{FIRSt|SECond|THIRd|FOURth}?`

Arguments None

Returns `#<num_digit><num_byte><data(1)><data(2)>...<data(n)>`

Where
<num_digit> is the number of digits in <num_byte>.
<num_byte> is the number of bytes of data that follow.
<data(n)> is the amplitude in dBm for the point #n,
4-byte little endian floating-point format specified in IEEE 488.2.
The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples `FETCH:AVTIME:FIRST?` might return `#3156xxxx...` (156-byte data) for Trace 1.

FETCh:AVTime:MAXimum? (Query Only)

Returns the maximum value for the selected trace in the Amplitude versus Time measurement. Select the trace using the TRACe<x>:AVTime:SELect command.

Conditions Measurement views: Amplitude versus Time

Group Fetch commands

Syntax `FETCh:AVTime:MAXimum?`

Related Commands FETCh:AVTime:MAXLocation?

Arguments None

Returns `<max>::=<NRf>` is the maximum Amplitude in dBm.
The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples FETCH:AVTIME:MAXIMUM? might return -2.84, indicating the maximum amplitude is -2.84 dBm.

FETCh:AVTime:MAXLocation? (Query Only)

Returns the time at which the amplitude is maximum for the selected trace in the Amplitude versus Time measurement. Select the trace using the TRACe<x>:AVTime:SELect command.

Conditions Measurement views: Amplitude versus Time

Group Fetch commands

Syntax `FETCh:AVTime:MAXLocation?`

Related Commands FETCh:AVTime:MAXimum?

Arguments None

Returns `<max_time>::=<NRf>` is the time at the maximum in seconds.

Examples FETCH:AVTIME:MAXLOCATION? might return 25.03E-9, indicating the amplitude is maximum at 25.03 ns.

FETCh:AVTime:MINimum? (Query Only)

Returns the minimum value for the selected trace in the Amplitude versus Time measurement. Select the trace using the TRACe<x>:AVTime:SELect command.

Conditions Measurement views: Amplitude versus Time

Group Fetch commands

Syntax `FETCh:AVTime:MINimum?`

Related Commands FETCh:AVTime:MINLocation?

Arguments None

Returns `<min>::=<NRf>` is the minimum amplitude in dBm.
The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples `FETCH:AVTIME:MINIMUM?` might return `-57.64`, indicating the minimum amplitude is -57.64 dBm.

FETCh:AVTime:MINLocation? (Query Only)

Returns the time at which the amplitude is minimum for the selected trace in the Amplitude versus Time measurement. Select the trace using the TRACe<x>:AVTime:SELect command.

Conditions Measurement views: Amplitude versus Time

Group Fetch commands

Syntax `FETCh:AVTime:MINLocation?`

Related Commands FETCh:AVTime:MINimum?

Arguments None

Returns `<min_time>::=<NRf>` is the time at the minimum in seconds.

Examples `FETCH:AVTIME:MINLOCATION?` might return `450.7E-9`, indicating the amplitude is minimum at 450.7 ns.

FETCh:AVTime:RESult? (Query Only)

Returns the measurement results for the selected trace in the Amplitude versus Time measurement. Select the trace using the TRACe<x>:AVTime:SELect command.

Conditions Measurement views: Amplitude versus Time

Group Fetch commands

Syntax FETCh:AVTime:RESult?

Arguments None

Returns <max>,<max_time>,<min>,<min_time>,<rms>

Where
<max>::=<NRf> is the maximum amplitude in dBm.
<max_time>::=<NRf> is the time at the maximum in seconds.
<min>::=<NRf> is the minimum amplitude in dBm.
<min_time>::=<NRf> is the time at the minimum in seconds.
<rms>::=<NRf> is the RMS amplitude in dBm.
The unit of amplitude can be changed by the [SENSe]:POWer:UNITs command.

Examples FETCH:AVTIME:RESULT? might return
-2.68,48.62E-6,-82.47,22.11E-6,-8.24, indicating
that
the maximum amplitude is -2.68 dBm at 48.62 µs,
the minimum amplitude is -82.47 dBm at 22.11 µs, and
the RMS amplitude is -8.24 dBm.

FETCh:CCDF? (Query Only)

Returns the CCDF measurement results.

Conditions Measurement views: CCDF

Group Fetch commands

Syntax FETCh:CCDF?

Related Commands

Arguments None

Returns <avg_ampl>,<avg_ccdf>,<crest_factor>,<ampl_10>,<ampl_1>,
<ampl_p1>,<ampl_p01>,<ampl_p001>,<ampl_p0001>

Where
<avg_ampl> is the average amplitude in dBm.
The unit can be changed by the [SENSe]:POWer:UNITs command.
<avg_ccdf> is the average CCDF in percent.
<crest_factor> is the crest factor in dB.
<ampl_10> is the amplitude at CCDF of 10% in dB.
<ampl_1> is the amplitude at CCDF of 1% in dB.
<ampl_p1> is the amplitude at CCDF of 0.1% in dB.
<ampl_p01> is the amplitude at CCDF of 0.01% in dB.
<ampl_p001> is the amplitude at CCDF of 0.001% in dB.
<ampl_p0001> is the amplitude at CCDF of 0.0001% in dB.

Examples

FETCH:CCDF? might return
-33.35,35.8,9.75,3.88,7.07,8.50,9.25,9.72,9.74, indicating
(average amplitude) = -33.35 dBm,
(average CCDF) = 35.8%,
(crest factor) = 9.75 dB,
(amplitude at CCDF of 10%) = 3.88 dB,
(amplitude at CCDF of 1%) = 7.07 dB,
(amplitude at CCDF of 0.1%) = 8.50 dB,
(amplitude at CCDF of 0.01%) = 9.25 dB,
(amplitude at CCDF of 0.001%) = 9.72 dB, and
(amplitude at CCDF of 0.0001%) = 9.74 dB.

FETCh:CCDF:{FIRSt|SECond|THIRd}:X? (Query Only)

Returns the horizontal values of the specified trace in the CCDF measurement.

The mnemonics FIRSt, SECond, and THIRd represent Trace 1, Trace 2, and Gaussian reference curve, respectively.

Conditions

Measurement views: CCDF

Group

Fetch commands

Syntax

FETCh:CCDF:{FIRSt|SECond|THIRd}:X?

Arguments

None

Returns

#<num_digit><num_byte><x(1)><x(2)>...<x(n)>

Where
<num_digit> is the number of digits in <num_byte>.

<num_byte> is the number of bytes of data that follow.
<x(n)> is the horizontal value (dB) of the CCDF graph at the point #n,
4-byte little endian floating-point format specified in IEEE 488.2.

Examples `FETCH:CCDF:FIRST:X?` might return `#41024xxxx...` (1024-byte data) for the horizontal values of Trace 1.

FETCh:CCDF:{FIRSt|SECond|THIRd}:XY? (Query Only)

Returns the horizontal and vertical value pairs of the specified trace in the CCDF measurement.

The mnemonics FIRSt, SECond, and THIRd represent Trace 1, Trace 2, and Gaussian reference curve, respectively.

Conditions Measurement views: CCDF

Group Fetch commands

Syntax `FETCh:CCDF:{FIRSt|SECond|THIRd}:XY?`

Arguments None

Returns `#<num_digit><num_byte><x(1)><y(1)><x(2)><y(2)>...<x(n)><y(n)>`

Where
<num_digit> is the number of digits in <num_byte>.
<num_byte> is the number of bytes of data that follow.
<x(n)><y(n)> is the horizontal value (dB) and vertical value (%) pair for the point #n,
4-byte little endian floating-point format specified in IEEE 488.2.

Examples `FETCH:CCDF:FIRST:XY?` might return `#41024xxxx...` (1024-byte data) for the horizontal and vertical value pairs of Trace 1.

FETCh:CCDF:{FIRSt|SECond|THIRd}:XY? (Query Only)

Returns the horizontal and vertical value pairs of the specified trace in the CCDF measurement.

The mnemonics FIRSt, SECond, and THIRd represent Trace 1, Trace 2, and Gaussian reference curve, respectively.

***NOTE.** This query is invalid when [SENSe]:CCDF:TIME:TYPE is set to CONTinuous or TOTal.*

Conditions Measurement views: CCDF

Group Read commands

Syntax `FETCh:CCDF:{FIRSt|SECond|THIRd}:XY?`

Arguments None

Returns `#<num_digit><num_byte><x(1)><y(1)><x(2)><y(2)>... <x(n)><y(n)>`

Where
`<num_digit>` is the number of digits in <num_byte>.
`<num_byte>` is the number of bytes of data that follow.
`<x(n)><y(n)>` is the horizontal value (dB) and vertical value (%) pair for the point #n,
4-byte little endian floating-point format specified in IEEE 488.2.

Examples `READ:CCDF:FIRST:XY?` might return `#41024xxxx...` (1024-byte data) for the horizontal and vertical value pairs of Trace 1.

FETCh:CCDF:{FIRSt|SECond|THIRd}[:Y]? (Query Only)

Returns the vertical values of the specified trace in the CCDF measurement.

The mnemonics FIRSt, SECond, and THIRd represent Trace 1, Trace 2, and Gaussian reference curve, respectively.

Conditions Measurement views: CCDF

Group Fetch commands

Syntax `FETCh:CCDF:{FIRSt|SECond|THIRd}[:Y]?`

Arguments None

Returns `#<num_digit><num_byte><y(1)><y(2)>...<y(n)>`

Where
`<num_digit>` is the number of digits in <num_byte>.
`<num_byte>` is the number of bytes of data that follow.
`<y(n)>` is the vertical value (%) of the CCDF graph at the point #n,
4-byte little endian floating-point format specified in IEEE 488.2.

Examples `FETCH:CCDF:FIRST:Y?` might return `#41024xxxx...` (1024-byte data) for the vertical values of Trace 1.

FETCh:CONSte:RESults? (Query Only)

Returns the constellation measurement results of EVM RMS, peak and location displayed on the bottom of the screen.

Conditions Measurement views: Constellation

Group Fetch commands

Syntax `FETCh:CONSte:RESults?`

Arguments None

Returns `<EVM_RMS>,<EVM_peak>,<location>`

Where
`<EVM_RMS>::=<NRf>` is the RMS EVM in percent (%).
`<EVM_peak>::=<NRf>` is the peak EVM in percent (%).
`<location>::=<NRf>` is the peak location in symbol number.
The time unit can be changed by the [SENSe]:DDEMod:TIMe:UNITs command.

Examples `FETCH:CONSTE:RESULTS?` might return `2.841,3.227,68.000`, indicating that the RMS EVM is 2.841% and the peak EVM is 3.227% at symbol #68.

FETCh:CONSte:TRACe? (Query Only)

Returns the constellation trace data.

Conditions Measurement views: Constellation

Group Fetch commands

Syntax FETCh:CONSte:TRACe?

Arguments None

Returns #<num_digit><num_byte><I(1)><Q(1)><I(2)><Q(2)>...
<I(n)><Q(n)>

Where
<num_digit> is the number of digits in <num_byte>.
<num_byte> is the number of bytes of data that follow.
<I(n)> and <Q(n)> are the normalized I- and Q-coordinate values at the n^{th} data point. 4-byte little endian floating-point format specified in IEEE 488.2.

Examples FETCH:CONSTE:TRACE? might return #43848xxxx... (3848-byte data) for the constellation trace data.

FETCh:DDEMod:STABle? (Query Only)

Returns the symbol table data.

Conditions Measurement views: Symbol table

Group Fetch commands

Syntax FETCh:DDEMod:STABle?

Arguments None

Returns #<num_digit><num_byte><data(1)><data(2)>...<data(n)>

Where
<num_digit> is the number of digits in <num_byte>.
<num_byte> is the number of bytes of data that follow.
<data(n)> is the symbol table data for the point #n,
4-byte little endian floating-point format specified in IEEE 488.2.

Examples `FETCH:DDEMOD:STABLE?` might return `#3512xxxx...` (512-byte data) for the symbol table.

FETCh:DDEMod:SYNCh:WORD:LENGth? (Query Only)

Returns the length of the synch word in the symbol table.

Conditions Measurement views: Symbol table

Group Fetch commands

Syntax `FETCh:DDEMod:SYNCh:WORD:LENGth?`

Related Commands FETCh:DDEMod:SYNCh:WORD:POSition?

Arguments None

Returns `<NR1>` indicates the length of the synch word in symbols.

Examples `FETCH:DDEMOD:SYNCH:WORD:LENGTH?` might return `3`, indicating the length of the synch word is three symbols.

FETCh:DDEMod:SYNCh:WORD:POSition? (Query Only)

Returns the position of the synch word in the symbol table.

Conditions Measurement views: Symbol table

Group Fetch commands

Syntax `FETCh:DDEMod:SYNCh:WORD:POSition?`

Related Commands FETCh:DDEMod:SYNCh:WORD:LENGth?

Arguments None

Returns `<NR1>` indicates what symbol number the synch word begins at in the table. Zero (0) represents the first symbol in the table.

Examples `FETCH:DDEMOD:SYNCH:WORD:POSITION?` might return `10`, indicating the synch word begins at 11th symbol in the table.

FETCh:DPSA:TRACe:AVERage? (Query Only)

Returns waveform data of the average trace in the DPX spectrum measurement.

Conditions Measurement views: DPX spectrum

Group Fetch commands

Syntax `FETCh:DPSA:TRACe:AVERage?`

Arguments None

Returns `#<num_digit><num_byte><data(1)><data(2)>...<data(n)>`

Where
`<num_digit>` is the number of digits in <num_byte>.
`<num_byte>` is the number of bytes of data that follow.
`<data(n)>` is the amplitude of the average trace for the point #n in dBm, 4-byte little endian floating-point format specified in IEEE 488.2.
The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples `FETCH:DPSA:TRACE:AVERAGE?` might return `#42004xxxx...` (2004-byte data) for the waveform data of the average trace.

FETCh:DPSA:TRACe:MATH? (Query Only)

Returns waveform data of the math trace in the DPX spectrum measurement.

Conditions Measurement views: DPX spectrum

Group Fetch commands

Syntax `FETCh:DPSA:TRACe:MATH?`

Arguments None

Returns `#<num_digit><num_byte><data(1)><data(2)>...<data(n)>`

Where
`<num_digit>` is the number of digits in <num_byte>.
`<num_byte>` is the number of bytes of data that follow.
`<data(n)>` is the amplitude of the math trace for the point #n in dBm, 4-byte little endian floating-point format specified in IEEE 488.2.
The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples `FETCH:DPSA:TRACE:MATH?` might return `#42004xxxx...` (2004-byte data) for the waveform data of the math trace.

FETCh:DPSA:TRACe:MAXimum? (Query Only)

Returns waveform data of the maximum trace in the DPX spectrum measurement.

Conditions Measurement views: DPX spectrum

Group Fetch commands

Syntax `FETCh:DPSA:TRACe:MAXimum?`

Arguments None

Returns `#<num_digit><num_byte><data(1)><data(2)>...<data(n)>`

Where
`<num_digit>` is the number of digits in <num_byte>.
`<num_byte>` is the number of bytes of data that follow.
`<data(n)>` is the amplitude of the maximum trace for the point #n in dBm, 4-byte little endian floating-point format specified in IEEE 488.2.
The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples `FETCH:DPSA:TRACE:MAXIMUM?` might return `#42004xxxx...` (2004-byte data) for the waveform data of the maximum trace.

FETCh:DPSA:TRACe:MINimum? (Query Only)

Returns waveform data of the minimum trace in the DPX spectrum measurement.

Conditions Measurement views: DPX spectrum

Group Fetch commands

Syntax `FETCh:DPSA:TRACe:MINimum?`

Arguments None

Returns `#<num_digit><num_byte><data(1)><data(2)>...<data(n)>`

Where
`<num_digit>` is the number of digits in <num_byte>.
`<num_byte>` is the number of bytes of data that follow.
`<data(n)>` is the amplitude data of the minimum trace for the point #n in dBm, 4-byte little endian floating-point format specified in IEEE 488.2.
The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples `FETCH:DPSA:TRACE:MINIMUM?` might return `#42004xxxx...` (2004-byte data) for the waveform data of the minimum trace.

FETCh:EVM:FERRor? (Query Only)

Returns the frequency error in the EVM versus Time measurement.

Conditions Measurement views: EVM versus Time

Group Fetch commands

Syntax `FETCh:EVM:FERRor?`

Arguments None

Returns `<freq_error>::=<NRf>` is the frequency error in Hz.

Examples `FETCH:EVM:FERROR?` might return `-10.7E+3`, indicating the frequency error is -10.7 kHz.

FETCh:EVM:PEAK? (Query Only)

Returns the peak value in the EVM versus Time measurement.

Conditions Measurement views: EVM versus Time

Group Fetch commands

Syntax `FETCh:EVM:PEAK?`

Related Commands FETCh:EVM:PINDex?

Arguments None

Returns `<peak>::=<NRf>` is the peak EVM value in percent (%).

Examples `FETCH:EVM:PEAK?` might return `1.32`, indicating the peak EVM value is 1.32%.

FETCh:EVM:PINDex? (Query Only)

Returns the time at the EVM peak.

Conditions Measurement views: EVM versus Time

Group Fetch commands

Syntax `FETCh:EVM:PINDex?`

Related Commands FETCh:EVM:PEAK?

Arguments None

Returns `<peak_time>::=<NRf>` is the time at the EVM peak in symbol number. The unit can be changed by the [SENSe]:DDEMod:TIMe:UNITs command.

Examples `FETCH:EVM:PINDEX?` might return `68.000`, indicating that the EVM peak is at symbol #68.

FETCh:EVM:RMS? (Query Only)

Returns the RMS (Root-Mean-Square) value in the EVM versus Time measurement.

Conditions Measurement views: EVM versus Time

Group Fetch commands

Syntax `FETCh:EVM:RMS?`

Arguments None

Returns `<rms>::=<NRf>` is the RMS EVM value in percent (%).

Examples `FETCH:EVM:RMS?` might return `0.582`, indicating the RMS EVM value is 0.582%.

FETCh:EVM:TRACe? (Query Only)

Returns the EVM versus Time trace data.

Conditions Measurement views: EVM versus Time

Group Fetch commands

Syntax `FETCh:EVM:TRACe?`

Arguments None

Returns `#<num_digit><num_byte><data(1)><data(2)>...<data(n)>`

Where
`<num_digit>` is the number of digits in <num_byte>.
`<num_byte>` is the number of bytes of data that follow.
`<data(n)>` is the EVM value for the point #n in percent (%),
4-byte little endian floating-point format specified in IEEE 488.2.

Examples `FETCH:EVM:TRACE?` might return `#42036xxxx...` (2036-byte data) for the EVM versus Time trace.

FETCh:FVTime? (Query Only)

Returns the Frequency versus Time trace data.

Conditions Measurement views: Frequency versus Time

Group Fetch commands

Syntax `FETCh:FVTime?`

Arguments None

Returns `#<num_digit><num_byte><data(1)><data(2)>...<data(n)>`

Where
`<num_digit>` is the number of digits in <num_byte>.
`<num_byte>` is the number of bytes of data that follow.
`<data(n)>` is the frequency in Hz for the point #n,
4-byte little endian floating-point format specified in IEEE 488.2.

Examples `FETCH:FVTIME?` might return `#3156xxxx...` (156-byte data) for the Frequency versus Time trace.

FETCh:FVTime:MAXimum? (Query Only)

Returns the maximum value in the Frequency versus Time measurement.

Conditions Measurement views: Frequency versus Time

Group Fetch commands

Syntax `FETCh:FVTime:MAXimum?`

Related Commands FETCh:FVTime:MAXLocation?

Arguments None

Returns `<max>::=<NRf>` is the maximum frequency drift in Hz.

Examples `FETCH:FVTIME:MAXIMUM?` might return `2.625E+6`, indicating the maximum frequency drift is 2.625 MHz.

FETCh:FVTime:MAXLocation? (Query Only)

Returns the time at which the frequency drift is maximum.

Conditions Measurement views: Frequency versus Time

Group Fetch commands

Syntax `FETCh:FVTime:MAXLocation?`

Related Commands FETCh:FVTime:MAXimum?

Arguments None

Returns `<max_time>::=<NRf>` is the time in seconds at which the frequency drift is maximum.

Examples `FETCH:FVTIME:MAXLOCATION?` might return `25.03E-9`, indicating the frequency drift is maximum at 25.03 ns.

FETCh:FVTime:MINimum? (Query Only)

Returns the minimum value in the Frequency versus Time measurement.

Conditions Measurement views: Frequency versus Time

Group Fetch commands

Syntax `FETCh:FVTime:MINimum?`

Related Commands FETCh:FVTime:MINLocation?

Arguments None

Returns `<min>::=<NRf>` is the minimum frequency drift in Hz.

Examples `FETCH:FVTIME:MINIMUM?` might return `-6.618E+6`, indicating the minimum frequency drift is -6.618 MHz.

FETCh:FVTime:MINLocation? (Query Only)

Returns the time at which the frequency drift is minimum.

Conditions Measurement views: Frequency versus Time

Group Fetch commands

Syntax `FETCh:FVTime:MINLocation?`

Related Commands FETCh:FVTime:MINimum?

Arguments None

Returns `<min_time>::=<NRf>` is the time in seconds at which the frequency drift is minimum.

Examples `FETCH:FVTIME:MINLOCATION?` might return `450.7E-9`, indicating the frequency drift is minimum at 450.7 ns.

FETCh:FVTime:RESult? (Query Only)

Returns the Frequency versus Time measurement results.

Conditions Measurement views: Frequency versus Time

Group Fetch commands

Syntax `FETCh:FVTime:RESult?`

Arguments None

Returns `<max>,<max_time>,<min>,<min_time>`

Where
`<max>::=<NRf>` is the maximum frequency drift in Hz.
`<max_time>::=<NRf>` is the time in seconds at which the frequency drift is maximum.
`<min>::=<NRf>` is the minimum frequency drift in Hz.
`<min_time>::=<NRf>` is the time in seconds at which the frequency drift is minimum.

Examples `FETCH:FVTIME:RESULT?` might return
`2.625E+6,25.03E-9,-6.618E+6,450.7E-9`,
indicating
the maximum frequency drift is 2.625 MHz at 25.03 ns and
the minimum frequency drift is -6.618 MHz at 450.7 ns.

FETCh:IQVTime:I? (Query Only)

Returns the I versus Time trace data.

Conditions Measurement views: RF I&Q versus Time

Group Fetch commands

Syntax `FETCh:IQVTime:I?`

Arguments None

Returns `#<num_digit><num_byte><data(1)><data(2)>...<data(n)>`

Where
`<num_digit>` is the number of digits in <num_byte>.
`<num_byte>` is the number of bytes of data that follow.
`<data(n)>` is the I level in volts for the point #n,
4-byte little endian floating-point format specified in IEEE 488.2.

Examples `FETCH:IQVTIME:I?` might return `#3160xxxx...` (160-byte data) for the I versus Time trace.

FETCh:IQVTime:MAXimum? (Query Only)

Returns the maximum value in the RF I&Q versus Time measurement.

Conditions Measurement views: RF I&Q versus Time

Group Fetch commands

Syntax `FETCh:IQVTime:MAXimum?`

Related Commands FETCh:IQVTime:MAXLocation?

Arguments None

Returns `<max>::=<NRf>` is the maximum I or Q level in volts.

Use the TRACe:IQVTime:SELect:I or TRACe:IQVTime:SELect:Q command to select the trace.

Examples `FETCH:IQVTIME:MAXIMUM?` might return `1.214`, indicating the maximum I or Q level is 1.214 V.

FETCh:IQVTime:MAXLocation? (Query Only)

Returns the time at which the I or Q level is maximum.

Conditions Measurement views: RF I&Q versus Time

Group Fetch commands

Syntax `FETCh:IQVTime:MAXLocation?`

Related Commands FETCh:IQVTime:MAXimum?

Arguments None

Returns `<max_time>::=<NRf>` is the time in seconds at which the I or Q level is maximum.

Examples `FETCH:IQVTIME:MAXLOCATION?` might return `175.3E-9`, indicating the I or Q level is maximum at 175.3 ns.

FETCh:IQVTime:MINimum? (Query Only)

Returns the minimum value in the RF I&Q versus Time measurement.

Conditions Measurement views: RF I&Q versus Time

Group Fetch commands

Syntax `FETCh:IQVTime:MINimum?  imum`

Related Commands FETCh:IQVTime:MINLocation?

Arguments None

Returns `<min>::=<NRf>` is the minimum I or Q level in volts.

Use the TRACe:IQVTime:SELect:I or TRACe:IQVTime:SELect:Q command to select the trace.

Examples `FETCH:IQVTIME:MINIMUM?` might return `-370.5E-3`, indicating the minimum I or Q level is -370.5 mV.

FETCh:IQVTime:MINLocation? (Query Only)

Returns the time at which the I or Q level is minimum.

Conditions Measurement views: RF I&Q versus Time

Group Fetch commands

Syntax `FETCh:IQVTime:MINLocation?`

Related Commands FETCh:IQVTime:MINimum?

Arguments None

Returns `<min_time>::=<NRf>` is the time in seconds at which the I or Q level is minimum.

Examples `FETCH:IQVTIME:MINLOCATION?` might return `450.7E-9`, indicating the I or Q level is minimum at 450.7 ns.

FETCh:IQVTime:Q? (Query Only)

Returns the Q versus Time trace data.

Conditions Measurement views: RF I&Q versus Time

Group Fetch commands

Syntax `FETCh:IQVTime:Q?`

Arguments None

Returns `#<num_digit><num_byte><data(1)><data(2)>...<data(n)>`

Where
`<num_digit>` is the number of digits in <num_byte>.
`<num_byte>` is the number of bytes of data that follow.
`<data(n)>` is the Q level in volts for the point #n,

4-byte little endian floating-point format specified in IEEE 488.2.

Examples `FETCH:IQVTIME:Q?` might return `#3160xxxx...` (160-byte data) for the Q versus Time trace.

FETCh:IQVTime:RESult? (Query Only)

Returns the RF I&Q versus Time measurement results.

Conditions Measurement views: RF I&Q versus Time

Group Fetch commands

Syntax `FETCh:IQVTime:RESult?`

Arguments None

Returns `<max>,<max_time>,<min>,<min_time>`

Where
`<max>::=<NRf>` is the maximum I or Q level in volts.
`<max_time>::=<NRf>` is the time in seconds at which the I or Q level is maximum.
`<min>::=<NRf>` is the minimum I or Q level in volts.
`<min_time>::=<NRf>` is the time in seconds at which the I or Q level is minimum.

Use the TRACe:IQVTime:SELect:I or TRACe:IQVTime:SELect:Q command to select the trace.

Examples `FETCH:IQVTIME:RESULT?` might return `1.214,175.3E-9,-370.5E-3,450.7E-9`, indicating the maximum I or Q level is 1.214 V at 175.3 ns and the minimum I or Q level is -370.5 mV at 450.7 ns.

FETCh:MCPower:ADJacent:CHANnels? (Query Only)

Returns the power of adjacent channels in order of increasing frequency.

Conditions Measurement views: MCPR

Group Fetch commands

Syntax FETCh:MCPower:ADJacent:CHANnels?

Arguments None

Returns <acpr_lower(n)>,...<acpr_lower(2)>,<acpr_lower(1)>, <acpr_upper(1)>,<acpr_upper(2)>,...<acpr_upper(n)>

Where
<acpr_lower(n)> is the ACPR for the lower channel #n in dB.
<acpr_upper(n)> is the ACPR for the upper channel #n in dB.

To add a pair of upper and lower adjacent channels, use the [SENSe]:MCPower:CHANnel:ADJacent:ADD command.

Examples FETCH:MCPOWER:ADJACENT:CHANNELS? might return -4.420,-4.847,-4.316,-4.225, indicating
(ACPR for the lower channel 2) = -4.420 dB,
(ACPR for the lower channel 1) = -4.847 dB,
(ACPR for the upper channel 1) = -4.316 dB, and
(ACPR for the upper channel 2) = -4.225 dB.

FETCh:MCPower:CHANnel:POWer? (Query Only)

Returns the reference power in the MCPR measurement.

Conditions Measurement views: MCPR

Group Fetch commands

Syntax FETCh:MCPower:CHANnel:POWer?

Arguments None

Returns <ref_power>:<NRf> is the reference power in dBm.
The unit can be changed by the [SENSe]:POWer:UNITs command.

To select the power reference, use the [SENSe]:MCPower:RCHannels? commands.

Examples `FETCH:MCPOWER:CHANNEL:POWER?` might return `4.227`, indicating that the reference power is 4.227 dBm.

FETCh:MCPower:MAIN:CHANnels? (Query Only)

Returns the power of main channels in order of increasing frequency.

Conditions Measurement views: MCPR

Group Fetch commands

Syntax `FETCh:MCPower:MAIN:CHANnels?`

Arguments None

Returns <power_main(1)>,<power_main(2)>,...<power_main(n)>

Where
<power_main(n)> is the power of main channel #n in dBm.
The unit can be changed by the [SENSe]:POWer:UNITs command.

To specify the main channels, use the [SENSe]:MCPower:CHANnel:MAIN commands.

Examples `FETCH:MCPOWER:MAIN:CHANNELS?` might return `-2.420,-2.847,-2.316,-2.225`, indicating
(power of the main channel 1) = -2.420 dBm,
(power of the main channel 2) = -2.847 dBm,
(power of the main channel 3) = -2.316 dBm, and
(power of the main channel 4) = -2.225 dBm.

FETCh:MCPower:SPECtrum? (Query Only)

Returns spectrum trace data of the MCPR measurement.

Conditions Measurement views: MCPR

Group Fetch commands

Syntax `FETCh:MCPower:SPECtrum?`

Arguments None

Returns `#<num_digit><num_byte><data(1)><data(2)>...<data(n)>`

Where
`<num_digit>` is the number of digits in <num_byte>.
`<num_byte>` is the number of bytes of data that follow.
`<data(n)>` is the amplitude in dBm for the point #n,
4-byte little endian floating-point format specified in IEEE 488.2.
The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples `FETCH:MCPOWER:SPECTRUM?` might return `#43204xxxx...` (3204-byte data) for the spectrum trace data of the MCPR measurement.

FETCh:MERRor:FERRor? (Query Only)

Returns the frequency error in the Magnitude error versus Time measurement.

Conditions Measurement views: Magnitude error versus Time

Group Fetch commands

Syntax `FETCh:MERRor:FERRor?`

Arguments None

Returns `<freq_error>::=<NRf>` is the frequency error in Hz.

Examples `FETCH:MERROR:FERROR?` might return `-10.7E+3`, indicating the frequency error is -10.7 kHz.

FETCh:MERRor:PEAK? (Query Only)

Returns the peak value in the Magnitude error versus Time measurement.

Conditions Measurement views: Magnitude error versus Time

Group Fetch commands

Syntax `FETCh:MERRor:PEAK?`

Related Commands FETCh:MERRor:PINDex?

Arguments None

Returns `<peak>::=<NRf>` is the peak magnitude error in percent (%).

Examples `FETCH:MERROR:PEAK?` might return `1.57`, indicating the peak magnitude error is 1.57%.

FETCh:MERRor:PINDex? (Query Only)

Returns the time at the magnitude error peak.

Conditions Measurement views: Magnitude error versus Time

Group Fetch commands

Syntax `FETCh:MERRor:PINDex?`

Related Commands FETCh:MERRor:PEAK?

Arguments None

Returns `<peak_time>::=<NRf>` is the time at the magnitude error peak in symbol number. The unit can be changed by the [SENSe]:DDEMod:TIMe:UNITs command.

Examples `FETCH:MERROR:PINDEX?` might return `68.000`, indicating that the magnitude error peak is at symbol #68.

FETCh:MERRor:RMS? (Query Only)

Returns the RMS (Root-Mean-Square) value in the Magnitude error versus Time measurement.

Conditions Measurement views: Magnitude error versus Time

Group Fetch commands

Syntax `FETCh:MERRor:RMS?`

Arguments None

Returns `<rms>::=<NRf>` is the RMS magnitude error in percent (%).

Examples `FETCH:MERROR:RMS?` might return `0.382`, indicating the magnitude error is 0.382% RMS.

FETCh:MERRor:TRACe? (Query Only)

Returns the Magnitude error versus Time trace data.

Conditions Measurement views: Magnitude error versus Time

Group Fetch commands

Syntax `FETCh:MERRor:TRACe?`

Arguments None

Returns `#<num_digit><num_byte><data(1)><data(2)>...<data(n)>`

Where
`<num_digit>` is the number of digits in <num_byte>.
`<num_byte>` is the number of bytes of data that follow.
`<data(n)>` is the magnitude error in percent (%) for the point #n,
4-byte little endian floating-point format specified in IEEE 488.2.

Examples `FETCH:MERROR:TRACE?` might return `#42036xxxx...` (2036-byte data) for the Magnitude error versus Time trace.

FETCh:OBWidth:FREQuency:ERRor? (Query Only)

Returns the frequency error in the Occupied Bandwidth measurement.

Conditions Measurement views: Occupied Bandwidth

Group Fetch commands

Syntax `FETCh:OBWidth:FREQuency:ERRor?`

Arguments None

Returns `<freq_error>::=<NRf>` is the frequency error in Hz.

Examples `FETCH:OBWIDTH:FREQUENCY:ERROR?` might return `-10.7E+3`, indicating the frequency error is -10.7 kHz.

FETCh:OBWidth:OBWidth:BANDwidth? (Query Only)

Returns the occupied bandwidth in the Occupied Bandwidth measurement.

Conditions Measurement views: Occupied Bandwidth

Group Fetch commands

Syntax `FETCh:OBWidth:OBWidth:BANDwidth?`

Arguments None

Returns `<OBW>::=<NRf>` is the occupied bandwidth in Hz.

Examples `FETCH:OBWIDTH:OBWIDTH:BANDWIDTH?` might return `4.0E+6`, indicating the occupied bandwidth is 4 MHz.

FETCh:OBWidth:OBWidth:LEFT:FREQuency? (Query Only)

Returns the left (lower) frequency of the occupied bandwidth.

Conditions Measurement views: Occupied Bandwidth

Group Fetch commands

Syntax `FETCh:OBWidth:OBWidth:LEFT:FREQuency?`

Related Commands FETCh:OBWidth:OBWidth:RIGHt:FREQuency?

Arguments None

Returns `<OBW_left_freq>::=<NRf>` is the left frequency in Hz.

Examples `FETCH:OBWIDTH:OBWIDTH:LEFT:FREQUENCY?` might return `1.498E+9`, indicating the left frequency is 1.498 GHz.

FETCh:OBWidth:OBWidth:LEFT:LEVel? (Query Only)

Returns the level at the left frequency of the occupied bandwidth.

Conditions Measurement views: Occupied Bandwidth

Group Fetch commands

Syntax `FETCh:OBWidth:OBWidth:LEFT:LEVel?`

Related Commands FETCh:OBWidth:OBWidth:RIGHt:LEVel?

Arguments None

Returns `<OBW_left_level>::=<NRf>` is the level at the left frequency in dB.

Examples
`FETCH:OBWIDTH:OBWIDTH:LEFT:LEVEL?` might return `-23.5`, indicating the level at the left frequency is -23.5 dB.

FETCh:OBWidth:OBWidth:POWer? (Query Only)

Returns the reference power in the Occupied Bandwidth measurement.

Conditions
Measurement views: Occupied Bandwidth

Group
Fetch commands

Syntax
`FETCh:OBWidth:OBWidth:POWer?`

Arguments
None

Returns
`<OBW_ref_power>::=<NRf>` is the reference power in dBm.
The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples
`FETCH:OBWIDTH:OBWIDTH:POWER?` might return `-10.0`, indicating the reference power is -10 dBm.

FETCh:OBWidth:OBWidth:RIGHt:FREQuency? (Query Only)

Returns the right (higher) frequency of the occupied bandwidth.

Conditions
Measurement views: Occupied Bandwidth

Group
Fetch commands

Syntax
`FETCh:OBWidth:OBWidth:RIGHt:FREQuency?`

Related Commands
FETCh:OBWidth:OBWidth:LEFT:FREQuency?

Arguments
None

Returns
`<OBW_right_freq>::=<NRf>` is the right frequency in Hz.

Examples `FETCH:OBWIDTH:OBWIDTH:RIGHT:FREQUENCY?` might return `1.502E+9`, indicating the right frequency is 1.502 GHz.

FETCh:OBWidth:OBWidth:RIGHt:LEVel? (Query Only)

Returns the level at the right frequency of the occupied bandwidth.

Conditions Measurement views: Occupied Bandwidth

Group Fetch commands

Syntax `FETCh:OBWidth:OBWidth:RIGHt:LEVel?`

Related Commands FETCh:OBWidth:OBWidth:LEFT:LEVel?

Arguments None

Returns `<OBW_right_level>::=<NRf>` is the level at the right frequency in dB.

Examples `FETCH:OBWIDTH:OBWIDTH:RIGHT:LEVEL?` might return `-23.5`, indicating the level at the right frequency is -23.5 dB.

FETCh:OBWidth:SPECtrum? (Query Only)

Returns spectrum trace data of the Occupied Bandwidth measurement.

Conditions Measurement views: Occupied Bandwidth

Group Fetch commands

Syntax `FETCh:OBWidth:SPECtrum?`

Arguments None

Returns `#<num_digit><num_byte><data(1)><data(2)>...<data(n)>`

Where

`<num_digit>` is the number of digits in <num_byte>.
`<num_byte>` is the number of bytes of data that follow.
`<data(n)>` is the amplitude in dBm for the point #n,
4-byte little endian floating-point format specified in IEEE 488.2.
The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples

`FETCH:OBWIDTH:SPECTRUM?` might return `#43204xxxx...` (3204-byte data) for the spectrum trace data of the Occupied Bandwidth measurement.

FETCh:OBWidth:XDBBandwidth:BANDwidth? (Query Only)

Returns the x dB bandwidth in the Occupied Bandwidth measurement.

Conditions

Measurement views: Occupied Bandwidth

Group

Fetch commands

Syntax

`FETCh:OBWidth:XDBBandwidth:BANDwidth?`

Arguments

None

Returns

`<xdBBW>::=<NRf>` is the x dB bandwidth in Hz.

Examples

`FETCH:OBWIDTH:XDBBANDWIDTH:BANDWIDTH?` might return `2.0E+6`, indicating the x dB bandwidth is 2 MHz.

FETCh:OBWidth:XDBBandwidth:LEFT:FREQuency? (Query Only)

Returns the left (lower) frequency of the x dB bandwidth.

Conditions

Measurement views: Occupied Bandwidth

Group

Fetch commands

Syntax

`FETCh:OBWidth:XDBBandwidth:LEFT:FREQuency?`

Related Commands

FETCh:OBWidth:XDBBandwidth:RIGHt:FREQuency?

Arguments None

Returns `<xdBBW_left_freq>::=<NRf>` is the left frequency in Hz.

Examples `FETCH:OBWIDTH:XDBBANDWIDTH:LEFT:FREQUENCY?` might return `1.498E+9`, indicating the left frequency is 1.498 GHz.

FETCh:OBWidth:XDBBandwidth:LEFT:LEVel? (Query Only)

Returns the level at the left frequency of the x dB bandwidth.

Conditions Measurement views: Occupied Bandwidth

Group Fetch commands

Syntax `FETCh:OBWidth:XDBBandwidth:LEFT:LEVel?`

Related Commands FETCh:OBWidth:XDBBandwidth:RIGHt:LEVel?

Arguments None

Returns `<xdBBW_left_level>::=<NRf>` is the level at the left frequency in dB.

Examples `FETCH:OBWIDTH:XDBBANDWIDTH:LEFT:LEVEL?` might return `-23.5`, indicating the level at the left frequency is -23.5 dB.

FETCh:OBWidth:XDBBandwidth:POWer? (Query Only)

Returns the reference power in the x dB bandwidth measurement.

Conditions Measurement views: Occupied Bandwidth

Group Fetch commands

Syntax `FETCh:OBWidth:XDBBandwidth:POWer?`

Arguments None

Returns `<xdBBW_ref_power>::=<NRf>` is the reference power in dBm.
The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples `FETCH:OBWIDTH:XDBBANDWIDTH:POWER?` might return `-10.0`, indicating the reference power is -10 dBm.

FETCh:OBWidth:XDBBandwidth:RIGHt:FREQuency? (Query Only)

Returns the right (higher) frequency of the x dB bandwidth.

Conditions Measurement views: Occupied Bandwidth

Group Fetch commands

Syntax `FETCh:OBWidth:XDBBandwidth:RIGHt:FREQuency?`

Related Commands FETCh:OBWidth:XDBBandwidth:LEFT:FREQuency?

Arguments None

Returns `<xdBBW_right_freq>::=<NRf>` is the right frequency in Hz.

Examples `FETCH:OBWIDTH:XDBBANDWIDTH:RIGHT:FREQUENCY?` might return `1.502E+9`, indicating the right frequency is 1.502 GHz.

FETCh:OBWidth:XDBBandwidth:RIGHt:LEVel? (Query Only)

Returns the level at the right frequency of the x dB bandwidth.

Conditions Measurement views: Occupied Bandwidth

Group Fetch commands

Syntax `FETCh:OBWidth:XDBBandwidth:RIGHt:LEVel?`

Related Commands FETCh:OBWidth:XDBBandwidth:LEFT:LEVel?

Arguments None

Returns `<xdBBW_right_level>::=<NRf>` is the level at the right frequency in dB.

Examples `FETCH:OBWIDTH:XDBBANDWIDTH:RIGHT:LEVEL?` might return `-23.5`, indicating the level at the right frequency is -23.5 dB.

FETCh:PERRor:FERRor? (Query Only)

Returns the frequency error in the Phase error versus Time measurement.

Conditions Measurement views: Phase error versus Time

Group Fetch commands

Syntax `FETCh:PERRor:FERRor?`

Arguments None

Returns `<freq_error>::=<NRf>` is the frequency error in Hz.

Examples `FETCH:PERROR:FERROR?` might return `-10.7E+3`, indicating the frequency error is -10.7 kHz.

FETCh:PERRor:PEAK? (Query Only)

Returns the peak value in the Phase error versus Time measurement.

Conditions Measurement views: Phase error versus Time

Group Fetch commands

Syntax `FETCh:PERRor:PEAK?`

Related Commands FETCh:PERRor:PINDex?

Arguments None

Returns `<peak>::=<NRf>` is the peak phase error in degrees.

Examples `FETCH:PERROR:PEAK?` might return `0.683`, indicating the peak phase error is 0.683°.

FETCh:PERRor:PINDex? (Query Only)

Returns the time at the phase error peak.

Conditions Measurement views: Phase error versus Time

Group Fetch commands

Syntax `FETCh:PERRor:PINDex?`

Related Commands FETCh:PERRor:PEAK?, [SENSe]:DDEMod:TIMe:UNITs

Arguments None

Returns `<peak_time>::=<NRf>` is the time at the phase error peak in symbol number. The unit can be changed by the [SENSe]:DDEMod:TIMe:UNITs command.

Examples `FETCH:PERROR:PINDEX?` might return `68.000`, indicating that the phase error peak is at symbol #68.

FETCh:PERRor:RMS? (Query Only)

Returns the RMS (Root-Mean-Square) value in the Phase error versus Time measurement.

Conditions Measurement views: Phase error versus Time

Group Fetch commands

Syntax `FETCh:PERRor:RMS?`

Arguments None

Returns `<rms>::=<NRf>` is the RMS phase error in degrees.

Examples `FETCH:PERROR:RMS?` might return `0.746`, indicating the phase error is 0.746° RMS.

FETCh:PERRor:TRACe? (Query Only)

Returns the Phase error versus Time trace data.

Conditions Measurement views: Phase error versus Time

Group Fetch commands

Syntax `FETCh:PERRor:TRACe?`

Arguments None

Returns `#<num_digit><num_byte><data(1)><data(2)>...<data(n)>`

Where
`<num_digit>` is the number of digits in <num_byte>.
`<num_byte>` is the number of bytes of data that follow.
`<data(n)>` is the phase error in degrees for the point #n,
4-byte little endian floating-point format specified in IEEE 488.2.

Examples `FETCH:PERROR:TRACE?` might return `#42036xxxx...` (2036-byte data) for the Phase error versus Time trace.

FETCh:PHVTime? (Query Only)

Returns the Phase versus Time trace data.

Conditions Measurement views: Phase versus Time

Group Fetch commands

Syntax `FETCh:PHVTime?`

Arguments None

Returns `#<num_digit><num_byte><data(1)><data(2)>...<data(n)>`

Where
`<num_digit>` is the number of digits in <num_byte>.
`<num_byte>` is the number of bytes of data that follow.
`<data(n)>` is the phase in degrees for the point #n,
4-byte little endian floating-point format specified in IEEE 488.2.

Examples `FETCH:PHVTIME?` might return `#3160xxxx...` (160-byte data) for the Phase versus Time trace.

FETCh:PHVTime:MAXimum? (Query Only)

Returns the maximum value in the Phase versus Time measurement.

Conditions Measurement views: Phase versus Time

Group Fetch commands

Syntax `FETCh:PHVTime:MAXimum?`

Related Commands FETCh:PHVTime:MAXLocation?

Arguments None

Returns `<max>::=<NRf>` is the maximum phase in degrees.

Examples `FETCH:PHVTIME:MAXIMUM?` might return `153.8`, indicating the maximum phase is 153.8°.

FETCh:PHVTime:MAXLocation? (Query Only)

Returns the time at which the phase is maximum.

Conditions Measurement views: Phase versus Time

Group Fetch commands

Syntax `FETCh:PHVTime:MAXLocation?`

Related Commands FETCh:PHVTime:MAXimum?

Arguments None

Returns `<max_time>::=<NRf>` is the time in seconds at which the phase is maximum.

Examples `FETCH:PHVTIME:MAXLOCATION?` might return `175.3E-9`, indicating the I or Q level is maximum at 175.3 ns.

FETCh:PHVTime:MINimum? (Query Only)

Returns the minimum value in the Phase versus Time measurement.

Conditions Measurement views: Phase versus Time

Group Fetch commands

Syntax `FETCh:PHVTime:MINimum?`

Related Commands FETCh:PHVTime:MINLocation?

Arguments None

Returns `<min>::=<NRf>` is the minimum phase in degrees.

Examples `FETCH:PHVTIME:MINIMUM?` might return `-176.3`, indicating the minimum phase is -176.3°.

FETCh:PHVTime:MINLocation? (Query Only)

Returns the time at which the phase is minimum.

Conditions Measurement views: Phase versus Time

Group Fetch commands

Syntax `FETCh:PHVTime:MINLocation?`

Related Commands FETCh:PHVTime:MINimum?

Arguments None

Returns `<min_time>::=<NRf>` is the time in seconds at which the phase is minimum.

Examples `FETCH:PHVTIME:MINLOCATION?` might return `450.7E-9`, indicating the phase is minimum at 450.7 ns.

FETCh:PHVTime:RESult? (Query Only)

Returns the Phase versus Time measurement results.

Conditions Measurement views: Phase versus Time

Group Fetch commands

Syntax `FETCH:PHVTime:RESult?`

Arguments None

Returns `<max>,<max_time>,<min>,<min_time>`

Where
`<max>::=<NRf>` is the maximum phase in degrees.
`<max_time>::=<NRf>` is the time in seconds at which the phase is maximum.
`<min>::=<NRf>` is the minimum phase in degrees.
`<min_time>::=<NRf>` is the time in seconds at which the phase is minimum.

Examples

`FETCH:PHVTIME:RESULT?` might return
`153.8,175.3E-9,-176.3,450.7E-9`, indicating the maximum phase is
153.8° at 175.3 ns and the minimum phase is -176.3° at 450.7 ns.

FETCh:PNOise:ALL? (Query Only)

Returns all results of the phase noise measurement.

Conditions

Measurement views: Phase noise

Group

Fetch commands

Syntax

`FETCh:PNOise:ALL?`

Arguments

None

Returns

`<Cpower>,<Ferror>,<Pnoise>,<Tjitter>,<Rjitter>,<RFM>`

Where
`<Cpower>::=<NRf>` is the carrier power in dBm.
`<Ferror>::=<NRf>` is the frequency error in Hz.
`<Pnoise>::=<NRf>` is the RMS phase noise in degrees.
`<Tjitter>::=<NRf>` is the total jitter in seconds.
`<Rjitter>::=<NRf>` is the random jitter in seconds.
`<RFM>::=<NRf>` is the residual FM in Hz.

Examples

`FETCH:PNOISE:ALL?` might return
`-9.455,1.235E+6,51.43,2.312E-9,4.178E-9,14.58`,
indicating
Carrier power: -9.455 dBm,
Frequency error: 1.235 MHz,
RMS phase noise: 51.43°,
Total jitter: 2.312 ns,
Random jitter: 4.178 ns, and

Residual FM: 14.58 Hz.

FETCh:PNOise:CARRier:FERRor? (Query Only)

Returns the carrier frequency error in the phase noise measurement.

Conditions Measurement views: Phase noise

Group Fetch commands

Syntax `FETCh:PNOise:CARRier:FERRor?`

Arguments None

Returns `<NRf>` Carrier frequency error in Hz.

Examples `FETCH:PNOISE:CARRIER:FERROR?` might return `1.235E+6`, indicating that the carrier frequency error is 1.235 MHz.

FETCh:PNOise:CARRier:POWer? (Query Only)

Returns the carrier power in the phase noise measurement.

Conditions Measurement views: Phase noise

Group Fetch commands

Syntax `FETCh:PNOise:CARRier:POWer?`

Arguments None

Returns `<NRf>` Carrier power in dBm.
The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples `FETCH:PNOISE:CARRIER:POWER?` might return `-9.455`, indicating that the carrier power is -9.455 dBm.

FETCh:PNOise:JITTer? (Query Only)

Returns the jitter in the phase noise measurement.

Conditions Measurement views: Phase noise

Group Fetch commands

Syntax `FETCh:PNOise:JITTer?`

Arguments None

Returns `<NRf>` Jitter in seconds.

Examples `FETCH:PNOISE:JITTER?` might return `2.312E-9`, indicating that the jitter is 2.312 ns.

FETCh:PNOise:RESidual:FM? (Query Only)

Returns the residual FM in the phase noise measurement.

Conditions Measurement views: Phase noise

Group Fetch commands

Syntax `FETCh:PNOise:RESidual:FM?`

Arguments None

Returns `<NRf>` Residual FM in Hz.

Examples `FETCH:PNOISE:RESIDUAL:FM?` might return `14.58`, indicating that the residual FM is 14.58 Hz.

FETCh:PNOise:RMS:PNOise? (Query Only)

Returns the RMS phase noise in the phase noise measurement.

Conditions Measurement views: Phase noise

Group Fetch commands

Syntax `FETCh:PNOise:RMS:PNOise?`

Arguments None

Returns `<NRf>` RMS phase noise in degrees.

Examples `FETCH:PNOISE:RMS:PNOISE?` might return `51.43`, indicating that the RMS phase noise is 51.43°.

FETCh:PNOise:SPECtrum<x>:X? (Query Only)

Returns the frequencies of the specified trace.

The parameter <x> = 1 and 2, representing Trace 1 and Trace 2, respectively.

Conditions Measurement views: Phase noise

Group Fetch commands

Syntax `FETCh:PNOise:SPECtrum<x>:X?`

Arguments None

Returns `#<num_digit><num_byte><x(1)><x(2)>...<x(n)>`

Where
`<num_digit>` is the number of digits in <num_byte>.
`<num_byte>` is the number of bytes of data that follow.
`<x(n)>` is the frequency (Hz) at the point #n,
4-byte little endian floating-point format specified in IEEE 488.2.

Examples `FETCH:PNOISE:SPECTRUM1:X?` might return `#516020xxxx...` (16020-byte data) for the frequencies of Trace 1.

FETCh:PNOise:SPECtrum<x>:XY? (Query Only)

Returns the frequency and phase noise pairs of the specified trace.

The parameter <x> = 1 and 2, representing Trace 1 and Trace 2, respectively.

Conditions Measurement views: Phase noise

Group Fetch commands

Syntax `FETCh:PNOise:SPECtrum<x>:XY?`

Arguments None

Returns `#<num_digit><num_byte><x(1)><y(1)><x(2)><y(2)>...<x(n)><y(n)>`

Where
`<num_digit>` is the number of digits in <num_byte>.
`<num_byte>` is the number of bytes of data that follow.
`<x(n)><y(n)>` is the frequency (Hz) and phase noise (dBc/Hz) pair for the point #n, 4-byte little endian floating-point format specified in IEEE 488.2.

Examples `FETCH:PNOISE:SPECTRUM1:XY?` might return `#516020xxxx...` (16020-byte data) for the frequency and phase noise pairs of the Trace 1.

FETCh:PNOise:SPECtrum<x>[:Y]? (Query Only)

Returns the phase noise values of the specified trace.

The parameter <x> = 1 and 2, representing Trace 1 and Trace 2, respectively.

Conditions Measurement views: Phase noise

Group Fetch commands

Syntax `FETCh:PNOise:SPECtrum<x>[:Y]?`

Arguments None

Returns `#<num_digit><num_byte><y(1)><y(2)>...<y(n)>`

Where
`<num_digit>` is the number of digits in <num_byte>.
`<num_byte>` is the number of bytes of data that follow.
`<y(n)>` is the phase noise (dBc/Hz) at the point #n,
4-byte little endian floating-point format specified in IEEE 488.2.

Examples `FETCH:PNOISE:SPECTRUM1:Y` might return `#516020xxxx...` (16020-byte data) for the phase noise values of Trace 1.

FETCh:PULSe[:RESult]:ATX? (Query Only)

Returns the average transmitted power in the pulse table.

Conditions Measurement views: Pulse table

Group Fetch commands

Syntax `FETCh:PULSe[:RESult]:ATX?`

Arguments None

Returns `<first_pulse_num>,<ATX(1)>,< ATX(2)>,...<ATX(n)>`

Where
`<first_pulse_num>::=<NR1>` is the first pulse number.
`<ATX(n)>::=<NRf>` is the average transmitted power for the pulse with the number of [first_pulse_num + n - 1] in dBm.
The unit can be changed to watts by the [SENSe]:POWer:UNITs command.
Volt is invalid in the average transmitted power measurement.

Examples `FETCH:PULSE:RESULT:ATX?` might return `0,-18.57,-18.73,-18.20,` `-18.53` for Pulse 0 to 3.

FETCh:PULSe[:RESult]:AVERage? (Query Only)

Returns the average on power in the pulse table.

Conditions Measurement views: Pulse table

Group Fetch commands

Syntax `FETCh:PULSe[:RESult]:AVERage?`

Arguments None

Returns `<first_pulse_num>,< avg(1)>,< avg(2)>,...<avg(n)>`

Where
`<first_pulse_num>::=<NR1>` is the first pulse number.
`<avg(n)>::=<NRf>` is the average on power for the pulse with the number of [first_pulse_num + n - 1] in dBm.
The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples `FETCH:PULSE:RESULT:AVERAGE?` might return
`0,-2.354,-2.368,-2.343,-2.358` for Pulse 0 to 3.

FETCh:PULSe[:RESult]:DROop? (Query Only)

Returns the droop in the pulse table.

Conditions Measurement views: Pulse table

Group Fetch commands

Syntax `FETCh:PULSe[:RESult]:DROop?`

Arguments None

Returns `<first_pulse_num>,<droop(1)>,<droop(2)>,...<droop(n)>`

Where
`<first_pulse_num>::=<NR1>` is the first pulse number.

<droop(n)>::=<NRf> is the wattage droop for the pulse with the number of [first_pulse_num + n - 1] in percent (%).

Examples `FETCH:PULSE:RESULT:DROOP?` might return `0,-270.9E-3,-193.0E-3,-242.7E-3,-177.5E-3` for Pulse 0 to 3.

FETCh:PULSe[:RESult]:DUTPct? (Query Only)

Returns the duty factor (%) in the pulse table.

Conditions Measurement views: Pulse table

Group Fetch commands

Syntax `FETCh:PULSe[:RESult]:DUTPct?`

Arguments None

Returns `<first_pulse_num>,<duty_pct(1)>,<duty_pct(2)>,...<duty_pct(n)>`

Where
<first_pulse_num>::=<NR1> is the first pulse number.
<duty_pct(n)>::=<NRf> is the duty factor for the pulse with the number of [first_pulse_num + n - 1] in percent (%).

Examples `FETCH:PULSE:RESULT:DUTPCT?` might return `0,28.94,28.96,29.00,29.01` for Pulse 0 to 3.

FETCh:PULSe[:RESult]:DUTRatio? (Query Only)

Returns the duty factor (ratio) in the pulse table.

Conditions Measurement views: Pulse table

Group Fetch commands

Syntax `FETCh:PULSe[:RESult]:DUTRatio?`

Arguments None

Returns `<first_pulse_num>,<duty_ratio(1)>,<duty_ratio(2)>,... <duty_ratio(n)>`

Where
`<first_pulse_num>::=<NR1>` is the first pulse number.
`<duty_ratio(n)>::=<NRf>` is the duty factor for the pulse with the number of [first_pulse_num + n - 1] (no unit).

Examples `FETCH:PULSE:RESULT:DUTRATIO?` might return `0,289.4E-3,289.6E-3, 290.0E-3,290.1E-3` for Pulse 0 to 3.

FETCh:PULSe[:RESult]:FALL? (Query Only)

Returns the fall time in the pulse table.

Conditions Measurement views: Pulse table

Group Fetch commands

Syntax `FETCh:PULSe[:RESult]:FALL?`

Arguments None

Returns `<first_pulse_num>,<fall(1)>,<fall(2)>,...<fall(n)>`

Where
`<first_pulse_num>::=<NR1>` is the first pulse number.
`<fall(n)>::=<NRf>` is the fall time for the pulse with the number of [first_pulse_num + n - 1] in seconds.

Examples `FETCH:PULSE:RESULT:FALL?` might return `0,110.3E-9,90.45E-9, 95.03E-9,111.9E-9` for Pulse 0 to 3.

FETCh:PULSe[:RESult]:FRDeviation? (Query Only)

Returns the frequency deviation in the pulse table.

Conditions Measurement views: Pulse table

Group Fetch commands

Syntax `FETCh:PULSe[:RESult]:FRDeviation?`

Arguments None

Returns `<first_pulse_num>,<freq_dev(1)>,<freq_dev(2)>,... <freq_dev(n)>`

Where
`<first_pulse_num>::=<NR1>` is the first pulse number.
`<freq_dev(n)>::=<NRf>` is the frequency deviation for the pulse with the number of [first_pulse_num + n - 1] in Hz.

Examples `FETCH:PULSE:RESULT:FRDEVIATION?` might return `1,740.6E+3, 736.5E+3,718.3E+3,672.2E+3` for Pulse 1 to 4.

FETCh:PULSe[:RESult]:MFReqerror? (Query Only)

Returns the maximum frequency error in the pulse table.

Conditions Measurement views: Pulse table

Group Fetch commands

Syntax `FETCh:PULSe[:RESult]:MFReqerror?`

Arguments None

Returns `<first_pulse_num>,<max_freq_err(1)>,<max_freq_err(2)>,... <max_freq_err(n)>`

Where
`<first_pulse_num>::=<NR1>` is the first pulse number.
`<max_freq_err(n)>::=<NRf>` is the maximum frequency error for the pulse with the number of [first_pulse_num + n - 1] in Hz.

Examples FETCH:PULSE:RESULT:MFREQERROR? might return 1,597.5E+3,675.8E+3,642.8E+3,598.2E+3 for Pulse 1 to 4.

FETCh:PULSe[:RESult]:MPHerror? (Query Only)

Returns the maximum phase error in the pulse table.

Conditions Measurement views: Pulse table

Group Fetch commands

Syntax FETCh:PULSe[:RESult]:MPHerror?

Arguments None

Returns <first_pulse_num>,<max_phase_err(1)>,<max_phase_err(2)>,... <max_phase_err(n)>

Where
<first_pulse_num>::=<NR1> is the first pulse number.
<max_phase_err(n)>::=<NRf> is the maximum phase error for the pulse with the number of [first_pulse_num + n - 1] in degrees.

Examples FETCH:PULSE:RESULT:MPHERROR? might return 1,-9.221,-8.413, -11.853,-10.258 for Pulse 1 to 4.

FETCh:PULSe[:RESult]:PHDeviation? (Query Only)

Returns the phase deviation in the pulse table.

Conditions Measurement views: Pulse table

Group Fetch commands

Syntax FETCh:PULSe[:RESult]:PHDeviation?

Arguments None

Returns
<first_pulse_num>,<phase_dev(1)>,<phase_dev(2)>,... <phase_dev(n)>

Where
<first_pulse_num>::=<NR1> is the first pulse number.
<phase_dev(n)>::=<NRf> is the phase deviation for the pulse with the number of [first_pulse_num + n - 1] in degrees.

Examples
FETCH:PULSE:RESULT:PHDEVIATION? might return 1,11.658,9.640,10.509,8.272 for Pulse 1 to 4.

FETCh:PULSe[:RESult]:PPFRequency? (Query Only)

Returns the pulse-pulse carrier frequency in the pulse table.

Conditions
Measurement views: Pulse table

Group
Fetch commands

Syntax
FETCh:PULSe[:RESult]:PPFRequency?

Arguments
None

Returns
<first_pulse_num>,<pp_freq(1)>,<pp_freq(2)>,...<pp_freq(n)>

Where
<first_pulse_num>::=<NR1> is the first pulse number.
<pp_freq(n)>::=<NRf> is the pulse-pulse carrier frequency for the pulse with the number of [first_pulse_num + n - 1] in Hz.

Examples
FETCH:PULSE:RESULT:PPFREQUENCY? might return 0,0.000,1.258E+3,-3.121E+3,1.862E+3 for Pulse 0 to 3.

FETCh:PULSe[:RESult]:PPOWer? (Query Only)

Returns the peak power in the pulse table.

Conditions
Measurement views: Pulse table

Group Fetch commands

Syntax `FETCh:PULSe[:RESult]:PPOWer?`

Arguments None

Returns `<first_pulse_num>,<pk_power(1)>,<pk_power(2)>,... <pk_power(n)>`

Where
`<first_pulse_num>::=<NR1>` is the first pulse number.
`<pk_power(n)>::=<NRf>` is the peak power for the pulse with the number of [first_pulse_num + n - 1] in dBm.
The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples `FETCH:PULSE:RESULT:PPOWER?` might return `0,-2.26,-2.27,-2.23,-2.25` for Pulse 0 to 3.

FETCh:PULSe[:RESult]:PPPHase? (Query Only)

Returns the pulse-pulse carrier phase in the pulse table.

Conditions Measurement views: Pulse table

Group Fetch commands

Syntax `FETCh:PULSe[:RESult]:PPPHase?`

Arguments None

Returns `<first_pulse_num>,<pp_phase(1)>,<pp_phase(2)>,... <pp_phase(n)>`

Where
`<first_pulse_num>::=<NR1>` is the first pulse number.
`<pp_phase(n)>::=<NRf>` is the pulse-pulse carrier phase for the pulse with the number of [first_pulse_num + n - 1] in degrees.

Examples `FETCH:PULSE:RESULT:PPPHASE?` might return `0,0.000,21.66,46.76,57.56` for Pulse 0 to 3.

FETCh:PULSe[:RESult]:RINTerval? (Query Only)

Returns the repetition interval in the pulse table.

Conditions Measurement views: Pulse table

Group Fetch commands

Syntax `FETCh:PULSe[:RESult]:RINTerval?`

Arguments None

Returns `<first_pulse_num>,<rep_int(1)>,<rep_int(2)>,...<rep_int(n)>`

Where
`<first_pulse_num>::=<NR1>` is the first pulse number.
`<rep_int(n)>::=<NRf>` is the repetition interval for the pulse with the number of [first_pulse_num + n - 1] in seconds.

Examples `FETCH:PULSE:RESULT:RINTERVAL?` might return `0,16.03E-6,16.08E-6,16.07E-6,16.02E-6` for Pulse 0 to 3.

FETCh:PULSe[:RESult]:RIPPle? (Query Only)

Returns the ripple in the pulse table.

Conditions Measurement views: Pulse table

Group Fetch commands

Syntax `FETCh:PULSe[:RESult]:RIPPle?`

Arguments None

Returns `<first_pulse_num>,<ripple(1)>,<ripple(2)>,...<ripple(n)>`

Where
`<first_pulse_num>::=<NR1>` is the first pulse number.
`<ripple(n)>::=<NRf>` is the voltage ripple for the pulse with the number of [first_pulse_num + n - 1] in percent (%).

Examples `FETCH:PULSE:RESULT:RIPPLE?` might return `0,106.5E-3,177.6E-3,148.3E-3,148.5E-3` for Pulse 0 to 3.

FETCh:PULSe[:RESult]:RISE? (Query Only)

Returns the rise time in the pulse table.

Conditions Measurement views: Pulse table

Group Fetch commands

Syntax `FETCh:PULSe[:RESult]:RISE?`

Arguments None

Returns `<first_pulse_num>,<rise(1)>,<rise(2)>,...<rise(n)>`

Where
`<first_pulse_num>::=<NR1>` is the first pulse number.
`<rise(n)>::=<NRf>` is the rise time for the pulse with the number of [first_pulse_num + n - 1] in seconds.

Examples `FETCH:PULSE:RESULT:RISE?` might return `0,92.94E-9,115.9E-9,115.1E-9,97.45E-9` for Pulse 0 to 3.

FETCh:PULSe[:RESult]:RMSFreqerror? (Query Only)

Returns the RMS frequency error in the pulse table.

Conditions Measurement views: Pulse table

Group Fetch commands

Syntax `FETCh:PULSe[:RESult]:RMSFreqerror?`

Arguments None

Returns `<first_pulse_num>,<RMS_freq_err(1)>,<RMS_freq_err(2)>,... <RMS_freq_err(n)>`

Where
`<first_pulse_num>::=<NR1>` is the first pulse number.
`<RMS_freq_err(n)>::=<NRf>` is the RMS frequency error for the pulse with the number of [first_pulse_num + n - 1] in Hz.

Examples `FETCH:PULSE:RESULT:RMSFREQERROR?` might return `1,51.54E+3, 69.20E+3,64.21E+3,51.02E+3` for Pulse 1 to 4.

FETCh:PULSe[:RESult]:RMSPherror? (Query Only)

Returns the RMS phase error in the pulse table.

Conditions Measurement views: Pulse table

Group Fetch commands

Syntax `FETCh:PULSe[:RESult]:RMSPherror?`

Arguments None

Returns `<first_pulse_num>,<RMS_phase_err(1)>,<RMS_phase_err(2)>,... <RMS_phase_err(n)>`

Where
`<first_pulse_num>::=<NR1>` is the first pulse number.
`<RMS_phase_err(n)>::=<NRf>` is the RMS phase error for the pulse with the number of [first_pulse_num + n - 1] in degrees.

Examples `FETCH:PULSE:RESULT:RMSPHERROR?` might return `1,908.4E-3,752.8E-3,981.7E-3,886.4E-3` for Pulse 1 to 4.

FETCh:PULSe[:RESult]:RRATe? (Query Only)

Returns the repetition rate in the pulse table.

Conditions Measurement views: Pulse table

Group Fetch commands

Syntax `FETCh:PULSe[:RESult]:RRATe?`

Arguments None

Returns `<first_pulse_num>,<rep_rate(1)>,<rep_rate(2)>,...<rep_rate(n)>`

Where
`<first_pulse_num>::=<NR1>` is the first pulse number.
`<rep_rate(n)>::=<NRf>` is the repetition rate for the pulse with the number of [first_pulse_num + n - 1] in Hz.

Examples `FETCH:PULSE:RESULT:RRATE?` might return `0,62.50E+3,62.52E+3,62.51E+3,62.49E+3` for Pulse 0 to 3.

FETCh:PULSe[:RESult]:TIME? (Query Only)

Returns the time in the pulse table.

Conditions Measurement views: Pulse table

Group Fetch commands

Syntax `FETCh:PULSe[:RESult]:TIME?`

Arguments None

Returns `<first_pulse_num>,<time(1)>,<time(2)>,...<time(n)>`

Where
`<first_pulse_num>::=<NR1>` is the first pulse number.

`<time(n)>::=<NRf>` is the time for the pulse with the number of [first_pulse_num + n - 1] in seconds.

Examples

`FETCH:PULSE:RESULT:TIME?` might return `1,7.937E-3,8.436E-3, 6.504E-3,9.876E-3` for Pulse 1 to 4.

FETCh:PULSe[:RESult]:WIDTh? (Query Only)

Returns the pulse width in the pulse table.

Conditions

Measurement views: Pulse table

Group

Fetch commands

Syntax

`FETCh:PULSe[:RESult]:WIDTh?`

Arguments

None

Returns

`<first_pulse_num>,<width(1)>,<width(2)>,...<width(n)>`

Where
`<first_pulse_num>::=<NR1>` is the first pulse number.
`<width(n)>::=<NRf>` is the pulse width for the pulse with the number of [first_pulse_num + n - 1] in seconds.

Examples

`FETCH:PULSE:RESULT:WIDTH?` might return `0,4.630E-6,4.632E-6, 4.639E-6,4.642E-6` for Pulse 0 to 3.

FETCh:PULSe:STATistics? (Query Only)

Returns the trace data of the pulse statistics measurement selected by the DISPlay:PULSe:SELect:RESult command.

NOTE. *Select the plot type (Trend or FFT) using the DISPlay:PULSe:STATistics:PLOT command before executing this query.*

Conditions

Measurement views: Pulse statistics

Group Fetch commands

Syntax `FETCh:PULSe:STATistics?`

Arguments None

Returns `#<num_digit><num_byte><data(1)><data(2)>...<data(n)>`

Where
`<num_digit>` is the number of digits in <num_byte>.
`<num_byte>` is the number of bytes of data that follow.
`<data(n)>` is the amplitude for the point #n.
The unit is dBm (Plot = Trend) or dB (Plot = FFT).
4-byte little endian floating-point format specified in IEEE 488.2.

The unit of power is selected by the [SENSe]:POWer:UNITs command.

Examples `FETCH:PULSE:STATISTICS?` might return `#264xxxx...` (64-byte data) for the statistics trace of the pulse width measurement when DISPlay:PULSe:SELect:RESult is set to WIDTh.

FETCh:PULSe:STATistics:ATX? (Query Only)

Returns the average transmitted power in the pulse statistics. This command is valid when DISPlay:PULSe:STATistics:PLOT is set to TRENd.

Conditions Measurement views: Pulse statistics

Group Fetch commands

Syntax `FETCh:PULSe:STATistics:ATX?`

Arguments None

Returns `<ATX_avg>,<ATX_min>,<ATX_max>`

Where
`<ATX_avg>::=<NRf>` is the average of the average transmitted power.
`<ATX_min>::=<NRf>` is the minimum of the average transmitted power.
`<ATX_max>::=<NRf>` is the maximum of the average transmitted power.
Unit: dBm.

The unit can be changed to watts by the [SENSe]:POWer:UNITs command.
Volt is invalid in the average transmitted power measurement.

Examples FETCH:PULSE:STATISTICS:ATX? might return -18.51,-18.74,-18.12 for the average transmitted power in the pulse statistics.

FETCh:PULSe:STATistics:AVERage? (Query Only)

Returns the average on power in the pulse statistics. This command is valid when DISPlay:PULSe:STATistics:PLOT is set to TRENd.

Conditions Measurement views: Pulse statistics

Group Fetch commands

Syntax `FETCh:PULSe:STATistics:AVERage?`

Arguments None

Returns `<avg_avg>,<avg_min>,<avg_max>`

Where
`<avg_avg>::=<NRf>` is the average of the average on power.
`<avg_min>::=<NRf>` is the minimum of the average on power.
`<avg_max>::=<NRf>` is the maximum of the average on power.
Unit: dBm.
The unit can be changed to watts by the [SENSe]:POWer:UNITs command.

Examples FETCH:PULSE:STATISTICS:AVERAGE? might return -2.35,-2.36,-2.34 for the average on power in the pulse statistics.

FETCh:PULSe:STATistics:DROop? (Query Only)

Returns the droop in the pulse statistics. This command is valid when DISPlay:PULSe:STATistics:PLOT is set to TRENd.

Conditions Measurement views: Pulse statistics

Group Fetch commands

Syntax `FETCh:PULSe:STATistics:DROop?`

Arguments None

Returns `<droop_avg>,<droop_min>,<droop_max>`

Where
`<droop_avg>::=<NRf>` is the average droop.
`<droop_min>::=<NRf>` is the minimum droop.
`<droop_max>::=<NRf>` is the maximum droop.
Unit: Percent (%) by watts.

Examples `FETCH:PULSE:STATISTICS:DROOP?` might return `22.67E-3,-613.5E-3,633.8E-3` for the droop in the pulse statistics.

FETCh:PULSe:STATistics:DUTPct? (Query Only)

Returns the duty factor (%) in the pulse statistics. This command is valid when DISPlay:PULSe:STATistics:PLOT is set to TRENd.

Conditions Measurement views: Pulse statistics

Group Fetch commands

Syntax `FETCh:PULSe:STATistics:DUTPct?`

Arguments None

Returns `<duty_pct_avg>,<duty_pct_min)>,<duty_pct_max>`

Where
`<duty_pct_avg>::=<NRf>` is the average duty factor.
`<duty_pct_min>::=<NRf>` is the minimum duty factor.
`<duty_pct_max>::=<NRf>` is the maximum duty factor.
Unit: Percent (%).

Examples `FETCH:PULSE:STATISTICS:DUTPCT?` might return `2.437,2.310,2.657` for the duty factor (%) in the pulse statistics.

FETCh:PULSe:STATistics:DUTRatio? (Query Only)

Returns the duty factor (ratio) in the pulse statistics. This command is valid when DISPlay:PULSe:STATistics:PLOT is set to TRENd.

Conditions Measurement views: Pulse statistics

Group Fetch commands

Syntax `FETCh:PULSe:STATistics:DUTRatio?`

Arguments None

Returns `<duty_ratio_avg>,<duty_ratio_min>,<duty_ratio_max)>`

Where
`<duty_ratio_avg>::=<NRf>` is the average duty factor.
`<duty_ratio_min>::=<NRf>` is the minimum duty factor.
`<duty_ratio_max>::=<NRf>` is the maximum duty factor.
Unit: None.

Examples `FETCH:PULSE:STATISTICS:DUTRATIO?` might return `24.37E-3,23.11E-3,26.57E-3` for the duty factor (ratio) in the pulse statistics.

FETCh:PULSe:STATistics:FALL? (Query Only)

Returns the fall time in the pulse statistics. This command is valid when DISPlay:PULSe:STATistics:PLOT is set to TRENd.

Conditions Measurement views: Pulse statistics

Group Fetch commands

Syntax `FETCh:PULSe:STATistics:FALL?`

Arguments None

Returns `<fall_avg>,<fall_min>,<fall_max>`

Where
`<fall_avg>::=<NRf>` is the average fall time.
`<fall_min>::=<NRf>` is the minimum fall time.
`<fall_max>::=<NRf>` is the maximum fall time.
Unit: Seconds.

Examples `FETCH:PULSE:STATISTICS:FALL?` might return `70.27E-9,69.62E-9,71.27E-9` for the fall time in the pulse statistics.

FETCh:PULSe:STATistics:FRDeviation? (Query Only)

Returns the frequency deviation in the pulse statistics. This command is valid when DISPlay:PULSe:STATistics:PLOT is set to TRENd.

Conditions Measurement views: Pulse statistics

Group Fetch commands

Syntax `FETCh:PULSe:STATistics:FRDeviation?`

Arguments None

Returns `<freq_dev_avg>,<freq_dev_min>,<freq_dev_max>`

Where
`<freq_dev_avg>::=<NRf>` is the average frequency deviation.
`<freq_dev_min>::=<NRf>` is the minimum frequency deviation.
`<freq_dev_max>::=<NRf>` is the maximum frequency deviation.
Unit: Hz.

Examples `FETCH:PULSE:STATISTICS:FRDEVIATION?` might return `754.1E+3,660.5E+3,835.2E+3` for the frequency deviation in the pulse statistics.

FETCh:PULSe:STATistics:MFReqerror? (Query Only)

Returns the maximum frequency error in the pulse statistics. This command is valid when DISPlay:PULSe:STATistics:PLOT is set to TRENd.

Conditions Measurement views: Pulse statistics

Group Fetch commands

Syntax `FETCh:PULSe:STATistics:MFReqerror?`

Arguments None

Returns `<max_freq_err_avg>,<max_freq_err_min>,<max_freq_err_max>`

Where
`<max_freq_err_avg>::=<NRf>` is the average of maximum frequency error.
`<max_freq_err_min>::=<NRf>` is the minimum of maximum frequency error.
`<max_freq_err_max>::=<NRf>` is the maximum of maximum frequency error.
Unit: Hz.

Examples `FETCH:PULSE:STATISTICS:MFREQERROR?` might return `645.0E+3,555.6E+3,738.8E+3` for the maximum frequency error in the pulse statistics.

FETCh:PULSe:STATistics:MPHerror? (Query Only)

Returns the maximum phase error in the pulse statistics. This command is valid when DISPlay:PULSe:STATistics:PLOT is set to TRENd.

Conditions Measurement views: Pulse statistics

Group Fetch commands

Syntax `FETCh:PULSe:STATistics:MPHerror?`

Arguments None

Returns `<max_phase_err_avg>,<max_phase_err_min>,<max_phase_err_max>`

Where
`<max_phase_err_avg>::=<NRf>` is the average of maximum phase error.
`<max_phase_err_min>::=<NRf>` is the minimum of maximum phase error.
`<max_phase_err_max>::=<NRf>` is the maximum of maximum phase error.
Unit: Degrees.

Examples `FETCH:PULSE:STATISTICS:MPHERROR?` might return `-11.47,-17.18, -7.61` for the maximum phase error in the pulse statistics.

FETCh:PULSe:STATistics:PHDeviation? (Query Only)

Returns the phase deviation in the pulse statistics. This command is valid when DISPlay:PULSe:STATistics:PLOT is set to TRENd.

Conditions Measurement views: Pulse statistics

Group Fetch commands

Syntax `FETCh:PULSe:STATistics:PHDeviation?`

Arguments None

Returns `<phase_dev_avg>,<phase_dev_min>,<phase_dev_max>`

Where
`<phase_dev_avg>::=<NRf>` is the average phase deviation.
`<phase_dev_min>::=<NRf>` is the minimum phase deviation.
`<phase_dev_max>::=<NRf>` is the maximum phase deviation.
Unit: Degrees.

Examples `FETCH:PULSE:STATISTICS:PHDEVIATION?` might return `11.678,7.694, 17.374` for the phase deviation in the pulse statistics.

FETCh:PULSe:STATistics:PPFRequency? (Query Only)

Returns the pulse-pulse carrier frequency in the pulse statistics. This command is valid when DISPlay:PULSe:STATistics:PLOT is set to TRENd.

Conditions Measurement views: Pulse statistics

Group Fetch commands

Syntax `FETCh:PULSe:STATistics:PPFRequency?`

Arguments None

Returns `<pp_freq_avg>,<pp_freq_min)>,<pp_freq_max>`

Where
`<pp_freq_avg>::=<NRf>` is the average pulse-pulse carrier frequency.
`<pp_freq_min>::=<NRf>` is the minimum pulse-pulse carrier frequency.
`<pp_freq_max>::=<NRf>` is the maximum pulse-pulse carrier frequency.
Unit: Hz.

Examples `FETCH:PULSE:STATISTICS:PPFREQUENCY?` might return `1.135E+3,311.3E+3,-262.8E+3` for the pulse-pulse carrier frequency in the pulse statistics.

FETCh:PULSe:STATistics:PPOWer? (Query Only)

Returns the peak power in the pulse statistics. This command is valid when DISPlay:PULSe:STATistics:PLOT is set to TRENd.

Conditions Measurement views: Pulse statistics

Group Fetch commands

Syntax `FETCh:PULSe:STATistics:PPOWer?`

Arguments None

Returns `<pk_power_avg>,<pk_power_min>,<pk_power_max>`

Where
`<pk_power_avg>::=<NRf>` is the average peak power.
`<pk_power_min>::=<NRf>` is the minimum peak power.
`<pk_power_max>::=<NRf>` is the maximum peak power.
Unit: dBm.
The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples `FETCH:PULSE:STATISTICS:PPOWER?` might return `-2.273,-2.313,-2.235` for the peak power in the pulse statistics.

FETCh:PULSe:STATistics:PPPHase? (Query Only)

Returns the pulse-pulse carrier phase in the pulse statistics. This command is valid when DISPlay:PULSe:STATistics:PLOT is set to TRENd.

Conditions Measurement views: Pulse statistics

Group Fetch commands

Syntax `FETCh:PULSe:STATistics:PPPHase?`

Arguments None

Returns `<pp_phase_avg>,<pp_phase_min)>,<pp_phase_max>`

Where
`<pp_phase_avg>::=<NRf>` is the average pulse-pulse carrier phase.
`<pp_phase_min>::=<NRf>` is the minimum pulse-pulse carrier phase.
`<pp_phase_max>::=<NRf>` is the maximum pulse-pulse carrier phase.
Unit: Degrees.

Examples `FETCH:PULSE:STATISTICS:PPPHASE?` might return `-9.298E-3,-254.3E-3,311.7E-3` for the pulse-pulse carrier phase in the pulse statistics.

FETCh:PULSe:STATistics:RINTerval? (Query Only)

Returns the repetition interval in the pulse statistics. This command is valid when DISPlay:PULSe:STATistics:PLOT is set to TRENd.

Conditions Measurement views: Pulse statistics

Group Fetch commands

Syntax `FETCh:PULSe:STATistics:RINTerval?`

Arguments None

Returns `<rep_int_avg>,<rep_int_min>,<rep_int_max>`

Where
`<rep_int_avg>::=<NRf>` is the average repetition interval.
`<rep_int_min>::=<NRf>` is the minimum repetition interval.
`<rep_int_max>::=<NRf>` is the maximum repetition interval.
Unit: Seconds.

Examples `FETCH:PULSE:STATISTICS:RINTERVAL?` might return `240.5E-6,217.9E-6,281.2E-6` for the repetition interval in the pulse statistics.

FETCh:PULSe:STATistics:RIPPle? (Query Only)

Returns the ripple in the pulse statistics. This command is valid when DISPlay:PULSe:STATistics:PLOT is set to TRENd.

Conditions Measurement views: Pulse statistics

Group Fetch commands

Syntax `FETCh:PULSe:STATistics:RIPPle?`

Arguments None

Returns `<ripple_avg>,<ripple_min>,<ripple_max>`

Where
`<ripple_avg>::=<NRf>` is the average ripple.
`<ripple_min>::=<NRf>` is the minimum ripple.
`<ripple_max>::=<NRf>` is the maximum ripple.
Unit: Percent (%) by volts.

Examples `FETCH:PULSE:STATISTICS:RIPPLE?` might return `160.4E-3,83.78E-3,287.7E-3` for the ripple in the pulse statistics.

FETCh:PULSe:STATistics:RISE? (Query Only)

Returns the rise time in the pulse statistics. This command is valid when DISPlay:PULSe:STATistics:PLOT is set to TRENd.

Conditions Measurement views: Pulse statistics

Group Fetch commands

Syntax `FETCh:PULSe:STATistics:RISE?`

Arguments None

Returns `<rise_avg>,<rise_min>,<rise_max>`

Where
`<rise_avg>::=<NRf>` is the average rise time.
`<rise_min>::=<NRf>` is the minimum rise time.
`<rise_max>::=<NRf>` is the maximum rise time.
Unit: Seconds.

Examples `FETCH:PULSE:STATISTICS:RISE?` might return `105.4E-9,91.65E-9,116.2E-9` for the rise time in the pulse statistics.

FETCh:PULSe:STATistics:RMSFreqerror? (Query Only)

Returns the RMS frequency error in the pulse statistics. This command is valid when DISPlay:PULSe:STATistics:PLOT is set to TRENd.

Conditions Measurement views: Pulse statistics

Group Fetch commands

Syntax `FETCh:PULSe:STATistics:RMSFreqerror?`

Arguments None

Returns `<RMS_freq_err_avg>,<RMS_freq_err_min>,<RMS_freq_err_max>`

Where
`<RMS_freq_err_avg>::=<NRf>` is the average of RMS frequency error.
`<RMS_freq_err_min>::=<NRf>` is the minimum of RMS frequency error.
`<RMS_freq_err_max>::=<NRf>` is the maximum of RMS frequency error.
Unit: Hz.

Examples FETCH:PULSE:STATISTICS:RMSFREQERROR? might return 63.67E+3, 45.49E+3,81.28E+3 for the RMS frequency error in the pulse statistics.

FETCh:PULSe:STATistics:RMSPherror? (Query Only)

Returns the RMS phase error in the pulse statistics. This command is valid when DISPlay:PULSe:STATistics:PLOT is set to TRENd.

Conditions Measurement views: Pulse statistics

Group Fetch commands

Syntax FETCh:PULSe:STATistics:RMSPherror?

Arguments None

Returns <RMS_phase_err_avg>,<RMS_phase_err_min>,<RMS_phase_err_max>

Where
<RMS_phase_err_avg>::=<NRf> is the average of RMS phase error.
<RMS_phase_err_min>::=<NRf> is the minimum of RMS phase error.
<RMS_phase_err_max>::=<NRf> is the maximum of RMS phase error.
Unit: Degrees.

Examples FETCH:PULSE:STATISTICS:RMSPHERROR? might return 1.032,604.5E-3, 1.606 for the RMS phase error in the pulse statistics.

FETCh:PULSe:STATistics:RRATe? (Query Only)

Returns the repetition rate in the pulse statistics. This command is valid when DISPlay:PULSe:STATistics:PLOT is set to TRENd.

Conditions Measurement views: Pulse trace

Group Fetch commands

Syntax FETCh:PULSe:STATistics:RRATe?

Arguments None

Returns `<rep_rate_avg>,<rep_rate_min>,<rep_rate_max>`

Where
`<rep_rate_avg>::=<NRf>` is the average repetition rate.
`<rep_rate_min>::=<NRf>` is the minimum repetition rate.
`<rep_rate_max>::=<NRf>` is the maximum repetition rate.
Unit: Hz.

Examples `FETCH:PULSE:STATISTICS:RRATE?` might return `62.50E+3,62.49E+3,62.52E+3` for the repetition rate in the pulse statistics.

FETCh:PULSe:STATistics:WIDTh? (Query Only)

Returns the pulse width in the pulse statistics. This command is valid when DISPlay:PULSe:STATistics:PLOT is set to TRENd.

Conditions Measurement views: Pulse trace

Group Fetch commands

Syntax `FETCh:PULSe:STATistics:WIDTh?`

Arguments None

Returns `<width_avg>,<width_min>,<width_max>`

Where
`<width_avg>::=<NRf>` is the average pulse width.
`<width_min>::=<NRf>` is the minimum pulse width.
`<width_max>::=<NRf>` is the maximum pulse width.
Unit: Seconds.

Examples `FETCH:PULSE:STATISTICS:WIDTH?` might return `4.636E-6,4.630E-6,4.643E-6` for the pulse width in the pulse statistics.

FETCh:PULSe:TRACe:X? (Query Only)

Returns the time values of the pulse trace. Use the DISPlay:PULSe:SELect:NUMBer command to select the pulse, and the DISPlay:PULSe:SELect:RESult command to select the measurement result.

Conditions Measurement views: Pulse trace

Group Fetch commands

Syntax `FETCh:PULSe:TRACe:X?`

Arguments None

Returns `#<num_digit><num_byte><X(1)><X(2)>...<X(n)>`

Where
`<num_digit>`is the number of digits in <num_byte>.
`<num_byte>` is the number of bytes of data that follow.
`<X(n)>` is the time in seconds at the point #n,
4-byte little endian floating-point format specified in IEEE 488.2.

Examples `FETCH:PULSE:TRACE:X?` might return `#43204xxxx...` (3204-byte data) for the time values of the trace.

FETCh:PULSe:TRACe:XY? (Query Only)

Returns the horizontal (time) and vertical value pairs of the pulse trace. Use the DISPlay:PULSe:SELect:NUMBer command to select the pulse, and the DISPlay:PULSe:SELect:RESult command to select the measurement result.

Conditions Measurement views: Pulse trace

Group Fetch commands

Syntax `FETCh:PULSe:TRACe:XY?`

Arguments None

Returns `#<num_digit><num_byte><x(1)><y(1)><x(2)><y(2)>...<x(n)><y(n)>`

Where
`<num_digit>` is the number of digits in <num_byte>.
`<num_byte>` is the number of bytes of data that follow.
`<x(n)><y(n)>` is the horizontal value (time in seconds) and vertical value pair for the point #n, 4-byte little endian floating-point format specified in IEEE 488.2.

The vertical unit depends on measurement results: Hz for frequency error and deviation, degrees for phase error and deviation, otherwise dBm. The vertical unit can be changed by the [SENSe]:POWer:UNITs command.

Examples `FETCH:PULSE:TRACE:XY?` might return `#43204xxxx...` (3204-byte data) for the horizontal (time) and vertical value pairs of the pulse trace.

FETCh:PULSe:TRACe[:Y]? (Query Only)

Returns the vertical values of the pulse trace. Use the DISPlay:PULSe:SELect:NUMBer command to select the pulse, and the DISPlay:PULSe:SELect:RESult command to select the measurement result.

Conditions Measurement views: Pulse trace

Group Fetch commands

Syntax `FETCh:PULSe:TRACe[:Y]?`

Arguments None

Returns `#<num_digit><num_byte><y(1)><y(2)>...<y(n)>`

Where
`<num_digit>` is the number of digits in <num_byte>.
`<num_byte>` is the number of bytes of data that follow.
`<y(n)>` is the amplitude (dBm) at the point #n, 4-byte little endian floating-point format specified in IEEE 488.2.

The unit depends on measurement results: Hz for frequency error and deviation, degrees for phase error and deviation, otherwise dBm. The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples `FETCH:PULSE:TRACE:Y?` might return `#43204xxxx...` (3204-byte data) for the vertical values of the pulse trace.

FETCh:RFIN:IQ? (Query Only)

Returns time-domain IQ data for a specific acquisition data record. You can set a range of IQ pairs optionally. The instrument needs to be in stopped mode. If not in stopped mode, it returns the execution error (-200).

Conditions Measurement views: All

Group Fetch commands

Syntax `FETCh:RFIN:IQ? <rec_ID>[,<begin_num>,<end_num>]`

Arguments `<rec_ID>` specifies the acquisition data record ID number.
`<begin_num>` specifies the beginning number of IQ pairs.
`<end_num>` specifies the end number of IQ pairs.

Use the FETCh:RFIN:RECord:IDS? query to get the beginning and end ID's of acquisition data records.

Returns `#<num_digit><num_byte><I(1)><Q(1)><I(2)><Q(2)>...<I(n)><Q(n)>`

Where
`<num_digit>` is the number of digits in <num_byte>.
`<num_byte>` is the number of bytes of data that follow.
`<I(n)>` and `<Q(n)>` are the time-domain IQ data pair. 4-byte little endian floating-point format specified in IEEE 488.2.

To find out the range of IQ pairs, use the FETCh:RFIN:IQ:HEADer? query. The returned value of <num_sample> is the number of IQ pairs.

NOTE. *Do not fetch a large number of IQ pairs because of memory limitation.*

Examples `FETCH:RFIN:IQ? 10,25,350` might return `#43848xxxx...` (3848-byte data) for the record #10, ranging from 25th to 350th IQ pair.

FETCh:RFIN:IQ:HEADer? (Query Only)

Returns the header information for a specific acquisition data record. The instrument needs to be in stopped mode. If not in stopped mode, it returns the execution error (-200).

Conditions Measurement views: All

Group Fetch commands

Syntax `FETCh:RFIN:IQ:HEADer? <rec_ID>`

Related Commands

Arguments `<rec_ID>` specifies the acquisition data record ID number.

Use the FETCh:RFIN:RECord:IDS? query to get the beginning and end ID's of acquisition data records.

Returns `<span>,<sampling_freq>,<num_sample>,<center_freq>,<acq_BW>,<ref_level>,<trig_pos>,<time_stamp>,<FastFrame_ID>`

Where
`<span>::=<NR3>` is the span in Hz.
`<sampling_freq>::=<NR3>` is the sampling frequency in Hz.
`<num_sample>::=<NR3>` is the number of samples.
`<center_freq>::=<NR3>` is the center frequency in Hz.
`<acq_BW>::=<NR3>` is the acquisition bandwidth in Hz.
`<ref_level>::=<NR3>` is the reference level in dBm.
`<trig_pos>::=<NR3>` is the trigger position in seconds.
`<time_stamp>::=<string>` is the time stamp.
`<FastFrame_ID>::=<NR1>` is the Fast Frame ID: Zero or positive number. If the Fast Frame is disabled, the ID will be negative.

Examples `FETCH:RFIN:IQ:HEADER? 10` might return `40.0E+6,50.0E+6,4.027E+3,1.5E+9,40.0E+6,0.0,20.242E-6,"10/31/2007 1118:32 AM",-1`, indicating

Span: 40 MHz,
Sampling frequency: 50 MHz,
Number of samples: 4027,
Center frequency: 1.5 GHz,
Acquisition bandwidth: 40 MHz,
Reference level: 0 dBm,
Trigger position: 20.242 µs,
Time stamp: 10/31/2007 1118:32 AM, and
Fast Frame ID: -1

FETCh:RFIN:IQ:SCALe? (Query Only)

Returns the internal RF linear data scaling factor contained in the .tiq file header. The scaling factor can be used to convert digital IQ output (Option 05) values into real IQ values.

Conditions Measurement views: All

Group Fetch commands

Syntax `FETCh:RFIN:IQ:SCALe?`

Related Commands FETCh:RFIN:RECord:IDS?

Arguments None

Returns `<NR3>` The RF linear data scaling factor.

Examples `FETCH:RFIN:IQ:SCALE?` might return `19.553E-6` for the scaling factor.

FETCh:RFIN:RECord:IDS? (Query Only)

Returns the beginning and end ID numbers of acquisition data.

NOTE. *The instrument needs to be in stopped mode. If not in stopped mode, it returns the execution error (-200).*

Conditions Measurement views: All

Group Fetch commands

Syntax `FETCh:RFIN:RECord:IDS?`

Arguments None

Returns `<begin_ID>,<end_ID>`

Where
`<begin_ID>::=<NR1>` is the beginning ID of acquisition data.
`<end_ID>::=<NR1>` is the end ID of acquisition data.

NOTE. *"-1,-1" is returned when the span changes and the acquisition is armed, but the acquisition has not yet occurred.*

Examples `FETCH:RFIN:RECORD:IDS?` might return `1,147`, indicating the beginning and end ID's of acquisition data are 1 and 147, respectively.

FETCh:SGRam? (Query Only)

Returns trace data of a line in the spectrogram. The line is selected using the TRACe:SGRam:SELect:LINE command.

Conditions Measurement views: Spectrogram

Group Fetch commands

Syntax `FETCh:SGRam?`

Arguments None

Returns `#<num_digit><num_byte><data(1)><data(2)>...<data(n)>`

Where
`<num_digit>` is the number of digits in <num_byte>.
`<num_byte>` is the number of bytes of data that follow.
`<data(n)>` is the amplitude in dBm for the point #n,
4-byte little endian floating-point format specified in IEEE 488.2.
The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples `FETCH:SGRAM?` might return `#43204xxxx...` (3204-byte data) for the line in the spectrogram.

FETCh:SPECtrum:TRACe<x>? (Query Only)

Returns the trace data in the Spectrum measurement.

The parameter <x> = 1 to 5.

NOTE. *TRACe5 (spectrogram) is valid when the spectrum and spectrogram measurements are running.*

Conditions Measurement views: Spectrum

Group Fetch commands

Syntax `FETCh:SPECtrum:TRACe<x>?`

Related Commands TRACe<x>:SPECtrum command subgroup

Arguments None

Returns `#<num_digit><num_byte><data(1)><data(2)>...<data(n)>`

Where
`<num_digit>` is the number of digits in <num_byte>.
`<num_byte>` is the number of bytes of data that follow.
`<data(n)>` is the amplitude in dBm for the point #n,
4-byte little endian floating-point format specified in IEEE 488.2.
The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples `FETCH:SPECTRUM:TRACE1?` might return `#43204xxxx...` (3204-byte data) for Trace 1 in the Spectrum measurement.

FETCh:SPURious:CARRier:POWer? (Query Only)

Returns the carrier power in the Spurious measurement.

Conditions Measurement views: Spurious

Group Fetch commands

Syntax `FETCh:SPURious:CARRier:POWer?`

Arguments None

Returns `<NRf>` Carrier power in dBm.
The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples `FETCH:SPURIOUS:CARRIER:POWER?` might return `4.227`, indicating that the carrier power is 4.227 dBm.

FETCh:SPURious:COUNt? (Query Only)

Returns the number of spurious signals in the Spurious measurement.

Conditions Measurement views: Spurious

Group Fetch commands

Syntax `FETCh:SPURious:COUNt?`

Arguments None

Returns `<NRf>` The number of spurious signals.

Examples `FETCH:SPURIOUS:COUNT?` might return `4`, indicating that the spurious count is 4.

FETCh:SPURious:PASS? (Query Only)

Returns the pass/fail limit test result in the Spurious measurement.

Conditions Measurement views: Spurious

Group Fetch commands

Syntax `FETCh:SPURious:PASS?`

Arguments None

Returns `0` (fail) or `1` (pass).

Examples `FETCH:SPURIOUS:PASS?` might return `1`, indicating that the limit test was successful.

FETCh:SPURious:SPECtrum:X? (Query Only)

Returns the frequencies of the spectrum trace in the Spurious measurement.

Conditions Measurement views: Spurious

Group Fetch commands

Syntax `FETCh:SPURious:SPECtrum:X?`

Arguments None

Returns `#<num_digit><num_byte><x(1)><x(2)>...<x(n)>`

Where
`<num_digit>` is the number of digits in <num_byte>.
`<num_byte>` is the number of bytes of data that follow.
`<x(n)>` is the frequency (Hz) at the point #n,
4-byte little endian floating-point format specified in IEEE 488.2.

Examples `FETCH:SPURIOUS:SPECTRUM:X?` might return `#516020xxxx...` (16020-byte data) for the frequencies of the spectrum trace in the Spurious measurement.

FETCh:SPURious:SPECtrum:XY? (Query Only)

Returns the frequency and amplitude pairs of the spectrum trace in the Spurious measurement.

Conditions Measurement views: Spurious

Group Fetch commands

Syntax `FETCh:SPURious:SPECtrum:XY?`

Arguments None

Returns `#<num_digit><num_byte><x(1)><y(1)><x(2)><y(2)>...<x(n)><y(n)>`

Where
`<num_digit>` is the number of digits in <num_byte>.
`<num_byte>` is the number of bytes of data that follow.
`<x(n)><y(n)>` is the frequency (Hz) and amplitude (dBm) pair for the point #n,
4-byte little endian floating-point format specified in IEEE 488.2.

The amplitude unit can be changed by the [SENSe]:POWer:UNITs command.

Examples `FETCH:SPURIOUS:SPECTRUM:XY?` might return `#516020xxxx...` (16020-byte data) for the frequency and amplitude pairs of the spectrum trace in the Spurious measurement.

FETCh:SPURious:SPECtrum[:Y]? (Query Only)

Returns the amplitudes of the spectrum trace in the Spurious measurement.

Conditions Measurement views: Spurious

Group Fetch commands

Syntax `FETCh:SPURious:SPECtrum[:Y]?`

Arguments None

Returns `#<num_digit><num_byte><y(1)><y(2)>...<y(n)>`

Where
`<num_digit>` is the number of digits in <num_byte>.
`<num_byte>` is the number of bytes of data that follow.
`<y(n)>` is the amplitude (dBm) at the point #n,
4-byte little endian floating-point format specified in IEEE 488.2.
The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples `FETCH:SPURIOUS:SPECTRUM:Y?` might return `#516020xxxx...` (16020-byte data) for the amplitudes of the spectrum trace in the Spurious measurement.

FETCh:SPURious:SPUR<x>:AMPLitude:ABSolute? (Query Only)

Returns the absolute amplitude of the specified spurious signal in the Spurious measurement.

Conditions Measurement views: Spurious

Group Fetch commands

Syntax `FETCh:SPURious:SPUR<x>:AMPLitude:ABSolute?`

Arguments None

Returns `<NRf>` Absolute amplitude of the specified spurious signal in dBm.
The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples `FETCH:SPURIOUS:SPUR1:AMPLITUDE:ABSOLUTE?` might return `-19.782`, indicating that the absolute amplitude of Spurious #1 is -19.782 dBm.

FETCh:SPURious:SPUR<x>:AMPLitude:RELative? (Query Only)

Returns the relative amplitude of the specified spurious signal in the Spurious measurement.

Conditions Measurement views: Spurious

Group Fetch commands

Syntax `FETCh:SPURious:SPUR<x>:AMPLitude:RELative?`

Arguments None

Returns `<NRf>` Relative amplitude of the specified spurious signal in dB. Use the [SENSe]:SPURious:REFerence command to set the power reference.

Examples `FETCH:SPURIOUS:SPUR1:AMPLITUDE:RELATIVE?` might return `-9.782`, indicating that the relative amplitude of Spurious #1 is -9.782 dB.

FETCh:SPURious:SPUR<x>:FREQuency:ABSolute? (Query Only)

Returns the absolute frequency of the specified spurious signal in the Spurious measurement.

Conditions Measurement views: Spurious

Group Fetch commands

Syntax `FETCh:SPURious:SPUR<x>:FREQuency:ABSolute?`

Arguments None

Returns `<NRf>` Absolute frequency of the spurious signal in Hz.

Examples `FETCH:SPURIOUS:SPUR1:FREQUENCY:ABSOLUTE?` might return `2.235E+9`, indicating that the absolute frequency of Spurious #1 is 2.235 GHz.

FETCh:SPURious:SPUR<x>:FREQuency:RELative? (Query Only)

Returns the relative frequency of the specified spurious signal to the carrier in the Spurious measurement. This command is valid when [SENSe]:SPURious:REFerence is set to CARRier.

Conditions Measurement views: Spurious

Group Fetch commands

Syntax `FETCh:SPURious:SPUR<x>:FREQuency:RELative?`

Arguments None

Returns `<NRf>` Relative frequency of the spurious signal to the carrier in Hz.

Examples `FETCH:SPURIOUS:SPUR1:FREQUENCY:RELATIVE` might return `3.634E+6`, indicating that the relative frequency of Spurious #1 is 3.634 MHz.

FETCh:SPURious:SPUR<x>:LIMit:ABSolute? (Query Only)

Returns the absolute amplitude of the limit for the specified spurious signal in the Spurious measurement.

Conditions Measurement views: Spurious

Group Fetch commands

Syntax `FETCh:SPURious:SPUR<x>:LIMit:ABSolute?`

Arguments None

Returns `<NRf>` Absolute amplitude of the limit for the specified spurious signal in dBm. The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples `FETCH:SPURIOUS:SPUR1:LIMIT:ABSOLUTE?` might return `-50.0`, indicating that the absolute amplitude of the limit for Spurious #1 is -50 dBm.

FETCh:SPURious:SPUR<x>:LIMit:RELative? (Query Only)

Returns the relative amplitude of the limit for the specified spurious signal in the Spurious measurement.

Conditions Measurement views: Spurious

Group Fetch commands

Syntax `FETCh:SPURious:SPUR<x>:LIMit:RELative?`

Arguments None

Returns `<NRf>` Relative amplitude of the limit for the specified spurious signal in dB. Use the [SENSe]:SPURious:REFerence command to set the power reference.

Examples `FETCH:SPURIOUS:SPUR1:LIMIT:RELATIVE?` might return `-10.0`, indicating that the relative amplitude of the limit for Spurious #1 is -10 dB.

FETCh:SPURious:SPUR<x>:LIMit:VIOLation? (Query Only)

Returns whether the specified spurious signal exceeds the limit or not.

Conditions Measurement views: Spurious

Group Fetch commands

Syntax `FETCh:SPURious:SPUR<x>:LIMit:VIOLation?`

Arguments None

Returns `0` (under the limit) or `1` (over the limit).

Examples `FETCH:SPURIOUS:SPUR1:LIMIT:VIOLATION?` might return `1`, indicating that Spurious #1 exceeds the limit.

FETCh:SPURious:SPUR<x>:RANGe? (Query Only)

Returns the frequency range in which the specified spurious signal occurred.

Conditions Measurement views: Spurious

Group Fetch commands

Syntax `FETCh:SPURious:SPUR<x>:RANGe?`

Arguments None

Returns `<string>` "A" to "T" representing Range A to T, respectively.

Examples `FETCH:SPURIOUS:SPUR1:RANGE` might return `"E"`, indicating that Spurious #1 is in Range E.

FETCh:SQUality:FREQuency:ERRor? (Query Only)

Returns the frequency error in the signal quality measurement.

Conditions Measurement views: Signal quality

Group Fetch commands

Syntax `FETCh:SQUality:FREQuency:ERRor?`

Arguments None

Returns <NRf> Frequency error in Hz.

Examples `FETCH:SQUALITY:FREQUENCY:ERROR?` might return `612.043E+3`, indicating that the frequency error is 612.043 kHz.

FETCh:SQUality:GAIN:IMBalance? (Query Only)

Returns the gain imbalance in the signal quality measurement.

Conditions Measurement views: Signal quality

Group Fetch commands

Syntax `FETCh:SQUality:GAIN:IMBalance?`

Arguments None

Returns <NRf> Gain imbalance in dB.

Examples `FETCH:SQUALITY:GAIN:IMBALANCE?` might return `-57.746E-3`, indicating that the gain imbalance is -0.057746 dB.

FETCh:SQUality:ORIGin:OFFSet? (Query Only)

Returns the origin offset in the signal quality measurement.

Conditions Measurement views: Signal quality

Group Fetch commands

Syntax `FETCh:SQUality:ORIGin:OFFSet?`

Arguments None

Returns `<NRf>` Origin offset in dB.

Examples `FETCH:SQUALITY:ORIGIN:OFFSET?` might return `-44.968`, indicating that the origin offset is -44.968 dB.

FETCh:SQUality:PEAK:EVM? (Query Only)

Returns the peak EVM (%) in the signal quality measurement.

Conditions Measurement views: Signal quality

Group Fetch commands

Syntax `FETCh:SQUality:PEAK:EVM?`

Arguments None

Returns `<NRf>` Peak EVM in percent (%).

Examples `FETCH:SQUALITY:PEAK:EVM?` might return `4.276`, indicating that the peak EVM is 4.276%.

FETCh:SQUality:PEAK:EVM:DB? (Query Only)

Returns the peak EVM (dB) in the signal quality measurement.

Conditions Measurement views: Signal quality

Group Fetch commands

Syntax `FETCh:SQUality:PEAK:EVM:DB?`

Arguments None

Returns `<NRf>` Peak EVM in dB.

Examples `FETCH:SQUALITY:PEAK:EVM:DB?` might return `-27.358`, indicating that the peak EVM is -27.358 dB.

FETCh:SQUality:PEAK:EVM:LOCation? (Query Only)

Returns the time at which the EVM is peak.

Conditions Measurement views: Signal quality

Group Fetch commands

Syntax `FETCh:SQUality:PEAK:EVM:LOCation?`

Arguments None

Returns `<NRf>` The time in symbol number at which the EVM is peak. The unit can be changed by the [SENSe]:DDEMod:TIMe:UNITs command.

Examples `FETCH:SQUALITY:PEAK:EVM:LOCATION?` might return `68.000`, indicating that the EVM is peak at symbol #68.000.

FETCh:SQUality:PEAK:MERRor? (Query Only)

Returns the peak magnitude error (%) in the signal quality measurement.

Conditions Measurement views: Signal quality

Group Fetch commands

Syntax `FETCh:SQUality:PEAK:MERRor?`

Arguments None

Returns `<NRf>` Peak magnitude error in percent (%).

Examples `FETCH:SQUALITY:PEAK:MERROR?` might return `3.595`, indicating that the peak magnitude error is 3.595%.

FETCh:SQUality:PEAK:MERRor:DB? (Query Only)

Returns the peak magnitude error (dB) in the signal quality measurement.

Conditions Measurement views: Signal quality

Group Fetch commands

Syntax `FETCh:SQUality:PEAK:MERRor:DB?`

Arguments None

Returns `<NRf>` Peak magnitude error in dB.

Examples `FETCH:SQUALITY:PEAK:MERROR:DB?` might return `-28.583`, indicating that the magnitude error is -28.583 dB.

FETCh:SQUality:PEAK:MERRor:LOCation? (Query Only)

Returns the time at which the magnitude error is peak.

Conditions Measurement views: Signal quality

Group Fetch commands

Syntax `FETCh:SQUality:PEAK:MERRor:LOCation?`

Arguments None

Returns `<NRf>` The time in symbol number at which the magnitude error is peak. The unit can be changed by the [SENSe]:DDEMod:TIMe:UNITs command.

Examples `FETCH:SQUALITY:PEAK:MERROR:LOCATION?` might return `68.000`, indicating that the magnitude error is peak at symbol #68.

FETCh:SQUality:PEAK:PERRor? (Query Only)

Returns the peak phase error in the signal quality measurement.

Conditions Measurement views: Signal quality

Group Fetch commands

Syntax `FETCh:SQUality:PEAK:PERRor?`

Arguments None

Returns `<NRf>` Peak phase error in degrees.

Examples `FETCH:SQUALITY:PEAK:PERROR?` might return `1.907`, indicating that the peak phase error is 1.907°.

FETCh:SQUality:PEAK:PERRor:LOCation? (Query Only)

Returns the time at which the phase error is peak.

Conditions Measurement views: Signal quality

Group Fetch commands

Syntax `FETCh:SQUality:PEAK:PERRor:LOCation?`

Arguments None

Returns `<NRf>` The time in symbol number at which the phase error is peak.
The unit can be changed by the [SENSe]:DDEMod:TIMe:UNITs command.

Examples `FETCH:SQUALITY:PEAK:PERROR:LOCATION?` might return `68.000`, indicating that the phase error is peak at symbol #68.

FETCh:SQUality:QUADrature:ERRor? (Query Only)

Returns the quadrature error in the signal quality measurement.

Conditions Measurement views: Signal quality

Group Fetch commands

Syntax `FETCh:SQUality:QUADrature:ERRor?`

Arguments None

Returns `<NRf>` Quadrature error in degrees.

Examples `FETCH:SQUALITY:QUADRATURE:ERROR?` might return `-14.264E-3`, indicating that the quadrature error is -0.014264°.

FETCh:SQUality:RHO? (Query Only)

Returns the ρ (waveform quality) value in the signal quality measurement.

Conditions Measurement views: Signal quality

Group Fetch commands

Syntax `FETCh:SQUality:RHO?`

Arguments None

Returns `<NRf>` ρ value.

Examples `FETCH:SQUALITY:RHO?` might return `998.703E-3`, indicating that ρ is 0.998703.

FETCh:SQUality:RMS:EVM? (Query Only)

Returns the RMS EVM (%) in the signal quality measurement.

Conditions Measurement views: Signal quality

Group Fetch commands

Syntax `FETCh:SQUality:RMS:EVM?`

Arguments None

Returns `<NRf>` RMS EVM in percent (%).

Examples `FETCH:SQUALITY:RMS:EVM?` might return `2.417`, indicating that the RMS EVM is 2.417%.

FETCh:SQUality:RMS:EVM:DB? (Query Only)

Returns the RMS EVM (dB) in the signal quality measurement.

Conditions Measurement views: Signal quality

Group Fetch commands

Syntax `FETCh:SQUality:RMS:EVM:DB?`

Arguments None

Returns `<NRf>` RMS EVM in dB.

Examples `FETCH:SQUALITY:RMS:EVM:DB?` might return `-32.356`, indicating that the RMS EVM is -32.356 dB.

FETCh:SQUality:RMS:MER:DB? (Query Only)

Returns the RMS MER (Modulation Error Ratio) in dB in the signal quality measurement.

Conditions Measurement views: Signal quality

Group Fetch commands

Syntax `FETCh:SQUality:RMS:MER:DB?`

Arguments None

Returns `<NRf>` RMS MER in dB.

Examples `FETCH:SQUALITY:RMS:MER:DB?` might return `27.394`, indicating that the RMS MER is 27.394 dB.

FETCh:SQUality:RMS:MERRor? (Query Only)

Returns the RMS magnitude error (%) in the signal quality measurement.

Conditions Measurement views: Signal quality

Group Fetch commands

Syntax `FETCh:SQUality:RMS:MERRor?`

Arguments None

Returns `<NRf>` RMS magnitude error in percent (%).

Examples `FETCH:SQUALITY:RMS:MERROR?` might return `1.837`, indicating that the RMS magnitude error is 1.837%.

FETCh:SQUality:RMS:MERRor:DB? (Query Only)

Returns the RMS magnitude error (dB) in the signal quality measurement.

Conditions Measurement views: Signal quality

Group Fetch commands

Syntax `FETCh:SQUality:RMS:MERRor:DB?`

Arguments None

Returns `<NRf>` RMS MERRor in dB.

Examples `FETCH:SQUALITY:RMS:MERROR:DB?` might return `-34.706`, indicating that the magnitude error is -34.706 dB.

FETCh:SQUality:RMS:PERRor? (Query Only)

Returns the RMS phase error in the signal quality measurement.

Conditions Measurement views: Signal quality

Group Fetch commands

Syntax `FETCh:SQUality:RMS:PERRor?`

Arguments None

Returns `<NRf>` RMS phase error in degrees.

Examples `FETCH:SQUALITY:RMS:PERROR?` might return `893.472E-3`, indicating that the RMS phase error is 0.893472°.

FETCh:TOVerview? (Query Only)

Returns the trace data in the time overview.

Conditions Measurement views: Time overview

Group Fetch commands

Syntax `FETCh:TOVerview?`

Arguments None

Returns `#<num_digit><num_byte><data(1)><data(2)>...<data(n)>`

Where
`<num_digit>` is the number of digits in <num_byte>.
`<num_byte>` is the number of bytes of data that follow.
`<data(n)>` is the amplitude in dBm for the point #n,
4-byte little endian floating-point format specified in IEEE 488.2.
The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples `FETCH:TOVERVIEW?` might return `#43204xxxx...` (3204-byte data) for the trace in the time overview.

*IDN? (Query Only)

Returns the analyzer identification code.

Conditions Measurement views: All

Group IEEE common commands

Syntax `*IDN?`

Arguments None

Returns The analyzer identification code in the following format

`TEKTRONIX,RSA61XXA,<serial_number>,<firmware_version>`

Where
`TEKTRONIX` indicates that the manufacturer is Tektronix.
`RSA61XXA` is the model number (RSA6106A or RSA6114A).
<serial_number> is the serial number.
<firmware_version> is the firmware version.

Examples

`*IDN?` might return the response `TEKTRONIX,RSA6114A,B000111,FV1.0.1500`.

INITiate:CONTinuous

Determines whether to place the analyzer in the single or the continuous acquisition mode.

Conditions

Measurement views: All

Group

Initiate commands

Syntax

```
INITiate:CONTinuous { OFF | ON | 0 | 1 }
INITiate:CONTinuous?
```

Related Commands

INITiate[:IMMediate]

Arguments

`OFF` or `0` places the analyzer in the single acquisition mode. To initiate the acquisition, use the INITiate[:IMMediate] command. To stop the acquisition because the trigger does not occur in the single mode, send the following command INITiate:CONTinuous OFF.

`ON` or `1` places the analyzer in the continuous acquisition mode. To initiate the acquisition, use the INITiate[:IMMediate] command. To stop the acquisition in the continuous mode, send the following command: INITiate:CONTinuous OFF

Examples

`INITIATE:CONTINUOUS ON` places the analyzer in the continuous acquisition mode.

INITiate[:IMMediate] (No Query Form)

Starts input signal acquisition.

*NOTE. It is an overlapped command, which does not finish executing before the next command starts executing. Use the *OPC(?) and *WAI commands to synchronize all pending operations to the execution of this command.*

Conditions Measurement views: All

Group Initiate commands

Syntax `INITiate[:IMMediate]`

Related Commands *OPC, *TRG, *WAI, INITiate:CONTinuous

Arguments None

Examples `INITIATE:IMMEDIATE` starts input signal acquisition.

INPut:{MLEVel|RLEVel}

Sets or queries the reference level.

Conditions Measurement views: All

Group Input commands

Syntax
```
INPut:{MLEVel|RLEVel} <value>
INPut:{MLEVel|RLEVel}?
```

Arguments `<value>::=<NRf>` specifies the reference level. Range: -170 to +50 dBm.

Examples `INPUT:RLEVEL -10` sets the reference level to -10 dBm.

INPut[:RF]:ATTenuation

Sets or queries the input attenuation. Programming a specified attenuation sets INPut[:RF]:ATTenuation:AUTO OFF.

Conditions Measurement views: All

Group Input commands

Syntax

```
INPut[:RF]:ATTenuation <value>
INPut[:RF]:ATTenuation?
```

Related Commands INPut[:RF]:ATTenuation:AUTO

Arguments `<value>::=<NR1>` specifies the input attenuation.
Range: 0 to 75 dB in 5 dB steps.

Examples `INPUT:RF:ATTENUATION 20` sets the input attenuation to 20 dB.

INPut[:RF]:ATTenuation:AUTO

Determines whether to set the input attenuation automatically or manually.

Conditions Measurement views: All

Group Input commands

Syntax

```
INPut[:RF]:ATTenuation:AUTO { OFF | ON | 0 | 1 }
INPut[:RF]:ATTenuation:AUTO?
```

Arguments `OFF` or `0` specifies that the input attenuation is set manually.
To set it, use the INPut[:RF]:ATTenuation command.

`ON` or `1` specifies that the input attenuation is set automatically according to the reference level.

Examples `INPUT:RF:ATTENUATION:AUTO ON` specifies that the input attenuation is set automatically.

INPut[:RF]:ATTenuation:MONitor:STATe

Determines whether to enable or disable to monitor attenuator use.

Conditions Measurement views: All

Group Input commands

Syntax
```
INPut[:RF]:ATTenuation:MONitor:STATe { OFF | ON | 0 | 1 }
INPut[:RF]:ATTenuation:MONitor:STATe?
```

Arguments `OFF` or `0` disables to monitor attenuator use.

`ON` or `1` enables to monitor attenuator use.

Examples `INPUT:RF:ATTENUATION:MONITOR:STATE ON` enables to monitor attenuator use.

INPut[:RF]:GAIN:STATe

Determines whether to enable or disable the internal pre-amp.

Conditions Measurement views: All (Option 01 only)

Group Input commands

Syntax
```
INPut[:RF]:GAIN:STATe { OFF | ON | 0 | 1 }
INPut[:RF]:GAIN:STATe?
```

Arguments `OFF` or `0` disables the internal pre-amp.

`ON` or `1` enables the internal pre-amp.

Examples `INPUT:RF:GAIN:STATE ON` enables the internal pre-amp.

MMEMory:AVTime:LOAD:TRACe<x> (No Query Form)

Loads the Amplitude versus Time trace data from the specified file.

The parameter <x> = 1 to 3; Trace 4 (math trace) is invalid.

Conditions Measurement views: Amplitude versus Time

Group Mass memory commands

Syntax `MMEMory:AVTime:LOAD:TRACe<x> <file_name>`

Arguments `<file_name>::=<string>` specifies the file from which to load trace data. The file extension is .AmplVsTime. You can omit the extension.

Examples `MMEMORY:AVTIME:LOAD:TRACE1 "Sample1"` loads Trace 1 data from the *Sample1.AmplVsTime* file.

MMEMory:AVTime:STORe:TRACe<x> (No Query Form)

Stores the Amplitude versus Time trace data in the specified file.

The parameter <x> = 1 to 4; All traces are valid.

Conditions Measurement views: Amplitude versus Time

Group Mass memory commands

Syntax `MMEMory:AVTime:STORe:TRACe<x> <file_name>`

Arguments `<file_name>::=<string>` specifies the file in which to store trace data. The file extension is .AmplVsTime. You can omit the extension.

Examples `MMEMORY:AVTIME:STORE:TRACE1 "Sample1"` stores Trace 1 data in the *Sample1.AmplVsTime* file.

MMEMory:CALibration:LOAD:CORRection:EXTernal:EDIT<x> (No Query Form)

Loads the external loss table from a specified file.

The parameter <x> = 1 to 3 represent External Loss Table 1 to 3, respectively.

Conditions Measurement views: All

Group Mass memory commands

Syntax `MMEMory:CALibration:LOAD:CORRection:EXTernal:EDIT<x> <file_name>`

Arguments `<file_name>::=<string>` specifies the file to load the external loss table from. The file extension is .csv. You can omit the extension.

Examples `MMEMORY:CALIBRATION:LOAD:CORRECTION:EXTERNAL:EDIT1 "Table1"` loads the External Loss Table 1 from the *Table1.csv* file.

MMEMory:CALibration:STORe:CORRection:EXTernal:EDIT<x> (No Query Form)

Stores the external loss table to a specified file.

The parameter <x> = 1 to 3 represent External Loss Table 1 to 3, respectively.

Conditions Measurement views: All

Group Mass memory commands

Syntax `MMEMory:CALibration:STORe:CORRection:EXTernal:EDIT<x> <file_name>`

Arguments `<file_name>::=<string>` specifies the file to store the external loss table to. The file extension is .csv. You can omit the extension.

Examples `MMEMORY:CALIBRATION:STORE:CORRECTION:EXTERNAL:EDIT1 "Table1"` stores the External Loss Table 1 to the *Table1.csv* file.

MMEMory:CCDF:LOAD:TRACe<x> (No Query Form)

Loads the CCDF trace data from the specified file.

Conditions Measurement views: CCDF

Group Mass memory commands

Syntax `MMEMory:CCDF:LOAD:TRACe<x> <file_name>`

Arguments `<file_name>::=<string>` specifies the file from which to load trace data. The file extension is .CCDF. You can omit the extension.

Examples `MMEMORY:CCDF:LOAD:TRACE1 "Sample1"` loads Trace 1 data from the *Sample1.CCDF* file.

MMEMory:CCDF:STORe:TRACe<x> (No Query Form)

Stores the CCDF trace data in the specified file.

Conditions Measurement views: CCDF

Group Mass memory commands

Syntax `MMEMory:CCDF:STORe:TRACe<x> <file_name>`

Arguments `<file_name>::=<string>` specifies the file in which to store trace data. The file extension is .CCDF. You can omit the extension.

Examples `MMEMORY:CCDF:STORE:TRACE1 "Sample1"` stores Trace 1 data in the *Sample1.CCDF* file.

MMEMory:DPSA:LOAD:TRACe<x> (No Query Form)

Loads the DPX spectrum trace data from the specified file.

The parameter <x> = 1, 2, 3, or 5; TRACe4 (math trace) is invalid.

Conditions Measurement views: DPX spectrum

Group Mass memory commands

Syntax `MMEMory:DPSA:LOAD:TRACe<x> <file_name>`

Arguments `<file_name>::=<string>` specifies the file from which to load trace data. The file extension is .dpt. You can omit the extension.

Examples `MMEMORY:DPSA:LOAD:TRACE1 "Sample1"` loads Trace 1 data from the *Sample1.dpt* file.

MMEMory:DPSA:STORe:TRACe<x> (No Query Form)

Stores the DPX spectrum trace data in the specified file.

The parameter <x> = 1 to 5; All traces are valid.

Conditions Measurement views: DPX spectrum

Group Mass memory commands

Syntax `MMEMory:DPSA:STORe:TRACe<x> <file_name>`

Arguments `<file_name>::=<string>` specifies the file in which to store trace data. The file extension is .dpt. You can omit the extension.

Examples `MMEMORY:DPSA:STORE:TRACE1 "Sample1"` stores Trace 1 data in the *Sample1.dpt* file.

MMEMory:FVTime:LOAD:TRACe (No Query Form)

Loads the Frequency versus Time trace data from the specified file.

Conditions Measurement views: Frequency versus Time

Group Mass memory commands

Syntax `MMEMory:FVTime:LOAD:TRACe <file_name>`

Arguments `<file_name>::=<string>` specifies the file from which to load trace data. The file extension is .FreqVsTime. You can omit the extension.

Examples `MMEMORY:FVTIME:LOAD:TRACE "Sample1"` loads the Frequency versus Time trace data from the *Sample1.FreqVsTime* file.

MMEMory:FVTime:STORe:TRACe (No Query Form)

Stores the Frequency versus Time trace data in the specified file.

Conditions Measurement views: Frequency versus Time

Group Mass memory commands

Syntax `MMEMory:FVTime:STORe:TRACe <file_name>`

Arguments `<file_name>::=<string>` specifies the file in which to store trace data. The file extension is .FreqVsTime. You can omit the extension.

Examples `MMEMORY:FVTIME:STORE:TRACE "Sample1"` stores the Frequency versus Time trace data in the *Sample1.FreqVsTime* file.

MMEMory:IQVTime:LOAD:TRACe:I (No Query Form)

Loads I trace data from the specified file.

Conditions Measurement views: RF I&Q versus Time

Group Mass memory commands

Syntax `MMEMory:IQVTime:LOAD:TRACe:I <file_name>`

Arguments `<file_name>::=<string>` specifies the file from which to load trace data. The file extension is .RFIQVsTime. You can omit the extension.

Examples `MMEMORY:IQVTIME:LOAD:TRACE:I "Sample1"` loads the I trace data from the *Sample1.RFIQVsTime* file.

MMEMory:IQVTime:LOAD:TRACe:Q (No Query Form)

Loads Q trace data from the specified file.

Conditions Measurement views: RF I&Q versus Time

Group Mass memory commands

Syntax `MMEMory:IQVTime:LOAD:TRACe:Q <file_name>`

Arguments `<file_name>::=<string>` specifies the file from which to load trace data. The file extension is .RFIQVsTime. You can omit the extension.

Examples `MMEMORY:IQVTIME:LOAD:TRACE:Q "Sample2"` loads the I trace data from the *Sample2.RFIQVsTime* file.

MMEMory:IQVTime:STORe:TRACe:I (No Query Form)

Stores I trace data in the specified file.

Conditions Measurement views: RF I&Q versus Time

Group Mass memory commands

Syntax `MMEMory:IQVTime:STORe:TRACe:I <file_name>`

Arguments `<file_name>::=<string>` specifies the file in which to store trace data. The file extension is .RFIQVsTime. You can omit the extension.

Examples `MMEMORY:IQVTIME:STORE:TRACE:I "Sample1"` stores the I trace data in the *Sample1.RFIQVsTime* file.

MMEMory:IQVTime:STORe:TRACe:Q (No Query Form)

Stores Q trace data in the specified file.

Conditions Measurement views: RF I&Q versus Time

Group Mass memory commands

Syntax `MMEMory:IQVTime:STORe:TRACe:Q <file_name>`

Arguments `<file_name>::=<string>` specifies the file in which to store trace data. The file extension is .RFIQVsTime. You can omit the extension.

Examples `MMEMORY:IQVTIME:STORE:TRACE:Q "Sample2"` stores the Q trace data in the *Sample2.RFIQVsTime* file.

MMEMory:LOAD:IQ (No Query Form)

Loads time-domain IQ waveform into the acquisition memory from a file.

Conditions Measurement views: All

Group Mass memory commands

Syntax `MMEMory:LOAD:IQ <file_name>`

Arguments `<file_name>::=<string>` specifies the file to load IQ data from. The file extension is .tiq. You can omit the extension.

Examples `MMEMORY:LOAD:IQ "IQ1"` loads IQ data from the *IQ1.tiq* file.

MMEMory:LOAD:STATe (No Query Form)

Loads the instrument setup from a specified file for the currently selected view.

Conditions Measurement views: All

Group Mass memory commands

Syntax `MMEMory:LOAD:STATe <file_name>`

Arguments <file_name>::=<string> specifies the file to load the instrument setup from. The file extension is .setup. You can omit the extension.

Examples MMEMORY:LOAD:STATE "STATE1" loads the instrument setup from the *STATE1.setup* file.

MMEMory:LOAD:TRACe (No Query Form)

Loads the trace data from a specified file for the currently selected view.

Conditions Measurement views (See Table 2-32.)

Group Mass memory commands

Syntax

```
MMEMory:LOAD:TRACe <file_name>
```

Arguments <file_name>::=<string> specifies the file to load the trace data from. The file extension is named for the measurement view as shown in the following table. The trace file is not available in some views which are indicated by "NA" in the file extension column.

For the views that allow multiple-trace display, select the trace to load or store using the command listed in the trace selection column in the following table. You can load the specified trace with any trace data that you saved in the view if the traces have the same units. For example, you can load Trace 2 with Trace 1 data that you saved.

Table 2-32: Trace file extension and trace selection command

Display group	Measurement view	File extension	Trace selection
General signal viewing	Spectrum	Specan	TRACe<x>:SPECtrum:SELect
	DPX spectrum	dpt	TRACe<x>:DPSA:SELect
	Amplitude versus Time	AmplVsTime	TRACe<x>:AVTime:SELect
	Frequency versus Time	FreqVsTime	NA
	Phase versus Time	PhaseVsTime	NA
	RF I&Q versus Time	RFIQVsTime	NA
	Spectrogram	Sogram	NA
	Time overview	NA	NA

Table 2-32: Trace file extension and trace selection command (cont.)

Display group	Measurement view	File extension	Trace selection
General purpose digital modulation (Option 21 only)	Constellation	NA	NA
	EVM versus Time	NA	NA
	Magnitude error versus Time	NA	NA
	Phase error versus Time	NA	NA
	Signal quality	NA	NA
	Symbol table	NA	NA
RF measurements	CCDF	CCDF	TRACe<x>:CCDF:SELect
	Channel power and ACPR	NA	NA
	MCPR	NA	NA
	Occupied Bandwidth	NA	NA
	Phase noise (Option 11 only)	PhaseNoise	TRACe<x>:PNOise:SELect
	Spurious	NA	NA
Pulsed RF (Option 20 only)	Pulse statistics	NA	NA
	Pulse table	NA	NA
	Pulse trace	NA	NA

Examples

`MMEMORY:LOAD:TRACE "TRACE1"` loads the trace data from the *TRACE1.Speca(checkmark)* file when the spectrum view is selected.

MMEMory:PHVTime:LOAD:TRACe (No Query Form)

Loads the Phase versus Time trace data from the specified file.

Conditions

Measurement views: Phase versus Time

Group

Mass memory commands

Syntax

```
MMEMory:PHVTime:LOAD:TRACe <file_name>
```

Arguments

`<file_name>::=<string>` specifies the file from which to load trace data. The file extension is .PhaseVsTime. You can omit the extension.

Examples

`MMEMORY:PHVTIME:LOAD:TRACE "Sample1"` loads the Phase versus Time trace data from the *Sample1.PhaseVsTime* file.

MMEMory:PHVTime:STORe:TRACe (No Query Form)

Stores the Phase versus Time trace data in the specified file.

Conditions Measurement views: Phase versus Time

Group Mass memory commands

Syntax `MMEMory:PHVTime:STORe:TRACe <file_name>`

Arguments `<file_name>::=<string>` specifies the file in which to store trace data. The file extension is .PhaseVsTime. You can omit the extension.

Examples `MMEMORY:PHVTIME:STORE:TRACE "Sample1"` stores the Phase versus Time trace data in the *Sample1.PhaseVsTime* file.

MMEMory:SPECtrum:LOAD:TRACe<x> (No Query Form)

Loads the spectrum trace data from the specified file.

The parameter <x> = 1 to 3; Trace 4 (math trace) and Trace 5 (spectrogram) are invalid.

Conditions Measurement views: Spectrum

Group Mass memory commands

Syntax `MMEMory:SPECtrum:LOAD:TRACe<x> <file_name>`

Arguments `<file_name>::=<string>` specifies the file from which to load trace data. The file extension is .Specan. You can omit the extension.

Examples `MMEMORY:SPECTRUM:LOAD:TRACE1 "Sample1"` loads Trace 1 data from the *Sample1. Speca(checkmark)* file.

MMEMory:SPECtrum:STORe:TRACe<x> (No Query Form)

Stores the spectrum trace data in the specified file.

The parameter <x> = 1 to 5; All traces are valid.

NOTE. *TRACe5 (spectrogram) is valid when the spectrum and spectrogram measurements are running.*

Conditions Measurement views: Spectrum

Group Mass memory commands

Syntax

```
MMEMory:SPECtrum:STORe:TRACe<x> <file_name>
```

Arguments `<file_name>::=<string>` specifies the file in which to store trace data. The file extension is .Specan. You can omit the extension.

Examples `MMEMORY:SPECTRUM:STORE:TRACE1 "Sample1"` stores Trace 1 data in the *Sample1.Speca(checkmark)* file.

MMEMory:SPURious:LOAD:TABLe (No Query Form)

Loads the spurious table containing the limits for enabled ranges from the specified file.

Conditions Measurement views: Spurious

Group Mass memory commands

Syntax

```
MMEMory:SPURious:LOAD:TABLe <file_name>
```

Arguments `<file_name>::=<string>` specifies the file to load the spurious table from. The file extension is .csv. You can omit the extension.

Examples `MMEMORY:SPURIOUS:LOAD:TRACE1 "Table1"` loads the spurious table from the *Table1.csv* file.

MMEMory:SPURious:STORe:TABLe (No Query Form)

Stores the spurious table containing the limits for enabled ranges in a specified file in the CSV (Comma Separated Values) format, allowing you to export the file into Microsoft Excel or other database systems.

Conditions Measurement views: Spurious

Group Mass memory commands

Syntax `MMEMory:SPURious:STORe:TABLe <file_name>`

Arguments `<file_name>::=<string>` specifies the file to store the spurious table in. The file extension is .csv. You can omit the extension.

Examples `MMEMORY:SPURIOUS:STORE:TABLE "Table1"` stores the spurious table in the *Table1.csv* file.

MMEMory:STORe:IQ (No Query Form)

Saves time-domain IQ waveform in the acquisition memory to a specified file.

Conditions Measurement views: All

Group Mass memory commands

Syntax `MMEMory:STORe:IQ <file_name>`

Arguments `<file_name>::=<string>` specifies the file to save IQ data. The file extension is .tiq. You can omit the extension.

Examples `MMEMORY:STORE:IQ "IQ1"` saves IQ data to the *IQ1.tiq* file.

MMEMory:STORe:IQ:CSV (No Query Form)

Saves time-domain IQ waveform in the acquisition memory to a specified file in the CSV (Comma Separated Values) format, allowing you to export the file into Microsoft Excel or other database systems.

Conditions Measurement views: All

Group Mass memory commands

Syntax `MMEMory:STORe:IQ:CSV <file_name>`

Arguments `<file_name>::=<string>` specifies the file to save IQ data.
The file extension is .csv. You can omit the extension.

Examples `MMEMORY:STORE:IQ:CSV "IQ2"` saves IQ data to the *IQ2.cs(less t eq)* file.

MMEMory:STORe:IQ:MAT (No Query Form)

Saves time-domain IQ waveform in the acquisition memory to a specified file in the MATLAB format, allowing you to export the file into the MATLAB technical computing environment.

Conditions Measurement views: All

Group Mass memory commands

Syntax `MMEMory:STORe:IQ:MAT <file_name>`

Arguments `<file_name>::=<string>` specifies the file to save IQ data.
The file extension is .mat. You can omit the extension.

Examples `MMEMORY:STORE:IQ:MAT "IQ3"` saves IQ data to the *IQ3.mat* file.

MMEMory:STORe:MSTate (No Query Form)

Stores the measurement parameters to a specified file in the ASCII text format for the currently selected view, allowing you to export the file into other applications.

Conditions Measurement views: All

Group Mass memory commands

Syntax `MMEMory:STORe:MSTate <file_name>`

Arguments `<file_name>::=<string>` specifies the file to store the measurement parameters. The file extension is .txt. You can omit the extension.

Examples `MMEMORY:STORE:MSTATE "MSTATE1"` stores the measurement parameters to the *MSTATE1.txt* file.

MMEMory:STORe:RESults (No Query Form)

Stores the measurement results including measurement parameters and trace data to a specified file in the CSV (Comma Separated Values) format for the currently selected view, allowing you to export the file into Microsoft Excel or other database systems.

Conditions Measurement views: All

Group Mass memory commands

Syntax `MMEMory:STORe:RESults <file_name>`

Arguments `<file_name>::=<string>` specifies the file to store the measurement results. The file extension is .csv. You can omit the extension.

Examples `MMEMORY:STORE:RESULTS "RESULT1"` stores the measurement results to the *RESULT1.csv* file.

MMEMory:STORe:STATe (No Query Form)

Stores the instrument setup to a specified file for the currently selected view.

Conditions Measurement views: All

Group Mass memory commands

Syntax `MMEMory:STORe:STATe <file_name>`

Arguments `<file_name>::=<string>` specifies the file to store the instrument setup. The file extension is .setup. You can omit the extension.

Examples `MMEMORY:STORE:STATE "STATE1"` stores the instrument setup in the *STATE1.setup* file.

MMEMory:STORe:TRACe (No Query Form)

Stores the trace data in a specified file for the currently selected view.

Conditions Measurement views (See Table 2-32.)

Group Mass memory commands

Syntax `MMEMory:STORe:TRACe <file_name>`

Arguments `<file_name>::=<string>` specifies the file to store the trace data in. The file extension is named for the measurement view as shown in the table (See Table 2-32.) The trace file is not available in some views which are indicated by "NA" in the file extension column. For the views that allow multiple-trace display, select the trace to load or store using the command listed in the trace selection column.

Examples `MMEMORY:STORE:TRACE "TRACE1"` stores the trace data in the *TRACE1.Speca(checkmark)* file when the spectrum view is selected.

*OPC

Generates the operation complete message in the Standard Event Status Register (SESR) when all pending operations finish. The *OPC? query places the ASCII character "1" into the output queue when all pending operations are finished. The *OPC? response is not available to read until all pending operations finish.

The *OPC command allows you to synchronize the operation of the analyzer with your application program. Refer to *Synchronizing Execution*(See page 3-12.) for the details.

Conditions Measurement views: All

Group IEEE common commands

Syntax
```
*OPC
*OPC?
```

Arguments None

*OPT? (Query Only)

Returns a list of options installed in your analyzer.

Conditions Measurement views: All

Group IEEE common commands

Syntax `*OPT?`

Arguments None

Returns The numbers of all the options installed in the analyzer, separated by commas. If no options have been installed, 0 is returned. The following table lists the options for the RSA6100A Series analyzers.

Table 2-33: Instrument options

Option	Description
01	Internal preamp, 5 MHz - 3 GHz, 30 dB gain, 8 dB noise figure at 2 GHz, typical
02	256 Msample deep memory, frequency mask trigger
05	Digital IQ output and 500 MHz analog IF output
06 [1]	Removable HDD, 80 GB
07 [1]	DVD-RW
20	Advanced signal analysis (including pulse measurements)
21	General purpose modulation analysis
110	110 MHz real-time capture bandwidth
1R [1]	Rackmount

[1] These options are not returned from this query.

Examples `*OPT?` might return `02,05,21`, indicating that Option 02, 05, and 21 are currently installed in the analyzer.

OUTPut:IF:{BANDwidth|BWIDth}

Selects or queries the IF output filter when OUTPut:IF[:STATe] is set to ON.

Conditions Measurement views: All (Option 05 only)

Group Output commands

Syntax

```
OUTPut:IF:{BANDwidth|BWIDth} { FLATtop | GAUSsian }
OUTPut:IF:{BANDwidth|BWIDth}?
```

Arguments `FLATtop` selects the flattop filter.

`GAUSsian` selects the Gaussian filter.

Examples `OUTPUT:IF:BANDWIDTH GAUSsian` selects the Gaussian filter for the IF output.

OUTPut:IF[:STATe]

Determines whether to turn on or off the 500 MHz IF Out on the rear panel.

Conditions Measurement views: All (Option 05 only)

Group Output commands

Syntax

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

Related Commands OUTPut:IF:{BANDwidth|BWIDth}

Arguments `OFF` or `0` turns off IF Out.

`ON` or `1` turns on IF Out.

Examples `OUTPUT:IF:STATE ON` turns on IF Out.

OUTPut:IQ[:STATe]

Determines whether to enable or disable the digital IQ output data stream from the rear panel connectors.

Conditions Measurement views: All (Option 05 only)

Group Output commands

Syntax

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

Arguments `OFF` or `0` disables the digital IQ output.

`ON` or `1` enables the digital IQ output.

At *RST, this value is set to OFF.

Examples `OUTPUT:IQ:STATE ON` enables the digital IQ output.

OUTPut:NOISe[:STATe]

Determines whether to turn on or off the +28 V DC Out on the rear panel.

Conditions Measurement views: All

Group Output commands

Syntax

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

Arguments `OFF` or `0` turns off +28 V DC Out.

`ON` or `1` turns on +28 V DC Out.

Examples `OUTPUT:NOISE:STATE ON` turns on +28 V DC Out.

READ:ACPower? (Query Only)

Returns the Channel power and ACPR measurement results for all available channels.

Conditions Measurement views: Channel power and ACPR

Group Read commands

Syntax `READ:ACPower?`

Arguments None

Returns
```
<chan_power>,<acpr_lower(1)>,<acpr_upper(1)>,
<acpr_lower(2)>,<acpr_upper(2)>,...
<acpr_lower(n)>,<acpr_upper(n)>
```

Where
`<chan_power>` is the average power of the main channel as the power reference in dBm. The unit can be changed by the [SENSe]:POWer:UNITs command.
`<acpr_lower(n)>` is the ACPR for the lower channel #n in dB.
`<acpr_upper(n)>` is the ACPR for the upper channel #n in dB.

The number of n depends on the setting of the [SENSe]:ACPower:CHANnel:PAIRs command.

Examples `READ:ACPOWER?` might return `4.227,-28.420,-23.847,-22.316,-29.225`, indicating
(average power of the main channel) = 4.227 dBm,
(ACPR for the lower channel 1) = -28.420 dB,
(ACPR for the upper channel 1) = -23.847 dB,
(ACPR for the lower channel 2) = -22.316 dB, and
(ACPR for the upper channel 2) = -29.225 dB.

READ:ACPower:CHANnel:POWer? (Query Only)

Returns the average power of the main channel (power reference) in the Channel power and ACPR measurement.

Conditions Measurement views: Channel power and ACPR

Group Read commands

Syntax `READ:ACPower:CHANnel:POWer?`

Arguments None

Returns `<chan_power>::=<NRf>` is the average power of the main channel in dBm. The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples `READ:ACPOWER:CHANNEL:POWER?` might return `4.227`, indicating that the average power of the main channel is 4.227 dBm.

READ:ACPower:SPECtrum? (Query Only)

Returns spectrum trace data of the Channel power and ACPR measurement.

Conditions Measurement views: Channel power and ACPR

Group Read commands

Syntax `READ:ACPower:SPECtrum?`

Arguments None

Returns `#<num_digit><num_byte><data(1)><data(2)>...<data(n)>`

Where
`<num_digit>` is the number of digits in <num_byte>.
`<num_byte>` is the number of bytes of data that follow.
`<data(n)>` is the spectrum trace data in dBm for the point n,
4-byte little endian floating-point format specified in IEEE 488.2.
The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples `READ:ACPOWER:SPECTRUM?` might return `#43204xxxx...` (3204-byte data) for the spectrum trace data of the Channel power and ACPR measurement.

READ:AVTime:AVERage? (Query Only)

Returns the RMS (root-mean-square) value for the selected trace in the Amplitude versus Time measurement. Select the trace using the TRACe<x>:AVTime:SELect command.

Conditions Measurement views: Amplitude versus Time

Group Read commands

Syntax `READ:AVTime:AVERage?`

Arguments None

Returns `<avg>::=<NRf>` is the RMS amplitude in dBm.
The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples `READ:AVTIME:AVERAGE?` might return `-2.53`, indicating the RMS amplitude is -2.53 dBm.

READ:AVTime:{FIRSt|SECond|THIRd|FOURth}? (Query Only)

Returns the trace data in the Amplitude versus Time measurement.

The mnemonics FIRSt, SECond, THIRd, and FOURth represent Trace 1, Trace 2, Trace 3, and Math trace, respectively. The traces can be specified by the TRACe<x>:AVTime command subgroup.

Conditions Measurement views: Amplitude versus Time

Group Read commands

Syntax `READ:AVTime:{FIRSt|SECond|THIRd|FOURth}?`

Arguments None

Returns `#<num_digit><num_byte><data(1)><data(2)>...<data(n)>`

Where

<num_digit> is the number of digits in <num_byte>.
<num_byte> is the number of bytes of data that follow.
<data(n)> is the amplitude in dBm for the point #n,
4-byte little endian floating-point format specified in IEEE 488.2.
The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples READ:AVTIME:FIRST? might return #3156xxxx... (156-byte data) for Trace 1.

READ:AVTime:MAXimum? (Query Only)

Returns the maximum value for the selected trace in the Amplitude versus Time measurement. Select the trace using the TRACe<x>:AVTime:SELect command.

Conditions Measurement views: Amplitude versus Time

Group Read commands

Syntax READ:AVTime:MAXimum?

Related Commands READ:AVTime:MAXLocation?

Arguments None

Returns <max>::=<NRf> is the maximum Amplitude in dBm.
The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples READ:AVTIME:MAXIMUM? might return -2.84, indicating the maximum amplitude is -2.84 dBm.

READ:AVTime:MAXLocation? (Query Only)

Returns the time at which the amplitude is maximum for the selected trace in the Amplitude versus Time measurement. Select the trace using the TRACe<x>:AVTime:SELect command.

Conditions Measurement views: Amplitude versus Time

Group Read commands

Syntax `READ:AVTime:MAXLocation?`

Related Commands READ:AVTime:MAXimum?

Arguments None

Returns `<max_time>::=<NRf>` is the time at the maximum in seconds.

Examples `READ:AVTIME:MAXLOCATION?` might return `25.03E-9`, indicating the amplitude is maximum at 25.03 ns.

READ:AVTime:MINimum? (Query Only)

Returns the minimum value for the selected trace in the Amplitude versus Time measurement. Select the trace using the TRACe<x>:AVTime:SELect command.

Conditions Measurement views: Amplitude versus Time

Group Read commands

Syntax `READ:AVTime:MINimum?`

Related Commands READ:AVTime:MINLocation?

Arguments None

Returns `<min>::=<NRf>` is the minimum amplitude in dBm.
The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples `READ:AVTIME:MINIMUM?` might return `-57.64`, indicating the minimum amplitude is -57.64 dBm.

READ:AVTime:MINLocation? (Query Only)

Returns the time at which the amplitude is minimum for the selected trace in the Amplitude versus Time measurement. Select the trace using the TRACe<x>:AVTime:SELect command.

Conditions Measurement views: Amplitude versus Time

Group Read commands

Syntax `READ:AVTime:MINLocation?`

Related Commands READ:AVTime:MINimum?

Arguments None

Returns `<min_time>::=<NRf>` is the time at the minimum in seconds.

Examples `READ:AVTIME:MINLOCATION?` might return `450.7E-9`, indicating the amplitude is minimum at 450.7 ns.

READ:AVTime:RESult? (Query Only)

Returns the measurement results for the selected trace in the Amplitude versus Time measurement. Select the trace using the TRACe<x>:AVTime:SELect command.

Conditions Measurement views: Amplitude versus Time

Group Read commands

Syntax `READ:AVTime:RESult?`

Arguments None

Returns `<max>,<max_time>,<min>,<min_time>,<rms>`

Where
`<max>::=<NRf>` is the maximum amplitude in dBm.
`<max_time>::=<NRf>` is the time at the maximum in seconds.
`<min>::=<NRf>` is the minimum amplitude in dBm.
`<min_time>::=<NRf>` is the time at the minimum in seconds.
`<rms>::=<NRf>` is the RMS amplitude in dBm.
The unit of amplitude can be changed by the [SENSe]:POWer:UNITs command.

Examples

`READ:AVTIME:RESULT?` might return
`-2.68,48.62E-6,-82.47,22.11E-6,-8.24`, indicating
that
the maximum amplitude is -2.68 dBm at 48.62 µs,
the minimum amplitude is -82.47 dBm at 22.11 µs, and
the RMS amplitude is -8.24 dBm.

READ:CCDF? (Query Only)

Returns the CCDF measurement results.

Conditions

Measurement views: CCDF

Group

Read commands

Syntax

`READ:CCDF?`

Arguments

None

Returns

<avg_ampl>,<avg_ccdf>,<crest_factor>,<ampl_10>,<ampl_1>,
<ampl_p1>,<ampl_p01>,<ampl_p001>,<ampl_p0001>

Where
<avg_ampl> is the average amplitude in dBm.
The unit can be changed by the [SENSe]:POWer:UNITs command.
<avg_ccdf> is the average CCDF in percent.
<crest_factor> is the crest factor in dB.
<ampl_10> is the amplitude at CCDF of 10% in dB.
<ampl_1> is the amplitude at CCDF of 1% in dB.
<ampl_p1> is the amplitude at CCDF of 0.1% in dB.
<ampl_p01> is the amplitude at CCDF of 0.01% in dB.
<ampl_p001> is the amplitude at CCDF of 0.001% in dB.
<ampl_p0001> is the amplitude at CCDF of 0.0001% in dB.

Examples

`READ:CCDF?` might return
`-33.35,35.8,9.75,3.88,7.07,8.50,9.25,9.72,9.74`, indicating
(average amplitude) = -33.35 dBm,
(average CCDF) = 35.8%,
(crest factor) = 9.75 dB,
(amplitude at CCDF of 10%) = 3.88 dB,
(amplitude at CCDF of 1%) = 7.07 dB,
(amplitude at CCDF of 0.1%) = 8.50 dB,

(amplitude at CCDF of 0.01%) = 9.25 dB,
(amplitude at CCDF of 0.001%) = 9.72 dB, and
(amplitude at CCDF of 0.0001%) = 9.74 dB.

READ:CCDF:{FIRSt|SECond|THIRd}:X? (Query Only)

Returns the horizontal values of the specified trace in the CCDF measurement.

The mnemonics FIRSt, SECond, and THIRd represent Trace 1, Trace 2, and Gaussian reference curve, respectively.

NOTE. *This query is invalid when [SENSe]:CCDF:TIME:TYPE is set to CONTinuous or TOTal.*

Conditions Measurement views: CCDF

Group Read commands

Syntax `READ:CCDF:{FIRSt|SECond|THIRd}:X?`

Related Commands READ:CCDF:{FIRSt|SECond|THIRd}[:Y]?

Arguments None

Returns `#<num_digit><num_byte><x(1)><x(2)>...<x(n)>`

Where
`<num_digit>` is the number of digits in <num_byte>.
`<num_byte>` is the number of bytes of data that follow.
`<x(n)>` is the horizontal value (dB) of the CCDF graph at the point #n, 4-byte little endian floating-point format specified in IEEE 488.2.

Examples `READ:CCDF:FIRST:X` might return `#41024xxxx...` (1024-byte data) for the horizontal values of Trace 1.

READ:CCDF:{FIRSt|SECond|THIRd}:XY? (Query Only)

Returns the horizontal and vertical value pairs of the specified trace in the CCDF measurement.

The mnemonics FIRSt, SECond, and THIRd represent Trace 1, Trace 2, and Gaussian reference curve, respectively.

***NOTE.** This query is invalid when [SENSe]:CCDF:TIME:TYPE is set to CONTinuous or TOTal.*

Conditions Measurement views: CCDF

Group Read commands

Syntax `READ:CCDF:{FIRSt|SECond|THIRd}:XY?`

Arguments None

Returns `#<num_digit><num_byte><x(1)><y(1)><x(2)><y(2)>...<x(n)><y(n)>`

Where
`<num_digit>` is the number of digits in <num_byte>.
`<num_byte>` is the number of bytes of data that follow.
`<x(n)><y(n)>` is the horizontal value (dB) and vertical value (%) pair for the point #n, 4-byte little endian floating-point format specified in IEEE 488.2.

Examples `READ:CCDF:FIRST:XY?` might return `#41024xxxx...` (1024-byte data) for the horizontal and vertical value pairs of Trace 1.

READ:CCDF:{FIRSt|SECond|THIRd}[:Y]? (Query Only)

Returns the vertical values of the specified trace in the CCDF measurement.

The mnemonics FIRSt, SECond, and THIRd represent Trace 1, Trace 2, and Gaussian reference curve, respectively.

***NOTE.** This query is invalid when [SENSe]:CCDF:TIME:TYPE is set to CONTinuous or TOTal.*

Conditions Measurement views: CCDF

Group Read commands

Syntax READ:CCDF:{FIRSt|SECond|THIRd}[:Y]?

Arguments None

Returns #<num_digit><num_byte><y(1)><y(2)>...<y(n)>
Where
<num_digit> is the number of digits in <num_byte>.
<num_byte> is the number of bytes of data that follow.
<y(n)> is the vertical value (%) of the CCDF graph at the point #n,
4-byte little endian floating-point format specified in IEEE 488.2.

Examples READ:CCDF:FIRST:Y might return #41024xxxx... (1024-byte data) for the vertical values of Trace 1.

READ:CONSte:RESults? (Query Only)

Returns the constellation measurement results of EVM RMS, peak and location displayed on the bottom of the screen.

Conditions Measurement views: Constellation

Group Read commands

Syntax READ:CONSte:RESults?

Arguments None

Returns <EVM_RMS>,<EVM_peak>,<location>

Where
<EVM_RMS>::=<NRf> is the RMS EVM in percent (%).
<EVM_peak>::=<NRf> is the peak EVM in percent (%).
<location>::=<NRf> is the peak location in symbol number.
The time unit can be changed by the [SENSe]:DDEMod:TIMe:UNITs command.

Examples READ:CONSTE:RESULTS? might return 2.841,3.227,68.000, indicating that the RMS EVM is 2.841% and the peak EVM is 3.227% at symbol #68.

READ:CONSte:TRACe? (Query Only)

Returns the constellation trace data.

Conditions Measurement views: Constellation

Group Read commands

Syntax `READ:CONSte:TRACe?`

Arguments None

Returns `#<num_digit><num_byte><I(1)><Q(1)><I(2)><Q(2)>...<I(n)><Q(n)>`

Where
`<num_digit>` is the number of digits in <num_byte>.
`<num_byte>` is the number of bytes of data that follow.
`<I(n)>` and `<Q(n)>` are the normalized I- and Q-coordinate values at the nth data point. 4-byte little endian floating-point format specified in IEEE 488.2.

Examples `READ:CONSTE:TRACE?` might return `#43848xxxx...` (3848-byte data) for the constellation trace data.

READ:DDEMod:STABle? (Query Only)

Returns the symbol table data.

Conditions Measurement views: Symbol table

Group Read commands

Syntax `READ:DDEMod:STABle?`

Arguments None

Returns `#<num_digit><num_byte><data(1)><data(2)>...<data(n)>`

Where
`<num_digit>` is the number of digits in <num_byte>.

<num_byte> is the number of bytes of data that follow.
<data(n)> is the symbol table data for the point n,
4-byte little endian floating-point format specified in IEEE 488.2.

Examples READ:DDEMOD:STABLE? might return #3512xxxx... (512-byte data) for the symbol table.

READ:DPSA:TRACe:AVERage? (Query Only)

Returns waveform data of the average trace in the DPX spectrum measurement.

Conditions Measurement views: DPX spectrum

Group Read commands

Syntax READ:DPSA:TRACe:AVERage?

Arguments None

Returns #<num_digit><num_byte><data(1)><data(2)>...<data(n)>

Where
<num_digit> is the number of digits in <num_byte>.
<num_byte> is the number of bytes of data that follow.
<data(n)> is the waveform data of the average trace for the point n in dBm,
4-byte little endian floating-point format specified in IEEE 488.2.
The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples READ:DPSA:TRACE:AVERAGE? might return #42004xxxx... (2004-byte data) for the waveform data of the average trace.

READ:DPSA:TRACe:MATH? (Query Only)

Returns waveform data of the math trace in the DPX spectrum measurement.

Conditions Measurement views: DPX spectrum

Group Read commands

Syntax `READ:DPSA:TRACe:MATH?`

Arguments None

Returns `#<num_digit><num_byte><data(1)><data(2)>...<data(n)>`

Where
`<num_digit>` is the number of digits in <num_byte>.
`<num_byte>` is the number of bytes of data that follow.
`<data(n)>` is the waveform data of the math trace for the point n in dBm, 4-byte little endian floating-point format specified in IEEE 488.2.
The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples `READ:DPSA:TRACE:MATH?` might return `#42004xxxx...` (2004-byte data) for the waveform data of the math trace.

READ:DPSA:TRACe:MAXimum? (Query Only)

Returns waveform data of the maximum trace in the DPX spectrum measurement.

Conditions Measurement views: DPX spectrum

Group Read commands

Syntax `READ:DPSA:TRACe:MAXimum?`

Arguments None

Returns `#<num_digit><num_byte><data(1)><data(2)>...<data(n)>`

Where
`<num_digit>` is the number of digits in <num_byte>.
`<num_byte>` is the number of bytes of data that follow.
`<data(n)>` is the waveform data of the maximum trace for the point n in dBm, 4-byte little endian floating-point format specified in IEEE 488.2.
The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples `READ:DPSA:TRACE:MAXIMUM?` might return `#42004xxxx...` (2004-byte data) for the waveform data of the maximum trace.

READ:DPSA:TRACe:MINimum? (Query Only)

Returns waveform data of the minimum trace in the DPX spectrum measurement.

Conditions Measurement views: DPX spectrum

Group Read commands

Syntax `READ:DPSA:TRACe:MINimum?`

Arguments None

Returns `#<num_digit><num_byte><data(1)><data(2)>...<data(n)>`

Where
`<num_digit>` is the number of digits in <num_byte>.
`<num_byte>` is the number of bytes of data that follow.
`<data(n)>` is the waveform data of the minimum trace for the point n in dBm, 4-byte little endian floating-point format specified in IEEE 488.2.
The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples `READ:DPSA:TRACE:MINIMUM?` might return `#42004xxxx...` (2004-byte data) for the waveform data of the minimum trace.

READ:EVM:FERRor? (Query Only)

Returns the frequency error in the EVM versus Time measurement.

Conditions Measurement views: EVM versus Time

Group Read commands

Syntax `READ:EVM:FERRor?`

Arguments None

Returns `<freq_error>::=<NRf>` is the frequency error in Hz.

Examples READ:EVM:FERROR? might return -10.7E+3, indicating the frequency error is -10.7 kHz.

READ:EVM:PEAK? (Query Only)

Returns the peak value in the EVM versus Time measurement.

Conditions Measurement views: EVM versus Time

Group Read commands

Syntax READ:EVM:PEAK?

Related Commands READ:EVM:PINDex?

Arguments None

Returns <peak>::=<NRf> is the peak EVM value in percent (%).

Examples READ:EVM:PEAK? might return 1.32, indicating the peak EVM value is 1.32%.

READ:EVM:PINDex? (Query Only)

Returns the time at the EVM peak.

Conditions Measurement views: EVM versus Time

Group Read commands

Syntax READ:EVM:PINDex?

Related Commands READ:EVM:PEAK?

Arguments None

Returns `<peak_time>::=<NRf>` is the time at the EVM peak in symbol number. The unit can be changed by the [SENSe]:DDEMod:TIMe:UNITs command.

Examples `READ:EVM:PINDEX?` might return `68.000`, indicating that the EVM peak is at symbol #68.

READ:EVM:RMS? (Query Only)

Returns the RMS (Root-Mean-Square) value in the EVM versus Time measurement.

Conditions Measurement views: EVM versus Time

Group Read commands

Syntax `READ:EVM:RMS?`

Arguments None

Returns `<rms>::=<NRf>` is the RMS EVM value in percent (%).

Examples `READ:EVM:RMS?` might return `0.582`, indicating the RMS EVM value is 0.582%.

READ:EVM:TRACe? (Query Only)

Returns the EVM versus Time trace data.

Conditions Measurement views: EVM versus Time

Group Read commands

Syntax `READ:EVM:TRACe?`

Arguments None

Returns `#<num_digit><num_byte><data(1)><data(2)>...<data(n)>`

Where
`<num_digit>` is the number of digits in <num_byte>.
`<num_byte>` is the number of bytes of data that follow.
`<data(n)>` is the EVM versus Time trace data for the point n in percent (%), 4-byte little endian floating-point format specified in IEEE 488.2.

Examples `READ:EVM:TRACE?` might return `#42036xxxx...` (2036-byte data) for the EVM versus Time trace.

READ:FVTime? (Query Only)

Returns the Frequency versus Time trace data.

Conditions Measurement views: Frequency versus Time

Group Read commands

Syntax `READ:FVTime?`

Arguments None

Returns `#<num_digit><num_byte><data(1)><data(2)>...<data(n)>`

Where
`<num_digit>` is the number of digits in <num_byte>.
`<num_byte>` is the number of bytes of data that follow.
`<data(n)>` is the frequency drift data for the point n in Hz, 4-byte little endian floating-point format specified in IEEE 488.2.

Examples `READ:FVTIME?` might return `#3156xxxx...` (156-byte data) for the Frequency versus Time trace.

READ:FVTime:MAXimum? (Query Only)

Returns the maximum value in the Frequency versus Time measurement.

Conditions Measurement views: Frequency versus Time

Group Read commands

Syntax READ:FVTime:MAXimum? imum

Related Commands READ:FVTime:MAXLocation?

Arguments None

Returns <max>::=<NRf> is the maximum frequency drift in Hz.

Examples READ:FVTIME:MAXIMUM? might return 2.625E+6, indicating the maximum frequency drift is 2.625 MHz.

READ:FVTime:MAXLocation? (Query Only)

Returns the time at which the frequency drift is maximum.

Conditions Measurement views: Frequency versus Time

Group Read commands

Syntax READ:FVTime:MAXLocation?

Related Commands READ:FVTime:MAXimum?

Arguments None

Returns <max_time>::=<NRf> is the time in seconds at which the frequency drift is maximum.

Examples READ:FVTIME:MAXLOCATION? might return 25.03E-9, indicating the frequency drift is maximum at 25.03 ns.

READ:FVTime:MINimum? (Query Only)

Returns the minimum value in the Frequency versus Time measurement.

Conditions Measurement views: Frequency versus Time

Group Read commands

Syntax `READ:FVTime:MINimum?`

Related Commands READ:FVTime:MINLocation?

Arguments None

Returns `<min>::=<NRf>` is the minimum frequency drift in Hz.

Examples `READ:FVTIME:MINIMUM?` might return `-6.618E+6`, indicating the minimum frequency drift is -6.618 MHz.

READ:FVTime:MINLocation? (Query Only)

Returns the time at which the frequency drift is minimum.

Conditions Measurement views: Frequency versus Time

Group Read commands

Syntax `READ:FVTime:MINLocation?`

Related Commands READ:FVTime:MINimum?

Arguments None

Returns `<min_time>::=<NRf>` is the time in seconds at which the frequency drift is minimum.

Examples `READ:FVTIME:MINLOCATION?` might return `450.7E-9`, indicating the frequency drift is minimum at 450.7 ns.

READ:FVTime:RESult? (Query Only)

Returns the Frequency versus Time measurement results.

Conditions Measurement views: Frequency versus Time

Group Read commands

Syntax `READ:FVTime:RESult?`

Arguments None

Returns `<max>,<max_time>,<min>,<min_time>`

Where
`<max>::=<NRf>` is the maximum frequency drift in Hz.
`<max_time>::=<NRf>` is the time in seconds at which the frequency drift is maximum.
`<min>::=<NRf>` is the minimum frequency drift in Hz.
`<min_time>::=<NRf>` is the time in seconds at which the frequency drift is minimum.

Examples `READ:FVTIME:RESULT?` might return
`2.625E+6,25.03E-9,-6.618E+6,450.7E-9`,
indicating
the maximum frequency drift is 2.625 MHz at 25.03 ns and
the minimum frequency drift is -6.618 MHz at 450.7 ns.

READ:IQVTime:I? (Query Only)

Returns the I versus Time trace data.

Conditions Measurement views: RF I&Q versus Time

Group Read commands

Syntax `READ:IQVTime:I?`

Arguments None

Returns `#<num_digit><num_byte><data(1)><data(2)>...<data(n)>`

Where

`<num_digit>` is the number of digits in <num_byte>.
`<num_byte>` is the number of bytes of data that follow.
`<data(n)>` is the I level data for the point n in volts,
4-byte little endian floating-point format specified in IEEE 488.2.

Examples `READ:IQVTIME:I?` might return `#3160xxxx...` (160-byte data) for the I versus Time trace.

READ:IQVTime:MAXimum? (Query Only)

Returns the maximum value in the RF I&Q versus Time measurement.

Conditions Measurement views: RF I&Q versus Time

Group Read commands

Syntax `READ:IQVTime:MAXimum?  imum`

Related Commands READ:IQVTime:MAXLocation?

Arguments None

Returns `<max>::=<NRf>` is the maximum I or Q level in volts.

Examples `READ:IQVTIME:MAXIMUM?` might return `1.214`, indicating the maximum I or Q level is 1.214 V.

READ:IQVTime:MAXLocation? (Query Only)

Returns the time at which the I or Q level is maximum.

Conditions Measurement views: RF I&Q versus Time

Group Read commands

Syntax `READ:IQVTime:MAXLocation?`

Related Commands READ:IQVTime:MAXimum?

Arguments None

Returns `<max_time>::=<NRf>` is the time in seconds at which the I or Q level is maximum.

Examples `READ:IQVTIME:MAXLOCATION?` might return `175.3E-9`, indicating the I or Q level is maximum at 175.3 ns.

READ:IQVTime:MINimum? (Query Only)

Returns the minimum value in the RF I&Q versus Time measurement.

Conditions Measurement views: RF I&Q versus Time

Group Read commands

Syntax `READ:IQVTime:MINimum?`

Related Commands READ:IQVTime:MINLocation?

Arguments None

Returns `<min>::=<NRf>` is the minimum I or Q level in volts.

Examples `READ:IQVTIME:MINIMUM?` might return `-370.5E-3`, indicating the minimum I or Q level is -370.5 mV.

READ:IQVTime:MINLocation? (Query Only)

Returns the time at which the I or Q level is minimum.

Conditions Measurement views: RF I&Q versus Time

Group Read commands

Syntax `READ:IQVTime:MINLocation?`

Related Commands READ:IQVTime:MINimum?

Arguments None

Returns `<min_time>::=<NRf>` is the time in seconds at which the I or Q level is minimum.

Examples `READ:IQVTIME:MINLOCATION?` might return `450.7E-9`, indicating the I or Q level is minimum at 450.7 ns.

READ:IQVTime:Q? (Query Only)

Returns the Q versus Time trace data.

Conditions Measurement views: IQ versus Time

Group Read commands

Syntax `READ:IQVTime:Q?`

Arguments None

Returns `#<num_digit><num_byte><data(1)><data(2)>...<data(n)>`

Where
`<num_digit>` is the number of digits in <num_byte>.
`<num_byte>` is the number of bytes of data that follow.
`<data(n)>` is the Q level data for the point n in volts,
4-byte little endian floating-point format specified in IEEE 488.2.

Examples `READ:IQVTIME:Q?` might return `#3160xxxx...` (160-byte data) for the Q versus Time trace.

READ:IQVTime:RESult? (Query Only)

Returns the RF I&Q versus Time measurement results.

Conditions Measurement views: RF I&Q versus Time

Group Read commands

Syntax `READ:IQVTime:RESult?`

Arguments None

Returns `<max>,<max_time>,<min>,<min_time>`

Where
`<max>::=<NRf>` is the maximum I or Q level in volts.
`<max_time>::=<NRf>` is the time in seconds at which the I or Q level is maximum.
`<min>::=<NRf>` is the minimum I or Q level in volts.
`<min_time>::=<NRf>` is the time in seconds at which the I or Q level is minimum.

Examples `READ:IQVTIME:RESULT?` might return `1.214,175.3E-9,-370.5E-3,450.7E-9`, indicating the maximum I or Q level is 1.214 V at 175.3 ns and the minimum I or Q level is -370.5 mV at 450.7 ns.

READ:MCPower:ADJacent:CHANnels? (Query Only)

Returns the power of adjacent channels in order of increasing frequency.

Conditions Measurement views: MCPR

Group Read commands

Syntax `READ:MCPower:ADJacent:CHANnels?`

Arguments None

Returns `<acpr_lower(n)>,...<acpr_lower(2)>,<acpr_lower(1)>,`
`<acpr_upper(1)>,<acpr_upper(2)>,...<acpr_upper(n)>`

Where

<acpr_lower(n)> is the ACPR for the lower channel #n in dB.
<acpr_upper(n)> is the ACPR for the upper channel #n in dB.

To add a pair of upper and lower adjacent channels, use the [SENSe]:MCPower:CHANnel:ADJacent:ADD command.

Examples READ:MCPOWER:ADJACENT:CHANNELS? might return -4.420,-4.847,-4.316,-4.225, indicating
(ACPR for the lower channel 2) = -4.420 dB,
(ACPR for the lower channel 1) = -4.847 dB,
(ACPR for the upper channel 1) = -4.316 dB, and
(ACPR for the upper channel 2) = -4.225 dB.

READ:MCPower:CHANnel:POWer? (Query Only)

Returns the reference power in the MCPR measurement.

Conditions Measurement views: MCPR

Group Read commands

Syntax READ:MCPower:CHANnel:POWer?

Arguments None

Returns <ref_power>:<NRf> is the reference power in dBm.
The unit can be changed by the [SENSe]:POWer:UNITs command.

To select the power reference, use the [SENSe]:MCPower:RCHannels commands.

Examples READ:MCPOWER:CHANNEL:POWER? might return 4.227, indicating that the reference power is 4.227 dBm.

READ:MCPower:MAIN:CHANnels? (Query Only)

Returns the power of main channels in order of increasing frequency.

Conditions Measurement views: MCPR

Group Read commands

Syntax `READ:MCPower:MAIN:CHANnels?`

Related Commands [:SENSe]:MCPower:CHANnel:MAIN commands

Arguments None

Returns `<power_main(1)>,<power_main(2)>,...<power_main(n)>`

Where
`<power_main(n)>` is the power of main channel #n in dBm.
The unit can be changed by the[SENSe]:POWer:UNITs command.

To specify the main channels, use the [SENSe]:MCPower:CHANnel:MAIN commands.

Examples `READ:MCPOWER:MAIN:CHANNELS?` might return
`-2.420,-2.847,-2.316,-2.225`, indicating
(power of the main channel 1) = -2.420 dBm,
(power of the main channel 2) = -2.847 dBm,
(power of the main channel 3) = -2.316 dBm, and
(power of the main channel 4) = -2.225 dBm.

READ:MCPower:SPECtrum? (Query Only)

Returns spectrum trace data of the MCPR measurement.

Conditions Measurement views: MCPR

Group Read commands

Syntax `READ:MCPower:SPECtrum?`

Arguments None

Returns `#<num_digit><num_byte><data(1)><data(2)>...<data(n)>`

Where
`<num_digit>` is the number of digits in <num_byte>.

`<num_byte>` is the number of bytes of data that follow.
`<data(n)>` is the spectrum trace data in dBm for the point n,
4-byte little endian floating-point format specified in IEEE 488.2.
The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples `READ:MCPOWER:SPECTRUM?` might return `#43204xxxx...` (3204-byte data) for the spectrum trace data of the MCPR measurement.

READ:MERRor:FERRor? (Query Only)

Returns the frequency error in the Magnitude error versus Time measurement.

Conditions Measurement views: Magnitude error versus Time

Group Read commands

Syntax `READ:MERRor:FERRor?`

Arguments None

Returns `<freq_error>::=<NRf>` is the frequency error in Hz.

Examples `READ:MERROR:FERROR?` might return `-10.7E+3`, indicating the frequency error is -10.7 kHz.

READ:MERRor:PEAK? (Query Only)

Returns the peak value in the Magnitude error versus Time measurement.

Conditions Measurement views: Magnitude error versus Time

Group Read commands

Syntax `READ:MERRor:PEAK?`

Related Commands READ:MERRor:PINDex?

Arguments None

Returns `<peak>::=<NRf>` is the peak magnitude error in percent (%).

Examples `READ:MERROR:PEAK?` might return `1.57`, indicating the peak magnitude error is 1.57%.

READ:MERRor:PINDex? (Query Only)

Returns the time at the magnitude error peak.

Conditions Measurement views: Magnitude error versus Time

Group Read commands

Syntax `READ:MERRor:PINDex?`

Related Commands READ:MERRor:PEAK?

Arguments None

Returns `<peak_time>::=<NRf>` is the time at the magnitude error peak in symbol number. The unit can be changed by the [SENSe]:DDEMod:TIMe:UNITs command.

Examples `READ:MERROR:PINDEX?` might return `68.000`, indicating that the magnitude error peak is at symbol #68.

READ:MERRor:RMS? (Query Only)

Returns the RMS (Root-Mean-Square) value in the Magnitude error versus Time measurement.

Conditions Measurement views: Magnitude error versus Time

Group Read commands

Syntax READ:MERRor:RMS?

Arguments None

Returns <rms>::=<NRf> is the RMS magnitude error in percent (%).

Examples READ:MERROR:RMS? might return 0.382, indicating the magnitude error is 0.382% RMS.

READ:MERRor:TRACe? (Query Only)

Returns the Magnitude error versus Time trace data.

Conditions Measurement views: Magnitude error versus Time

Group Read commands

Syntax READ:MERRor:TRACe?

Arguments None

Returns #<num_digit><num_byte><data(1)><data(2)>...<data(n)>

Where
<num_digit> is the number of digits in <num_byte>.
<num_byte> is the number of bytes of data that follow.
<data(n)> is the magnitude error data for the point n in percent (%), 4-byte little endian floating-point format specified in IEEE 488.2.

Examples READ:MERROR:TRACE? might return #42036xxxx... (2036-byte data) for the Magnitude error versus Time trace.

READ:OBWidth:FREQuency:ERRor? (Query Only)

Returns the frequency error in the Occupied Bandwidth measurement.

Conditions Measurement views: Occupied Bandwidth

Group Read commands

Syntax `READ:OBWidth:FREQuency:ERRor?`

Arguments None

Returns `<freq_error>::=<NRf>` is the frequency error in Hz.

Examples `READ:OBWIDTH:FREQUENCY:ERROR?` might return `-10.7E+3`, indicating the frequency error is -10.7 kHz.

READ:OBWidth:OBWidth:BANDwidth? (Query Only)

Returns the occupied bandwidth in the Occupied Bandwidth measurement.

Conditions Measurement views: Occupied Bandwidth

Group Read commands

Syntax `READ:OBWidth:OBWidth:BANDwidth?`

Arguments None

Returns `<OBW>::=<NRf>` is the occupied bandwidth in Hz.

Examples `READ:OBWIDTH:OBWIDTH:BANDWIDTH?` might return `4.0E+6`, indicating the occupied bandwidth is 4 MHz.

READ:OBWidth:OBWidth:LEFT:FREQuency? (Query Only)

Returns the left (lower) frequency of the occupied bandwidth.

Conditions Measurement views: Occupied Bandwidth

Group Read commands

Syntax READ:OBWidth:OBWidth:LEFT:FREQuency?

Related Commands READ:OBWidth:OBWidth:RIGHt:FREQuency?

Arguments None

Returns <OBW_left_freq>::=<NRf> is the left frequency in Hz.

Examples READ:OBWIDTH:OBWIDTH:LEFT:FREQUENCY? might return 1.498E+9, indicating the left frequency is 1.498 GHz.

READ:OBWidth:OBWidth:LEFT:LEVel? (Query Only)

Returns the level at the left frequency of the occupied bandwidth.

Conditions Measurement views: Occupied Bandwidth

Group Read commands

Syntax READ:OBWidth:OBWidth:LEFT:LEVel?

Related Commands READ:OBWidth:OBWidth:RIGHt:LEVel?

Arguments None

Returns <OBW_left_level>::=<NRf> is the level at the left frequency in dB.

Examples READ:OBWIDTH:OBWIDTH:LEFT:LEVEL? might return -23.5, indicating the level at the left frequency is -23.5 dB.

READ:OBWidth:OBWidth:POWer? (Query Only)

Returns the reference power in the Occupied Bandwidth measurement.

Conditions Measurement views: Occupied Bandwidth

Group Read commands

Syntax `READ:OBWidth:OBWidth:POWer?`

Arguments None

Returns `<OBW_ref_power>::=<NRf>` is the reference power in dBm.
The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples `READ:OBWIDTH:OBWIDTH:POWER?` might return `-10.0`, indicating the reference power is -10 dBm.

READ:OBWidth:OBWidth:RIGHt:FREQuency? (Query Only)

Returns the right (higher) frequency of the occupied bandwidth.

Conditions Measurement views: Occupied Bandwidth

Group Read commands

Syntax `READ:OBWidth:OBWidth:RIGHt:FREQuency?`

Related Commands READ:OBWidth:OBWidth:LEFT:FREQuency?

Arguments None

Returns `<OBW_right_freq>::=<NRf>` is the right frequency in Hz.

Examples `READ:OBWIDTH:OBWIDTH:RIGHT:FREQUENCY?` might return `1.502E+9`, indicating the right frequency is 1.502 GHz.

READ:OBWidth:OBWidth:RIGHt:LEVel? (Query Only)

Returns the level at the right frequency of the occupied bandwidth.

Conditions Measurement views: Occupied Bandwidth

Group Read commands

Syntax `READ:OBWidth:OBWidth:RIGHt:LEVel?`

Related Commands READ:OBWidth:OBWidth:LEFT:LEVel?

Arguments None

Returns `<OBW_right_level>::=<NRf>` is the level at the right frequency in dB.

Examples `READ:OBWIDTH:OBWIDTH:RIGHT:LEVEL?` might return `-23.5`, indicating the level at the right frequency is -23.5 dB.

READ:OBWidth:SPECtrum? (Query Only)

Returns spectrum trace data of the Occupied Bandwidth measurement.

Conditions Measurement views: Occupied Bandwidth

Group Read commands

Syntax `READ:OBWidth:SPECtrum?`

Arguments None

Returns `#<num_digit><num_byte><data(1)><data(2)>...<data(n)>`

Where
`<num_digit>` is the number of digits in <num_byte>.
`<num_byte>` is the number of bytes of data that follow.
`<data(n)>` is the amplitude in dBm for the point #n,
4-byte little endian floating-point format specified in IEEE 488.2.
The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples `READ:OBWIDTH:SPECTRUM?` might return `#43204xxxx...` (3204-byte data) for the spectrum trace data of the Occupied Bandwidth measurement.

READ:OBWidth:XDBBandwidth:BANDwidth? (Query Only)

Returns the x dB bandwidth in the Occupied Bandwidth measurement.

Conditions Measurement views: Occupied Bandwidth

Group Read commands

Syntax `READ:OBWidth:XDBBandwidth:BANDwidth?`

Arguments None

Returns `<xdBBW>::=<NRf>` is the x dB bandwidth in Hz.

Examples `READ:OBWIDTH:XDBBANDWIDTH:BANDWIDTH?` might return `2.0E+6`, indicating the x dB bandwidth is 2 MHz.

READ:OBWidth:XDBBandwidth:LEFT:FREQuency? (Query Only)

Returns the left (lower) frequency of the x dB bandwidth.

Conditions Measurement views: Occupied Bandwidth

Group Read commands

Syntax `READ:OBWidth:XDBBandwidth:LEFT:FREQuency?`

Related Commands READ:OBWidth:XDBBandwidth:RIGHt:FREQuency?

Arguments None

Returns `<xdBBW_left_freq>::=<NRf>` is the left frequency in Hz.

Examples `READ:OBWIDTH:XDBBANDWIDTH:LEFT:FREQUENCY?` might return `1.498E+9`, indicating the left frequency is 1.498 GHz.

READ:OBWidth:XDBBandwidth:LEFT:LEVel? (Query Only)

Returns the level at the left frequency of the x dB bandwidth.

Conditions Measurement views: Occupied Bandwidth

Group Read commands

Syntax `READ:OBWidth:XDBBandwidth:LEFT:LEVel?`

Related Commands READ:OBWidth:XDBBandwidth:RIGHt:LEVel?

Arguments None

Returns `<xdBBW_left_level>::=<NRf>` is the level at the left frequency in dB.

Examples `READ:OBWIDTH:XDBBANDWIDTH:LEFT:LEVEL?` might return `-23.5`, indicating the level at the left frequency is -23.5 dB.

READ:OBWidth:XDBBandwidth:POWer? (Query Only)

Returns the reference power in the x dB bandwidth measurement.

Conditions Measurement views: Occupied Bandwidth

Group Read commands

Syntax `READ:OBWidth:XDBBandwidth:POWer?`

Arguments None

Returns `<xdBBW_ref_power>::=<NRf>` is the reference power in dBm.
The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples `READ:OBWIDTH:XDBBANDWIDTH:POWER?` might return `-10.0`, indicating the reference power is -10 dBm.

READ:OBWidth:XDBBandwidth:RIGHt:FREQuency? (Query Only)

Returns the right (higher) frequency of the x dB bandwidth.

Conditions Measurement views: Occupied Bandwidth

Group Read commands

Syntax `READ:OBWidth:XDBBandwidth:RIGHt:FREQuency?`

Related Commands READ:OBWidth:XDBBandwidth:LEFT:FREQuency?

Arguments None

Returns `<xdBBW_right_freq>::=<NRf>` is the right frequency in Hz.

Examples `READ:OBWIDTH:XDBBANDWIDTH:RIGHT:FREQUENCY?` might return `1.502E+9`, indicating the right frequency is 1.502 GHz.

READ:OBWidth:XDBBandwidth:RIGHt:LEVel? (Query Only)

Returns the level at the right frequency of the x dB bandwidth.

Conditions Measurement views: Occupied Bandwidth

Group Read commands

Syntax `READ:OBWidth:XDBBandwidth:RIGHt:LEVel?`

Related Commands READ:OBWidth:XDBBandwidth:LEFT:LEVel?

Arguments None

Returns `<xdBBW_right_level>::=<NRf>` is the level at the right frequency in dB.

Examples `READ:OBWIDTH:XDBBANDWIDTH:RIGHT:LEVEL?` might return `-23.5`, indicating the level at the right frequency is -23.5 dB.

READ:PERRor:FERRor? (Query Only)

Returns the frequency error in the Phase error versus Time measurement.

Conditions Measurement views: Phase error versus Time

Group Read commands

Syntax `READ:PERRor:FERRor?`

Arguments None

Returns `<freq_error>::=<NRf>` is the frequency error in Hz.

Examples `READ:PERROR:FERROR?` might return `-10.7E+3`, indicating the frequency error is -10.7 kHz.

READ:PERRor:PEAK? (Query Only)

Returns the peak value in the Phase error versus Time measurement.

Conditions Measurement views: Phase error versus Time

Group Read commands

Syntax `READ:PERRor:PEAK?`

Related Commands READ:PERRor:PINDex?

Arguments None

Returns `<peak>::=<NRf>` is the peak phase error in degrees.

Examples READ:PERROR:PEAK? might return 0.683, indicating the peak phase error is 0.683°.

READ:PERRor:PINDex? (Query Only)

Returns the time at the phase error peak.

Conditions Measurement views: Phase error versus Time

Group Read commands

Syntax `READ:PERRor:PINDex?`

Related Commands READ:PERRor:PEAK?

Arguments None

Returns <peak_time>::=<NRf> is the time at the phase error peak in symbol number. The unit can be changed by the [SENSe]:DDEMod:TIMe:UNITs command.

Examples READ:PERROR:PINDEX? might return 68.000, indicating that the phase error peak is at symbol #68.

READ:PERRor:RMS (Query Only)

Returns the RMS (Root-Mean-Square) value in the Phase error versus Time measurement.

Conditions Measurement views: Phase error versus Time

Group Read commands

Syntax `READ:PERRor:RMS`

Arguments None

Returns `<rms>::=<NRf>` is the RMS phase error in degrees.

Examples `READ:PERROR:RMS` might return `0.746`, indicating the phase error is 0.746° RMS.

READ:PERRor:TRACe? (Query Only)

Returns the Phase error versus Time trace data.

Conditions Measurement views: Phase error versus Time

Group Read commands

Syntax `READ:PERRor:TRACe?`

Arguments None

Returns `#<num_digit><num_byte><data(1)><data(2)>...<data(n)>`

Where
`<num_digit>` is the number of digits in <num_byte>.
`<num_byte>` is the number of bytes of data that follow.
`<data(n)>` is the phase error data for the point n in degrees,
4-byte little endian floating-point format specified in IEEE 488.2.

Examples `READ:PERROR:TRACE?` might return `#42036xxxx...` (2036-byte data) for the Phase error versus Time trace.

READ:PHVTime? (Query Only)

Returns the Phase versus Time trace data.

Conditions Measurement views: Phase versus Time

Group Read commands

Syntax `READ:PHVTime?`

Arguments None

Returns `#<num_digit><num_byte><data(1)><data(2)>...<data(n)>`

Where
`<num_digit>` is the number of digits in <num_byte>.
`<num_byte>` is the number of bytes of data that follow.
`<data(n)>` is the phase in degrees for the point #n,
4-byte little endian floating-point format specified in IEEE 488.2.

Examples `READ:PHVTIME?` might return `#3160xxxx...` (160-byte data) for the Phase versus Time trace.

READ:PHVTime:MAXimum? (Query Only)

Returns the maximum value in the Phase versus Time measurement.

Conditions Measurement views: Phase versus Time

Group Read commands

Syntax `READ:PHVTime:MAXimum?`

Related Commands READ:PHVTime:MAXLocation?

Arguments None

Returns `<max>::=<NRf>` is the maximum phase in degrees.

Examples `READ:PHVTIME:MAXIMUM?` might return `153.8`, indicating the maximum phase is 153.8°.

READ:PHVTime:MAXLocation? (Query Only)

Returns the time at which the phase is maximum.

Conditions Measurement views: Phase versus Time

Group Read commands

Syntax `READ:PHVTime:MAXLocation?`

Related Commands READ:PHVTime:MAXimum?

Arguments None

Returns `<max_time>::=<NRf>` is the time in seconds at which the phase is maximum.

Examples `READ:PHVTIME:MAXLOCATION?` might return `175.3E-9`, indicating the I or Q level is maximum at 175.3 ns.

READ:PHVTime:MINimum? (Query Only)

Returns the minimum value in the Phase versus Time measurement.

Conditions Measurement views: Phase versus Time

Group Read commands

Syntax `READ:PHVTime:MINimum?`

Related Commands READ:PHVTime:MINLocation?

Arguments None

Returns `<min>::=<NRf>` is the minimum phase in degrees.

Examples `READ:PHVTIME:MINIMUM?` might return `-176.3`, indicating the minimum phase is -176.3°.

READ:PHVTime:MINLocation? (Query Only)

Returns the time at which the phase is minimum.

Conditions Measurement views: Phase versus Time

Group Read commands

Syntax `READ:PHVTime:MINLocation?`

Related Commands READ:PHVTime:MINimum?

Arguments None

Returns `<min_time>::=<NRf>` is the time in seconds at which the phase is minimum.

Examples `READ:PHVTIME:MINLOCATION?` might return `450.7E-9`, indicating the phase is minimum at 450.7 ns.

READ:PHVTime:RESult? (Query Only)

Returns the Phase versus Time measurement results.

Conditions Measurement views: Phase versus Time

Group Read commands

Syntax `READ:PHVTime:RESult?`

Arguments None

Returns `<max>,<max_time>,<min>,<min_time>`

Where
`<max>::=<NRf>` is the maximum phase in degrees.
`<max_time>::=<NRf>` is the time in seconds at which the phase is maximum.
`<min>::=<NRf>` is the minimum phase in degrees.
`<min_time>::=<NRf>` is the time in seconds at which the phase is minimum.

Examples `READ:PHVTIME:RESULT?` might return `153.8,175.3E-9,-176.3,450.7E-9`, indicating the maximum phase is 153.8° at 175.3 ns and the minimum phase is -176.3° at 450.7 ns.

READ:PNOise:ALL? (Query Only)

Returns all results of the phase noise measurement.

Conditions Measurement views: Phase noise

Group Read commands

Syntax `READ:PNOise:ALL?`

Arguments None

Returns `<Cpower>,<Ferror>,<Pnoise>,<Tjitter>,<Rjitter>,<RFM>`

Where
`<Cpower>::=<NRf>` is the carrier power in dBm.
`<Ferror>::=<NRf>` is the frequency error in Hz.
`<Pnoise>::=<NRf>` is the RMS phase noise in degrees.
`<Tjitter>::=<NRf>` is the total jitter in seconds.
`<Rjitter>::=<NRf>` is the random jitter in seconds.
`<RFM>::=<NRf>` is the residual FM in Hz.

Examples `READ:PNOISE:ALL?` might return
`-9.455,1.235E+6,51.43,2.312E-9,4.178E-9,14.58`,
indicating
Carrier power: -9.455 dBm,
Frequency error: 1.235 MHz,
RMS phase noise: 51.43°,
Total jitter: 2.312 ns,
Random jitter: 4.178 ns, and
Residual FM: 14.58 Hz.

READ:PNOise:CARRier:FERRor? (Query Only)

Returns the carrier frequency error in the phase noise measurement.

Conditions Measurement views: Phase noise

Group Read commands

Syntax `READ:PNOise:CARRier:FERRor?`

Arguments None

Returns `<NRf>` Carrier frequency error in Hz.

Examples `READ:PNOISE:CARRIER:FERROR?` might return `1.235E+6`, indicating that the carrier frequency error is 1.235 MHz.

READ:PNOise:CARRier:POWer? (Query Only)

Returns the carrier power in the phase noise measurement.

Conditions Measurement views: Phase noise

Group Read commands

Syntax `READ:PNOise:CARRier:POWer?`

Arguments None

Returns `<NRf>` Carrier power in dBm.
The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples `READ:PNOISE:CARRIER:POWER?` might return `-9.455`, indicating that the carrier power is -9.455 dBm.

READ:PNOise:JITTer? (Query Only)

Returns the jitter in the phase noise measurement.

Conditions Measurement views: Phase noise

Group Read commands

Syntax `READ:PNOise:JITTer?`

Arguments None

Returns `<NRf>` Jitter in seconds.

Examples `READ:PNOISE:JITTER?` might return `2.312E-9`, indicating that the jitter is 2.312 ns.

READ:PNOise:RESidual:FM? (Query Only)

Returns the residual FM in the phase noise measurement.

Conditions Measurement views: Phase noise

Group Read commands

Syntax `READ:PNOise:RESidual:FM?`

Arguments None

Returns `<NRf>` Residual FM in Hz.

Examples `READ:PNOISE:RESIDUAL:FM?` might return `14.58`, indicating that the residual FM is 14.58 Hz.

READ:PNOise:RMS:PNOise? (Query Only)

Returns the RMS phase noise in the phase noise measurement.

Conditions Measurement views: Phase noise

Group Read commands

Syntax `READ:PNOise:RMS:PNOise?`

Arguments None

Returns `<NRf>` RMS phase noise in degrees.

Examples `READ:PNOISE:RMS:PNOISE?` might return `51.43`, indicating that the RMS phase noise is 51.43°.

READ:PNOise:SPECtrum<x>:X? (Query Only)

Returns the frequencies of the specified trace.

The parameter <x> = 1 and 2, representing Trace 1 and Trace 2, respectively.

Conditions Measurement views: Phase noise

Group Read commands

Syntax `READ:PNOise:SPECtrum<x>:X?`

Arguments None

Returns `#<num_digit><num_byte><x(1)><x(2)>...<x(n)>`

Where
`<num_digit>` is the number of digits in <num_byte>.
`<num_byte>` is the number of bytes of data that follow.
`<x(n)>` is the frequency (Hz) at the point #n,
4-byte little endian floating-point format specified in IEEE 488.2.

Examples `READ:PNOISE:SPECTRUM1:X?` might return `#516020xxxx...` (16020-byte data) for the frequencies of Trace 1.

READ:PNOise:SPECtrum<x>:XY? (Query Only)

Returns the frequency and phase noise pairs of the specified trace.

The parameter <x> = 1 and 2, representing Trace 1 and Trace 2, respectively.

Conditions Measurement views: Phase noise

Group Read commands

Syntax READ:PNOise:SPECtrum<x>:XY?

Arguments None

Returns #<num_digit><num_byte><x(1)><y(1)><x(2)><y(2)>...<x(n)><y(n)>

Where
<num_digit> is the number of digits in <num_byte>.
<num_byte> is the number of bytes of data that follow.
<x(n)><y(n)> is the frequency (Hz) and phase noise (dBc/Hz) pair for the point #n, 4-byte little endian floating-point format specified in IEEE 488.2.

Examples READ:PNOISE:SPECTRUM1:XY? might return #516020xxxx... (16020-byte data) for the frequency and phase noise pairs of the Trace 1.

READ:PNOise:SPECtrum<x>[:Y]? (Query Only)

Returns the phase noise values of the specified trace.

The parameter <x> = 1 and 2, representing Trace 1 and Trace 2, respectively.

Conditions Measurement views: Phase noise

Group Read commands

Syntax READ:PNOise:SPECtrum<x>[:Y]?

Arguments None

Returns #<num_digit><num_byte><y(1)><y(2)>...<y(n)>

Where <num_digit> is the number of digits in <num_byte>.
<num_byte> is the number of bytes of data that follow.
<y(n)> is the phase noise (dBc/Hz) at the point #n,
4-byte little endian floating-point format specified in IEEE 488.2.

Examples `READ:PNOISE:SPECTRUM1:Y` might return `#516020xxxx...` (16020-byte data) for the phase noise values of Trace 1.

READ:PULSe[:RESult]:ATX? (Query Only)

Returns the average transmitted power in the pulse table.

Conditions Measurement views: Pulse table

Group Read commands

Syntax `READ:PULSe[:RESult]:ATX?`

Arguments None

Returns `<first_pulse_num>,<ATX(1)>,< ATX(2)>,...<ATX(n)>`

Where
`<first_pulse_num>::=<NR1>` is the first pulse number.
`<ATX(n)>::=<NRf>` is the average transmitted power for the pulse with the number of [first_pulse_num + n - 1] in dBm.
The unit can be changed to watts by the [SENSe]:POWer:UNITs command.
Volt is invalid in the average transmitted power measurement.

Examples `READ:PULSE:RESULT:ATX?` might return `0,-18.57,-18.73,-18.20, -18.53` for Pulse 0 to 3.

READ:PULSe[:RESult]:AVERage? (Query Only)

Returns the average on power in the pulse table.

Conditions Measurement views: Pulse table

Group Read commands

Syntax `READ:PULSe[:RESult]:AVERage?`

Arguments None

Returns `<first_pulse_num>,<avg(1)>,< avg(2)>,...<avg(n)>`

Where
`<first_pulse_num>::=<NR1>` is the first pulse number.
`<avg(n)>::=<NRf>` is the average on power for the pulse with the number of [first_pulse_num + n - 1] in dBm.
The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples `READ:PULSE:RESULT:AVERAGE?` might return `0,-2.354,-2.368,-2.343,-2.358` for Pulse 0 to 3.

READ:PULSe[:RESult]:DROop? (Query Only)

Returns the average on power in the pulse table.

Conditions Measurement views: Pulse table

Group Read commands

Syntax `READ:PULSe[:RESult]:DROop?`

Arguments None

Returns `<first_pulse_num>,<droop(1)>,<droop(2)>,...<droop(n)>`

Where
`<first_pulse_num>::=<NR1>` is the first pulse number.
`<droop(n)>::=<NRf>` is the wattage droop for the pulse with the number of [first_pulse_num + n - 1] in percent (%).

Examples `READ:PULSE:RESULT:DROOP?` might return `0,-270.9E-3,-193.0E-3,-242.7E-3,-177.5E-3` for Pulse 0 to 3.

READ:PULSe[:RESult]:DUTPct? (Query Only)

Returns the duty factor (%) in the pulse table.

Conditions Measurement views: Pulse table

Group Read commands

Syntax `READ:PULSe[:RESult]:DUTPct?`

Arguments None

Returns `<first_pulse_num>,<duty_pct(1)>,<duty_pct(2)>,...`
`<duty_pct(n)>`

Where
`<first_pulse_num>::=<NR1>` is the first pulse number.
`<duty_pct(n)>::=<NRf>` is the duty factor for the pulse with the number of [first_pulse_num + n - 1] in percent (%).

Examples `READ:PULSE:RESULT:DUTPCT?` might return `0,28.94,28.96,29.00,29.01` for Pulse 0 to 3.

READ:PULSe[:RESult]:DUTRatio? (Query Only)

Returns the duty factor (ratio) in the pulse table.

Conditions Measurement views: Pulse table

Group Read commands

Syntax `READ:PULSe[:RESult]:DUTRatio?`

Arguments None

Returns `<first_pulse_num>,<duty_ratio(1)>,<duty_ratio(2)>,...`
`<duty_ratio(n)>`

Where
`<first_pulse_num>::=<NR1>` is the first pulse number.
`<duty_ratio(n)>::=<NRf>` is the duty factor for the pulse with the number of [first_pulse_num + n - 1] (no unit).

Examples `READ:PULSE:RESULT:DUTRATIO?` might return `0,289.4E-3,289.6E-3,290.0E-3,290.1E-3` for Pulse 0 to 3.

READ:PULSe[:RESult]:FALL? (Query Only)

Returns the fall time in the pulse table.

Conditions Measurement views: Pulse table

Group Read commands

Syntax `READ:PULSe[:RESult]:FALL?`

Arguments None

Returns `<first_pulse_num>,<fall(1)>,<fall(2)>,...<fall(n)>`

Where
`<first_pulse_num>::=<NR1>` is the first pulse number.
`<fall(n)>::=<NRf>` is the fall time for the pulse with the number of [first_pulse_num + n - 1] in seconds.

Examples `READ:PULSE:RESULT:FALL?` might return `0,110.3E-9,90.45E-9, 95.03E-9,111.9E-9` for Pulse 0 to 3.

READ:PULSe[:RESult]:FRDeviation? (Query Only)

Returns the frequency deviation in the pulse table.

Conditions Measurement views: Pulse table

Group Read commands

Syntax `READ:PULSe[:RESult]:FRDeviation?`

Arguments None

Returns `<first_pulse_num>,<freq_dev(1)>,<freq_dev(2)>,... <freq_dev(n)>`

Where
`<first_pulse_num>::=<NR1>` is the first pulse number.

<freq_dev(n)>::=<NRf> is the frequency deviation for the pulse with the number of [first_pulse_num + n - 1] in Hz.

Examples READ:PULSE:RESULT:FRDEVIATION? might return 1,740.6E+3,736.5E+3,718.3E+3,672.2E+3 for Pulse 1 to 4.

READ:PULSe[:RESult]:MFReqerror? (Query Only)

Returns the maximum frequency error in the pulse table.

Conditions Measurement views: Pulse table

Group Read commands

Syntax READ:PULSe[:RESult]:MFReqerror?

Arguments None

Returns <first_pulse_num>,<max_freq_err(1)>,<max_freq_err(2)>,... <max_freq_err(n)>

Where
<first_pulse_num>::=<NR1> is the first pulse number.
<max_freq_err(n)>::=<NRf> is the maximum frequency error for the pulse with the number of [first_pulse_num + n - 1] in Hz.

Examples READ:PULSE:RESULT:MFREQERROR? might return 1,597.5E+3,675.8E+3,642.8E+3,598.2E+3 for Pulse 1 to 4.

READ:PULSe[:RESult]:MPHerror? (Query Only)

Returns the maximum phase error in the pulse table.

Conditions Measurement views: Pulse table

Group Read commands

Syntax READ:PULSe[:RESult]:MPHerror?

Arguments None

Returns <first_pulse_num>,<max_phase_err(1)>,<max_phase_err(2)>,...
<max_phase_err(n)>

Where
<first_pulse_num>::=<NR1> is the first pulse number.
<max_phase_err(n)>::=<NRf> is the maximum phase error for the pulse with the number of [first_pulse_num + n - 1] in degrees.

Examples READ:PULSE:RESULT:MPHERROR? might return 1,-9.221,-8.413,
-11.853,-10.258 for Pulse 1 to 4.

READ:PULSe[:RESult]:PHDeviation? (Query Only)

Returns the phase deviation in the pulse table.

Conditions Measurement views: Pulse table

Group Read commands

Syntax READ:PULSe[:RESult]:PHDeviation?

Arguments None

Returns <first_pulse_num>,<phase_dev(1)>,<phase_dev(2)>,...
<phase_dev(n)>

Where
<first_pulse_num>::=<NR1> is the first pulse number.
<phase_dev(n)>::=<NRf> is the phase deviation for the pulse with the number of [first_pulse_num + n - 1] in degrees.

Examples READ:PULSE:RESULT:PHDEVIATION? might return
1,11.658,9.640,10.509,8.272 for Pulse 1 to 4.

READ:PULSe[:RESult]:PPFRequency? (Query Only)

Returns the pulse-pulse carrier frequency in the pulse table.

Conditions Measurement views: Pulse table

Group Read commands

Syntax `READ:PULSe[:RESult]:PPFRequency?`

Arguments None

Returns `<first_pulse_num>,<pp_freq(1)>,<pp_freq(2)>,... <pp_freq(n)>`

Where
`<first_pulse_num>::=<NR1>` is the first pulse number.
`<pp_freq(n)>::=<NRf>` is the pulse-pulse carrier frequency for the pulse with the number of [first_pulse_num + n - 1] in Hz.

Examples `READ:PULSE:RESULT:PPFREQUENCY?` might return `0,0.000,1.258E+3,-3.121E+3,1.862E+3` for Pulse 0 to 3.

READ:PULSe[:RESult]:PPOWer? (Query Only)

Returns the peak power in the pulse table.

Conditions Measurement views: Pulse table

Group Read commands

Syntax `READ:PULSe[:RESult]:PPOWer?`

Arguments None

Returns `<first_pulse_num>,<pk_power(1)>,<pk_power(2)>,... <pk_power(n)>`

Where

`<first_pulse_num>::=<NR1>` is the first pulse number.
`<pk_power(n)>::=<NRf>` is the peak power for the pulse with the number of [first_pulse_num + n - 1] in dBm.
The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples `READ:PULSE:RESULT:PPOWER?` might return `0,-2.26,-2.27,-2.23,-2.25` for Pulse 0 to 3.

READ:PULSe[:RESult]:PPPHase? (Query Only)

Returns the pulse-pulse carrier phase in the pulse table.

Conditions Measurement views: Pulse table

Group Read commands

Syntax `READ:PULSe[:RESult]:PPPHase?`

Arguments None

Returns `<first_pulse_num>,<pp_phase(1)>,<pp_phase(2)>,...<pp_phase(n)>`

Where
`<first_pulse_num>::=<NR1>` is the first pulse number.
`<pp_phase(n)>::=<NRf>` is the pulse-pulse carrier phase for the pulse with the number of [first_pulse_num + n - 1] in degrees.

Examples `READ:PULSE:RESULT:PPPHASE?` might return `0,0.000,21.66,46.76,57.56` for Pulse 0 to 3.

READ:PULSe[:RESult]:RINTerval? (Query Only)

Returns the repetition interval in the pulse table.

Conditions Measurement views: Pulse table

Group Read commands

Syntax READ:PULSe[:RESult]:RINTerval?

Arguments None

Returns <first_pulse_num>,<rep_int(1)>,<rep_int(2)>,...<rep_int(n)>

Where
<first_pulse_num>::=<NR1> is the first pulse number.
<rep_int(n)>::=<NRf> is the repetition interval for the pulse with the number of [first_pulse_num + n - 1] in seconds.

Examples READ:PULSE:RESULT:RINTERVAL? might return 0,16.03E-6,16.08E-6, 16.07E-6,16.02E-6 for Pulse 0 to 3.

READ:PULSe[:RESult]:RIPPle? (Query Only)

Returns the ripple in the pulse table.

Conditions Measurement views: Pulse table

Group Read commands

Syntax READ:PULSe[:RESult]:RIPPle?

Arguments None

Returns <first_pulse_num>,<ripple(1)>,<ripple(2)>,...<ripple(n)>

Where
<first_pulse_num>::=<NR1> is the first pulse number.
<ripple(n)>::=<NRf> is the voltage ripple for the pulse with the number of [first_pulse_num + n - 1] in percent (%).

Examples READ:PULSE:RESULT:RIPPLE? might return 0,106.5E-3,177.6E-3, 148.3E-3,148.5E-3 for Pulse 0 to 3.

READ:PULSe[:RESult]:RISE? (Query Only)

Returns the rise time in the pulse table.

Conditions Measurement views: Pulse table

Group Read commands

Syntax `READ:PULSe[:RESult]:RISE?`

Arguments None

Returns `<first_pulse_num>,<rise(1)>,<rise(2)>,...<rise(n)>`

Where
`<first_pulse_num>::=<NR1>` is the first pulse number.
`<rise(n)>::=<NRf>` is the rise time for the pulse with the number of [first_pulse_num + n - 1] in seconds.

Examples `READ:PULSE:RESULT:RISE?` might return `0,92.94E-9,115.9E-9,115.1E-9,97.45E-9` for Pulse 0 to 3.

READ:PULSe[:RESult]:RMSFreqerror? (Query Only)

Returns the RMS frequency error in the pulse table.

Conditions Measurement views: Pulse table

Group Read commands

Syntax `READ:PULSe[:RESult]:RMSFreqerror?`

Arguments None

Returns `<first_pulse_num>,<RMS_freq_err(1)>,<RMS_freq_err(2)>,...<RMS_freq_err(n)>`

Where
`<first_pulse_num>::=<NR1>` is the first pulse number.
`<RMS_freq_err(n)>::=<NRf>` is the RMS frequency error for the pulse with the number of [first_pulse_num + n - 1] in Hz.

Examples `READ:PULSE:RESULT:RMSFREQERROR?` might return `1,51.54E+3, 69.20E+3,64.21E+3,51.02E+3` for Pulse 1 to 4.

READ:PULSe[:RESult]:RMSPherror? (Query Only)

Returns the RMS phase error in the pulse table.

Conditions Measurement views: Pulse table

Group Read commands

Syntax `READ:PULSe[:RESult]:RMSPherror?`

Arguments None

Returns `<first_pulse_num>,<RMS_phase_err(1)>,<RMS_phase_err(2)>,... <RMS_phase_err(n)>`

Where
`<first_pulse_num>::=<NR1>` is the first pulse number.
`<RMS_phase_err(n)>::=<NRf>` is the RMS phase error for the pulse with the number of [first_pulse_num + n - 1] in degrees.

Examples `READ:PULSE:RESULT:RMSPHERROR?` might return `1,908.4E-3,752.8E-3,981.7E-3,886.4E-3` for Pulse 1 to 4.

READ:PULSe[:RESult]:RRATe? (Query Only)

Returns the repetition rate in the pulse table.

Conditions Measurement views: Pulse table

Group Read commands

Syntax `READ:PULSe[:RESult]:RRATe?`

Arguments None

Returns `<first_pulse_num>,<rep_rate(1)>,<rep_rate(2)>,...<rep_rate(n)>`

Where
`<first_pulse_num>::=<NR1>` is the first pulse number.
`<rep_rate(n)>::=<NRf>` is the repetition rate for the pulse with the number of [first_pulse_num + n - 1] in Hz.

Examples `READ:PULSE:RESULT:RRATE?` might return `0,62.50E+3,62.52E+3,62.51E+3,62.49E+3` for Pulse 0 to 3.

READ:PULSe[:RESult]:TIME? (Query Only)

Returns the time in the pulse table.

Conditions Measurement views: Pulse table

Group Read commands

Syntax `READ:PULSe[:RESult]:TIME?`

Arguments None

Returns `<first_pulse_num>,<time(1)>,<time(2)>,...<time(n)>`

Where
`<first_pulse_num>::=<NR1>` is the first pulse number.
`<time(n)>::=<NRf>` is the time for the pulse with the number of [first_pulse_num + n - 1] in seconds.

Examples `READ:PULSE:RESULT:TIME?` might return `1,7.937E-3,8.436E-3,6.504E-3,9.876E-3` for Pulse 1 to 4.

READ:PULSe[:RESult]:WIDTh? (Query Only)

Returns the pulse width in the pulse table.

Conditions Measurement views: Pulse table

Group Read commands

Syntax `READ:PULSe[:RESult]:WIDTh?`

Arguments None

Returns `<first_pulse_num>,<width(1)>,<width(2)>,...<width(n)>`

Where `<first_pulse_num>::=<NR1>` is the first pulse number.
`<width(n)>::=<NRf>` is the pulse width for the pulse with the number of [first_pulse_num + n - 1] in seconds.

Examples `READ:PULSE:RESULT:WIDTH?` might return `0,4.630E-6,4.632E-6, 4.639E-6,4.642E-6` for Pulse 0 to 3.

READ:PULSe:STATistics? (Query Only)

Returns the trace data of the pulse statistics measurement selected by the DISPlay:PULSe:SELect:RESult command.

***NOTE.** Select the plot type (Trend or FFT) using the DISPlay:PULSe:STATistics:PLOT command before executing this query.*

Conditions Measurement views: Pulse statistics

Group Read commands

Syntax `READ:PULSe:STATistics?`

Related Commands DISPlay:PULSe:SELect:RESult

Arguments None

Returns `#<num_digit><num_byte><data(1)><data(2)>...<data(n)>`

Where
`<num_digit>` is the number of digits in <num_byte>.
`<num_byte>` is the number of bytes of data that follow.
`<data(n)>` is the amplitude for the point #n.

The unit is dBm (Plot = Trend) or dB (Plot = FFT).
4-byte little endian floating-point format specified in IEEE 488.2.

The unit of power is selected by the [SENSe]:POWer:UNITs command.

Examples

`READ:PULSE:STATISTICS?` might return `#264xxxx...` (64-byte data) for the statistics trace of the pulse width measurement when DISPlay:PULSe:SELect:RESult is set to WIDTh.

READ:PULSe:STATistics:ATX? (Query Only)

Returns the average transmitted power in the pulse statistics. This command is valid when DISPlay:PULSe:STATistics:PLOT is set to TRENd.

Conditions

Measurement views: Pulse statistics

Group

Read commands

Syntax

`READ:PULSe:STATistics:ATX?`

Arguments

None

Returns

`<ATX_avg>,<ATX_min>,<ATX_max>`

Where
`<ATX_avg>::=<NRf>` is the average of the average transmitted power.
`<ATX_min>::=<NRf>` is the minimum of the average transmitted power.
`<ATX_max>::=<NRf>` is the maximum of the average transmitted power.
Unit: dBm.

The unit can be changed to watts by the [SENSe]:POWer:UNITs command.
Volt is invalid in the average transmitted power measurement.

Examples

`READ:PULSE:STATISTICS:ATX?` might return `-18.51,-18.74,-18.12` for the average transmitted power in the pulse statistics.

READ:PULSe:STATistics:AVERage? (Query Only)

Returns the average on power in the pulse statistics. This command is valid when DISPlay:PULSe:STATistics:PLOT is set to TRENd.

Conditions Measurement views: Pulse statistics

Group Read commands

Syntax `READ:PULSe:STATistics:AVERage?`

Arguments None

Returns `<avg_avg>,<avg_min>,< avg_max>`

Where
`<avg_avg>::=<NRf>` is the average of the average on power.
`<avg_min>::=<NRf>` is the minimum of the average on power.
`<avg_max>::=<NRf>` is the maximum of the average on power.
Unit: dBm.
The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples `READ:PULSE:STATISTICS:AVERAGE?` might return `-2.35,-2.36,-2.34` for the average on power in the pulse statistics.

READ:PULSe:STATistics:DROop? (Query Only)

Returns the droop in the pulse statistics. This command is valid when DISPlay:PULSe:STATistics:PLOT is set to TRENd.

Conditions Measurement views: Pulse statistics

Group Read commands

Syntax `READ:PULSe:STATistics:DROop?`

Arguments None

Returns `<droop_avg>,<droop_min>,<droop_max>`

Where
`<droop_avg>::=<NRf>` is the average droop.
`<droop_min>::=<NRf>` is the minimum droop.
`<droop_max>::=<NRf>` is the maximum droop.

Unit: Percent (%) by watts.

Examples `READ:PULSE:STATISTICS:DROOP?` might return `22.67E-3,-613.5E-3,633.8E-3` for the droop in the pulse statistics.

READ:PULSe:STATistics:DUTPct? (Query Only)

Returns the duty factor (%) in the pulse statistics. This command is valid when DISPlay:PULSe:STATistics:PLOT is set to TRENd.

Conditions Measurement views: Pulse statistics

Group Read commands

Syntax `READ:PULSe:STATistics:DUTPct?`

Related Commands

Arguments None

Returns `<duty_pct_avg>,<duty_pct_min)>,<duty_pct_max>`

Where
`<duty_pct_avg>::=<NRf>` is the average duty factor.
`<duty_pct_min>::=<NRf>` is the minimum duty factor.
`<duty_pct_max>::=<NRf>` is the maximum duty factor.
Unit: Percent (%).

Examples `READ:PULSE:STATISTICS:DUTPCT?` might return `2.437,2.310,2.657` for the duty factor (%) in the pulse statistics.

READ:PULSe:STATistics:DUTRatio? (Query Only)

Returns the duty factor (ratio) in the pulse statistics. This command is valid when DISPlay:PULSe:STATistics:PLOT is set to TRENd.

Conditions Measurement views: Pulse statistics

Group Read commands

Syntax READ:PULSe:STATistics:DUTRatio?

Arguments None

Returns <duty_ratio_avg>,<duty_ratio_min>,<duty_ratio_max)>

Where
<duty_ratio_avg>::=<NRf> is the average duty factor.
<duty_ratio_min>::=<NRf> is the minimum duty factor.
<duty_ratio_max>::=<NRf> is the maximum duty factor.
Unit: None.

Examples READ:PULSE:STATISTICS:DUTRATIO? might return 24.37E-3,23.11E-3, 26.57E-3 for the duty factor (ratio) in the pulse statistics.

READ:PULSe:STATistics:FALL? (Query Only)

Returns the fall time in the pulse statistics. This command is valid when DISPlay:PULSe:STATistics:PLOT is set to TRENd.

Conditions Measurement views: Pulse statistics

Group Read commands

Syntax READ:PULSe:STATistics:FALL?

Arguments None

Returns <fall_avg>,<fall_min>,<fall_max>

Where
<fall_avg>::=<NRf> is the average fall time.
<fall_min>::=<NRf> is the minimum fall time.
<fall_max>::=<NRf> is the maximum fall time.
Unit: Seconds.

Examples READ:PULSE:STATISTICS:FALL? might return 70.27E-9,69.62E-9, 71.27E-9 for the fall time in the pulse statistics.

READ:PULSe:STATistics:FRDeviation? (Query Only)

Returns the frequency deviation in the pulse statistics. This command is valid when DISPlay:PULSe:STATistics:PLOT is set to TRENd.

Conditions Measurement views: Pulse statistics

Group Read commands

Syntax `READ:PULSe:STATistics:FRDeviation?`

Arguments None

Returns `<freq_dev_avg>,<freq_dev_min>,<freq_dev_max>`

Where
`<freq_dev_avg>::=<NRf>` is the average frequency deviation.
`<freq_dev_min>::=<NRf>` is the minimum frequency deviation.
`<freq_dev_max>::=<NRf>` is the maximum frequency deviation.
Unit: Hz.

Examples `READ:PULSE:STATISTICS:FRDEVIATION?` might return `754.1E+3,660.5E+3,835.2E+3` for the frequency deviation in the pulse statistics.

READ:PULSe:STATistics:MFReqerror? (Query Only)

Returns the maximum frequency error in the pulse statistics. This command is valid when DISPlay:PULSe:STATistics:PLOT is set to TRENd.

Conditions Measurement views: Pulse statistics

Group Read commands

Syntax `READ:PULSe:STATistics:MFReqerror?`

Arguments None

Returns `<max_freq_err_avg>,<max_freq_err_min>,<max_freq_err_max>`

Where
<max_freq_err_avg>::=<NRf> is the average of maximum frequency error.
<max_freq_err_min>::=<NRf> is the minimum of maximum frequency error.
<max_freq_err_max>::=<NRf> is the maximum of maximum frequency error.
Unit: Hz.

Examples

READ:PULSE:STATISTICS:MFREQERROR? might return 645.0E+3,555.6E+3,738.8E+3 for the maximum frequency error in the pulse statistics.

READ:PULSe:STATistics:MPHerror? (Query Only)

Returns the maximum phase error in the pulse statistics. This command is valid when DISPlay:PULSe:STATistics:PLOT is set to TRENd.

Conditions

Measurement views: Pulse statistics

Group

Read commands

Syntax

```
READ:PULSe:STATistics:MPHerror?
```

Arguments

None

Returns

<max_phase_err_avg>,<max_phase_err_min>,<max_phase_err_max>

Where
<max_phase_err_avg>::=<NRf> is the average of maximum phase error.
<max_phase_err_min>::=<NRf> is the minimum of maximum phase error.
<max_phase_err_max>::=<NRf> is the maximum of maximum phase error.
Unit: Degrees.

Examples

READ:PULSE:STATISTICS:MPHERROR? might return -11.47,-17.18,-7.61 for the maximum phase error in the pulse statistics.

READ:PULSe:STATistics:PHDeviation? (Query Only)

Returns the phase deviation in the pulse statistics. This command is valid when DISPlay:PULSe:STATistics:PLOT is set to TRENd.

Conditions

Measurement views: Pulse statistics

Group Read commands

Syntax `READ:PULSe:STATistics:PHDeviation?`

Arguments None

Returns `<phase_dev_avg>,<phase_dev_min>,<phase_dev_max>`

Where
`<phase_dev_avg>::=<NRf>` is the average phase deviation.
`<phase_dev_min>::=<NRf>` is the minimum phase deviation.
`<phase_dev_max>::=<NRf>` is the maximum phase deviation.
Unit: Degrees.

Examples `READ:PULSE:STATISTICS:PHDEVIATION?` might return `11.678,7.694,17.374` for the phase deviation in the pulse statistics.

READ:PULSe:STATistics:PPFRequency? (Query Only)

Returns the pulse-pulse carrier frequency in the pulse statistics. This command is valid when DISPlay:PULSe:STATistics:PLOT is set to TRENd.

Conditions Measurement views: Pulse statistics

Group Read commands

Syntax `READ:PULSe:STATistics:PPFRequency?`

Arguments None

Returns `<pp_freq_avg>,<pp_freq_min)>,<pp_freq_max>`

Where
`<pp_freq_avg>::=<NRf>` is the average pulse-pulse carrier frequency.
`<pp_freq_min>::=<NRf>` is the minimum pulse-pulse carrier frequency.
`<pp_freq_max>::=<NRf>` is the maximum pulse-pulse carrier frequency.
Unit: Hz.

Examples `READ:PULSE:STATISTICS:PPFREQUENCY?` might return `1.135E+3,311.3E+3,-262.8E+3` for the pulse-pulse carrier frequency in the pulse statistics.

READ:PULSe:STATistics:PPOWer? (Query Only)

Returns the peak power in the pulse statistics. This command is valid when DISPlay:PULSe:STATistics:PLOT is set to TRENd.

Conditions Measurement views: Pulse statistics

Group Read commands

Syntax `READ:PULSe:STATistics:PPOWer?`

Arguments None

Returns <pk_power_avg>,<pk_power_min>,<pk_power_max>

Where
<pk_power_avg>::=<NRf> is the average peak power.
<pk_power_min>::=<NRf> is the minimum peak power.
<pk_power_max>::=<NRf> is the maximum peak power.
Unit: dBm.
The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples `READ:PULSe:STATistics:PPOWer?` might return `-2.273,-2.313,-2.235` for the peak power in the pulse statistics.

READ:PULSe:STATistics:PPPHase? (Query Only)

Returns the pulse-pulse carrier phase in the pulse statistics. This command is valid when DISPlay:PULSe:STATistics:PLOT is set to TRENd.

Conditions Measurement views: Pulse statistics

Group Read commands

Syntax `READ:PULSe:STATistics:PPPHase?`

Arguments None

Returns `<pp_phase_avg>,<pp_phase_min>,<pp_phase_max>`

Where
`<pp_phase_avg>::=<NRf>` is the average pulse-pulse carrier phase.
`<pp_phase_min>::=<NRf>` is the minimum pulse-pulse carrier phase.
`<pp_phase_max>::=<NRf>` is the maximum pulse-pulse carrier phase.
Unit: Degrees.

Examples `READ:PULSE:STATISTICS:PPPHASE?` might return `-9.298E-3,-254.3E-3,311.7E-3` for the pulse-pulse carrier phase in the pulse statistics.

READ:PULSe:STATistics:RINTerval? (Query Only)

Returns the repetition interval in the pulse statistics. This command is valid when DISPlay:PULSe:STATistics:PLOT is set to TRENd.

Conditions Measurement views: Pulse statistics

Group Read commands

Syntax `READ:PULSe:STATistics:RINTerval?`

Arguments None

Returns `<rep_int_avg>,<rep_int_min>,<rep_int_max>`

Where
`<rep_int_avg>::=<NRf>` is the average repetition interval.
`<rep_int_min>::=<NRf>` is the minimum repetition interval.
`<rep_int_max>::=<NRf>` is the maximum repetition interval.
Unit: Seconds.

Examples `READ:PULSE:STATISTICS:RINTERVAL?` might return `240.5E-6,217.9E-6,281.2E-6` for the repetition interval in the pulse statistics.

READ:PULSe:STATistics:RIPPle? (Query Only)

Returns the ripple in the pulse statistics. This command is valid when DISPlay:PULSe:STATistics:PLOT is set to TRENd.

Conditions Measurement views: Pulse statistics

Group Read commands

Syntax `READ:PULSe:STATistics:RIPPle?`

Arguments None

Returns <ripple_avg>,<ripple_min>,<ripple_max>

Where
<ripple_avg>::=<NRf> is the average ripple.
<ripple_min>::=<NRf> is the minimum ripple.
<ripple_max>::=<NRf> is the maximum ripple.
Unit: Percent (%) by volts.

Examples `READ:PULSE:STATISTICS:RIPPLE?` might return `160.4E-3,83.78E-3,287.7E-3` for the ripple in the pulse statistics.

READ:PULSe:STATistics:RISE? (Query Only)

Returns the rise time in the pulse statistics. This command is valid when DISPlay:PULSe:STATistics:PLOT is set to TRENd.

Conditions Measurement views: Pulse statistics

Group Read commands

Syntax `READ:PULSe:STATistics:RISE?`

Arguments None

Returns `<rise_avg>,<rise_min>,<rise_max>`

Where
`<rise_avg>::=<NRf>` is the average rise time.
`<rise_min>::=<NRf>` is the minimum rise time.
`<rise_max>::=<NRf>` is the maximum rise time.
Unit: Seconds.

Examples `READ:PULSE:STATISTICS:RISE?` might return `105.4E-9,91.65E-9,116.2E-9` for the rise time in the pulse statistics.

READ:PULSe:STATistics:RMSFreqerror? (Query Only)

Returns the RMS frequency error in the pulse statistics. This command is valid when DISPlay:PULSe:STATistics:PLOT is set to TRENd.

Conditions Measurement views: Pulse statistics

Group Read commands

Syntax `READ:PULSe:STATistics:RMSFreqerror?`

Arguments None

Returns `<RMS_freq_err_avg>,<RMS_freq_err_min>,<RMS_freq_err_max>`

Where
`<RMS_freq_err_avg>::=<NRf>` is the average of RMS frequency error.
`<RMS_freq_err_min>::=<NRf>` is the minimum of RMS frequency error.
`<RMS_freq_err_max>::=<NRf>` is the maximum of RMS frequency error.
Unit: Hz.

Examples `READ:PULSE:STATISTICS:RMSFREQERROR?` might return `63.67E+3,45.49E+3,81.28E+3` for the RMS frequency error in the pulse statistics.

READ:PULSe:STATistics:RMSPherror? (Query Only)

Returns the RMS phase error in the pulse statistics. This command is valid when DISPlay:PULSe:STATistics:PLOT is set to TRENd.

Conditions Measurement views: Pulse statistics

Group Read commands

Syntax `READ:PULSe:STATistics:RMSPherror?`

Arguments None

Returns `<RMS_phase_err_avg>,<RMS_phase_err_min>,<RMS_phase_err_max>`

Where
`<RMS_phase_err_avg>::=<NRf>` is the average of RMS phase error.
`<RMS_phase_err_min>::=<NRf>` is the minimum of RMS phase error.
`<RMS_phase_err_max>::=<NRf>` is the maximum of RMS phase error.
Unit: Degrees.

Examples `READ:PULSE:STATISTICS:RMSPHERROR?` might return `1.032,604.5E-3,1.606` for the RMS phase error in the pulse statistics.

READ:PULSe:STATistics:RRATe? (Query Only)

Returns the repetition rate in the pulse statistics. This command is valid when DISPlay:PULSe:STATistics:PLOT is set to TRENd.

Conditions Measurement views: Pulse statistics

Group Read commands

Syntax `READ:PULSe:STATistics:RRATe?`

Arguments None

Returns `<rep_rate_avg>,<rep_rate_min>,<rep_rate_max>`

Where
`<rep_rate_avg>::=<NRf>` is the average repetition rate.
`<rep_rate_min>::=<NRf>` is the minimum repetition rate.
`<rep_rate_max>::=<NRf>` is the maximum repetition rate.
Unit: Hz.

Examples

`READ:PULSE:STATISTICS:RRATE?` might return `62.50E+3,62.49E+3,62.52E+3` for the repetition rate in the pulse statistics.

READ:PULSe:STATistics:WIDTh? (Query Only)

Returns the pulse width in the pulse statistics. This command is valid when DISPlay:PULSe:STATistics:PLOT is set to TRENd.

Conditions

Measurement views: Pulse statistics

Group

Read commands

Syntax

`READ:PULSe:STATistics:WIDTh?`

Arguments

None

Returns

<width_avg>,<width_min>,<width_max>

Where
<width_avg>::=<NRf> is the average pulse width.
<width_min>::=<NRf> is the minimum pulse width.
<width_max>::=<NRf> is the maximum pulse width.
Unit: Seconds.

Examples

`READ:PULSE:STATISTICS:WIDTH?` might return `4.636E-6,4.630E-6,4.643E-6` for the pulse width in the pulse statistics.

READ:PULSe:TRACe:X? (Query Only)

Returns the time values of the pulse trace. Use the DISPlay:PULSe:SELect:NUMBer command to select the pulse, and the DISPlay:PULSe:SELect:RESult command to select the measurement result.

Conditions

Measurement views: Pulse trace

Group

Read commands

Syntax `READ:PULSe:TRACe:X?`

Arguments None

Returns `#<num_digit><num_byte><X(1)><X(2)>...<X(n)>`

Where
`<num_digit>` is the number of digits in <num_byte>.
`<num_byte>` is the number of bytes of data that follow.
`<X(n)>` is the time in seconds at the point #n,
4-byte little endian floating-point format specified in IEEE 488.2.

Examples `READ:PULSE:TRACE:X?` might return `#43204xxxx...` (3204-byte data) for the time values of the trace.

READ:PULSe:TRACe:XY? (Query Only)

Returns the horizontal (time) and vertical value pairs of the pulse trace. Use the DISPlay:PULSe:SELect:NUMBer command to select the pulse, and the DISPlay:PULSe:SELect:RESult command to select the measurement result.

Conditions Measurement views: Pulse trace

Group Read commands

Syntax `READ:PULSe:TRACe:XY?`

Arguments None

Returns `#<num_digit><num_byte><x(1)><y(1)><x(2)><y(2)>...<x(n)><y(n)>`

Where
`<num_digit>` is the number of digits in <num_byte>.
`<num_byte>` is the number of bytes of data that follow.
`<x(n)><y(n)>` is the horizontal value (time in seconds) and vertical value pair for the point #n, 4-byte little endian floating-point format specified in IEEE 488.2.

The vertical unit depends on measurement results: Hz for frequency error and deviation, degrees for phase error and deviation, otherwise dBm. The vertical unit can be changed by the [SENSe]:POWer:UNITs command.

Examples `READ:PULSE:TRACE:XY?` might return `#43204xxxx...` (3204-byte data) for the horizontal (time) and vertical value pairs of the pulse trace.

READ:PULSe:TRACe[:Y]? (Query Only)

Returns the vertical values of the pulse trace. Use the DISPlay:PULSe:SELect:NUMBer command to select the pulse, and the DISPlay:PULSe:SELect:RESult command to select the measurement result.

Conditions Measurement views: Pulse trace

Group Read commands

Syntax `READ:PULSe:TRACe[:Y]?`

Arguments None

Returns `#<num_digit><num_byte><y(1)><y(2)>...<y(n)>`

Where
`<num_digit>` is the number of digits in <num_byte>.
`<num_byte>` is the number of bytes of data that follow.
`<y(n)>` is the vertical value of the pulse trace at the point #n, 4-byte little endian floating-point format specified in IEEE 488.2.

The unit depends on measurement results: Hz for frequency error and deviation, degrees for phase error and deviation, otherwise dBm. The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples `READ:PULSE:TRACE:Y?` might return `#43204xxxx...` (3204-byte data) for the vertical values of the pulse trace.

READ:SGRam? (Query Only)

Returns the spectrogram trace data.The line is selected using the TRACe:SGRam:SELect:LINE command.

Conditions Measurement views: Spectrogram

Group Read commands

Syntax READ:SGRam?

Arguments None

Returns #<num_digit><num_byte><data(1)><data(2)>...<data(n)>

Where
<num_digit> is the number of digits in <num_byte>.
<num_byte> is the number of bytes of data that follow.
<data(n)> is the trace data in dBm for the point n,
4-byte little endian floating-point format specified in IEEE 488.2.
The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples READ:SGRAM? might return #43204xxxx... (3204-byte data) for the spectrogram trace.

READ:SPECtrum:TRACe<x>? (Query Only)

Returns the trace data in the Spectrum measurement.

The parameter <x> = 1 to 5.

NOTE. *TRACe5 (spectrogram) is valid when the spectrum and spectrogram measurements are running.*

Conditions Measurement views: Spectrum

Group Read commands

Syntax READ:SPECtrum:TRACe<x>?

Arguments None

Returns #<num_digit><num_byte><data(1)><data(2)>...<data(n)>

Where
<num_digit> is the number of digits in <num_byte>.
<num_byte> is the number of bytes of data that follow.
<data(n)> is the trace data in dBm for the point n,
4-byte little endian floating-point format specified in IEEE 488.2.
The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples READ:SPECTRUM:TRACE1 might return #43204xxxx... (3204-byte data) for Trace 1 in the Spectrum measurement.

READ:SPURious:CARRier:POWer? (Query Only)

Returns the carrier power in the Spurious measurement.

Conditions Measurement views: Spurious

Group Read commands

Syntax `READ:SPURious:CARRier:POWer?`

Arguments None

Returns <NRf> Carrier power in dBm.
The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples READ:SPURIOUS:CARRIER:POWER? might return 4.227, indicating that the carrier power is 4.227 dBm.

READ:SPURious:COUNt? (Query Only)

Returns the number of spurious signals in the Spurious measurement.

Conditions Measurement views: Spurious

Group Read commands

Syntax `READ:SPURious:COUNt?`

Arguments None

Returns <NRf> The spurious count.

Examples READ:SPURIOUS:COUNT? might return 4, indicating that the spurious count is 4.

READ:SPURious:PASS? (Query Only)

Returns the pass/fail limit test result in the Spurious measurement.

Conditions Measurement views: Spurious

Group Read commands

Syntax `READ:SPURious:PASS?`

Arguments None

Returns `0` (fail) or `1` (pass).

Examples `READ:SPURIOUS:PASS?` might return `1`, indicating that the limit test was successful.

READ:SPURious:SPECtrum:X? (Query Only)

Returns the frequencies of the spectrum trace in the Spurious measurement.

Conditions Measurement views: Spurious

Group Read commands

Syntax `READ:SPURious:SPECtrum:X?`

Arguments None

Returns `#<num_digit><num_byte><x(1)><x(2)>...<x(n)>`

Where
`<num_digit>` is the number of digits in <num_byte>.
`<num_byte>` is the number of bytes of data that follow.
`<x(n)>` is the frequency (Hz) at the point #n,
4-byte little endian floating-point format specified in IEEE 488.2.

Examples `READ:SPURIOUS:SPECTRUM:X?` might return `#516020xxxx...` (16020-byte data) for the frequencies of the spectrum trace in the Spurious measurement.

READ:SPURious:SPECtrum:XY? (Query Only)

Returns the frequency and amplitude pairs of the spectrum trace in the Spurious measurement.

Conditions Measurement views: Spurious

Group Read commands

Syntax `READ:SPURious:SPECtrum:XY?`

Arguments None

Returns `#<num_digit><num_byte><x(1)><y(1)><x(2)><y(2)>...<x(n)><y(n)>`

Where
`<num_digit>` is the number of digits in <num_byte>.
`<num_byte>` is the number of bytes of data that follow.
`<x(n)><y(n)>` is the frequency (Hz) and amplitude (dBm) pair for the point #n, 4-byte little endian floating-point format specified in IEEE 488.2.
The amplitude unit can be changed by the [SENSe]:POWer:UNITs command.

Examples `READ:SPURIOUS:SPECTRUM:XY?` might return `#516020xxxx...` (16020-byte data) for the frequency and amplitude pairs of the spectrum trace in the Spurious measurement.

READ:SPURious:SPECtrum[:Y]? (Query Only)

Returns the amplitudes of the spectrum trace in the Spurious measurement.

Conditions Measurement views: Spurious

Group Read commands

Syntax `READ:SPURious:SPECtrum[:Y]?`

Related Commands READ:SPURious:SPECtrum:X?

Arguments None

Returns `#<num_digit><num_byte><y(1)><y(2)>...<y(n)>`

Where
`<num_digit>` is the number of digits in <num_byte>.
`<num_byte>` is the number of bytes of data that follow.
`<y(n)>` is the amplitude (dBm) at the point #n,
4-byte little endian floating-point format specified in IEEE 488.2.
The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples `READ:SPURIOUS:SPECTRUM:Y` might return `#516020xxxx...` (16020-byte data) for the amplitudes of the spectrum trace in the Spurious measurement.

READ:SPURious:SPUR<x>:AMPLitude:ABSolute? (Query Only)

Returns the absolute amplitude of the specified spurious signal in the Spurious measurement.

Conditions Measurement views: Spurious

Group Read commands

Syntax `READ:SPURious:SPUR<x>:AMPLitude:ABSolute?`

Arguments None

Returns `<NRf>` Absolute amplitude of the specified spurious in dBm.
The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples `READ:SPURIOUS:SPUR1:AMPLITUDE:ABSOLUTE?` might return `-19.782`, indicating that the absolute amplitude of Spurious #1 is -19.782 dBm.

READ:SPURious:SPUR<x>:AMPLitude:RELative? (Query Only)

Returns the relative amplitude of the specified spurious signal in the Spurious measurement.

Conditions Measurement views: Spurious

Group Read commands

Syntax `READ:SPURious:SPUR<x>:AMPLitude:RELative?`

Arguments None

Returns `<NRf>` Relative amplitude of the specified spurious signal in dB.
Use the [SENSe]:SPURious:REFerence command to set the power reference.

Examples `READ:SPURIOUS:SPUR1:AMPLITUDE:RELATIVE?` might return `-9.782`, indicating that the relative amplitude of Spurious #1 is -9.782 dB.

READ:SPURious:SPUR<x>:FREQuency:ABSolute? (Query Only)

Returns the absolute frequency of the specified spurious signal in the Spurious measurement.

Conditions Measurement views: Spurious

Group Read commands

Syntax `READ:SPURious:SPUR<x>:FREQuency:ABSolute?`

Arguments None

Returns `<NRf>` Absolute frequency of the spurious signal in Hz.

Examples `READ:SPURIOUS:SPUR1:FREQUENCY:ABSOLUTE?` might return `2.235E+9`, indicating that the absolute frequency of Spurious #1 is 2.235 GHz.

READ:SPURious:SPUR<x>:FREQuency:RELative? (Query Only)

Returns the relative frequency of the specified spurious signal to the carrier in the Spurious measurement. This command is valid when [SENSe]:SPURious:REFerence is set to CARRier.

Conditions Measurement views: Spurious

Group Read commands

Syntax `READ:SPURious:SPUR<x>:FREQuency:RELative?`

Arguments None

Returns `<NRf>` Relative frequency of the spurious signal to the carrier in Hz.

Examples `READ:SPURIOUS:SPUR1:FREQUENCY:RELATIVE?` might return `3.634E+6`, indicating that the relative frequency of Spurious #1 is 3.634 MHz.

READ:SPURious:SPUR<x>:LIMit:ABSolute? (Query Only)

Returns the absolute amplitude of the limit for the specified spurious signal in the Spurious measurement.

Conditions Measurement views: Spurious

Group Read commands

Syntax `READ:SPURious:SPUR<x>:LIMit:ABSolute?`

Arguments None

Returns `<NRf>` Absolute amplitude of the limit for the specified spurious signal in dBm. The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples `READ:SPURIOUS:SPUR1:LIMIT:ABSOLUTE?` might return `-50.0`, indicating that the absolute amplitude of the limit for Spurious #1 is -50 dBm.

READ:SPURious:SPUR<x>:LIMit:RELative? (Query Only)

Returns the relative amplitude of the limit for the specified spurious signal in the Spurious measurement.

Conditions Measurement views: Spurious

Group Read commands

Syntax `READ:SPURious:SPUR<x>:LIMit:RELative?`

Arguments None

Returns `<NRf>` Relative amplitude of the limit for the specified spurious signal in dB. Use the [SENSe]:SPURious:REFerence command to set the power reference.

Examples `READ:SPURIOUS:SPUR1:LIMIT:RELATIVE?` might return `-10.0`, indicating that the relative amplitude of the limit for Spurious #1 is -10 dB.

READ:SPURious:SPUR<x>:LIMit:VIOLation? (Query Only)

Returns whether the specified spurious signal exceeds the limit or not.

Conditions Measurement views: Spurious

Group Read commands

Syntax `READ:SPURious:SPUR<x>:LIMit:VIOLation?`

Arguments None

Returns `0` (under the limit) or `1` (over the limit).

Examples `READ:SPURIOUS:SPUR1:LIMIT:VIOLATION?` might return `1`, indicating that Spurious #1 exceeds the limit.

READ:SPURious:SPUR<x>:RANGe? (Query Only)

Returns the frequency range in which the specified spurious signal occurred.

Conditions Measurement views: Spurious

Group Read commands

Syntax `READ:SPURious:SPUR<x>:RANGe?`

Arguments None

Returns `<string>` "A" to "T" representing Range A to T, respectively.

Examples `READ:SPURIOUS:SPUR1:RANGE?` might return `"E"`, indicating that Spurious #1 is in Range E.

READ:SQUality:FREQuency:ERRor? (Query Only)

Returns the frequency error in the signal quality measurement.

Conditions Measurement views: Signal quality

Group Read commands

Syntax `READ:SQUality:FREQuency:ERRor?`

Arguments None

Returns `<NRf>` Frequency error in Hz.

Examples `READ:SQUALITY:FREQUENCY:ERROR?` might return `612.043E+3`, indicating that the frequency error is 612.043 kHz.

READ:SQUality:GAIN:IMBalance? (Query Only)

Returns the gain imbalance in the signal quality measurement.

Conditions Measurement views: Signal quality

Group Read commands

Syntax `READ:SQUality:GAIN:IMBalance?`

Arguments None

Returns `<NRf>` Gain imbalance in dB.

Examples `READ:SQUALITY:GAIN:IMBALANCE?` might return `-57.746E-3`, indicating that the gain imbalance is -0.057746 dB.

READ:SQUality:ORIGin:OFFSet? (Query Only)

Returns the origin offset in the signal quality measurement.

Conditions Measurement views: Signal quality

Group Read commands

Syntax `READ:SQUality:ORIGin:OFFSet?`

Arguments None

Returns `<NRf>` Origin offset in dB.

Examples `READ:SQUALITY:ORIGIN:OFFSET?` might return `-44.968`, indicating that the origin offset is -44.968 dB.

READ:SQUality:PEAK:EVM? (Query Only)

Returns the peak EVM (%) in the signal quality measurement.

Conditions Measurement views: Signal quality

Group Read commands

Syntax `READ:SQUality:PEAK:EVM?`

Arguments None

Returns `<NRf>` Peak EVM in percent (%).

Examples `READ:SQUALITY:PEAK:EVM?` might return `4.276`, indicating that the peak EVM is 4.276%.

READ:SQUality:PEAK:EVM:DB? (Query Only)

Returns the peak EVM (dB) in the signal quality measurement.

Conditions Measurement views: Signal quality

Group Read commands

Syntax `READ:SQUality:PEAK:EVM:DB?`

Arguments None

Returns `<NRf>` Peak EVM in dB.

Examples `READ:SQUALITY:PEAK:EVM:DB?` might return `-27.358`, indicating that the peak EVM is -27.358 dB.

READ:SQUality:PEAK:EVM:LOCation? (Query Only)

Returns the time at which the EVM is peak.

Conditions Measurement views: Signal quality

Group Read commands

Syntax `READ:SQUality:PEAK:EVM:LOCation?`

Arguments None

Returns <NRf> The time in symbol number at which the EVM is peak.
The unit can be changed by the [SENSe]:DDEMod:TIMe:UNITs command.

Examples `READ:SQUALITY:PEAK:EVM:LOCATION?` might return `68.000`, indicating that the EVM is peak at symbol #68.000.

READ:SQUality:PEAK:MERRor? (Query Only)

Returns the peak magnitude error (%) in the signal quality measurement.

Conditions Measurement views: Signal quality

Group Read commands

Syntax `READ:SQUality:PEAK:MERRor?`

Arguments None

Returns <NRf> Peak magnitude error in percent (%).

Examples `READ:SQUALITY:PEAK:MERROR?` might return `3.595`, indicating that the peak magnitude error is 3.595%.

READ:SQUality:PEAK:MERRor:DB? (Query Only)

Returns the peak magnitude error (dB) in the signal quality measurement.

Conditions Measurement views: Signal quality

Group Read commands

Syntax `READ:SQUality:PEAK:MERRor:DB?`

Arguments None

Returns <NRf> Peak magnitude error in dB.

Examples `READ:SQUALITY:PEAK:MERROR:DB?` might return `-28.583`, indicating that the magnitude error is -28.583 dB.

READ:SQUality:PEAK:MERRor:LOCation? (Query Only)

Returns the time at which the magnitude error is peak.

Conditions Measurement views: Signal quality

Group Read commands

Syntax `READ:SQUality:PEAK:MERRor:LOCation?`

Arguments None

Returns `<NRf>` The time in symbol number at which the magnitude error is peak.
The unit can be changed by the [SENSe]:DDEMod:TIMe:UNITs command.

Examples `READ:SQUALITY:PEAK:MERROR:LOCATION?` might return `68.000`, indicating that the magnitude error is peak at symbol #68.

READ:SQUality:PEAK:PERRor? (Query Only)

Returns the peak phase error in the signal quality measurement.

Conditions Measurement views: Signal quality

Group Read commands

Syntax `READ:SQUality:PEAK:PERRor?`

Arguments None

Returns `<NRf>` Peak phase error in degrees.

Examples `READ:SQUALITY:PEAK:PERROR?` might return `1.907`, indicating that the peak phase error is 1.907°.

READ:SQUality:PEAK:PERRor:LOCation? (Query Only)

Returns the time at which the phase error is peak.

Conditions Measurement views: Signal quality

Group Read commands

Syntax `READ:SQUality:PEAK:PERRor:LOCation?`

Arguments None

Returns `<NRf>` The time in symbol number at which the phase error is peak. The unit can be changed by the [SENSe]:DDEMod:TIMe:UNITs command.

Examples `READ:SQUALITY:PEAK:PERROR:LOCATION?` might return `68.000`, indicating that the phase error is peak at symbol #68.

READ:SQUality:QUADrature:ERRor? (Query Only)

Returns the quadrature error in the signal quality measurement.

Conditions Measurement views: Signal quality

Group Read commands

Syntax `READ:SQUality:QUADrature:ERRor?`

Arguments None

Returns `<NRf>` Quadrature error in degrees.

Examples `READ:SQUALITY:QUADRATURE:ERROR?` might return `-14.264E-3`, indicating that the quadrature error is -0.014264°.

READ:SQUality:RHO? (Query Only)

Returns the ρ (waveform quality) value in the signal quality measurement.

Conditions Measurement views: Signal quality

Group Read commands

Syntax `READ:SQUality:RHO?`

Arguments None

Returns `<NRf>` ρ value.

Examples `READ:SQUALITY:RHO?` might return `998.703E-3`, indicating that ρ is 0.998703.

READ:SQUality:RMS:EVM? (Query Only)

Returns the RMS EVM (%) in the signal quality measurement.

Conditions Measurement views: Signal quality

Group Read commands

Syntax `READ:SQUality:RMS:EVM?`

Arguments None

Returns `<NRf>` RMS EVM in percent (%).

Examples `READ:SQUALITY:RMS:EVM?` might return `2.417`, indicating that the RMS EVM is 2.417%.

READ:SQUality:RMS:EVM:DB? (Query Only)

Returns the RMS EVM (dB) in the signal quality measurement.

Conditions Measurement views: Signal quality

Group Read commands

Syntax `READ:SQUality:RMS:EVM:DB?`

Arguments None

Returns `<NRf>` RMS EVM in dB.

Examples `READ:SQUALITY:RMS:EVM:DB?` might return `-32.356`, indicating that the RMS EVM is -32.356 dB.

READ:SQUality:RMS:MER:DB? (Query Only)

Returns the RMS MER (Modulation Error Ratio) in dB in the signal quality measurement.

Conditions Measurement views: Signal quality

Group Read commands

Syntax `READ:SQUality:RMS:MER:DB?`

Arguments None

Returns `<NRf>` RMS MER in dB.

Examples `READ:SQUALITY:RMS:MER:DB?` might return `27.394`, indicating that the RMS MER is 27.394 dB.

READ:SQUality:RMS:MERRor? (Query Only)

Returns the RMS magnitude error (%) in the signal quality measurement.

Conditions Measurement views: Signal quality

Group Read commands

Syntax `READ:SQUality:RMS:MERRor?`

Arguments None

Returns `<NRf>` RMS magnitude error in percent (%).

Examples `READ:SQUALITY:RMS:MERROR?` might return `1.837`, indicating that the RMS magnitude error is 1.837%.

READ:SQUality:RMS:MERRor:DB? (Query Only)

Returns the RMS magnitude error (dB) in the signal quality measurement.

Conditions Measurement views: Signal quality

Group Read commands

Syntax `READ:SQUality:RMS:MERRor:DB?`

Arguments None

Returns `<NRf>` RMS magnitude error in dB.

Examples `READ:SQUALITY:RMS:MERROR:DB?` might return `-34.706`, indicating that the magnitude error is -34.706 dB.

READ:SQUality:RMS:PERRor? (Query Only)

Returns the RMS phase error in the signal quality measurement.

Conditions Measurement views: Signal quality

Group Read commands

Syntax `READ:SQUality:RMS:PERRor?`

Arguments None

Returns `<NRf>` RMS phase error in degrees.

Examples `READ:SQUALITY:RMS:PERROR?` might return `893.472E-3`, indicating that the RMS phase error is 0.893472°.

READ:TOVerview? (Query Only)

Returns the trace data in the time overview.

Conditions Measurement views: Time overview

Group Read commands

Syntax `READ:TOVerview?`

Arguments None

Returns `#<num_digit><num_byte><data(1)><data(2)>...<data(n)>`

Where
`<num_digit>` is the number of digits in <num_byte>.
`<num_byte>` is the number of bytes of data that follow.
`<data(n)>` is the trace data in dBm for the point n,
4-byte little endian floating-point format specified in IEEE 488.2.
The unit can be changed by the [SENSe]:POWer:UNITs command.

Examples `READ:TOVERVIEW?` might return `#43204xxxx...` (3204-byte data) for the trace in the time overview.

*RST (No Query Form)

Returns the instrument settings to the factory defaults ((See page 3-1, *Factory Initialization Settings*.)

The *RST command does not alter the following

- The state of the GPIB interface.
- The selected GPIB address of the analyzer.
- Alignment data that affect device specifications.
- The Output Queue.
- The Service Request Enable Register setting.
- The Standard Event Status Enable Register setting.
- The Power-on status clear flag setting.
- Stored settings.

NOTE. *Execution of the *RST command is not complete until all changes from resetting the instrument are completed. Following commands and queries will not be executed until these actions are completed.*

Conditions Measurement views: All

Group IEEE common commands

Syntax `*RST`

Related Commands *CLS

Arguments None

Examples `*RST` returns the instrument settings to the factory defaults.

[SENSe]:ACPower:AVERage

Selects or queries how to average waveform in the Channel power and ACPR measurement.

Conditions Measurement views: Channel power and ACPR

Group Sense commands

Syntax

```
[SENSe]:ACPower:AVERage { OFF | TIMe | FREQuency }
[SENSe]:ACPower:AVERage?
```

Arguments `OFF` disables averaging.

`TIMe` averages waveform using time samples.

`FREQuency` averages waveform using frequency samples.

Examples `SENSE:ACPOWER:AVERAGE TIMe` averages waveform using time samples.

[SENSe]:ACPower:AVERage:COUNt

Sets or queries the number of traces for averaging in the Channel power and ACPR measurement.

Conditions Measurement views: Channel power and ACPR

Group Sense commands

Syntax

```
[SENSe]:ACPower:AVERage:COUNt <number>
[SENSe]:ACPower:AVERage:COUNt?
```

Arguments `<number>::=<NR1>` specifies the average count. Range: 2 to 10000.

Examples `SENSE:ACPOWER:AVERAGE:COUNT 64` sets the average count to 64.

[SENSe]:ACPower:{BANDwidth|BWIDth}[:RESolution]

Sets or queries the resolution bandwidth (RBW). Programming a specified RBW sets [SENSe]:ACPower:{BANDwidth|BWIDth}[:RESolution]:AUTO OFF.

Conditions Measurement views: Channel power and ACPR

Group Sense commands

Syntax

```
[SENSe]:ACPower:{BANDwidth|BWIDth}[:RESolution] <value>
[SENSe]:ACPower:{BANDwidth|BWIDth}[:RESolution]?
```

Related Commands [SENSe]:ACPower:{BANDwidth|BWIDth}[:RESolution]:AUTO

Arguments `<value>::=<NRf>` specifies the RBW. Range: 100 Hz to 5 MHz.

Examples `SENSE:ACPOWER:BANDWIDTH:RESOLUTION 200kHz` sets the RBW to 200 kHz.

[SENSe]:ACPower:{BANDwidth|BWIDth}[:RESolution]:ACTual? (Query Only)

Queries the actual resolution bandwidth (RBW) in the Channel power and ACPR measurement.

Conditions Measurement views: Channel power and ACPR

Group Sense commands

Syntax

```
[SENSe]:ACPower:{BANDwidth|BWIDth}[:RESolution]:ACTual?
```

Arguments None

Returns `<NRf>` The actual RBW in Hz.

Examples `SENSE:ACPOWER:BANDWIDTH:RESOLUTION:ACTUAL?` might return `299.624E+3`, indicating that the actual RBW is 299.624 kHz.

[SENSe]:ACPower:{BANDwidth|BWIDth}[:RESolution]:AUTO

Determines whether to set the resolution bandwidth (RBW) automatically or manually in the Channel power and ACPR measurement.

Conditions Measurement views: Channel power and ACPR

Group Sense commands

Syntax

```
[SENSe]:ACPower:{BANDwidth|BWIDth}[:RESolution]:AUTO { OFF |
ON | 0 | 1 }
[SENSe]:ACPower:{BANDwidth|BWIDth}[:RESolution]:AUTO?
```

Arguments OFF or 0 specifies that the RBW is set manually using the [SENSe]:ACPower:{BANDwidth|BWIDth}[:RESolution] command.

ON or 1 specifies that the RBW is set automatically.

Examples SENSE:ACPOWER:BANDWIDTH:AUTO ON sets the RBW automatically.

[SENSe]:ACPower:{BANDwidth|BWIDth}:VIDeo

Sets or queries the video bandwidth (VBW). Programming a specified VBW sets [SENSe]:ACPower:{BANDwidth|BWIDth}:VIDeo:STATe OFF.

Conditions Measurement views: Channel power and ACPR

Group Sense commands

Syntax

```
[SENSe]:ACPower:{BANDwidth|BWIDth}:VIDeo <value>
[SENSe]:ACPower:{BANDwidth|BWIDth}:VIDeo?
```

Related Commands [SENSe]:ACPower:{BANDwidth|BWIDth}:VIDeo:STATe

Arguments <value>::=<NRf> specifies the VBW.
Range: Current RBW/10^4 (1 Hz minimum) to Current RBW.

Examples SENSE:ACPOWER:BANDWIDTH:VIDEO 200kHz sets the VBW to 200 kHz.

[SENSe]:ACPower:{BANDwidth|BWIDth}:VIDeo:STATe

Determines whether to enable or disable the video bandwidth (VBW) in the Channel power and ACPR measurement.

Conditions Measurement views: Channel power and ACPR

Group Sense commands

Syntax

```
[SENSe]:ACPower:{BANDwidth|BWIDth}:VIDeo:STATe { OFF | ON
| 0 | 1 }
[SENSe]:ACPower:{BANDwidth|BWIDth}:VIDeo:STATe?
```

Arguments `OFF` or `0` disables the VBW.

`ON` or `1` enables the VBW.

Examples `SENSE:ACPOWER:BANDWIDTH:VIDEO:STATE ON` enables the VBW.

[SENSe]:ACPower:CHANnel:{BANDwidth|BWIDth}

Sets or queries frequency bandwidth of each channel (all share the same value) in the Channel power and ACPR measurement.

Conditions Measurement views: Channel power and ACPR

Group Sense commands

Syntax

```
[SENSe]:ACPower:CHANnel:{BANDwidth|BWIDth} <value>
[SENSe]:ACPower:CHANnel:{BANDwidth|BWIDth}?
```

Arguments `<value>::=<NRf>` specifies the channel bandwidth. Range: 1 Hz to full span.

Examples `SENSE:ACPOWER:CHANNEL:BANDWIDTH 1.5MHz` sets the channel bandwidth to 1.5 MHz.

[SENSe]:ACPower:CHANnel:FILTer

Selects or queries the adjacent channel filter in the Channel power and ACPR measurement.

Conditions Measurement views: Channel power and ACPR

Group Sense commands

Syntax

```
[SENSe]:ACPower:CHANnel:FILTer { RRCosine | NONE }
[SENSe]:ACPower:CHANnel:FILTer?
```

Arguments `RRCosine` uses the Root-Raised Cosine filter.

`NONE` uses no filter.

Examples `SENSE:ACPOWER:CHANNEL:FILTER RRCosine` uses the Root-Raised Cosine filter for the Channel power and ACPR measurement.

[SENSe]:ACPower:CHANnel:PAIRs

Sets or queries the number of adjacent channel pairs (upper and lower) in the Channel power and ACPR measurement.

Conditions Measurement views: Channel power and ACPR

Group Sense commands

Syntax

```
[SENSe]:ACPower:CHANnel:PAIRs <number>
[SENSe]:ACPower:CHANnel:PAIRs?
```

Arguments `<number>::=<NR1>` specifies the number of adjacent pairs. Range: 0 to 50.

Examples `SENSE:ACPOWER:CHANNEL:PAIRS 5` sets five adjacent channel pairs.

[SENSe]:ACPower:CHANnel:SPACing

Sets or queries frequency difference between centers of each channel in the Channel power and ACPR measurement.

Conditions Measurement views: Channel power and ACPR

Group Sense commands

Syntax
```
[SENSe]:ACPower:CHANnel:SPACing <value>
[SENSe]:ACPower:CHANnel:SPACing?
```

Arguments `<value>::=<NRf>` specifies the channel-to-channel spacing.
Range: 1 Hz to 1 GHz.

Examples `SENSE:ACPOWER:CHANNEL:SPACING 5MHz` sets the channel-to-channel spacing to 5 MHz.

[SENSe]:ACPower:CHIPrate

Sets or queries the chip rate when [SENSe]:ACPower:CHANnel:FILTer is set to RRCosine (Root Raised Cosine).

Conditions Measurement views: Channel power and ACPR

Group Sense commands

Syntax
```
[SENSe]:ACPower:CHIPrate <value>
[SENSe]:ACPower:CHIPrate?
```

Related Commands [SENSe]:ACPower:CHANnel:FILTer

Arguments `<value>::=<NRf>` specifies the chip rate. Range: 100 Hz to 105 MHz.

Examples `SENSE:ACPOWER:CHIPRATE 5kHz` sets the chip rate to 5 kHz.

[SENSe]:ACPower:CLEar:RESults (No Query Form)

Restarts the average trace.

Conditions Measurement views: Channel power and ACPR

Group Sense commands

Syntax `[SENSe]:ACPower:CLEar:RESults`

Arguments None

Examples `SENSE:ACPOWER:CLEAR:RESULTS` restarts the average trace.

[SENSe]:ACPower:FREQuency

Sets or queries the center frequency in the Channel power and ACPR measurement.

Conditions Measurement views: Channel power and ACPR

Group Sense commands

Syntax
```
[SENSe]:ACPower:FREQuency <value>
[SENSe]:ACPower:FREQuency?
```

Arguments `<value>::=<NRf>` specifies the center frequency.
Range: 0 to 6.2 GHz (RSA6106A) / 14 GHz (RSA6114A).

Examples `SENSE:ACPOWER:FREQUENCY 2.35GHz` sets the center frequency to 2.35 GHz.

[SENSe]:ACPower:FREQuency:STEP

Sets or queries the frequency step size in the Channel power and ACPR measurement. Programming a specified step size sets [SENSe]:ACPower FREQuency:STEP:AUTO OFF.

Conditions Measurement views: Channel power and ACPR

Group Sense commands

Syntax
```
[SENSe]:ACPower:FREQuency:STEP <value>
[SENSe]:ACPower:FREQuency:STEP?
```

Related Commands [SENSe]:ACPower:FREQuency:STEP:AUTO

Arguments `<value>::=<NRf>` specifies the frequency step size.
Range: 0 to 6.2 GHz (RSA6106A) / 14 GHz (RSA6114A).

Examples `SENSE:ACPOWER:FREQUENCY:STEP 1kHz` sets the frequency step size to 1 kHz.

[SENSe]:ACPower:FREQuency:STEP:AUTO

Determines whether to set the frequency step size automatically or manually in the Channel power and ACPR measurement.

Conditions Measurement views: Channel power and ACPR

Group Sense commands

Syntax

```
[SENSe]:ACPower:FREQuency:STEP:AUTO { OFF | ON | 0 | 1 }
[SENSe]:ACPower:FREQuency:STEP:AUTO?
```

Arguments `OFF` or `0` specifies that the frequency step size is set manually using the [SENSe]:ACPower:FREQuency:STEP command.

`ON` or `1` specifies that the frequency step size is set automatically.

Examples `SENSE:ACPOWER:FREQUENCY:STEP:AUTO ON` specifies that the frequency step size is set automatically.

[SENSe]:ACPower:NFLoor:STATe

Determines whether to enable or disable the correction for noise floor.

Conditions Measurement views: Channel power and ACPR

Group Sense commands

Syntax

```
[SENSe]:ACPower:NFLoor:STATe { OFF | ON | 0 | 1 }
[SENSe]:ACPower:NFLoor:STATe?
```

Arguments

OFF or 0 disables the correction for noise floor.

ON or 1 enables the correction for noise floor.

Examples

SENSE:ACPOWER:NFLOOR:STATE ON enables the correction for noise floor.

[SENSe]:ACPower:OPTimize:SPAN

Selects or queries the optimization method in the Channel power and ACPR measurement.

Conditions

Measurement views: Channel power and ACPR

Group

Sense commands

Syntax

```
[SENSe]:ACPower:OPTimize:SPAN { RTBandwidth | DRANge }
[SENSe]:ACPower:OPTimize:SPAN?
```

Arguments

RTBandwidth optimizes the measurement for real-time bandwidth.

DRANge optimizes the measurement for dynamic range.

Examples

SENSE:ACPOWER:OPTIMIZE:SPAN RTBandwidth optimizes the measurement for real-time bandwidth.

[SENSe]:ACPower:RRCRolloff

Sets or queries the filter parameter (roll-off ratio) for the Root Raised Cosine filter.

Conditions

Measurement views: Channel power and ACPR

Group

Sense commands

Syntax

```
[SENSe]:ACPower:RRCRolloff <value>
[SENSe]:ACPower:RRCRolloff?
```

Related Commands

[SENSe]:ACPower:CHANnel:FILTer

Arguments `<value>::=<NRf>` specifies the filter parameter.
Range: 0.0001 to 1 in 0.0001 steps.

Examples `SENSE:ACPOWER:RRCROLLOFF 0.3` sets the filter parameter to 0.3.

[SENSe]:ACQuisition:{BANDwidth|BWIDth}

Sets or queries the acquisition bandwidth (frequency range of the acquisition) when [SENSe]:ACQuisition:MODE is set to SAMPles or LENGth.

Conditions Measurement views: All

Group Sense commands

Syntax

```
[SENSe]:ACQuisition:{BANDwidth|BWIDth} <value>
[SENSe]:ACQuisition:{BANDwidth|BWIDth}?
```

Arguments `<value>::=<NRf>` specifies the acquisition bandwidth.
Range: 1 MHz to 40 MHz (Standard) / 110 MHz (Option 110).

Examples `SENSE:ACQUISITION:BANDWIDTH 30MHz` sets the acquisition bandwidth to 30 MHz.

[SENSe]:ACQuisition:FFRame:ACTual? (Query Only)

Queries the actual number of Fast Frames.

Conditions Measurement views: All

Group Sense commands

Syntax

```
[SENSe]:ACQuisition:FFRame:ACTual?
```

Arguments None

Returns `<NR1>` The actual number of Fast Frames.

Examples `SENSE:ACQUISITION:FFRAME:ACTUAL?` might return `178`, indicating that the actual number of Fast Frames is 178.

[SENSe]:ACQuisition:FFRame:LIMit

Sets or queries the limit number of Fast Frames.

Conditions Measurement views: All

Group Sense commands

Syntax

```
[SENSe]:ACQuisition:FFRame:LIMit
[SENSe]:ACQuisition:FFRame:LIMit?
```

Arguments `<value>::=<NR1>` specifies the fast frame limit. Range: 1 to 65535 frames.

Examples `SENSE:ACQUISITION:FFRAME:LIMIT 500` sets the Fast Frame limit to 500.

[SENSe]:ACQuisition:FFRame:STATe

Determines whether to enable or disable the Fast Frame.

Conditions Measurement views: All

Group Sense commands

Syntax

```
[SENSe]:ACQuisition:FFRame:STATe { OFF | ON | 0 | 1 }
[SENSe]:ACQuisition:FFRame:STATe?
```

Arguments `OFF` or `0` disables the Fast Frame.

`ON` or `1` enables the Fast Frame.

Examples `SENSE:ACQUISITION:FFRAME:STATE ON` enables the Fast Frame.

[SENSe]:ACQuisition:MEMory:AVAilable:SAMPles? (Query Only)

Returns the amount of acquisition memory available in the instrument.

Conditions Measurement views: All

Group Sense commands

Syntax `[SENSe]:ACQuisition:MEMory:AVAilable:SAMPles?`

Arguments None

Returns `<NRf>` The amount of acquisition memory available in samples.

Examples `SENSE:ACQUISITION:MEMORY:AVAILABLE:SAMPLES?` might return `999.424E+3`, indicating that 999424 samples are available.

[SENSe]:ACQuisition:MEMory:CAPacity[:TIME]? (Query Only)

Returns the acquisition memory capacity (maximum period of time that can be acquired with the acquisition memory).

Conditions Measurement views: All

Group Sense commands

Syntax `[SENSe]:ACQuisition:MEMory:CAPacity[:TIME]?`

Arguments None

Returns `<NRf>` The acquisition memory capacity in seconds.

Examples `SENSE:ACQUISITION:MEMORY:CAPACITY:TIME?` might return `26.651E-3`, indicating that 26.651 ms can be acquired.

[SENSe]:ACQuisition:MEMory:USED[:PERCent]? (Query Only)

Returns the percentage of the capacity used based on the current settings.

Conditions Measurement views: All

Group Sense commands

Syntax
```
[SENSe]:ACQuisition:MEMory:USED[:PERCent]?
```

Arguments None

Returns `<NRf>` The percentage of the capacity used.

Examples `SENSE:ACQUISITION:MEMORY:USED:PERCENT?` might return `50.0`, indicating that 50% is used.

[SENSe]:ACQuisition:MODE

Selects or queries the acquisition mode (how to determine the sampling parameters of acquisition bandwidth, samples, and length).

Conditions Measurement views: All

Group Sense commands

Syntax
```
[SENSe]:ACQuisition:MODE { AUTO | SAMPles | LENGth }
[SENSe]:ACQuisition:MODE?
```

Related Commands [SENSe]:ACQuisition:{BANDwidth|BWIDth}, [SENSe]:ACQuisition:SAMPles, [SENSe]:ACQuisition:SEConds

Arguments `AUTO` sets the all sampling parameters automatically.

`SAMPles` sets the acquisition bandwidth and samples manually, using the [SENSe]:ACQuisition:{BANDwidth|BWIDth} and :SAMPles commands.

`LENGth` sets the acquisition bandwidth and length manually, using the [SENSe]:ACQuisition:{BANDwidth|BWIDth} and :SEConds commands.

Examples `SENSE:ACQUISITION:MODE AUTO` sets the all sampling parameters automatically.

[SENSe]:ACQuisition:SAMPles

Sets or queries the acquisition samples (number of samples acquired over the acquisition time) when [SENSe]:ACQuisition:MODE is set to SAMPles.

Conditions Measurement views: All

Group Sense commands

Syntax
```
[SENSe]:ACQuisition:SAMPles <value>
[SENSe]:ACQuisition:SAMPles?
```

Arguments `<value>::=<NR1>` specifies the acquisition samples. Range: 2 to 1 G samples.

Examples `SENSE:ACQUISITION:SAMPLES 1114` sets the acquisition samples to 1114.

[SENSe]:ACQuisition:SEConds

Sets or queries the acquisition length (time over which the acquisition occurs) when [SENSe]:ACQuisition:MODE is set to LENGth.

Conditions Measurement views: All

Group Sense commands

Syntax
```
[SENSe]:ACQuisition:SEConds <value>
[SENSe]:ACQuisition:SEConds?
```

Arguments `<value>::=<NRf>` specifies the acquisition length.

Examples `SENSE:ACQUISITION:SAMPLES 12.5ms` sets the acquisition length to 12.5 ms.

[SENSe]:ANALysis:ADVanced:DITHer

Determines whether to enable or disable dithering, or set it automatically.

Dither is a random low-level signal consisting of white noise of one quantizing level peak-to-peak amplitude which may be added to an analog signal prior to sampling for the purpose of minimizing quantization error.

Conditions Measurement views: All

Group Sense commands

Syntax

```
[SENSe]:ANALysis:ADVanced:DITHer { AUTO | ON | OFF }
[SENSe]:ANALysis:ADVanced:DITHer?
```

Arguments `AUTO` specifies that the dither is set automatically.

`ON` enables dithering.

`OFF` disables dithering.

Examples `SENSE:ANALYSIS:ADVANCED:DITHER ON` enables dithering.

[SENSe]:ANALysis:ADVanced:DITHer:HWARe:STATus? (Query Only)

Queries the dithering hardware status.

Conditions Measurement views: All

Group Sense commands

Syntax

```
[SENSe]:ANALysis:ADVanced:DITHer:HWARe:STATus?
```

Arguments None

Returns One of the following status information.

Table 2-34: Dithering status

Status	Description
DUNaligned	Dithering is disabled and unaligned.
ON	Dithering is enabled
OFF	Dithering is disabled.

Examples `SENSE:ANALYSIS:ADVANCED:DITHER:HWARE:STATUS?` might return `OFF`, indicating that the dithering is disabled.

[SENSe]:ANALysis:LENGth

Sets or queries the analysis length. Programming a specified length sets [SENSe]:ANALysis:AUTO OFF.

Conditions Measurement views: All

Group Sense commands

Syntax
```
[SENSe]:ANALysis:LENGth <value>
[SENSe]:ANALysis:LENGth?
```

Related Commands [SENSe]:ANALysis:LENGth:AUTO

Arguments `<value>::=<NRf>` specifies the analysis length.
Range: 10 ns to [(acquisition length) - 400 ns].

If [(analysis start) + (analysis length)] > [(acquisition length) - 400 ns], the actual analysis length is reduced to [(acquisition length) - 200 ns].

Examples `SENSE:ANALYSIS:LENGTH 25.625us` sets the analysis length to 25.625 µs.

[SENSe]:ANALysis:LENGth:ACTual? (Query Only)

Queries the actual analysis length.

Conditions Measurement views: All

Group Sense commands

Syntax `[SENSe]:ANALysis:LENGth:ACTual?`

Arguments None

Returns `<NRf>` Actual analysis length in seconds.

Examples `SENSE:ANALYSIS:LENGTH:ACTUAL?` might return `25.625E-6`, indicating that the actual analysis length is 25.625 µs.

[SENSe]:ANALysis:LENGth:AUTO

Determines whether to set the analysis length automatically or manually.

Conditions Measurement views: All

Group Sense commands

Syntax
```
[SENSe]:ANALysis:LENGth:AUTO { OFF | ON | 0 | 1 }
[SENSe]:ANALysis:LENGth:AUTO?
```

Arguments `OFF` or `0` sets the analysis length manually, using the [SENSe]:ANALysis:LENGth command.

`ON` or `1` sets the analysis length automatically.

Examples `SENSE:ANALYSIS:LENGTH:AUTO ON` sets the analysis length automatically.

[SENSe]:ANALysis:REFerence

Selects or queries the analysis time reference.

Conditions Measurement views: All

Group Sense commands

Syntax
```
[SENSe]:ANALysis:REFerence { ACQSTART | TRIGGER }
[SENSe]:ANALysis:REFerence?
```

Arguments

`ACQSTART` specifies the acquisition start as the time zero reference.

`TRIGGER` specifies the trigger point as the time zero reference.

Examples

`SENSE:ANALYSIS:REFERENCE ACQSTART` specifies the acquisition start as the analysis time reference.

[SENSe]:ANALysis:STARt

Sets or queries the analysis offset time. Programming a specified offset time sets [SENSe]:ANALysis:STARt:AUTO OFF.

Conditions

Measurement views: All

Group

Sense commands

Syntax

```
[SENSe]:ANALysis:STARt <value>
[SENSe]:ANALysis:STARt?
```

Related Commands

[SENSe]:ANALysis:LENGth, [SENSe]:ANALysis:STARt:AUTO

Arguments

`<value>::=<NRf>` specifies the analysis offset time.
Range: 0 to [(acquisition length) - 200 ns].

If [(analysis start) + (analysis length)] > [(acquisition length) - 400 ns], the actual analysis length is reduced to [(acquisition length) - 200 ns].

Examples

`SENSE:ANALYSIS:START 23.5us` sets the analysis offset to 23.5 μs.

[SENSe]:ANALysis:STARt:AUTO

Determines whether to set the analysis offset automatically or manually.

Conditions

Measurement views: All

Group

Sense commands

Syntax

```
[SENSe]:ANALysis:STARt:AUTO { OFF | ON | 0 | 1 }
[SENSe]:ANALysis:STARt:AUTO?
```

Arguments OFF or 0 sets the analysis offset manually, using the [SENSe]:ANALysis:STARt command.

ON or 1 sets the analysis offset automatically.

Examples SENSE:ANALYSIS:START:AUTO ON sets the analysis offset automatically.

[SENSe]:AVTime:{BANDwidth|BWIDth}

Sets or queries the time-domain bandwidth filter in the Amplitude versus Time measurement. Programming a specified bandwidth disables the [SENSe]:AVTime:SPAN setting.

Conditions Measurement views: Amplitude versus Time

Group Sense commands

Syntax

```
[SENSe]:AVTime:{BANDwidth|BWIDth} <value>
[SENSe]:AVTime:{BANDwidth|BWIDth}?
```

Arguments <value>::=<NRf> specifies the filter bandwidth.
Range: 1 Hz to 20 MHz (Standard) / 60 MHz (Option 110).

Examples SENSE:AVTIME:BANDWITH 10MHz sets the filter bandwidth to 10 MHz.

[SENSe]:AVTime:{BANDwidth|BWIDth}:ACTual? (Query Only)

Queries the actual time-domain bandwidth in the Amplitude versus Time measurement.

Conditions Measurement views: Amplitude versus Time

Group Sense commands

Syntax

```
[SENSe]:AVTime:{BANDwidth|BWIDth}:ACTual?
```

Arguments None

Returns `<NRf>` The actual time-domain bandwidth in Hz.

Examples `SENSE:AVTIME:BANDWIDTH:ACTUAL?` might return `20E+6`, indicating that the actual time-domain bandwidth is 20 MHz.

[SENSe]:AVTime:CLEar:RESults (No Query Form)

Restarts multi-trace functions (Average and Max/Min Hold).

Conditions Measurement views: Amplitude versus Time

Group Sense commands

Syntax

```
[SENSe]:AVTime:CLEar:RESults
```

Arguments None

Examples `SENSE:AVTIME:CLEAR:RESULTS` restarts multi-trace functions.

[SENSe]:AVTime:MAXTracepoints

Selects or queries the maximum trace points in the Amplitude versus Time measurement.

Conditions Measurement views: Amplitude versus Time

Group Sense commands

Syntax

```
[SENSe]:AVTime:MAXTracepoints { ONEK | TENK | HUNDredk |
NEVerdecimate }
[SENSe]:AVTime:MAXTracepoints?
```

Arguments

`ONEK` sets the maximum trace points to 1 k.

`TENK` sets the maximum trace points to 10 k.

`HUNDredk` sets the maximum trace points to 100 k.

`NEVerdecimate` never decimates the trace points.

Examples `SENSE:AVTIME:MAXTRACEPOINTS TENK` sets the maximum trace points to 10 k.

[SENSe]:AVTime:METHod

Selects or queries the method to set the measurement bandwidth in the Amplitude versus Time measurement.

Conditions Measurement views: Amplitude versus Time

Group Sense commands

Syntax

```
[SENSe]:AVTime:METHod { SPAN | TDBW }
[SENSe]:AVTime:METHod?
```

Arguments `SPAN` specifies that the measurement bandwidth is set by the frequency span, using the [SENSe]:AVTime:SPAN command.

`TDBW` specifies that the measurement bandwidth is set by the time-domain bandwidth, using the [SENSe]:AVTime:{BANDwidth|BWIDth} command.

Examples `SENSE:AVTIME:METHOD SPAN` specifies that the measurement bandwidth is set by the frequency span.

[SENSe]:AVTime:SPAN

Sets or queries the frequency span in the Amplitude versus Time measurement. Programming a specified span disables the [SENSe]:AVTime:{BANDwidth|BWIDth} setting.

Conditions Measurement views: Amplitude versus Time

Group Sense commands

Syntax

```
[SENSe]:AVTime:SPAN <value>
[SENSe]:AVTime:SPAN?
```

Arguments `<value>::=<NRf>` specifies the frequency span.
Range: 10 Hz to 40 MHz (Standard) / 110 MHz (Option 110)

Examples SENSE:AVTIME:SPAN 5MHz sets the frequency span to 5 MHz.

[SENSe]:CCDF:{BANDwidth|BWIDth}

Sets or queries the CCDF measurement bandwidth (frequency span).

Conditions Measurement views: CCDF

Group Sense commands

Syntax
```
[SENSe]:CCDF:{BANDwidth|BWIDth} <value>
[SENSe]:CCDF:{BANDwidth|BWIDth}?
```

Arguments <value>::=<NRf> is the CCDF measurement bandwidth.
Range: 10 Hz to 40 MHz (Standard) / 60 MHz (Option 110).

Examples SENSE:CCDF:BANDWIDTH 1MHz sets the CCDF measurement bandwidth to 1 MHz.

[SENSe]:CCDF:CLEar (No Query Form)

Clears the CCDF accumulator and restarts the measurement.

Conditions Measurement views: CCDF

Group Sense commands

Syntax
```
[SENSe]:CCDF:CLEar
```

Arguments None

Examples SENSE:CCDF:CLEAR clears the CCDF accumulator and restarts the measurement.

[SENSe]:CCDF:TIME:TOTal:LENGth

Sets or queries the CCDF measurement time when [SENSe]:CCDF:TIME:TYPE is set to TOTal.

Conditions Measurement views: CCDF

Group Sense commands

Syntax

```
[SENSe]:CCDF:TIME:TOTal:LENGth <value>
[SENSe]:CCDF:TIME:TOTal:LENGth?
```

Arguments `<value>::=<NRf>` specifies the CCDF measurement time.
Range: 20 ms to 100 s.

Examples `SENSE:CCDF:TIME:TOTAL:LENGTH 10` sets the CCDF measurement time to 10 s.

[SENSe]:CCDF:TIME:TYPE

Determines how to repeat the CCDF measurement.

Conditions Measurement views: CCDF

Group Sense commands

Syntax

```
[SENSe]:CCDF:TIME:TYPE { SINGLe | TOTal | CONTinuous }
[SENSe]:CCDF:TIME:TYPE?
```

Related Commands INITiate commands

Arguments `SINGLe` specifies that the analyzer sets the analysis length to 1 ms and then acquire data once to calculate CCDF.

`TOTal` specifies that the analyzer sets the analysis length to 20 ms and then repeats data acquisition and CCDF calculation for the time specified by the [SENSe]:CCDF:TIME:TOTal:LENGth command.

`CONTinuous` specifies that the analyzer sets the analysis length to 1 ms and then repeats data acquisition and CCDF calculation continuously. To reset the process, use the [SENSe]:CCDF:CLEar command or the INITiate commands.

Examples `SENSE:CCDF:TIME:TYPE SINGLe` specifies that the analyzer sets the analysis length to 1 ms and then acquire data once to calculate CCDF.

[SENSe]:DDEMod:ANALysis:LENGth

Sets or queries the analysis length. Programming a specified length sets [SENSe]:DDEMod:ANALysis:AUTO OFF.

Conditions Measurement views: General purpose digital modulation

Group Sense commands

Syntax

```
[SENSe]:DDEMod:ANALysis:LENGth <value>
[SENSe]:DDEMod:ANALysis:LENGth?
```

Related Commands [SENSe]:DDEMod:ANALysis:LENGth:AUTO

Arguments `<value>::=<NRf>` specifies the analysis length.
Range: 200ns to [(acquisition length) - 400 ns].

If [(analysis start) + (analysis length)] > [(acquisition length) - 400 ns], the actual analysis length is reduced to [(acquisition length) - 200 ns].

Examples `SENSE:DDEMOD:ANALYSIS:LENGTH 25.625us` sets the analysis length to 25.625 µs.

[SENSe]:DDEMod:ANALysis:LENGth:ACTual? (Query Only)

Queries the actual analysis length.

Conditions Measurement views: General purpose digital modulation

Group Sense commands

Syntax

```
[SENSe]:DDEMod:ANALysis:LENGth:ACTual?
```

Arguments None

Returns `<NRf>` Actual analysis length in seconds.

Examples `SENSE:DDEMOD:ANALYSIS:LENGTH:ACTUAL?` might return `25.625E-6`, indicating that the actual analysis length is 25.625 µs.

[SENSe]:DDEMod:ANALysis:LENGth:AUTO

Determines whether to set the analysis length automatically or manually.

Conditions Measurement views: General purpose digital modulation

Group Sense commands

Syntax

```
[SENSe]:DDEMod:ANALysis:LENGth:AUTO { OFF | ON | 0 | 1 }
[SENSe]:DDEMod:ANALysis:LENGth:AUTO?
```

Arguments `OFF` or `0` sets the analysis length manually, using the [SENSe]:DDEMod:ANALysis:LENGth command.

`ON` or `1` sets the analysis length automatically.

Examples `SENSE:DDEMOD:ANALYSIS:LENGTH:AUTO ON` sets the analysis length automatically.

[SENSe]:DDEMod:BURSt:DETect

Determines how to detect bursts.

Conditions Measurement views: General purpose digital modulation

Group Sense commands

Syntax

```
[SENSe]:DDEMod:BURSt:DETect { ON | OFF }
[SENSe]:DDEMod:BURSt:DETect?
```

Related Commands [SENSe]:DDEMod:BURSt:THReshold

Arguments `ON` analyzes just that burst period if a burst is found. If a burst is not found, does not analyze but displays an error message.

`OFF` analyzes the whole analysis length.

***NOTE.** When selecting On and if the signal is not adequate for the demodulation, the measurement will fail and show an error message.*

Examples `SENSE:DDEMOD:BURST:DETECT OFF` analyzes the whole analysis length.

[SENSe]:DDEMod:BURSt:THReshold

Sets or queries the threshold level above which the input signal is determined to be a burst.

Conditions Measurement views: General purpose digital modulation

Group Sense commands

Syntax
```
[SENSe]:DDEMod:BURSt:THReshold <value>
[SENSe]:DDEMod:BURSt:THReshold?
```

Related Commands [SENSe]:DDEMod:BURSt:DETect

Arguments `<value>::=<NRf>` specifies the threshold level for detecting bursts.
Range: -100 to -10 dBc.

Examples `SENSE:DDEMOD:BURST:THRESHOLD -25` sets the threshold level to -25 dBc.

[SENSe]:DDEMod:CARRier:OFFSet

Sets or queries the carrier frequency offset in the digital modulation analysis.

Conditions Measurement views: General purpose digital modulation

Group Sense commands

Syntax
```
[SENSe]:DDEMod:CARRier:OFFSet <value>
[SENSe]:DDEMod:CARRier:OFFSet?
```

Arguments `<value>::=<NRf>` specifies the carrier frequency offset.
Range: -50 kHz to +50 kHz.

Examples `SENSE:DDEMOD:CARRIER:OFFSET 2kHz` sets the carrier frequency offset to 2 kHz.

[SENSe]:DDEMod:FILTer:ALPHa

Sets or queries the filter factor (α/BT) in the digital modulation analysis.

Conditions Measurement views: General purpose digital modulation

Group Sense commands

Syntax

```
[SENSe]:DDEMod:FILTer:ALPHa <value>
[SENSe]:DDEMod:FILTer:ALPHa?
```

Arguments `<value>::=<NRf>` specifies the filter factor. Range: 0.001 to 1.

Examples `SENSE:DDEMOD:FILTER:ALPHA 0.5` sets the filter factor to 0.5.

[SENSe]:DDEMod:FILTer:MEASurement

Selects or queries the measurement filter in the digital modulation analysis.

Conditions Measurement views: General purpose digital modulation

Group Sense commands

Syntax

```
[SENSe]:DDEMod:FILTer:MEASurement { OFF | RRCosine | RCOSine
| GAUSsian | RECTangular | IS95TXEQ_MEA | IS95TX_MEA |
IS95REF }
[SENSe]:DDEMod:FILTer:MEASurement?
```

Arguments The following table lists the arguments.

Table 2-35: Digital modulation measurement filter

Argument	Measurement fllter
OFF	No filter
RRCosine	Root Raised Cosine
RCOSine	Raised Cosine

Table 2-35: Digital modulation measurement filter (cont.)

Argument	Measurement fIlter
GAUSsian	Gaussian
RECTangular	Rectangular
IS95TXEQ_MEA	IS95 receive filter for the transmitter configured with both the transmit filter and the phase equalizer.
IS95TX_MEA	IS95 receive filter for the transmitter configured with only the transmit filter.
IS95REF	IS95 reference filter including the response of the transmit filter and phase equalizer as well as the receive (complementary) filter.

Examples `SENSE:DDEMOD:FILTER:MEASUREMENT RRCosine` selects the Root Raised Cosine filter as the measurement filter.

[SENSe]:DDEMod:FILTer:REFerence

Selects or queries the reference filter in the digital modulation analysis.

Conditions Measurement views: General purpose digital modulation

Group Sense commands

Syntax

```
[SENSe]:DDEMod:FILTer:REFerence { OFF | RRCosine | RCOSine |
GAUSsian | RECTangular | IS95TXEQ_MEA | IS95TX_MEA | IS95REF
}
[SENSe]:DDEMod:FILTer:REFerence?
```

Arguments (See Table 2-35.)

Examples `SENSE:DDEMOD:FILTER:REFERENCE RCOSine` selects the Raised Cosine filter as the reference filter.

[SENSe]:DDEMod:MAGNitude:NORMalize

Selects or queries the method for the magnitude normalization.

Conditions Measurement views: General purpose digital modulation

Group Sense commands

Syntax

```
[SENSe]:DDEMod:MAGNitude:NORMalize { RSYMbol | MSYMbol }
[SENSe]:DDEMod:MAGNitude:NORMalize?
```

Arguments `RSYMbol` normalizes the magnitude with the RMS symbol magnitude.

`MSYMbol` normalizes the magnitude with the maximum symbol magnitude.

Examples `SENSE:DDEMOD:MAGNITUDE:NORMALIZE RSYMbol` normalizes the magnitude with the RMS symbol magnitude.

[SENSe]:DDEMod:MODulation:TYPE

Selects or queries the modulation type in the digital modulation analysis.

Conditions Measurement views: General purpose digital modulation

Group Sense commands

Syntax

```
[SENSe]:DDEMod:MODulation:TYPE { QPSK | PSK8 | D8PSK | DQPSK
| PIOVER4DQPSK | BPSK | QAM16 | QAM32 | QAM64 | QAM128 |
QAM256 | MSK }
[SENSe]:DDEMod:MODulation:TYPE?
```

Arguments The following table lists the arguments and corresponding modulation type.

Table 2-36: Modulation type

Argument	Modulation type
QPSK	QPSK
PSK8	8PSK
D8PSK	D8PSK
DQPSK	DQPSK
PIOVER4DQPSK	π/4 QPSK
BPSK	BPSK
QAM16	16QAM
QAM32	32QAM
QAM64	64QAM
QAM128	128QAM

Table 2-36: Modulation type (cont.)

Argument	Modulation type
QAM256	256QAM
MSK	MSK

Examples `SENSE:DDEMOD:MODULATION:TYPE QPSK` selects QPSK modulation system.

[SENSe]:DDEMod:SRATe

Sets or queries the symbol rate in the digital modulation analysis.

Conditions Measurement views: General purpose digital modulation

Group Sense commands

Syntax

```
[SENSe]:DDEMod:SRATe <value>
[SENSe]:DDEMod:SRATe?
```

Arguments `<value>::=<NRf>` specifies the symbol rate. Range: 100 Hz to 122.9 MHz.

Examples `SENSE:DDEMOD:SRATE 21.0E3` sets the symbol rate to 21 kHz.

[SENSe]:DDEMod:SWAP:IQ

Determines whether or not to exchange I and Q data before demodulating.

Conditions Measurement views: General purpose digital modulation

Group Sense commands

Syntax

```
[SENSe]:DDEMod:SWAP:IQ { OFF | ON | 0 | 1 }
[SENSe]:DDEMod:SWAP:IQ?
```

Arguments `OFF` or `0` uses I and Q data as they are.

`ON` or `1` exchanges I and Q data.

Examples `SENSE:DDEMOD:SWAP:IQ ON` exchanges I and Q data before demodulating.

[SENSe]:DDEMod:SYMBol:POINts

Selects or queries the number of points per symbol (how many points to use between symbols when connecting the dots).

NOTE. *1 is not valid for the GMSK modulation.*

In the constellation view, select VECTors using the TRACe:CONSte:MODE command first to change Points/Symbol.

Conditions Measurement views: Constellation, EVM versus Time, Magnitude error versus Time, Phase error versus Time

Group Sense commands

Syntax

```
[SENSe]:DDEMod:SYMBol:POINts { ONE | TWO | FOUR | EIGHt }
[SENSe]:DDEMod:SYMBol:POINts?
```

Arguments `ONE`, `TWO`, `FOUR`, and `EIGHt` represent the number of points per symbol.

Examples `SENSE:DDEMOD:SYMBOL:POINTS FOUR` sets the number of points per symbol to four.

[SENSe]:DDEMod:SYNCh:WORD

Determines whether to enable or disable the synchronization word.

Conditions Measurement views: General purpose digital modulation

Group Sense commands

Syntax

```
[SENSe]:DDEMod:SYNCh:WORD { OFF | ON | 0 | 1 }
[SENSe]:DDEMod:SYNCh:WORD?
```

Related Commands [SENSe]:DDEMod:SYNCh:WORD:SYMBol

Arguments `OFF` or `0` disables the synchronization word.

`ON` or `1` enables the synchronization word.

Examples `SENSE:DDEMOD:SYNCH:WORD ON` enables the synchronization word.

[SENSe]:DDEMod:SYNCh:WORD:SYMBol

Sets or queries the synchronization word when [SENSe]:DDEMod:SYNCh:WORD is ON. The word depends on the modulation type selected by the [SENSe]:DDEMod:MODulation:TYPE command.

Conditions Measurement views: General purpose digital modulation

Group Sense commands

Syntax
```
[SENSe]:DDEMod:SYNCh:WORD:SYMBol <block>
[SENSe]:DDEMod:SYNCh:WORD:SYMBol?
```

Arguments `<block>::=#<num_digit><num_byte><sym(1)><sym(2)>...<sym(n)>`

Where
`<num_digit>` is the number of digits in <num_byte>.
`<num_byte>` is the number of bytes of data that follow.
`<sym(n)>` is the n^{th} symbol value of the sync word. 32-bit integer.
n: Max 256.

Examples `SENSE:DDEMOD:SYNCH:WORD:SYMBOL #216xxxx` (4 symbols) sets a sync word composed of four symbols.

[SENSe]:DDEMod:TIMe:UNITs

Selects or queries the fundamental unit of time.

Conditions Measurement views: General purpose digital modulation

Group Sense commands

Syntax
```
[SENSe]:DDEMod:TIMe:UNITs { SEConds | SYMBols }
[SENSe]:DDEMod:TIMe:UNITs?
```

Arguments

`SEConds` specifies the fundamental unit of time as seconds.

`SYMBols` specifies the fundamental unit of time as symbols.

Examples

`SENSE:DDEMOD:TIME:UNITS SEConds` specifies the fundamental unit of time as seconds.

[SENSe]:DPSA:AUDio:DEMod:GAIN

Sets or queries the audio gain.

***NOTE.** The sound level is also affected by the Windows volume control.*

Conditions

Measurement views: DPX spectrum

Group

Sense commands

Syntax

```
[SENSe]:DPSA:AUDio:DEMod:GAIN <value>
[SENSe]:DPSA:AUDio:DEMod:GAIN?
```

Arguments

`<value>::=<NR1>` specifies the audio gain. Range: 0 to 15 (integer).

Examples

`SENSE:DPSA:AUDIO:DEMOD:GAIN 7` sets the audio gain to 7.

[SENSe]:DPSA:AUDio:DEMod:RXBWidth

Sets or queries the receiver bandwidth in the audio demodulation.

Conditions

Measurement views: DPX spectrum

Group

Sense commands

Syntax

```
[SENSe]:DPSA:AUDio:DEMod:RXBWidth <value>
[SENSe]:DPSA:AUDio:DEMod:RXBWidth?
```

Arguments

`<value>::=<NRf>` specifies the receiver bandwidth in the audio demodulation. Range: 1 kHz to 500 kHz.

Examples SENSE:DPSA:AUDIO:DEMOD:RXBWIDTH 30kHz sets the receiver bandwidth to 30 kHz.

[SENSe]:DPSA:AUDio:DEMod:RXFRequency? (Query Only)

Returns the receiver frequency in the audio demodulation. The frequency depends on the setting of the [SENSe]:DPSA:AUDio:DEMod:TUNE command.

Conditions Measurement views: DPX spectrum

Group Sense commands

Syntax `[SENSe]:DPSA:AUDio:DEMod:RXFRequency?`

Arguments None

Returns <frequency>::=<NRf> is the receiver frequency in the audio demodulation.

Examples SENSE:DPSA:AUDIO:DEMOD:RXFREQUENCY? might return 80.3E+6, indicating that the receiver frequency is 80.3 MHz.

[SENSe]:DPSA:AUDio:DEMod:STATe

Determines whether to enable or disable the audio demodulation.

Conditions Measurement views: DPX spectrum

Group Sense commands

Syntax
```
[SENSe]:DPSA:AUDio:DEMod:STATe { OFF | ON | 0 | 1 }
[SENSe]:DPSA:AUDio:DEMod:STATe?
```

Arguments OFF or 0 disables the audio demodulation.

ON or 1 enables the audio demodulation.

Examples SENSE:DPSA:AUDIO:DEMOD:STATE ON enables the audio demodulation.

[SENSe]:DPSA:AUDio:DEMod:TUNE

Selects or queries how to determine the tuning frequency in the audio demodulation.

Conditions Measurement views: DPX spectrum

Group Sense commands

Syntax

```
[SENSe]:DPSA:AUDio:DEMod:TUNE { MR | MARK1 | MARK2 | MARK3|
MARK4 | SMARker | FREQcontrol }
[SENSe]:DPSA:AUDio:DEMod:TUNE?
```

Arguments The following table lists the arguments.

Table 2-37: Frequency tuning

Argument	Tune with
MR	Reference marker (MR)
MARK1	Marker 1 (M1)
MARK2	Marker 2 (M2)
MARK3	Marker 3 (M3)
MARK4	Marker 4 (M4)
SMARker	Selected marker
FREQcontrol	Center frequency setting

Examples `SENSE:DPSA:AUDIO:DEMOD:TUNE MARK1` sets the tuning frequency to the value at Marker 1.

[SENSe]:DPSA:AUDio:DEMod:TYPE

Selects or queries the modulation type in the audio demodulation.

Conditions Measurement views: DPX spectrum

Group Sense commands

Syntax

```
[SENSe]:DPSA:AUDio:DEMod:TYPE { AM | FM }
[SENSe]:DPSA:AUDio:DEMod:TYPE?
```

Arguments

AM selects the AM (Amplitude Modulation).

FM selects the FM (Frequency Modulation).

Examples

SENSE:DPSA:AUDIO:DEMOD:TYPE FM selects FM in the audio demodulation.

[SENSe]:DPSA:{BANDwidth|BWIDth}[:RESolution]

Sets or queries the resolution bandwidth (RBW) in the DPX spectrum measurement. Programming a specified RBW sets [SENSe]:DPSA:BANDwidth |BWIDth[:RESolution]:AUTO OFF.

Conditions

Measurement views: DPX spectrum

Group

Sense commands

Syntax

```
[SENSe]:DPSA:{BANDwidth|BWIDth}[:RESolution] <value>
[SENSe]:DPSA:{BANDwidth|BWIDth}[:RESolution]?
```

Related Commands

[SENSe]:DPSA:{BANDwidth|BWIDth}[:RESolution]:AUTO

Arguments

<value>::=<NRf> is the resolution bandwidth. Range: 1 Hz to 5 MHz.

Examples

SENSE:DPSA:BANDWIDTH:RESOLUTION 200kHz sets the resolution bandwidth to 200 kHz.

[SENSe]:DPSA:{BANDwidth|BWIDth}[:RESolution]:AUTO

Determines whether to set the resolution bandwidth (RBW) automatically or manually in the DPX spectrum measurement.

Conditions

Measurement views: DPX spectrum

Group

Sense commands

Syntax

```
[SENSe]:DPSA:{BANDwidth|BWIDth}[:RESolution]:AUTO { OFF |
ON | 0 | 1 }
[SENSe]:DPSA:{BANDwidth|BWIDth}[:RESolution]:AUTO?
```

Arguments

OFF or 0 specifies that the resolution bandwidth is set manually using the [SENSe]:DPSA:{BANDwidth|BWIDth}[:RESolution] command.

ON or 1 specifies that the resolution bandwidth is set automatically.

Examples

SENSE:DPSA:BANDWIDTH:AUTO ON sets the resolution bandwidth automatically.

[SENSe]:DPSA:CLEar:RESults (No Query Form)

Restarts multi-trace functions (Average and Max/Min Hold).

Conditions

Measurement views: DPX spectrum

Group

Sense commands

Syntax

```
[SENSe]:DPSA:CLEar:RESults
```

Arguments

None

Examples

SENSE:DPSA:CLEAR:RESULTS restarts multi-trace functions.

[SENSe]:DPSA:COLor

Selects or queries the color palette of three-dimensional graphs.

Conditions

Measurement views: DPX spectrum

Group

Sense commands

Syntax

```
[SENSe]:DPSA:COLor { RED | GREen | BLUe | CYAN | BCYan |
YELLow | MAGenta | GRAY | TEMPerature | SPECtral }
[SENSe]:DPSA:COLor?
```

Arguments The following table lists the arguments.

Table 2-38: Color palette for DPX spectrum

Argument	Palette
RED	Red
GREEn	Green
BLUe	Blue
CYAN	Cyan
BCYan	Binary cyan
YELLow	Yellow
MAGenta	Magenta
GRAY	Gray
TEMPerature	Temperature
SPECtral	Spectral

Examples `SENSE:DPSA:COLOR TEMPerature` selects the temperature color palette.

[SENSe]:DPSA:COLor:MAXimum

Sets or queries the maximum value of the color axis in the DPX spectrum measurement.

Conditions Measurement views: DPX spectrum

Group Sense commands

Syntax

```
[SENSe]:DPSA:COLor:MAXimum <value>
[SENSe]:DPSA:COLor:MAXimum?
```

Arguments `<value>::=<NRf>` specifies the maximum value of the color axis.
Range: The minimum value to 100%.

The minimum value is set using the [SENSe]:DPSA:COLor:MINimum command.

Examples `SENSE:DPSA:COLOR:MAXIMUM 90` sets the maximum value of the color axis to 90%.

[SENSe]:DPSA:COLor:MINimum

Sets or queries the minimum value of the color axis in the DPX spectrum measurement.

Conditions Measurement views: DPX spectrum

Group Sense commands

Syntax

```
[SENSe]:DPSA:COLor:MINimum <value>
[SENSe]:DPSA:COLor:MINimum?
```

Arguments `<value>::=<NRf>` specifies the minimum value of the color axis.
Range: 0% to the maximum value.

The maximum value is set using the [SENSe]:DPSA:COLor:MAXimum command.

Examples `SENSE:DPSA:COLOR:MINIMUM 10` sets the minimum value of the color axis to 10%.

[SENSe]:DPSA:FREQuency:CENTer

Sets or queries the center frequency in the DPX spectrum measurement.

NOTE. *The center, start and stop frequencies are set interlocking each other with the following relationships: (start frequency) = (center frequency) - (span)/2 and (stop frequency) = (center frequency) + (span)/2.*

Conditions Measurement views: DPX spectrum

Group Sense commands

Syntax

```
[SENSe]:DPSA:FREQuency:CENTer <value>
[SENSe]:DPSA:FREQuency:CENTer?
```

Related Commands [SENSe]:DPSA:FREQuency:STARt, [SENSe]:DPSA:FREQuency:STOP

Arguments

<value>::=<NRf> specifies the center frequency.
Range: 0 Hz to 6.2 GHz (RSA6106A) / 14 GHz (RSA6114A).

Examples

`SENSE:DPSA:FREQUENCY:CENTER 7.5GHz` sets the center frequency to 7.5 GHz.

[SENSe]:DPSA:FREQuency:SPAN

Sets or queries the frequency span in the DPX spectrum measurement.

Conditions

Measurement views: DPX spectrum

Group

Sense commands

Syntax

```
[SENSe]:DPSA:FREQuency:SPAN <value>
[SENSe]:DPSA:FREQuency:SPAN?
```

Arguments

<value>::=<NRf> is the frequency span.
Range: 10 Hz to 40 MHz (Standard) / 110 MHz (Option 110)

Examples

`SENSE:DPSA:FREQUENCY:SPAN 20MHz` sets the span to 20 MHz.

[SENSe]:DPSA:FREQuency:STARt

Sets or queries the measurement start frequency (left edge on the graph) in the DPX spectrum measurement.

The center, start and stop frequencies are set interlocking each other. Refer to the [SENSe]:DPSA:FREQuency:CENTer command.

Conditions

Measurement views: DPX spectrum

Group

Sense commands

Syntax

```
[SENSe]:DPSA:FREQuency:STARt <value>
[SENSe]:DPSA:FREQuency:STARt?
```

Related Commands

[SENSe]:DPSA:FREQuency:STOP

Arguments

<value>::=<NRf> is the measurement start frequency.
Range: (center frequency) ± (span)/2.

Examples

`SENSE:DPSA:FREQUENCY:START 6.95GHz` sets the start frequency to 6.95 GHz.

[SENSe]:DPSA:FREQuency:STEP

Sets or queries the frequency step size (the amount per press by which the up or down key changes the setting value). Programming a specified step size sets [SENSe]:DPSA:FREQuency:STEP:AUTO OFF.

Conditions

Measurement views: DPX spectrum

Group

Sense commands

Syntax

```
[SENSe]:DPSA:FREQuency:STEP <value>
[SENSe]:DPSA:FREQuency:STEP?
```

Related Commands

[SENSe]:DPSA:FREQuency:STEP:AUTO

Arguments

<value>::=<NRf> specifies the frequency step size.
Range: 0 Hz to 6.2 GHz (RSA6106A) / 14 GHz (RSA6114A).

Examples

`SENSE:DPSA:FREQUENCY:STEP 1.5kHz` sets the step size to 1.5 kHz.

[SENSe]:DPSA:FREQuency:STEP:AUTO

Determines whether to set the frequency step size automatically or manually.

Conditions

Measurement views: DPX spectrum

Group

Sense commands

Syntax

```
[SENSe]:DPSA:FREQuency:STEP:AUTO { OFF | ON | 0 | 1 }
[SENSe]:DPSA:FREQuency:STEP:AUTO?
```

Arguments `OFF` or `0` specifies that the frequency step size is set manually using the [SENSe]:DPSA:FREQuency:STEP command.

`ON` or `1` specifies that the frequency step size is set automatically.

Examples `SENSE:DPSA:BANDWIDTH:AUTO ON` sets the frequency step size automatically.

[SENSe]:DPSA:FREQuency:STOP

Sets or queries the measurement stop frequency (right edge of the graph) in the DPX spectrum measurement.

The center, start and stop frequencies are set interlocking each other. Refer to the [SENSe]:DPSA:FREQuency:CENTer command.

Conditions Measurement views: DPX spectrum

Group Sense commands

Syntax

```
[SENSe]:DPSA:FREQuency:STOP <value>
[SENSe]:DPSA:FREQuency:STOP?
```

Related Commands [SENSe]:DPSA:FREQuency:STARt

Arguments `<value>::=<NRf>` is the measurement stop frequency.
Range: (center frequency) ± (span)/2.

Examples `SENSE:DPSA:FREQUENCY:STOP 7.05GHz` sets the stop frequency to 7.05 GHz.

[SENSe]:FVTime:CLEar:RESults (No Query Form)

Restarts multi-trace functions (Average and Max/Min Hold).

Conditions Measurement views: Frequency versus Time

Group Sense commands

Syntax

```
[SENSe]:FVTime:CLEar:RESults
```

Arguments None

Examples `SENSE:FVTIME:CLEAR:RESULTS` restarts multi-trace functions.

[SENSe]:FVTime:FREQuency:CENTer

Sets or queries the center frequency in the Frequency versus Time measurement.

***NOTE.** The center, start and stop frequencies are set interlocking each other with the following relationships: (start frequency) = (center frequency) - (span)/2 and (stop frequency) = (center frequency) + (span)/2.*

Conditions Measurement views: Frequency versus Time

Group Sense commands

Syntax

```
[SENSe]:FVTime:FREQuency:CENTer <value>
[SENSe]:FVTime:FREQuency:CENTer?
```

Related Commands [SENSe]:FVTime:FREQuency:STARt, [SENSe]:FVTime:FREQuency:STOP

Arguments `<value>::=<NRf>` specifies the center frequency.
Range: 0 Hz to 6.2 GHz (RSA6106A) / 14 GHz (RSA6114A).

Examples `SENSE:FVTIME:FREQUENCY:CENTER 7.5GHz` sets the center frequency to 7.5 GHz.

[SENSe]:FVTime:FREQuency:SPAN

Sets or queries the frequency span in the Frequency versus Time measurement.

Conditions Measurement views: Frequency versus Time

Group Sense commands

Syntax

```
[SENSe]:FVTime:FREQuency:SPAN <value>
[SENSe]:FVTime:FREQuency:SPAN?
```

Arguments `<value>::=<NRf>` is the frequency span.
Range: 10 Hz to 40 MHz (Standard) / 110 MHz (Option 110)

Examples `SENSE:FVTIME:FREQUENCY:SPAN 20MHz` sets the span to 20 MHz.

[SENSe]:FVTime:FREQuency:STARt

Sets or queries the measurement start frequency (left edge on the graph) in the Frequency versus Time measurement.

The center, start and stop frequencies are set interlocking each other. Refer to the [SENSe]:FVTime:FREQuency:CENTer command.

Conditions Measurement views: Frequency versus Time

Group Sense commands

Syntax
```
[SENSe]:FVTime:FREQuency:STARt <value>
[SENSe]:FVTime:FREQuency:STARt?
```

Related Commands [SENSe]:FVTime:FREQuency:STOP

Arguments `<value>::=<NRf>` is the measurement start frequency.
Range: (center frequency) ± (span)/2.

Examples `SENSE:FVTIME:FREQUENCY:START 6.95GHz` sets the start frequency to 6.95 GHz.

[SENSe]:FVTime:FREQuency:STEP

Sets or queries the frequency step size (the amount per press by which the up or down key changes the setting value). Programming a specified step size sets [SENSe]:FVTime:FREQuency:STEP:AUTO OFF.

Conditions Measurement views: Frequency versus Time

Group Sense commands

Syntax

```
[SENSe]:FVTime:FREQuency:STEP <value>
[SENSe]:FVTime:FREQuency:STEP?
```

Related Commands [SENSe]:FVTime:FREQuency:STEP:AUTO

Arguments `<value>::=<NRf>` specifies the frequency step size.
Range: 0 Hz to 6.2 GHz (RSA6106A) / 14 GHz (RSA6114A).

Examples `SENSE:FVTIME:FREQUENCY:STEP 1.5kHz` sets the step size to 1.5 kHz.

[SENSe]:FVTime:FREQuency:STEP:AUTO

Determines whether to set the frequency step size automatically or manually.

Conditions Measurement views: Frequency versus Time

Group Sense commands

Syntax

```
[SENSe]:FVTime:FREQuency:STEP:AUTO { OFF | ON | 0 | 1 }
[SENSe]:FVTime:FREQuency:STEP:AUTO?
```

Arguments `OFF` or `0` specifies that the frequency step size is set manually using the [SENSe]:FVTime:FREQuency:STEP command.

`ON` or `1` specifies that the frequency step size is set automatically.

Examples `SENSE:FVTIME:BANDWIDTH:AUTO ON` sets the frequency step size automatically.

[SENSe]:FVTime:FREQuency:STOP

Sets or queries the measurement stop frequency (right edge of the graph) in the Frequency versus Time measurement.

The center, start and stop frequencies are set interlocking each other. Refer to the [SENSe]:FVTime:FREQuency:CENTer command.

Conditions Measurement views: Frequency versus Time

Group Sense commands

Syntax
```
[SENSe]:FVTime:FREQuency:STOP <value>
[SENSe]:FVTime:FREQuency:STOP?
```

Related Commands [SENSe]:FVTime:FREQuency:STARt

Arguments <value>::=<NRf> is the measurement stop frequency.
Range: (center frequency) ± (span)/2.

Examples SENSE:FVTIME:FREQUENCY:STOP 7.05GHz sets the stop frequency to 7.05 GHz.

[SENSe]:FVTime:MAXTracepoints

Selects or queries the maximum trace points in the Frequency versus Time measurement.

Conditions Measurement views: Frequency versus Time

Group Sense commands

Syntax
```
[SENSe]:FVTime:MAXTracepoints { ONEK | TENK | HUNDredk |
NEVerdecimate }
[SENSe]:FVTime:MAXTracepoints?
```

Arguments ONEK sets the maximum trace points to 1 k.

TENK sets the maximum trace points to 10 k.

HUNDredk sets the maximum trace points to 100 k.

NEVerdecimate never decimates the trace points.

Examples SENSE:FVTIME:MAXTRACEPOINTS TENK sets the maximum trace points to 10 k.

[SENSe]:IQVTime:CLEar:RESults (No Query Form)

Restarts multi-trace functions (Average and Max/Min Hold).

Conditions Measurement views: RF I&Q versus Time

Group Sense commands

Syntax `[SENSe]:IQVTime:CLEar:RESults`

Arguments None

Examples `SENSE:IQVTIME:CLEAR:RESULTS` restarts multi-trace functions.

[SENSe]:IQVTime:FREQuency:CENTer

Sets or queries the center frequency in the RF I&Q versus Time measurement.

***NOTE.** The center, start and stop frequencies are set interlocking each other with the following relationships: (start frequency) = (center frequency) - (span)/2 and (stop frequency) = (center frequency) + (span)/2.*

Conditions Measurement views: RF I&Q versus Time

Group Sense commands

Syntax

```
[SENSe]:IQVTime:FREQuency:CENTer <value>
[SENSe]:IQVTime:FREQuency:CENTer?
```

Related Commands [SENSe]:IQVTime:FREQuency:STARt, [SENSe]:IQVTime:FREQuency:STOP

Arguments `<value>::=<NRf>` specifies the center frequency.
Range: 0 Hz to 6.2 GHz (RSA6106A) / 14 GHz (RSA6114A).

Examples `SENSE:IQVTIME:FREQUENCY:CENTER 7.5GHz` sets the center frequency to 7.5 GHz.

[SENSe]:IQVTime:FREQuency:SPAN

Sets or queries the frequency span in the RF I&Q versus Time measurement.

Conditions Measurement views: RF I&Q versus Time

Group Sense commands

Syntax
```
[SENSe]:IQVTime:FREQuency:SPAN <value>
[SENSe]:IQVTime:FREQuency:SPAN?
```

Arguments `<value>::=<NRf>` is the frequency span.
Range: 10 Hz to 40 MHz (Standard) / 110 MHz (Option 110)

Examples `SENSE:IQVTIME:FREQUENCY:SPAN 20MHz` sets the span to 20 MHz.

[SENSe]:IQVTime:FREQuency:STARt

Sets or queries the measurement start frequency (left edge on the graph) in the RF I&Q versus Time measurement.

The center, start and stop frequencies are set interlocking each other. Refer to the [SENSe]:IQVTime:FREQuency:CENTer command.

Conditions Measurement views: RF I&Q versus Time

Group Sense commands

Syntax
```
[SENSe]:IQVTime:FREQuency:STARt <value>
[SENSe]:IQVTime:FREQuency:STARt?
```

Related Commands [SENSe]:IQVTime:FREQuency:STOP

Arguments `<value>::=<NRf>` is the measurement start frequency.
Range: (center frequency) ± (span)/2.

Examples `SENSE:IQVTIME:FREQUENCY:START 6.95GHz` sets the start frequency to 6.95 GHz.

[SENSe]:IQVTime:FREQuency:STEP

Sets or queries the frequency step size (the amount per press by which the up or down key changes the setting value). Programming a specified step size sets [SENSe]:IQVTime:FREQuency:STEP:AUTO OFF.

Conditions Measurement views: RF I&Q versus Time

Group Sense commands

Syntax

```
[SENSe]:IQVTime:FREQuency:STEP <value>
[SENSe]:IQVTime:FREQuency:STEP?
```

Related Commands [SENSe]:IQVTime:FREQuency:STEP:AUTO

Arguments `<value>::=<NRf>` specifies the frequency step size.
Range: 0 Hz to 6.2 GHz (RSA6106A) / 14 GHz (RSA6114A).

Examples `SENSE:IQVTIME:FREQUENCY:STEP 1.5kHz` sets the step size to 1.5 kHz.

[SENSe]:IQVTime:FREQuency:STEP:AUTO

Determines whether to set the frequency step size automatically or manually.

Conditions Measurement views: RF I&Q versus Time

Group Sense commands

Syntax

```
[SENSe]:IQVTime:FREQuency:STEP:AUTO { OFF | ON | 0 | 1 }
[SENSe]:IQVTime:FREQuency:STEP:AUTO?
```

Arguments `OFF` or `0` specifies that the frequency step size is set manually using the [SENSe]:IQVTime:FREQuency:STEP command.

`ON` or `1` specifies that the frequency step size is set automatically.

Examples `SENSE:IQVTIME:FREQUENCY:STEP:AUTO ON` sets the frequency step size automatically.

[SENSe]:IQVTime:FREQuency:STOP

Sets or queries the measurement stop frequency (right edge of the graph) in the RF I&Q versus Time measurement.

The center, start and stop frequencies are set interlocking each other. Refer to the [SENSe]:IQVTime:FREQuency:CENTer command.

Conditions Measurement views: RF I&Q versus Time

Group Sense commands

Syntax

```
[SENSe]:IQVTime:FREQuency:STOP <value>
[SENSe]:IQVTime:FREQuency:STOP?
```

Related Commands [SENSe]:IQVTime:FREQuency:STARt

Arguments `<value>::=<NRf>` is the measurement stop frequency.
Range: (center frequency) ± (span)/2.

Examples `SENSE:IQVTIME:FREQUENCY:STOP 7.05GHz` sets the stop frequency to 7.05 GHz.

[SENSe]:IQVTime:MAXTracepoints

Selects or queries the maximum trace points in the RF I&Q versus Time measurement.

Conditions Measurement views: RF I&Q versus Time

Group Sense commands

Syntax

```
[SENSe]:IQVTime:MAXTracepoints { ONEK | TENK | HUNDredk |
NEVerdecimate }
[SENSe]:IQVTime:MAXTracepoints?
```

Arguments `ONEK` sets the maximum trace points to 1 k.

`TENK` sets the maximum trace points to 10 k.

`HUNDredk` sets the maximum trace points to 100 k.

`NEVerdecimate` never decimates the trace points.

Examples `SENSE:IQVTIME:MAXTRACEPOINTS TENK` sets the maximum trace points to 10 k.

[SENSe]:MCPower:AVERage

Selects or queries the average method in the MCPR measurement.

Conditions Measurement views: MCPR

Group Sense commands

Syntax

```
[SENSe]:MCPower:AVERage { OFF | TIMe | FREQuency }
[SENSe]:MCPower:AVERage?
```

Arguments `OFF` disables averaging.

`TIMe` performs averaging for time samples.

`FREQuency` performs averaging for frequency samples.

Examples `SENSE:MCPOWER:AVERAGE TIMe` performs averaging for time samples.

[SENSe]:MCPower:AVERage:COUNt

Sets or queries the average count in the MCPR measurement.

Conditions Measurement views: MCPR

Group Sense commands

Syntax

```
[SENSe]:MCPower:AVERage:COUNT <value>
[SENSe]:MCPower:AVERage:COUNt?
```

Arguments `<value>::=<NR1>` specifies the average count. Range: 2 to 10000.

Examples `SENSE:MCPOWER:AVERAGE:COUNT 256` sets the average count to 256.

[SENSe]:MCPower:{BANDwidth|BWIDth}[:RESolution]

Sets or queries the resolution bandwidth (RBW). Programming a specified RBW sets [SENSe]:MCPower{BANDwidth|BWIDth}[:RESolution]:AUTO OFF.

Conditions Measurement views: MCPR

Group Sense commands

Syntax

```
[SENSe]:MCPower:{BANDwidth|BWIDth}[:RESolution] <value>
[SENSe]:MCPower:{BANDwidth|BWIDth}[:RESolution]?
```

Related Commands [SENSe]:MCPower:{BANDwidth|BWIDth}[:RESolution]:AUTO

Arguments `<value>::=<NRf>` specifies the RBW. Range: 100 Hz to 5 MHz.

Examples `SENSE:MCPOWER:BANDWIDTH:RESOLUTION 200kHz` sets the RBW to 200 kHz.

[SENSe]:MCPower:{BANDwidth|BWIDth}[:RESolution]:ACTual? (Query Only)

Queries the actual resolution bandwidth (RBW) in the MCPR measurement.

Conditions Measurement views: MCPR

Group Sense commands

Syntax

```
[SENSe]:MCPower:{BANDwidth|BWIDth}[:RESolution]:ACTual?
```

Arguments None

Returns `<NRf>` The actual RBW in Hz.

Examples `SENSE:MCPOWER:BANDWIDTH:RESOLUTION:ACTUAL?` might return `299.624E+3`, indicating that the actual RBW is 299.624 kHz.

[SENSe]:MCPower:{BANDwidth|BWIDth}[:RESolution]:AUTO

Determines whether to set the resolution bandwidth (RBW) automatically or manually.

Conditions Measurement views: MCPR

Group Sense commands

Syntax

```
[SENSe]:MCPower:{BANDwidth|BWIDth}[:RESolution]:AUTO { OFF |
ON | 0 | 1 }
[SENSe]:MCPower:{BANDwidth|BWIDth}[:RESolution]:AUTO?
```

Arguments `OFF` or `0` specifies that the resolution bandwidth is set manually using the [SENSe]:MCPower:{BANDwidth|BWIDth}[:RESolution] command.

`ON` or `1` specifies that the resolution bandwidth is set automatically.

Examples `SENSE:MCPOWER:BANDWIDTH:AUTO ON` sets the resolution bandwidth automatically.

[SENSe]:MCPower:{BANDwidth|BWIDth}:VIDeo

Sets or queries the video bandwidth (VBW). Programming a specified VBW sets [SENSe]:MCPower:{BANDwidth|BWIDth}:VIDeo:STATe OFF.

Conditions Measurement views: MCPR

Group Sense commands

Syntax

```
[SENSe]:MCPower:{BANDwidth|BWIDth}:VIDeo <value>
[SENSe]:MCPower:{BANDwidth|BWIDth}:VIDeo?
```

Related Commands [SENSe]:MCPower:{BANDwidth|BWIDth}:VIDeo:STATe

Arguments `<value>::=<NRf>` specifies the VBW.
Range: Current RBW/10^4 (1 Hz minimum) to Current RBW.

Examples `SENSE:MCPOWER:BANDWIDTH:VIDEO 200kHz` sets the VBW to 200 kHz.

[SENSe]:MCPower:{BANDwidth|BWIDth}:VIDeo:STATe

Determines whether to enable or disable the video bandwidth (VBW) in theMCPR measurement.

Conditions Measurement views: MCPR

Group Sense commands

Syntax
```
[SENSe]:MCPower:{BANDwidth|BWIDth}:VIDeo:STATe { OFF | ON
| 0 | 1 }
[SENSe]:MCPower:{BANDwidth|BWIDth}:VIDeo:STATe?
```

Arguments `OFF` or `0` disables the VBW.

`ON` or `1` enables the VBW.

Examples `SENSE:MCPOWER:BANDWIDTH:VIDEO:STATE ON` enables the VBW.

[SENSe]:MCPower:CHANnel:ADJacent:ADD (No Query Form)

Adds a pair of upper and lower adjacent channels in the MCPR measurement.

Conditions Measurement views: MCPR

Group Sense commands

Syntax
```
[SENSe]:MCPower:CHANnel:ADJacent:ADD <offset>,<bandwidth>
```

Arguments `<offset>::=<NRf>` specifies the offset from the center frequency for the adjacent channel. Range: 0 to 6.2 GHz (RSA6106A) / 14 GHz (RSA6114A).

`<bandwidth>::=<NRf>` specifies the bandwidth of the adjacent channel. Range: 0 to 6.2 GHz (RSA6106A) / 14 GHz (RSA6114A).

Examples `SENSE:MCPOWER:CHANNEL:ADJACENT:ADD 200kHz,80kHz` adds a pair of upper and lower adjacent channels with the offset of ±200 kHz and the bandwidth of 80 kHz.

[SENSe]:MCPower:CHANnel:ADJacent:DELete (No Query Form)

Deletes a selected adjacent channel in the MCPR measurement.

Conditions Measurement views: MCPR

Group Sense commands

Syntax `[SENSe]:MCPower:CHANnel:ADJacent:DELete <channel>`

Arguments `<channel>::=<string>` specifies the channel to be deleted.
Specify the channel with `"A<n>"` for the adjacent channel where <n> represents the channel number (<n> = 1, 2, 3,...). See the example below.

Examples `SENSe:MCPower:CHANnel:ADJacent:DELete "A2"` deletes A2 (the adjacent channel 2).

[SENSe]:MCPower:CHANnel:FILTer

Selects or queries the adjacent channel filter in the MCPR measurement.

Conditions Measurement views: MCPR

Group Sense commands

Syntax

```
[SENSe]:MCPower:CHANnel:FILTer { RRCosine | NONE }
[SENSe]:MCPower:CHANnel:FILTer?
```

Arguments `RRCosine` selects the Root-Raised-Cosine filter.

`NONE` uses no filter.

Examples `SENSE:MCPOWER:CHANNEL:FILTER RRCosine` selects Root-Raised-Cosine for the adjacent channel filter.

[SENSe]:MCPower:CHANnel:MAIN:{BANDwidth|BWIDth}

Sets or queries the frequency bandwidth of the main channels (all share the same value) in the MCPR measurement.

Conditions Measurement views: MCPR

Group Sense commands

Syntax

```
[SENSe]:MCPower:CHANnel:MAIN:{BANDwidth|BWIDth} <value>
[SENSe]:MCPower:CHANnel:MAIN:{BANDwidth|BWIDth}?
```

Arguments `<value>::=<NRf>` specifies the main channel bandwidth.
Range: 1 Hz to full span.

Examples `SENSE:MCPOWER:CHANNEL:MAIN:BANDWIDTH 4.5MHz` sets the main channel bandwidth to 4.5 MHz.

[SENSe]:MCPower:CHANnel:MAIN:COUNt

Sets or queries the number of main channels in the MCPR measurement. You can use this command to add and remove main channels.

Conditions Measurement views: MCPR

Group Sense commands

Syntax

```
[SENSe]:MCPower:CHANnel:MAIN:COUNt <value>
[SENSe]:MCPower:CHANnel:MAIN:COUNt?
```

Arguments `<value>::=<NRf>` specifies the number of main channels. Range: 1 to 99.

Examples `SENSE:MCPOWER:CHANNEL:MAIN:COUNT 3` sets the the number of main channels to 3.

[SENSe]:MCPower:CHANnel:MAIN:INACtive

Makes a specified main channel inactive. You can set it on or off. The query returns all inactive main channels.

Conditions Measurement views: MCPR

Group Sense commands

Syntax

```
[SENSe]:MCPower:CHANnel:MAIN:INACtive <channel>,<boolean>
[SENSe]:MCPower:CHANnel:MAIN:INACtive?
```

Arguments `<channel>::=<string>` specifies the channel to be inactive.
Specify the channel with `"M<n>"` for the main channel where <n> represents the channel number (<n> = 1, 2, 3,...). See the example below.

`<boolean>::={ OFF | ON | 0 | 1 }` specifies that the specified channel is inactive (On) or not (Off).

Examples `SENSE:MCPOWER:CHANNEL:MAIN:INACTIVE "M2",ON` makes the main channel 2 inactive.

[SENSe]:MCPower:CHANnel:MAIN:SPACing

Sets or queries frequency difference between centers of each main channel in the MCPR measurement.

Conditions Measurement views: MCPR

Group Sense commands

Syntax

```
[SENSe]:MCPower:CHANnel:MAIN:SPACing <value>
[SENSe]:MCPower:CHANnel:MAIN:SPACing?
```

Arguments `<value>::=<NRf>` specifies the spacing between two adjacent main channels.
Range: 1 Hz to 1 GHz.

Examples `SENSE:MCPOWER:CHANNEL:MAIN:SPACING 5MHz` sets the main channel spacing to 5 MHz.

[SENSe]:MCPower:CHIPrate

Sets or queries the chip rate in the MCPR measurement. This command is valid when [SENSe]:MCPower:CHANnel:FILTer is set to RRCosine (Root-Raised-Cosine).

Conditions Measurement views: MCPR

Group Sense commands

Syntax

```
[SENSe]:MCPower:CHIPrate <value>
[SENSe]:MCPower:CHIPrate?
```

Arguments `<value>::=<NRf>` specifies the chip rate. Range: 100 Hz to 105 MHz.

Examples `SENSE:MCPOWER:CHIPRATE 1kHz` sets the chip rate to 1 kHz.

[SENSe]:MCPower:CLEar:RESults (No Query Form)

Restarts the average trace.

Conditions Measurement views: MCPR

Group Sense commands

Syntax

```
[SENSe]:MCPower:CLEar:RESults
```

Arguments None

Examples `SENSE:MCPOWER:CLEAR:RESULTS` restarts the average trace.

[SENSe]:MCPower:FREQuency

Sets or queries the center frequency in the MCPR measurement.

Conditions Measurement views: MCPR

Group Sense commands

Syntax
```
[SENSe]:MCPower:FREQuency <value>
[SENSe]:MCPower:FREQuency?
```

Arguments `<value>::=<NRf>` specifies the center frequency.
Range: 0 to 6.2 GHz (RSA6106A) / 14 GHz (RSA6114A).

Examples `SENSE:MCPOWER:FREQUENCY 2.35GHz` sets the center frequency to 2.35 GHz.

[SENSe]:MCPower:FREQuency:STEP

Sets or queries the frequency step size. Programming a specified step size sets [SENSe]:MCPower:FREQuency:STEP:AUTO OFF.

Conditions Measurement views: MCPR

Group Sense commands

Syntax
```
[SENSe]:MCPower:FREQuency:STEP <value>
[SENSe]:MCPower:FREQuency:STEP?
```

Related Commands [SENSe]:MCPower:FREQuency:STEP:AUTO

Arguments `<value>::=<NRf>` specifies the frequency step size.
Range: 0 to 6.2 GHz (RSA6106A) / 14 GHz (RSA6114A).

Examples `SENSE:MCPOWER:FREQUENCY:STEP 50kHz` sets the frequency step size to 50 kHz.

[SENSe]:MCPower:FREQuency:STEP:AUTO

Determines whether to set the frequency step size automatically or manually in the MCPR measurement.

Conditions Measurement views: MCPR

Group Sense commands

Syntax
```
[SENSe]:MCPower:FREQuency:STEP:AUTO { OFF | ON | 0 | 1 }
[SENSe]:MCPower:FREQuency:STEP:AUTO?
```

Arguments OFF or 0 specifies that the frequency step size is set manually using the [SENSe]:MCPower:FREQuency:STEP command.

ON or 1 specifies that the frequency step size is set automatically.

Examples SENSE:MCPOWER:FREQUENCY:STEP:AUTO ON specifies that the frequency step size is set automatically.

[SENSe]:MCPower:NFLoor:STATe

Determines whether to enable or disable correction for noise floor.

Conditions Measurement views: MCPR

Group Sense commands

Syntax
```
[SENSe]:MCPower:NFLoor:STATe { OFF | ON | 0 | 1 }
[SENSe]:MCPower:NFLoor:STATe?
```

Arguments OFF or 0 disables correction for noise floor.

ON or 1 enables correction for noise floor.

Examples SENSE:MCPOWER:NFLOOR:STATE ON enables correction for noise floor.

[SENSe]:MCPower:OPTimize:SPAN

Selects or queries the optimization method in the MCPR measurement.

Conditions Measurement views: MCPR

Group Sense commands

Syntax `[SENSe]:MCPower:OPTimize:SPAN { RTBandwidth | DRANge }`
`[SENSe]:MCPower:OPTimize:SPAN?`

Arguments `RTBandwidth` optimizes the measurement for real-time bandwidth.

`DRANge` optimizes the measurement for dynamic range.

Examples `SENSE:MCPOWER:OPTIMIZE:SPAN RTBandwidth` optimizes the measurement for real-time bandwidth.

[SENSe]:MCPower:RCHannels? (Query Only)

Queries the power reference in the MCPR measurement.

Conditions Measurement views: MCPR

Group Sense commands

Syntax `[SENSe]:MCPower:RCHannels?`

Arguments None

Returns `<power_ref>::={ Total | M<x> }` where <x> = 1 to 99.

`Total` indicates that the power reference is the total power of all the active channels.

`M<x>` indicates that the power reference is the main channel with the index (<x>).

Examples `SENSE:MCPOWER:RCHANNELS?` might return `M3`, indicating that the power reference is the main channel 3.

[SENSe]:MCPower:RCHannels:MAIN<x> (No Query Form)

Sets the power reference to the main channel with the index (<x>) in the MCPR measurement.

The parameter <x> = 1 to 99, representing the main channel 1 to 99, respectively. The main channel must be defined using the [SENSe]:MCPower:CHANnel:MAIN commands.

Conditions Measurement views: MCPR

Group Sense commands

Syntax `[SENSe]:MCPower:RCHannels:MAIN<x>`

Related Commands [:SENSe]:MCPower:CHANnel:MAIN commands

Arguments None

Examples `SENSE:MCPOWER:RCHANNELS:MAIN3` selects Main 3 for the power reference channel.

[SENSe]:MCPower:RCHannels:TOTal (No Query Form)

Sets the power reference to the total power of all the active channels in the MCPR measurement.

Conditions Measurement views: MCPR

Group Sense commands

Syntax `[SENSe]:MCPower:RCHannels:TOTal`

Arguments None

Examples `SENSE:MCPOWER:RCHANNELS:TOTAL` sets the power reference to the total power of all the active channels.

[SENSe]:MCPower:RRCRolloff

Sets or queries the filter parameter (roll-off ratio) for the Root Raised Cosine filter.

Conditions Measurement views: MCPR

Group Sense commands

Syntax
```
[SENSe]:MCPower:RRCRolloff <value>
[SENSe]:MCPower:RRCRolloff?
```

Related Commands [SENSe]:MCPower:CHANnel:FILTer

Arguments `<value>::=<NRf>` specifies the filter parameter.
Range: 0.001 to 1, 0.0001 step.

Examples `SENSE:MCPOWER:RRCROLLOFF 0.3` sets the filter parameter to 0.3.

[SENSe]:MEASurement:FREQuency

Sets or queries the measurement frequency.

Conditions Measurement views: All

Group Sense commands

Syntax
```
[SENSe]:MEASurement:FREQuency <value>
[SENSe]:MEASurement:FREQuency?
```

Arguments `<value>::=<NRf>` specifies the measurement frequency.
Range: 0 Hz to 6.2 GHz (RSA6106A) / 14 GHz (RSA6114A).

Examples `SENSE:MEASUREMENT:FREQUENCY 7.5GHz` sets the measurement frequency to 7.5 GHz.

[SENSe]:OBWidth:AVERage

Selects or queries whether to enable or disable averaging in the Occupied Bandwidth measurement.

Conditions Measurement views: Occupied Bandwidth

Group Sense commands

Syntax
```
[SENSe]:OBWidth:AVERage { OFF | ON | 0 | 1 }
[SENSe]:OBWidth:AVERage?
```

Arguments `OFF` or `0` disables averaging.

`ON` or `1` enables averaging.

Examples `SENSE:OBWIDTH:AVERAGE ON` enables averaging.

[SENSe]:OBWidth:AVERage:COUNt

Sets or queries the number of measurements for averaging in the Occupied Bandwidth measurement.

Conditions Measurement views: Occupied Bandwidth

Group Sense commands

Syntax
```
[SENSe]:OBWidth:AVERage:COUNt <number>
[SENSe]:OBWidth:AVERage:COUNt?
```

Arguments `<number>::=<NR1>` specifies the average count. Range: 2 to 10000.

Examples `SENSE:OBWIDTH:AVERAGE:COUNT 64` sets the average count to 64.

[SENSe]:OBWidth:{BANDwidth|BWIDth}:MEASurement

Sets or queries the measurement bandwidth to determine the total power in the Occupied Bandwidth measurement.

Conditions Measurement views: Occupied Bandwidth

Group Sense commands

Syntax
```
[SENSe]:OBWidth:{BANDwidth|BWIDth}:MEASurement <value>
[SENSe]:OBWidth:{BANDwidth|BWIDth}:MEASurement?
```

Arguments <value>::=<NRf> specifies the measurement bandwidth.
Range: 100 Hz to 109 MHz.

Examples SENSE:OBWIDTH:BANDWIDTH:MEASUREMENT 10MHz sets the measurement bandwidth to 10 MHz.

[SENSe]:OBWidth:{BANDwidth|BWIDth}[:RESolution]

Sets or queries the resolution bandwidth (RBW). Programming a specified RBW sets [SENSe]:OBWidth{BANDwidth|BWIDth}[:RESolution]:AUTO OFF.

Conditions Measurement views: Occupied Bandwidth

Group Sense commands

Syntax
```
[SENSe]:OBWidth:{BANDwidth|BWIDth}[:RESolution] <value>
[SENSe]:OBWidth:{BANDwidth|BWIDth}[:RESolution]?
```

Related Commands [SENSe]:OBWidth:{BANDwidth|BWIDth}[:RESolution]:AUTO

Arguments <value>::=<NRf> specifies the RBW. Range: 100 Hz to 5 MHz.

Examples SENSE:OBWIDTH:BANDWIDTH:RESOLUTION 200kHz sets the RBW to 200 kHz.

[SENSe]:OBWidth:{BANDwidth|BWIDth}[:RESolution]:ACTual? (Query Only)

Queries the actual resolution bandwidth (RBW) in the Occupied Bandwidth measurement.

Conditions Measurement views: Occupied Bandwidth

Group Sense commands

Syntax
```
[SENSe]:OBWidth:{BANDwidth|BWIDth}[:RESolution]:ACTual?
```

Arguments None

Returns <NRf> The actual RBW in Hz.

Examples SENSE:OBWIDTH:BANDWIDTH:RESOLUTION:ACTUAL? might return 299.624E+3, indicating that the actual RBW is 299.624 kHz.

[SENSe]:OBWidth:{BANDwidth|BWIDth}[:RESolution]:AUTO

Determines whether to set the resolution bandwidth (RBW) automatically or manually in the Occupied Bandwidth measurement.

Conditions Measurement views: Occupied Bandwidth

Group Sense commands

Syntax

```
[SENSe]:OBWidth:{BANDwidth|BWIDth}[:RESolution]:AUTO { OFF |
ON | 0 | 1 }
[SENSe]:OBWidth:{BANDwidth|BWIDth}[:RESolution]:AUTO?
```

Arguments OFF or 0 specifies that the RBW is set manually using the [SENSe]:OBWidth:{BANDwidth|BWIDth}[:RESolution] command.

ON or 1 specifies that the RBW is set automatically.

Examples SENSE:OBWIDTH:BANDWIDTH:AUTO ON sets the RBW automatically.

[SENSe]:OBWidth:{BANDwidth|BWIDth}:VIDeo

Sets or queries the video bandwidth (VBW). Programming a specified VBW sets [SENSe]:OBWidth{BANDwidth|BWIDth}:VIDeo:STATe OFF.

Conditions Measurement views: Occupied Bandwidth

Group Sense commands

Syntax

```
[SENSe]:OBWidth:{BANDwidth|BWIDth}:VIDeo <value>
[SENSe]:OBWidth:{BANDwidth|BWIDth}:VIDeo?
```

Arguments <value>::=<NRf> specifies the VBW.
Range: Current RBW/10^4 (1 Hz minimum) to Current RBW.

Examples `SENSE:OBWIDTH:BANDWIDTH:VIDEO 200kHz` sets the VBW to 200 kHz.

[SENSe]:OBWidth:{BANDwidth|BWIDth}:VIDeo:STATe

Determines whether to enable or disable the video bandwidth (VBW) in the Occupied Bandwidth measurement.

Conditions Measurement views: Occupied Bandwidth

Group Sense commands

Syntax

```
[SENSe]:OBWidth:{BANDwidth|BWIDth}:VIDeo:STATe { OFF | ON
| 0 | 1 }
[SENSe]:OBWidth:{BANDwidth|BWIDth}:VIDeo:STATe?
```

Arguments `OFF` or `0` disables the VBW.

`ON` or `1` enables the VBW.

Examples `SENSE:OBWIDTH:BANDWIDTH:VIDEO:STATE ON` enables the VBW.

[SENSe]:OBWidth:CLEar:RESults (No Query Form)

Restarts the average trace. This command is valid when [SENSe]:OBWidth:AVERage is set to ON.

Conditions Measurement views: Occupied Bandwidth

Group Sense commands

Syntax `[SENSe]:OBWidth:CLEar:RESults`

Arguments None

Examples `SENSE:OBWIDTH:CLEAR:RESULTS` restarts the average trace.

[SENSe]:OBWidth:FREQuency:CENTer

Sets or queries the center frequency in the Occupied Bandwidth measurement.

Conditions Measurement views: Occupied Bandwidth

Group Sense commands

Syntax

```
[SENSe]:OBWidth:FREQuency:CENTer <value>
[SENSe]:OBWidth:FREQuency:CENTer?
```

Arguments `<value>::=<NRf>` specifies the center frequency.
Range: 0 Hz to 6.2 GHz (RSA6106A) / 14 GHz (RSA6114A).

Examples `SENSE:OBWIDTH:FREQUENCY:CENTER 7.5GHz` sets the center frequency to 7.5 GHz.

[SENSe]:OBWidth:FREQuency:STEP

Sets or queries the frequency step size in the Occupied Bandwidth measurement. Programming a specified step size sets [SENSe]:OBWidth:FREQuency:STEP AUTO OFF.

Conditions Measurement views: Occupied Bandwidth

Group Sense commands

Syntax

```
[SENSe]:OBWidth:FREQuency:STEP <value>
[SENSe]:OBWidth:FREQuency:STEP?
```

Related Commands [SENSe]:OBWidth:FREQuency:STEP:AUTO

Arguments `<value>::=<NRf>` specifies the frequency step size.
Range: 0 to 6.2 GHz (RSA6106A) / 14 GHz (RSA6114A).

Examples `SENSE:OBWIDTH:FREQUENCY:STEP 1kHz` sets the frequency step size to 1 kHz.

[SENSe]:OBWidth:FREQuency:STEP:AUTO

Determines whether to set the frequency step size automatically or manually in the Occupied Bandwidth measurement.

Conditions Measurement views: Occupied Bandwidth

Group Sense commands

Syntax

```
[SENSe]:OBWidth:FREQuency:STEP:AUTO { OFF | ON | 0 | 1 }
[SENSe]:OBWidth:FREQuency:STEP:AUTO?
```

Arguments `OFF` or `0` specifies that the frequency step size is set manually using the [SENSe]:OBWidth:FREQuency:STEP command.

`ON` or `1` specifies that the frequency step size is set automatically.

Examples `SENSE:OBWIDTH:FREQUENCY:STEP:AUTO ON` specifies that the frequency step size is set automatically.

[SENSe]:OBWidth:PERCent

Sets or queries the occupied bandwidth percent power (power ratio of the occupied bandwidth to the measurement bandwidth).

Conditions Measurement views: Occupied Bandwidth

Group Sense commands

Syntax

```
[SENSe]:OBWidth:PERCent <value>
[SENSe]:OBWidth:PERCent?
```

Arguments `<value>::=<NRf>` specifies the occupied bandwidth percent power.
Range: 50 to 99.9%.

Examples `SENSE:OBWIDTH:PERCENT 98` sets the occupied bandwidth percent power to 98%.

[SENSe]:OBWidth:XDBLevel

Sets or queries the x dB level (how far down from the peak level the bandwidth is measured) in the x dB bandwidth measurement.

Conditions Measurement views: Occupied Bandwidth

Group Sense commands

Syntax
```
[SENSe]:OBWidth:XDBLevel <value>
[SENSe]:OBWidth:XDBLevel?
```

Arguments `<value>::=<NRf>` specifies the x dB level. Range: -80 to -1 dB.

Examples `SENSE:OBWIDTH:XDBLEVEL -10` sets the x dB level to -10 dB.

[SENSe]:PHVTime:CLEar:RESults (No Query Form)

Restarts multi-trace functions (Average and Max/Min Hold).

Conditions Measurement views: Phase versus Time

Group Sense commands

Syntax
```
[SENSe]:PHVTime:CLEar:RESults
```

Arguments None

Examples `SENSE:PHVTIME:CLEAR:RESULTS` restarts multi-trace functions.

[SENSe]:PHVTime:FREQuency:CENTer

Sets or queries the center frequency in the Phase versus Time measurement.

NOTE. *The center, start and stop frequencies are set interlocking each other with the following relationships: (start frequency) = (center frequency) - (span)/2 and (stop frequency) = (center frequency) + (span)/2.*

Conditions Measurement views: Phase versus Time

Group Sense commands

Syntax

```
[SENSe]:PHVTime:FREQuency:CENTer <value>
[SENSe]:PHVTime:FREQuency:CENTer?
```

Related Commands [SENSe]:PHVTime:FREQuency:STARt, [SENSe]:PHVTime:FREQuency:STOP

Arguments `<value>::=<NRf>` specifies the center frequency.
Range: 0 Hz to 6.2 GHz (RSA6106A) / 14 GHz (RSA6114A).

Examples `SENSE:PHVTIME:FREQUENCY:CENTER 7.5GHz` sets the center frequency to 7.5 GHz.

[SENSe]:PHVTime:FREQuency:SPAN

Sets or queries the frequency span in the Phase versus Time measurement.

Conditions Measurement views: Phase versus Time

Group Sense commands

Syntax

```
[SENSe]:PHVTime:FREQuency:SPAN <value>
[SENSe]:PHVTime:FREQuency:SPAN?
```

Arguments `<value>::=<NRf>` is the frequency span.
Range: 10 Hz to 40 MHz (Standard) / 110 MHz (Option 110)

Examples `SENSE:PHVTIME:FREQUENCY:SPAN 20MHz` sets the span to 20 MHz.

[SENSe]:PHVTime:FREQuency:STARt

Sets or queries the measurement start frequency (left edge on the graph) in the Phase versus Time measurement.

The center, start and stop frequencies are set interlocking each other. Refer to the [SENSe]:PHVTime:FREQuency:CENTer command.

Conditions Measurement views: Phase versus Time

Group Sense commands

Syntax

```
[SENSe]:PHVTime:FREQuency:STARt <value>
[SENSe]:PHVTime:FREQuency:STARt?
```

Related Commands [SENSe]:PHVTime:FREQuency:STOP

Arguments <value>::=<NRf> is the measurement start frequency.
Range: (center frequency) ± (span)/2.

Examples SENSE:PHVTIME:FREQUENCY:START 6.95GHz sets the start frequency to 6.95 GHz.

[SENSe]:PHVTime:FREQuency:STEP

Sets or queries the frequency step size (the amount per press by which the up or down key changes the setting value). Programming a specified step size sets [SENSe]:PHVTime:FREQuency:STEP:AUTO OFF.

Conditions Measurement views: Phase versus Time

Group Sense commands

Syntax

```
[SENSe]:PHVTime:FREQuency:STEP <value>
[SENSe]:PHVTime:FREQuency:STEP?
```

Related Commands [SENSe]:PHVTime:FREQuency:STEP:AUTO

Arguments <value>::=<NRf> specifies the frequency step size.
Range: 0 Hz to 6.2 GHz (RSA6106A) / 14 GHz (RSA6114A).

Examples SENSE:PHVTIME:FREQUENCY:STEP 1.5kHz sets the step size to 1.5 kHz.

[SENSe]:PHVTime:FREQuency:STEP:AUTO

Determines whether to set the frequency step size automatically or manually.

Conditions Measurement views: Phase versus Time

Group Sense commands

Syntax

```
[SENSe]:PHVTime:FREQuency:STEP:AUTO { OFF | ON | 0 | 1 }
[SENSe]:PHVTime:FREQuency:STEP:AUTO?
```

Arguments `OFF` or `0` specifies that the frequency step size is set manually using the [SENSe]:PHVTime:FREQuency:STEP command.

`ON` or `1` specifies that the frequency step size is set automatically.

Examples `SENSE:PHVTIME:BANDWIDTH:AUTO ON` sets the frequency step size automatically.

[SENSe]:PHVTime:FREQuency:STOP

Sets or queries the measurement stop frequency (right edge of the graph) in the Phase versus Time measurement.

The center, start and stop frequencies are set interlocking each other. Refer to the [SENSe]:PHVTime:FREQuency:CENTer command.

Conditions Measurement views: Phase versus Time

Group Sense commands

Syntax

```
[SENSe]:PHVTime:FREQuency:STOP <value>
[SENSe]:PHVTime:FREQuency:STOP?
```

Related Commands [SENSe]:PHVTime:FREQuency:STARt

Arguments `<value>::=<NRf>` is the measurement stop frequency.
Range: (center frequency) ± (span)/2.

Examples `SENSE:PHVTIME:FREQUENCY:STOP 7.05GHz` sets the stop frequency to 7.05 GHz.

[SENSe]:PHVTime:MAXTracepoints

Selects or queries the maximum trace points in the Phase versus Time measurement.

Conditions Measurement views: Phase versus Time

Group Sense commands

Syntax

```
[SENSe]:PHVTime:MAXTracepoints { ONEK | TENK | HUNDredk |
NEVerdecimate }
[SENSe]:PHVTime:MAXTracepoints?
```

Arguments

`ONEK` sets the maximum trace points to 1 k.

`TENK` sets the maximum trace points to 10 k.

`HUNDredk` sets the maximum trace points to 100 k.

`NEVerdecimate` never decimates the trace points.

Examples `SENSE:PHVTIME:MAXTRACEPOINTS TENK` sets the maximum trace points to 10 k.

[SENSe]:PNOise:AVERage:COUNt

Sets or queries the number of traces to combine for averaging in the phase noise measurement. This command is effective when [SENSe]:PNOise:AVERage:ENABle is set to ON.

Conditions Measurement views: Phase noise

Group Sense commands

Syntax

```
[SENSe]:PNOise:AVERage:COUNt <number>
[SENSe]:PNOise:AVERage:COUNt?
```

Arguments <number>::=<NR1> specifies the average count. Range: 2 to 10000.

Examples `SENSE:PNOISE:AVERAGE:COUNT 64` sets the average count to 64.

[SENSe]:PNOise:AVERage:ENABle

Determines whether to enable or disable averaging trace in the phase noise measurement.

Conditions Measurement views: Phase noise

Group Sense commands

Syntax
```
[SENSe]:PNOise:AVERage:ENABle { OFF | ON | 0 | 1 }
[SENSe]:PNOise:AVERage:ENABle?
```

Arguments `OFF` disables averaging trace.

`ON` enables averaging trace.

Examples `SENSE:PNOISE:AVERAGE:ENABLE ON` enables averaging trace.

[SENSe]:PNOise:CARRier:FREQuency:TRACk

Determines whether to enable or disable tracking the carrier frequency in the phase noise measurement.

Conditions Measurement views: Phase noise

Group Sense commands

Syntax
```
[SENSe]:PNOise:CARRier:FREQuency:TRACK { OFF | ON | 0 | 1 }
[SENSe]:PNOise:CARRier:FREQuency:TRACK?
```

Arguments `OFF` or `0` disables tracking the carrier frequency.

`ON` or `1` enables tracking the carrier frequency.

Examples `SENSE:PNOISE:CARRIER:FREQUENCY:TRACK ON` enables tracking the carrier frequency.

[SENSe]:PNOise:CARRier:THReshold

Sets or queries the threshold level to detect the carrier in the phase noise measurement.

Conditions Measurement views: Phase noise

Group Sense commands

Syntax
```
[SENSe]:PNOise:CARRier:THReshold <value>
[SENSe]:PNOise:CARRier:THReshold?
```

Arguments `<value>::=<NRf>` specifies the threshold level above which the input signal is determined to be a carrier. Range: -60 to 0 dBm.

Examples `SENSE:PNOISE:CARRIER:THRESHOLD -25` sets the carrier threshold level to -25 dB.

[SENSe]:PNOise:CLEar:RESults (No Query Form)

Restarts the average process, clearing average data and counter.

Conditions Measurement views: Phase noise

Group Sense commands

Syntax
```
[SENSe]:PNOise:CLEar:RESults
```

Arguments None

Examples `SENSE:PNOISE:CLEAR:RESULTS` restarts the average process.

[SENSe]:PNOise:FREQuency:INTegration:OFFSet:STARt

Sets or queries the start offset frequency for integration in the phase noise measurement.

Conditions Measurement views: Phase noise

Group Sense commands

Syntax
```
[SENSe]:PNOise:FREQuency:INTegration:OFFSet:STARt <value>
[SENSe]:PNOise:FREQuency:INTegration:OFFSet:STARt?
```

Arguments `<value>::=<NRf>` specifies the start offset frequency for integration.
It must be less than the stop offset frequency.
Range: 10 Hz to Stop Offset Frequency - 1 Hz.

Examples `SENSE:PNOISE:FREQUENCY:INTEGRATION:OFFSET:START 100kHz` sets the start offset frequency for integration to 100 kHz.

[SENSe]:PNOise:FREQuency:INTegration:OFFSet:STOP

Sets or queries the stop offset frequency for integration in the phase noise measurement.

Conditions Measurement views: Phase noise

Group Sense commands

Syntax
```
[SENSe]:PNOise:FREQuency:INTegration:OFFSet:STOP <value>
[SENSe]:PNOise:FREQuency:INTegration:OFFSet:STOP?
```

Arguments `<value>::=<NRf>` specifies the stop offset frequency for integration.
It must be greater than the start offset frequency.
Range: 11 Hz to 1 GHz.

Examples `SENSE:PNOISE:FREQUENCY:INTEGRATION:OFFSET:STOP 100MHz` sets the stop offset frequency for integration to 100 MHz.

[SENSe]:PNOise:FREQuency:PLOT:OFFSet:STARt

Sets or queries the start offset frequency for plotting the phase noise trace.

Conditions Measurement views: Phase noise

Group Sense commands

Syntax
```
[SENSe]:PNOise:FREQuency:PLOT:OFFSet:START <value>
[SENSe]:PNOise:FREQuency:PLOT:OFFSet:STARt?
```

Arguments `<value>::=<NRf>` specifies the start offset frequency for plot.
It must be less than the stop offset frequency.
Range: 10 Hz to 100 MHz in a tenfold sequence.

Examples `SENSE:PNOISE:FREQUENCY:PLOT:OFFSET:START 100kHz` sets the start offset frequency for plot to 100 kHz.

[SENSe]:PNOise:FREQuency:PLOT:OFFSet:STOP

Sets or queries the stop offset frequency for plotting the phase noise trace.

Conditions Measurement views: Phase noise

Group Sense commands

Syntax
```
[SENSe]:PNOise:FREQuency:PLOT:OFFSet:STOP <value>
[SENSe]:PNOise:FREQuency:PLOT:OFFSet:STOP?
```

Arguments `<value>::=<NRf>` specifies the stop offset frequency for plot.
It must be greater than the start offset frequency.
Range: 100 Hz to 1 GHz in a tenfold sequence.

Examples `SENSE:PNOISE:FREQUENCY:PLOT:OFFSET:STOP 100MHz` sets the stop offset frequency for plot to 100 MHz.

[SENSe]:PNOise:OPTimization

Selects or queries the method of optimizing the gain and input bandwidth in the phase noise measurement.

Conditions Measurement views: Phase noise

Group Sense commands

Syntax

```
[SENSe]:PNOise:OPTimization { DRANge | SPEed }
[SENSe]:PNOise:OPTimization?
```

Arguments `DRANge` optimizes the gain and input bandwidth to maximize the dynamic range.

`SPEed` optimizes the gain and input bandwidth to speed the measurement.

Examples `SENSE:PNOISE:OPTIMIZATION DRANge` optimizes the gain and input bandwidth to maximize the dynamic range.

[SENSe]:POWer:UNITs

Selects or queries the fundamental unit of power.

Conditions Measurement views: All

Group Sense commands

Syntax

```
[SENSe]:POWer:UNITs { DBM | DBV | VOLTs | WATTs | DBUW | DBW
| DBUV | DBMV | DBUA | DBUV_M | DBUA_M | AMPS }
[SENSe]:POWer:UNITs?
```

Arguments The following table lists the arguments.

Table 2-39: Power units

Argument	Power unit
DBM	dBm
DBV	dBV
VOLTs	Volts
WATTs	Watts

Table 2-39: Power units (cont.)

Argument	Power unit
DBUW	dBµW
DBW	dBW
DBUV	dBµV
DBMV	dBmV
DBUA	dBµA
DBUV_M	dBµV/m
DBUA_M	dBµA/m
AMPS	Amps

NOTE. *Select dBµV/m or dBµA/m unit when using an antenna table.*

Examples

`SENSE:POWER:UNITS DBM` specifies the fundamental unit of power as dBm.

[SENSe]:PULSe:ANALyze:LEVel

Selects or queries how to determine the 50% level for the pulsed RF measurements.

Conditions

Measurement views: Pulse statistics, Pulse table, Pulse trace

Group

Sense commands

Syntax

```
[SENSe]:PULSe:ANALyze:LEVel { VOLTage | POWer }
[SENSe]:PULSe:ANALyze:LEVel?
```

Arguments

`VOLTage` uses -6 dB to determine the 50% level.

`POWer` uses -3 dB to determine the 50% level.

Examples

`SENSe:PULSe:ANALyze:LEVel POWer` uses -3 dB to determine the 50% level.

[SENSe]:PULSe:ANALyze:LEVel:FIFTy

Selects or queries how to determine the 50% level for the pulsed RF measurements. This command is equivalent to the [SENSe]:PULSe:ANALyze:LEVel command.

Conditions
Measurement views: Pulse statistics, Pulse table, Pulse trace

Group
Sense commands

Syntax
```
[SENSe]:PULSe:ANALyze:LEVel:FIFTy { VOLTage | POWer }
[SENSe]:PULSe:ANALyze:LEVel:FIFTy?
```

Arguments
`VOLTage` uses -6 dB to determine the 50% level.

`POWer` uses -3 dB to determine the 50% level.

Examples
`SENSE:PULSE:ANALYZE:LEVEL:FIFTY POWer` uses -3 dB to determine the 50% level.

[SENSe]:PULSe:ANALyze:LEVel:HUNDred

Selects or queries how to determine the 100% level in the pulsed RF measurements. This command is equivalent to the [SENSe]:PULSe:ANALyze:POINt:LOCation command.

Conditions
Measurement views: Pulse statistics, Pulse table, Pulse trace

Group
Sense commands

Syntax
```
[SENSe]:PULSe:ANALyze:LEVel:HUNDred { AVERage | INDependent
}
[SENSe]:PULSe:ANALyze:LEVel:HUNDred?
```

Related Commands

Arguments
`AVERage` uses the average amplitude calculated for the pulse-on as the 100% reference to measure the rise and fall times.

`INDependent` uses the amplitudes at the beginning and end of the pulse-on as the 100% references to measure the rise and fall times, respectively

Examples
`SENSE:PULSE:ANALYZE:LEVEL:HUNDRED AVERage` uses pulse average amplitude to set the 100% level.

[SENSe]:PULSe:ANALyze:MEASurement:TIME:AUTO

Determines whether to set the measurement time for frequency and phase results automatically or manually in the pulsed RF measurements.

Conditions Measurement views: Pulse statistics, Pulse table, Pulse trace

Group Sense commands

Syntax

```
[SENSe]:PULSe:ANALyze:MEASurement:TIME:AUTO { OFF | ON | 0 |
1 }
[SENSe]:PULSe:ANALyze:MEASurement:TIME:AUTO?
```

Arguments `OFF` or `0` sets the measurement time manually. Use the [SENSe]:PULSe:ANALyze:MEASurement:TIME:STARt and [SENSe]:PULSe:ANALyze:MEASurement:TIME:STOP commands to set the measurement start and stop time.

`ON` or `1` sets the measurement time automatically.

Examples `SENSE:PULSE:ANALYZE:MEASUREMENT:TIME:AUTO ON` sets the measurement time for frequency and phase results automatically.

[SENSe]:PULSe:ANALyze:MEASurement:TIME:STARt

Sets or queries the measurement start time for frequency and phase results in the pulsed RF measurements.

Conditions Measurement views: Pulse statistics, Pulse table, Pulse trace

Group Sense commands

Syntax

```
[SENSe]:PULSe:ANALyze:MEASurement:TIME:START <value>
[SENSe]:PULSe:ANALyze:MEASurement:TIME:STARt?
```

Related Commands [SENSe]:PULSe:ANALyze:MEASurement:TIME:STOP

Arguments `<value>::=<NRf>` specifies the measurement start time from the 50% level of.the pulse rising edge. Range: -100 to 100 ms.

Examples

`SENSe:PULSe:ANALyze:MEASurement:TIME:STARt 2.8us` sets the start time to 2.8 μs from the 50% level of the pulse rising edge.

[SENSe]:PULSe:ANALyze:MEASurement:TIME:STOP

Sets or queries the measurement stop time for frequency and phase results in the pulsed RF measurements.

Conditions

Measurement views: Pulse statistics, Pulse table, Pulse trace

Group

Sense commands

Syntax

```
[SENSe]:PULSe:ANALyze:MEASurement:TIME:STOP <value>
[SENSe]:PULSe:ANALyze:MEASurement:TIME:STOP?
```

Related Commands

[SENSe]:PULSe:ANALyze:MEASurement:TIME:STARt

Arguments

`<value>::=<NRf>` specifies the measurement stop time from the 50% level of the pulse falling edge. Range: -100 to 100 ms.

Examples

`SENSe:PULSe:ANALyze:MEASurement:TIME:STOP 1.2us` sets the stop time to 1.2 μs from the 50% level of the pulse falling edge.

[SENSe]:PULSe:ANALyze:PMLocation

Sets or queries the phase measurement location (the position along the pulse tops where the phase is measured) in the pulse-pulse phase measurement.

Conditions

Measurement views: Pulse statistics, Pulse table, Pulse trace

Group

Sense commands

Syntax

```
[SENSe]:PULSe:ANALyze:PMLocation <value>
[SENSe]:PULSe:ANALyze:PMLocation?
```

Arguments

`<value>::=<NRf>` specifies the pulse-pulse phase measurement location. Range: 5 ns to 100 ms.

Examples `SENSE:PULSE:ANALYZE:PMLOCATION 1.5ms` sets the phase measurement location to 1.5 ms.

[SENSe]:PULSe:ANALyze:POINt:LOCation

Selects or queries the point location method in the pulsed RF measurements.

Conditions Measurement views: Pulse statistics, Pulse table, Pulse trace

Group Sense commands

Syntax
```
[SENSe]:PULSe:ANALyze:POINt:LOCation { AVERage | INDependent
}
[SENSe]:PULSe:ANALyze:POINt:LOCation?
```

Arguments `AVERage` uses the average amplitude calculated for the pulse-on as the 100% reference to measure the rise and fall times.

`INDependent` uses the amplitudes at the beginning and end of the pulse-on as the 100% references to measure the rise and fall times, respectively.

Examples `SENSE:PULSE:ANALYZE:POINT:LOCATION AVERage` uses pulse average amplitude to locate points.

[SENSe]:PULSe:ANALyze:RFALl

Selects or queries the threshold levels to measure the rise/fall time.

Conditions Measurement views: Pulse statistics, Pulse table, Pulse trace

Group Sense commands

Syntax
```
[SENSe]:PULSe:ANALyze:RFALl { WIDE | NARRow }
[SENSe]:PULSe:ANALyze:RFALl?
```

Arguments `WIDE` selects 10 – 90% to measure the rise/fall time.

`NARRow` selects 20 – 80% to measure the rise/fall time.

Examples `SENSE:PULSE:ANALYZE:RFALL WIDE` selects 10 – 90% to measure the rise/fall time.

[SENSe]:PULSe:ANALyze:RIPPle

Sets or queries the ripple portion of the pulse top (that is, how much of the beginning and end of the pulse top is excluded from the ripple calculation).

Conditions Measurement views: Pulse statistics, Pulse table, Pulse trace

Group Sense commands

Syntax
```
[SENSe]:PULSe:ANALyze:RIPPle <value>
[SENSe]:PULSe:ANALyze:RIPPle?
```

Arguments `<value>::=<NRf>` specifies the ripple portion of the pulse top.
Range: 10 to 100% in 1% steps.

Examples `SENSE:PULSE:ANALYZE:RIPPLE 30` sets the ripple portion to 30% of the pulse top.

[SENSe]:PULSe:CARRier:OFFSet

Sets or queries the carrier frequency offset.

Conditions Measurement views: Pulse statistics, Pulse table, Pulse trace

Group Sense commands

Syntax
```
[SENSe]:PULSe:CARRier:OFFSet <value>
[SENSe]:PULSe:CARRier:OFFSet?
```

Related Commands [SENSe]:PULSe:CARRier:SEARch

Arguments `<value>::=<NRf>` specifies the carrier frequency offset.
Range: -50 kHz to +50 kHz.

Examples `SENSE:PULSE:CARRIER:OFFSET 2.5kHz` sets the offset frequency to 2.5 kHz to the carrier.

[SENSe]:PULSe:CARRier:SEARch

Selects or queries how to detect the carrier in the pulsed RF measurements.

Conditions Measurement views: Pulse statistics, Pulse table, Pulse trace

Group Sense commands

Syntax

```
[SENSe]:PULSe:CARRier:SEARch { AUTO | MANual }
[SENSe]:PULSe:CARRier:SEARch?
```

Arguments `AUTO` specifies that the carrier is detected automatically.

`MANual` specifies that the carrier frequency offset is set manually, using the [SENSe]:PULSe:CARRier:OFFSet command.

Examples `SENSE:PULSE:CARRIER:SEARCH AUTO` specifies that the carrier is detected automatically.

[SENSe]:PULSe:DETect:MEASurement

Determines whether or not to set the maximum number of pulses to measure within the analysis time.

Conditions Measurement views: Pulse statistics, Pulse table, Pulse trace

Group Sense commands

Syntax

```
[SENSe]:PULSe:DETect:MEASurement { OFF | ON | 0 | 1 }
[SENSe]:PULSe:DETect:MEASurement?
```

Arguments `OFF` or `0` measures all pulses (max. 1000) in the analysis time.

`ON` or `1` specifies that the maximum number of pulses is set manually, using the [SENSe]:PULSe:DETect:NUMBer command.

Examples `SENSE:PULSE:DETECT:MEASUREMENT ON` specifies that the maximum number of pulses is set manually.

[SENSe]:PULSe:DETect:NUMBer

Sets or queries the maximum number of pulses to measure within the analysis time when [SENSe]:PULSe:DETect:MEASurement is On.

Conditions Measurement views: Pulse statistics, Pulse table, Pulse trace

Group Sense commands

Syntax
```
[SENSe]:PULSe:DETect:NUMBer <value>
[SENSe]:PULSe:DETect:NUMBer?
```

Arguments `<value>::=<NRf>` specifies the maximum number of pulses to measure within the analysis time. Range: 1 to 1000.

If the analysis time contains fewer pulses than this number, all of these are measured.

Examples `SENSe:PULSe:DETect:NUMBer 850` sets the maximum number of pulses to 850.

[SENSe]:PULSe:DETect:POWer[:THReshold]

Sets or queries the power threshold to detect pulses.

Conditions Measurement views: Pulse statistics, Pulse table, Pulse trace

Group Sense commands

Syntax
```
[SENSe]:PULSe:DETect:POWer[:THReshold] <value>
[SENSe]:PULSe:DETect:POWer[:THReshold]?
```

Arguments `<value>::=<NRf>` specifies the power threshold to detect pulses. Range: -70 to 0 dB.

Examples `SENSE:PULSE:DETECT:POWER:THRESHOLD -20` sets the power threshold to -20 dB.

[SENSe]:PULSe:DETect:TIMe[:THReshold]

Sets or queries the minimum off-time between pulses.

Conditions Measurement views: Pulse statistics, Pulse table, Pulse trace

Group Sense commands

Syntax

```
[SENSe]:PULSe:DETect:TIMe[:THReshold] <value>
[SENSe]:PULSe:DETect:TIMe[:THReshold]?
```

Arguments `<value>::=<NRf>` specifies the minimum off-time between pulses.
Range: 1 ns to 100 ms.

Examples `SENSE:PULSE:DETECT:TIME:THRESHOLD 1.5ms` sets the time threshold to 1.5 ms.

[SENSe]:PULSe:FILTer:{BANDwidth|BWIDth}

Sets or queries the filter or acquisition bandwidth when [SENSe]:PULSe:FILTer:MEASurement is set to GAUSsian or NONE (No filter), respectively.

Conditions Measurement views: Pulse statistics, Pulse table, Pulse trace

Group Sense commands

Syntax

```
[SENSe]:PULSe:FILTer:{BANDwidth|BWIDth} <value>
[SENSe]:PULSe:FILTer:{BANDwidth|BWIDth}?
```

Related Commands [SENSe]:PULSe:FILTer:MEASurement

Arguments `<value>::=<NRf>` specifies the filter/acquisition bandwidth depending the [SENSe]:PULSe:FILTer:MEASurement setting. The table below shows the setting range. You can enter any value, but it is rounded up to the next valid number.

[SENSe]:PULSe:FILTer :MEASurement	Range
GAUSsian	100 Hz to 20 MHz (Standard) / 100 Hz to 55 MHz (Option 110) in 1-2-3-5 sequence.
NONE	152, 305, 610, 1.22 k, 2.44 k, 4.88 k, 9.76 k, 19.3 k, 39 k, 78 k, 156 k, 312 k, 625 k, 1.25 M, 2.5 M, 5 M, 10 M, 20 M, 40 MHz, and optionally 60 M and 110 MHz (Option 110).

Examples

`SENSE:PULSE:FILTER:BANDWIDTH 10MHz` sets the filter bandwidth to 10 MHz.

[SENSe]:PULSe:FILTer:MEASurement

Selects or queries the measurement filter in the pulsed RF measurements.

Conditions

Measurement views: Pulse statistics, Pulse table, Pulse trace

Group

Sense commands

Syntax

```
[SENSe]:PULSe:FILTer:MEASurement { GAUSsian | NONE | MAXRtbw
}
[SENSe]:PULSe:FILTer:MEASurement?
```

Related Commands

[SENSe]:PULSe:FILTer:{BANDwidth|BWIDth}

Arguments

`GAUSsian` uses the Gaussian filter in the pulsed RF measurements. Use the [SENSe]:PULSe:FILTer:{BANDwidth|BWIDth} command to set the filter bandwidth.

`NONE` uses no filter. Use the [SENSe]:PULSe:FILTer:{BANDwidth|BWIDth} command to set the acquisition bandwidth.

`MAXRtbw` uses no filter. The acquisition bandwidth is fixed to the maximum real-time bandwidth: 40 MHz (Standard) or 110 MHz (Option 110).

Examples

`SENSE:PULSE:FILTER:MEASUREMENT GAUSsian` uses the Gaussian filter in the pulsed RF measurements.

[SENSe]:PULSe:FREFerence:AUTO

Determines whether to estimate the pulse frequency reference automatically or manually in the pulsed RF measurements.

Conditions Measurement views: Pulse statistics, Pulse table, Pulse trace

Group Sense commands

Syntax

```
[SENSe]:PULSe:FREFerence:AUTO { OFF | ON | 0 | 1 }
[SENSe]:PULSe:FREFerence:AUTO?
```

Arguments `OFF` or `0` estimates the frequency reference manually. Use the [SENSe]:PULSe:FREFerence:OFFSet command to set the frequency offset. Use the [SENSe]:PULSe:FREFerence:CHIRpbw command to set the chirp bandwidth.

`ON` or `1` estimates the frequency reference automatically.

Examples `SENSE:PULSE:FREFERENCE:AUTO ON` specifies that the frequency reference is estimated automatically.

[SENSe]:PULSe:FREFerence:CHIRpbw

Sets or queries the chirp bandwidth. This command is valid when [SENSe]:PULSe:MODulation:TYPE is set to LCHirp and [SENSe]:PULSe:FREFerence:AUTO is set to OFF.

Conditions Measurement views: Pulse statistics, Pulse table, Pulse trace

Group Sense commands

Syntax

```
[SENSe]:PULSe:FREFerence:CHIRpbw <value>
[SENSe]:PULSe:FREFerence:CHIRpbw?
```

Arguments <value>::=<NRf> specifies the chirp bandwidth.
Range: 100 Hz to 40 MHz (Standard) / 110 MHz (Option 110).

Examples `SENSE:PULSE:FREFERENCE:CHIRPBW 1.5MHz` sets the chirp bandwidth to 1.5 MHz.

[SENSe]:PULSe:FREFerence:OFFSet

Sets or queries the frequency reference offset. This command is valid when [SENSe]:PULSe:FREFerence:AUTO is set to OFF.

Conditions Measurement views: Pulse statistics, Pulse table, Pulse trace

Group Sense commands

Syntax

```
[SENSe]:PULSe:FREFerence:OFFSet <value>
[SENSe]:PULSe:FREFerence:OFFSet?
```

Arguments `<value>::=<NRf>` specifies the frequency reference offset.
Range: -50 kHz to +50 kHz.

Examples `SENSE:PULSE:FREFERENCE:OFFSET 2.5kHz` sets the frequency offset to 2.5 kHz to the reference.

[SENSe]:PULSe:MODulation:TYPE

Selects or queries the modulation type in the pulsed RF measurements. This command is equivalent to the [SENSe]:PULSe:SIGNal:TYPE command.

Conditions Measurement views: Pulse statistics, Pulse table, Pulse trace

Group Sense commands

Syntax

```
[SENSe]:PULSe:MODulation:TYPE { CWConst | CWCHange | LCHirp
}
[SENSe]:PULSe:MODulation:TYPE?
```

Arguments `CWConst` selects the CW (continuous wave) with constant phase.

`CWCHange` selects the CW (continuous wave) with changing phase.

`LCHirp` selects the linear chirp.

Examples `SENSE:PULSE:MODULATION:TYPE CWCHange` selects the CW with changing phase as the modulation type.

[SENSe]:PULSe:SIGNal:TYPE

Selects or queries the signal type in the pulsed RF measurements.

Conditions Measurement views: Pulse statistics, Pulse table, Pulse trace

Group Sense commands

Syntax

```
[SENSe]:PULSe:SIGNal:TYPE { CWConst | CWCHange | LCHirp }
[SENSe]:PULSe:SIGNal:TYPE?
```

Arguments

`CWConst` selects the CW (continuous wave) with constant phase.

`CWCHange` selects the CW (continuous wave) with changing phase.

`LCHirp` selects the linear chirp.

Examples `SENSE:PULSE:SIGNAL:TYPE CWCHange` selects the CW with changing phase as the signal type.

[SENSe]:REANalyze (No Query Form)

Have all measurements reanalyze the current acquisition record.

NOTE. *It is an overlapped command, which does not finish executing before the next command starts executing. Use the *OPC(?) and *WAI commands to synchronize all pending operations to the execution of this command.*

Conditions Measurement views: All

Group Sense commands

Syntax

```
[SENSe]:REANalyze
```

Related Commands *OPC, *WAI

Arguments None

Examples `SENSE:REANALYZE` have all measurements reanalyze the current acquisition record.

[SENSe]:ROSCillator:SOURce

Selects or queries the frequency reference oscillator source.

Conditions Measurement views: All

Group Sense commands

Syntax
```
[SENSe]:ROSCillator:SOURce { INTernal | EXTernal }
[SENSe]:ROSCillator:SOURce?
```

Arguments `INTernal` specifies that the analyzer uses the 10 MHz internal oscillator.

`EXTernal` specifies that the analyzer uses the external reference signal. Use the Ref In connector on the rear panel to input the signal.

Selecting EXTernal initiates an attempt to lock the internal reference oscillator to the external reference signal. If the signal is not connected or is at an invalid frequency or amplitude, an error (2028, "External frequency reference signal not valid. Using internal reference) is returned. If not able to lock to the external reference, an error (2029, "Unable to lock to external frequency reference. Using internal reference.") is returned.

Examples `SENSE:ROSCILLATOR:SOURCE INTernal` uses the 10 MHz internal oscillator for the frequency reference.

[SENSe]:SGRam:{BANDwidth|BWIDth}:OPTimization

Selects or queries the method of optimizing the gain and input bandwidth in the spectrogram.

Conditions Measurement views: Spectrogram

Group Sense commands

Syntax
```
[SENSe]:SGRam:{BANDwidth|BWIDth}:OPTimization { AUTO |
MAXDynrange | MINNoise | MINTime }
[SENSe]:SGRam:{BANDwidth|BWIDth}:OPTimization?
```

Arguments

AUTO optimizes automatically the gain and input bandwidth.

MAXDynrange optimizes the gain and input bandwidth to maximize the dynamic range.

MINNoise optimizes the gain and input bandwidth to minimize noise.

MINTime optimizes the gain and input bandwidth to minimize sweep time.

Examples

SENSE:SGRAM:BANDWIDTH:OPTIMIZATION AUTO optimizes automatically the gain and input bandwidth.

[SENSe]:SGRam:{BANDwidth|BWIDth}:RESolution

Sets or queries the resolution bandwidth (RBW). Programming a specified RBW sets [SENSe]:SGRam{BANDwidth|BWIDth}:RESolution:AUTO OFF.

Conditions

Measurement views: Spectrogram

Group

Sense commands

Syntax

```
[SENSe]:SGRam:{BANDwidth|BWIDth}:RESolution <value>
[SENSe]:SGRam:{BANDwidth|BWIDth}:RESolution?
```

Related Commands

[SENSe]:SGRam:{BANDwidth|BWIDth}[:RESolution]:AUTO

Arguments

<value>::=<NRf> specifies the RBW. Range: 1 Hz to 10 MHz.

Examples

SENSE:SGRAM:BANDWIDTH:RESOLUTION 200kHz sets the RBW to 200 kHz.

[SENSe]:SGRam:{BANDwidth|BWIDth}[:RESolution]:ACTual? (Query Only)

Queries the actual resolution bandwidth (RBW) in the spectrogram.

Conditions

Measurement views: Spectrogram

Group

Sense commands

Syntax

```
[SENSe]:SGRam:{BANDwidth|BWIDth}[:RESolution]:ACTual?
```

Arguments None

Returns `<NRf>` The actual RBW in Hz.

Examples `SENSE:SGRAM:BANDWIDTH:RESOLUTION:ACTUAL?` might return `299.624E+3`, indicating that the actual RBW is 299.624 kHz.

[SENSe]:SGRam:{BANDwidth|BWIDth}[:RESolution]:AUTO

Determines whether to set the resolution bandwidth (RBW) automatically or manually in the spectrogram.

Conditions Measurement views: Spectrogram

Group Sense commands

Syntax

```
[SENSe]:SGRam:{BANDwidth|BWIDth}[:RESolution]:AUTO { OFF |
ON | 0 | 1 }
[SENSe]:SGRam:{BANDwidth|BWIDth}[:RESolution]:AUTO?
```

Arguments `OFF` or `0` specifies that the resolution bandwidth is set manually using the [SENSe]:DPSA:{BANDwidth|BWIDth}[:RESolution] command.

`ON` or `1` specifies that the resolution bandwidth is set automatically.

Examples `SENSE:SGRAM:BANDWIDTH:RESOLUTION:AUTO ON` sets the resolution bandwidth automatically.

[SENSe]:SGRam:{BANDwidth|BWIDth}[:RESolution]:MODE

Determines whether to enable or disable the RBW processing in the spectrogram.

Conditions Measurement views: Spectrogram

Group Sense commands

Syntax

```
[SENSe]:SGRam:{BANDwidth|BWIDth}[:RESolution]:MODE { OFF |
ON | 0 | 1 }
[SENSe]:SGRam:{BANDwidth|BWIDth}[:RESolution]:MODE?
```

Arguments

`OFF` or `0` disables the RBW processing. You can select the FFT window using the [SENSe]:SGRam:FFT:WINDow command.

`ON` or `1` enables the RBW processing. Refer to the [SENSe]:SGRam:{BANDwidth|BWIDth}:RESolution command to set the RBW.

Examples

`SENSE:SGRAM:BANDWIDTH:RESOLUTION:MODE ON` enables the RBW processing.

[SENSe]:SGRam:{BANDwidth|BWIDth}:VIDeo

Sets or queries the video bandwidth (VBW). Programming a specified VBW sets [SENSe]:SGRam:{BANDwidth|BWIDth}:VIDeo:STATe OFF.

Conditions

Measurement views: Spectrogram

Group

Sense commands

Syntax

```
[SENSe]:SGRam:{BANDwidth|BWIDth}:VIDeo <value>
[SENSe]:SGRam:{BANDwidth|BWIDth}:VIDeo?
```

Related Commands

[SENSe]:SGRam:{BANDwidth|BWIDth}:VIDeo:STATe

Arguments

`<value>::=<NRf>` specifies the VBW.
Range: Current RBW/10^4 (1 Hz minimum) to Current RBW.

Examples

`SENSE:SGRAM:BANDWIDTH:VIDEO 200kHz` sets the VBW to 200 kHz.

[SENSe]:SGRam:{BANDwidth|BWIDth}:VIDeo:STATe

Determines whether to enable or disable the video bandwidth (VBW) in the spectrogram measurement.

Conditions

Measurement views: Spectrogram

Group Sense commands

Syntax
```
[SENSe]:SGRam:{BANDwidth|BWIDth}:VIDeo:STATe { OFF | ON |
0 | 1 }
[SENSe]:SGRam:{BANDwidth|BWIDth}:VIDeo:STATe?
```

Arguments OFF or 0 disables the VBW.

ON or 1 enables the VBW.

Examples SENSE:SGRAM:BANDWIDTH:VIDEO:STATE ON enables the VBW.

[SENSe]:SGRam:COLor

Selects or queries the color palette of three-dimensional graphs.

Conditions Measurement views: Spectrogram

Group Sense commands

Syntax
```
[SENSe]:SGRam:COLor { RED | GREen | BLUe | CYAN | BCYan|
YELLow | MAGenta | GRAY | TEMPerature | SPECtral }
[SENSe]:SGRam:COLor?
```

Arguments The following table lists the arguments.

Table 2-40: Color palette for spectrogram

Argument	Palette
RED	Red
GREen	Green
BLUe	Blue
CYAN	Cyan
BCYan	Binary cyan
YELLow	Yellow
MAGenta	Magenta
GRAY	Gray

Table 2-40: Color palette for spectrogram (cont.)

Argument	Palette
TEMPerature	Temperature
SPECtral	Spectral

Examples `SENSE:SGRAM:COLOR TEMPerature` selects the temperature color palette.

[SENSe]:SGRam:COLor:MAXimum

Sets or queries the maximum value of the color axis in the spectrogram.

Conditions Measurement views: Spectrogram

Group Sense commands

Syntax

```
[SENSe]:SGRam:COLor:MAXimum <value>
[SENSe]:SGRam:COLor:MAXimum?
```

Related Commands [SENSe]:SGRam:COLor:MINimum

Arguments `<value>::=<NRf>` specifies the maximum value of the color axis.
Range: -100 to +100 dBm.

Examples `SENSE:SGRAM:COLOR:MAXIMUM 10` sets the maximum value of the color axis to 10 dBm.

[SENSe]:SGRam:COLor:MINimum

Sets or queries the minimum value of the color axis in the spectrogram.

Conditions Measurement views: Spectrogram

Group Sense commands

Syntax

```
[SENSe]:SGRam:COLor:MINimum <value>
[SENSe]:SGRam:COLor:MINimum?
```

Related Commands [SENSe]:SGRam:COLor:MAXimum

Arguments `<value>::=<NRf>` specifies the minimum value of the color axis.
Range: -100 to +100 dBm.

Examples `SENSE:SGRAM:COLOR:MINIMUM 10` sets the minimum value of the color axis to 10 dBm.

[SENSe]:SGRam:FFT:WINDow

Selects or queries the FFT window in the spectrogram.This command is equivalent to [SENSe]:SGRam:FILTer[:SHAPe].

Conditions Measurement views: Spectrogram

Group Sense commands

Syntax
```
[SENSe]:SGRam:FFT:WINDow { KAISer | MIL6db | CISPr | BH4B |
UNIForm | FLATtop | HANNing }
[SENSe]:SGRam:FFT:WINDow?
```

Arguments `KAISer` selects the Kaiser (RBW) window.

`MIL6db` selects the -6 dB RBW (MIL) window.

`CISPr` selects the CISPR window.

`FLATtop` selects the flat-top window.

`HANNing` selects the Hanning window.

`BH4B` selects the Blackman-Harris 4B type window.

`UNIForm` selects the uniform window.

Examples `SENSE:SGRAM:FFT:WINDOW HANNing` selects the Hanning window.

[SENSe]:SGRam:FILTer[:SHAPe]

Selects or queries the filter shape in the spectrogram. This command is equivalent to [SENSe]:SGRam:FFT:WINDow.

Conditions Measurement views: Spectrogram

Group Sense commands

Syntax
```
[SENSe]:SGRam:FILTer[:SHAPe] { KAISer | MIL6db | CISPr |
BH4B | UNIForm | FLATtop | HANNing }
[SENSe]:SGRam:FILTer[:SHAPe]?
```

Arguments `KAISer` selects the Kaiser (RBW) window.

`MIL6db` selects the -6 dB RBW (MIL) window.

`CISPr` selects the CISPR window.

`FLATtop` selects the flat-top window.

`HANNing` selects the Hanning window.

`BH4B` selects the Blackman-Harris 4B type window.

`UNIForm` selects the uniform window.

Examples `SENSE:SGRAM:FILTER:SHAPE HANNing` selects the Hanning window.

[SENSe]:SGRam:FREQuency:CENTer

Sets or queries the center frequency in the spectrogram.

NOTE. *The center, start and stop frequencies are set interlocking each other with the following relationships: (start frequency) = (center frequency) - (span)/2 and (stop frequency) = (center frequency) + (span)/2.*

Conditions Measurement views: Spectrogram

Group Sense commands

Syntax
```
[SENSe]:SGRam:FREQuency:CENTer <value>
[SENSe]:SGRam:FREQuency:CENTer?
```

Related Commands [SENSe]:SGRam:FREQuency:STARt, [SENSe]:SGRam:FREQuency:STOP

Arguments `<value>::=<NRf>` specifies the center frequency.
Range: 0 Hz to 6.2 GHz (RSA6106A) / 14 GHz (RSA6114A).

Examples `SENSE:SGRAM:FREQUENCY:CENTER 7.5GHz` sets the center frequency to 7.5 GHz.

[SENSe]:SGRam:FREQuency:SPAN

Sets or queries the frequency span in the spectrogram.

Conditions Measurement views: Spectrogram

Group Sense commands

Syntax
```
[SENSe]:SGRam:FREQuency:SPAN <value>
[SENSe]:SGRam:FREQuency:SPAN?
```

Arguments `<value>::=<NRf>` specifies the frequency span.
Range: 10 Hz to 6.2 GHz (RSA6106A) / 14 GHz (RSA6114A).

Examples `SENSE:SGRAM:FREQUENCY:SPAN 20MHz` sets the span to 20 MHz.

[SENSe]:SGRam:FREQuency:SPAN:BANDwidth[:RESolution]:RATio

Sets or queries the ratio of span to RBW (Resolution Bandwidth) in the spectrogram. This command is valid when [SENSe]:SGRam:{BANDwidth|BWIDth}[:RESolution]:AUTO is set to On.

Conditions Measurement views: Spectrogram

Group Sense commands

Syntax
```
[SENSe]:SGRam:FREQuency:SPAN:BANDwidth[:RESolution]:RATio
<value>
[SENSe]:SGRam:FREQuency:SPAN:BANDwidth[:RESolution]:RATio?
```

Arguments <value>::=<NRf> specifies the ratio of span to RBW. Range: 20 to 1000.

Programming a specified ratio sets the RBW (= span/ratio), which is rounded down to the nearest valid value.

Examples SENSE:SGRAM:FREQUENCY:SPAN:BANDWIDTH:RESOLUTION:RATIO 200 sets the ratio to 200, setting the RBW to 200 kHz for the span of 40 MHz.

[SENSe]:SGRam:FREQuency:SPAN:MAXimum (No Query Form)

Sets the frequency range to the maximum real-time span in the spectrogram.

Conditions Measurement views: Spectrogram

Group Sense commands

Syntax

```
[SENSe]:SGRam:FREQuency:SPAN:MAXimum <value>
```

Arguments None

Examples SENSE:SGRAM:FREQUENCY:SPAN:MAXIMUM sets the frequency range to the maximum real-time span.

[SENSe]:SGRam:FREQuency:STARt

Sets or queries the measurement start frequency (left edge of the graph) in the spectrogram.

The center, start and stop frequencies are set interlocking each other. Refer to the [SENSe]:SGRam:FREQuency:CENTer command.

Conditions Measurement views: Spectrogram

Group Sense commands

Syntax

```
[SENSe]:SGRam:FREQuency:STARt <value>
[SENSe]:SGRam:FREQuency:STARt?
```

Related Commands [SENSe]:SGRam:FREQuency:STOP

Arguments

<value>::=<NRf> is the measurement start frequency.
Range: (center frequency) ± (span)/2.

Examples

`SENSE:SGRAM:FREQUENCY:START 6.95GHz` sets the start frequency to 6.95 GHz.

[SENSe]:SGRam:FREQuency:STEP

Sets or queries the frequency step size. Programming a specified step size sets [SENSe]:SGRam:FREQuency:STEP:AUTO OFF.

Conditions

Measurement views: Spectrogram

Group

Sense commands

Syntax

```
[SENSe]:SGRam:FREQuency:STEP <value>
[SENSe]:SGRam:FREQuency:STEP?
```

Related Commands

[SENSe]:SGRam:FREQuency:STEP:AUTO

Arguments

<value>::=<NRf> specifies the frequency step size.
Range: 0 to 6.2 GHz (RSA6106A) / 14 GHz (RSA6114A).

Examples

`SENSE:SGRAM:FREQUENCY:STEP 1kHz` sets the frequency step size to 1 kHz.

[SENSe]:SGRam:FREQuency:STEP:AUTO

Determines whether to set the frequency step size automatically or manually in the spectrogram.

Conditions

Measurement views: Spectrogram

Group

Sense commands

Syntax

```
[SENSe]:SGRam:FREQuency:STEP:AUTO { OFF | ON | 0 | 1 }
[SENSe]:SGRam:FREQuency:STEP:AUTO?
```

Arguments

OFF or 0 specifies that the frequency step size is set manually using the [SENSe]:SGRam:FREQuency:STEP command.

ON or 1 specifies that the frequency step size is set automatically.

Examples

SENSE:SGRAM:FREQUENCY:STEP:AUTO ON specifies that the frequency step size is set automatically.

[SENSe]:SGRam:FREQuency:STOP

Sets or queries the measurement stop frequency (right edge of the graph) in the spectrogram.

The center, start and stop frequencies are set interlocking each other. Refer to the [SENSe]:SGRam:FREQuency:CENTer command.

Conditions

Measurement views: Spectrogram

Group

Sense commands

Syntax

```
[SENSe]:SGRam:FREQuency:STOP <value>
[SENSe]:SGRam:FREQuency:STOP?
```

Related Commands

[SENSe]:SGRam:FREQuency:STARt

Arguments

<value>::=<NRf> is the measurement start frequency.
Range: (center frequency) ± (span)/2.

Examples

SENSE:SGRAM:FREQUENCY:STOP 7.05GHz sets the stop frequency to 7.05 GHz.

[SENSe]:SPECtrum:{BANDwidth|BWIDth}:OPTimization

Selects or queries the method of optimizing the gain and input bandwidth in the spectrum measurement.

Conditions

Measurement views: Spectrum

Group

Sense commands

Syntax

```
[SENSe]:SPECtrum:{BANDwidth|BWIDth}:OPTimization { AUTO |
MAXDynrange | MINNoise | MINTime }
[SENSe]:SPECtrum:{BANDwidth|BWIDth}:OPTimization?
```

Arguments

`AUTO` optimizes automatically the gain and input bandwidth.

`MAXDynrange` optimizes the gain and input bandwidth to maximize the dynamic range.

`MINNoise` optimizes the gain and input bandwidth to minimize noise.

`MINTime` optimizes the gain and input bandwidth to minimize sweep time.

Examples

`SENSE:SPECTRUM:BANDWIDTH:OPTIMIZATION AUTO` optimizes automatically the gain and input bandwidth.

[SENSe]:SPECtrum:{BANDwidth|BWIDth}[:RESolution]

Sets or queries the resolution bandwidth (RBW). Programming a specified RBW sets [SENSe]:SPECtrum:{BANDwidth|BWIDth}[:RESolution]:AUTO OFF.

Conditions

Measurement views: Spectrum

Group

Sense commands

Syntax

```
[SENSe]:SPECtrum:{BANDwidth|BWIDth}[:RESolution] <value>
[SENSe]:SPECtrum:{BANDwidth|BWIDth}[:RESolution]?
```

Related Commands

[SENSe]:SPECtrum:{BANDwidth|BWIDth}[:RESolution]:AUTO

Arguments

`<value>::=<NRf>` specifies the RBW. Range: 1 Hz to 10 MHz.

Examples

`SENSE:SPECTRUM:BANDWIDTH:RESOLUTION 200kHz` sets the RBW to 200 kHz.

[SENSe]:SPECtrum:{BANDwidth|BWIDth}[:RESolution]:ACTual? (Query Only)

Queries the actual resolution bandwidth (RBW) in the spectrum measurement.

Conditions

Measurement views: Spectrum

Group Sense commands

Syntax `[SENSe]:SPECtrum:{BANDwidth|BWIDth}[:RESolution]:ACTual?`

Arguments None

Returns `<NRf>` The actual RBW in Hz.

Examples `SENSE:SPECTRUM:BANDWIDTH:RESOLUTION:ACTUAL?` might return `299.624E+3`, indicating that the actual RBW is 299.624 kHz.

[SENSe]:SPECtrum:{BANDwidth|BWIDth}[:RESolution]:AUTO

Determines whether to set the resolution bandwidth (RBW) automatically or manually in the spectrum measurement.

Conditions Measurement views: Spectrum

Group Sense commands

Syntax
```
[SENSe]:SPECtrum:{BANDwidth|BWIDth}[:RESolution]:AUTO { OFF
| ON | 0 | 1 }
[SENSe]:SPECtrum:{BANDwidth|BWIDth}[:RESolution]:AUTO?
```

Arguments `OFF` or `0` specifies that the resolution bandwidth is set manually using the [SENSe]:SPECtrum:{BANDwidth|BWIDth}[:RESolution] command.

`ON` or `1` specifies that the resolution bandwidth is set automatically.

Examples `SENSE:SPECTRUM:BANDWIDTH:RESOLUTION:AUTO ON` sets the resolution bandwidth automatically.

[SENSe]:SPECtrum:{BANDwidth|BWIDth}[:RESolution]:MODE

Determines whether to enable or disable the RBW process.

Conditions Measurement views: Spectrum

Group Sense commands

Syntax

```
[SENSe]:SPECtrum:{BANDwidth|BWIDth}[:RESolution]:MODE { OFF
| ON | 0 | 1 }
[SENSe]:SPECtrum:{BANDwidth|BWIDth}[:RESolution]:MODE?
```

Arguments `OFF` or `0` disables the RBW process. You can select the FFT window using the [SENSe]:SPECtrum:FFT:WINDow command.

`ON` or `1` enables the RBW process. Refer to the [SENSe]:SPECtrum:{BANDwidth|BWIDth}[:RESolution] command to set the RBW.

Examples `SENSE:SPECTRUM:BANDWIDTH:RESOLUTION:MODE ON` enables the RBW process.

[SENSe]:SPECtrum:{BANDwidth|BWIDth}:VIDeo

Sets or queries the video bandwidth (VBW). Programming a specified VBW sets [SENSe]:SPECtrum{BANDwidth|BWIDth}:VIDeo:STATe OFF.

Conditions Measurement views: Spectrum

Group Sense commands

Syntax

```
[SENSe]:SPECtrum:{BANDwidth|BWIDth}:VIDeo <value>
[SENSe]:SPECtrum:{BANDwidth|BWIDth}:VIDeo?
```

Arguments `<value>::=<NRf>` specifies the VBW.
Range: Current RBW/10^4 (1 Hz minimum) to Current RBW.

Examples `SENSE:SPECTRUM:BANDWIDTH:VIDEO 200kHz` sets the VBW to 200 kHz.

[SENSe]:SPECtrum:{BANDwidth|BWIDth}:VIDeo:STATe

Determines whether to enable or disable the video bandwidth (VBW) in the spectrum measurement.

Conditions Measurement views: Spectrum

Group Sense commands

Syntax
```
[SENSe]:SPECtrum:{BANDwidth|BWIDth}:VIDeo:STATe { OFF | ON
| 0 | 1 }
[SENSe]:SPECtrum:{BANDwidth|BWIDth}:VIDeo:STATe?
```

Arguments OFF or 0 disables the VBW.

ON or 1 enables the VBW.

Examples `SENSE:SPECTRUM:BANDWIDTH:VIDEO:STATE ON` enables the VBW.

[SENSe]:SPECtrum:CLEar:RESults (No Query Form)

Restarts multi-trace functions (Average and Max/Min Hold).

Conditions Measurement views: Spectrum

Group Sense commands

Syntax
```
[SENSe]:SPECtrum:CLEar:RESults
```

Arguments None

Examples `SENSE:SPECTRUM:CLEAR:RESULTS` restarts multi-trace functions.

[SENSe]:SPECtrum:FFT:WINDow

Selects or queries the FFT window in the spectrum measurement.This command is equivalent to [SENSe]:SPECtrum:FILTer[:SHAPe].

Conditions Measurement views: Spectrum

Group Sense commands

Syntax
```
[SENSe]:SPECtrum:FFT:WINDow { KAISer | MIL6db | CISPr | BH4B
| UNIForm | FLATtop | HANNing }
[SENSe]:SPECtrum:FFT:WINDow?
```

Arguments `KAISer` selects the Kaiser (RBW) window.

`MIL6db` selects the -6 dB RBW (MIL) window.

`CISPr` selects the CISPR window.

`BH4B` selects the Blackman-Harris 4B type window.

`UNIForm` selects the uniform window.

`FLATtop` selects the flat-top window.

`HANNing` selects the Hanning window.

Examples `SENSE:SPECTRUM:FFT:WINDOW HANNing` selects the Hanning window.

[SENSe]:SPECtrum:FILTer[:SHAPe]

Selects or queries the filter shape in the spectrum measurement. This command is equivalent to [SENSe]:SPECtrum:FFT:WINDow.

Conditions Measurement views: Spectrum

Group Sense commands

Syntax

```
[SENSe]:SPECtrum:FILTer[:SHAPe] { KAISer | MIL6db | CISPr |
BH4B | UNIForm | FLATtop | HANNing }
[SENSe]:SPECtrum:FILTer[:SHAPe]?
```

Arguments `KAISer` selects the Kaiser (RBW) window.

`MIL6db` selects the -6 dB RBW (MIL) window.

`CISPr` selects the CISPR window.

`BH4B` selects the Blackman-Harris 4B type window.

`UNIForm` selects the uniform window.

`FLATtop` selects the flat-top window.

`HANNing` selects the Hanning window.

Examples `SENSE:SPECTRUM:FILTER:SHAPE HANNing` selects the Hanning window.

[SENSe]:SPECtrum:FREQuency:CENTer

Sets or queries the center frequency in the spectrum measurement.

NOTE. *The center, start and stop frequencies are set interlocking each other with the following relationships: (start frequency) = (center frequency) - (span)/2 and (stop frequency) = (center frequency) + (span)/2.*

Conditions Measurement views: Spectrum

Group Sense commands

Syntax
```
[SENSe]:SPECtrum:FREQuency:CENTer <value>
[SENSe]:SPECtrum:FREQuency:CENTer?
```

Related Commands [SENSe]:SPECtrum:FREQuency:STARt, [SENSe]:SPECtrum:FREQuency:STOP

Arguments `<value>::=<NRf>` specifies the center frequency.
Range: 0 Hz to 6.2 GHz (RSA6106A) / 14 GHz (RSA6114A).

Examples `SENSE:SPECTRUM:FREQUENCY:CENTER 7.5GHz` sets the center frequency to 7.5 GHz.

[SENSe]:SPECtrum:FREQuency:SPAN

Sets or queries the frequency span in the spectrum measurement.

Conditions Measurement views: Spectrum

Group Sense commands

Syntax
```
[SENSe]:SPECtrum:FREQuency:SPAN <value>
[SENSe]:SPECtrum:FREQuency:SPAN?
```

Arguments `<value>::=<NRf>` specifies the frequency span.
Range: 10 Hz to 6.2 GHz (RSA6106A) / 14 GHz (RSA6114A).

Examples `SENSE:SPECTRUM:FREQUENCY:SPAN 20MHz` sets the span to 20 MHz.

[SENSe]:SPECtrum:FREQuency:SPAN:BANDwidth[:RESolution]:RATio

Sets or queries the ratio of span to RBW (Resolution Bandwidth) in the spectrum measurement. This command is valid when [SENSe]:SPECtrum:{BANDwidth|BWIDth}[:RESolution]:AUTO is set to On.

Conditions Measurement views: Spectrum

Group Sense commands

Syntax

```
[SENSe]:SPECtrum:FREQuency:SPAN:BANDwidth[:RESolution]:RATio
<value>
[SENSe]:SPECtrum:FREQuency:SPAN:BANDwidth[:RESolution]:
RATio?
```

Arguments `<value>::=<NRf>` specifies the ratio of span to RBW. Range: 20 to 1000.

Programming a specified ratio sets the RBW (= span/ratio), which is rounded down to the nearest valid value.

Examples `SENSE:SPECTRUM:FREQUENCY:SPAN:BANDWIDTH:RESOLUTION:RATIO 200` sets the ratio to 200, setting the RBW to 200 kHz for the span of 40 MHz.

[SENSe]:SPECtrum:FREQuency:STARt

Sets or queries the measurement start frequency (left edge on the graph) in the spectrum measurement.

The center, start and stop frequencies are set interlocking each other. Refer to the [SENSe]:SPECtrum:FREQuency:CENTer command.

Conditions Measurement views: Spectrum

Group Sense commands

Syntax

```
[SENSe]:SPECtrum:FREQuency:STARt <value>
[SENSe]:SPECtrum:FREQuency:STARt?
```

Related Commands [SENSe]:SPECtrum:FREQuency:STOP

Arguments

<value>::=<NRf> is the measurement start frequency.
Range: (center frequency) ±(span)/2.

Examples

`SENSE:SPECTRUM:FREQUENCY:START 6.95GHz` sets the start frequency to 6.95 GHz.

[SENSe]:SPECtrum:FREQuency:STEP

Sets or queries the frequency step size (the amount per press by which the up or down key changes the setting value). Programming a specified step size sets [SENSe]:SPECtrum:FREQuency:STEP:AUTO OFF.

Conditions

Measurement views: Spectrum

Group

Sense commands

Syntax

```
[SENSe]:SPECtrum:FREQuency:STEP <value>
[SENSe]:SPECtrum:FREQuency:STEP?
```

Related Commands

[SENSe]:SPECtrum:FREQuency:STEP:AUTO

Arguments

<value>::=<NRf> specifies the frequency step size.
Range: 0 Hz to 6.2 GHz (RSA6106A) / 14 GHz (RSA6114A).

Examples

`SENSE:SPECTRUM:FREQUENCY:STEP 1.5kHz` sets the step size to 1.5 kHz.

[SENSe]:SPECtrum:FREQuency:STEP:AUTO

Determines whether to set the frequency step size automatically or manually.

Conditions

Measurement views: Spectrum

Group

Sense commands

Syntax

```
[SENSe]:SPECtrum:FREQuency:STEP:AUTO { OFF | ON | 0 | 1 }
[SENSe]:SPECtrum:FREQuency:STEP:AUTO?
```

Arguments

OFF or 0 specifies that the frequency step size is set manually using the [SENSe]:SPECtrum:FREQuency:STEP command.

ON or 1 specifies that the frequency step size is set automatically.

Examples

SENSE:SPECTRUM:FREQUENCY:STEP:AUTO ON sets the frequency step size automatically.

[SENSe]:SPECtrum:FREQuency:STOP

Sets or queries the measurement stop frequency (right edge on the graph) in the spectrum measurement.

The center, start and stop frequencies are set interlocking each other. Refer to the [SENSe]:SPECtrum:FREQuency:CENTer command.

Conditions

Measurement views: Spectrum

Group

Sense commands

Syntax

```
[SENSe]:SPECtrum:FREQuency:STOP <value>
[SENSe]:SPECtrum:FREQuency:STOP?
```

Related Commands

[SENSe]:SPECtrum:FREQuency:STARt

Arguments

<value>::=<NRf> is the measurement start frequency.
Range: (center frequency) ± (span)/2.

Examples

SENSE:SPECTRUM:FREQUENCY:STOP 7.05GHz sets the stop frequency to 7.05 GHz.

[SENSe]:SPECtrum:LENGth

Sets or queries the spectrum length when [SENSe]:SPECtrum:TIMe:MODe is INDependent. Programming a specified length sets [SENSe]:SPECtrum:LENGth:AUTO OFF.

Conditions

Measurement views: All

Group

Sense commands

Syntax
```
[SENSe]:SPECtrum:LENGth <value>
[SENSe]:SPECtrum:LENGth?
```

Related Commands [SENSe]:SPECtrum:LENGth:AUTO, [SENSe]:SPECtrum:STARt

Arguments `<value>::=<NRf>` specifies the spectrum length.
Range: 0 to [(acquisition length) - (spectrum start)].

If [(spectrum start) + (spectrum length)] > (acquisition length), the actual spectrum length is reduced to the acquisition length.

Examples `SENSE:SPECTRUM:LENGTH 38.5us` sets the spectrum length to 38.5 μs.

[SENSe]:SPECtrum:LENGth:ACTual? (Query Only)

Queries the actual spectrum length.

Conditions Measurement views: All

Group Sense commands

Syntax
```
[SENSe]:SPECtrum:LENGth:ACTual?
```

Arguments None

Returns `<NRf>` Actual spectrum length in seconds.

Examples `SENSE:SPECTRUM:LENGTH:ACTUAL?` might return `6.337E-6`, indicating that the actual spectrum length is 6.337 μs.

[SENSe]:SPECtrum:LENGth:AUTO

Determines whether to set the spectrum length automatically or manually when [SENSe]:SPECtrum:TIMe:MODe is INDependent.

Conditions Measurement views: All

Group Sense commands

Syntax
```
[SENSe]:SPECtrum:LENGth:AUTO { OFF | ON | 0 | 1 }
[SENSe]:SPECtrum:LENGth:AUTO?
```

Arguments OFF or 0 sets the spectrum length manually, using the [SENSe]:SPECtrum:LENGth command.

ON or 1 sets the spectrum length automatically.

Examples SENSE:SPECTRUM:LENGTH:AUTO ON sets the spectrum length automatically.

[SENSe]:SPECtrum:MAX:SPAN (No Query Form)

Sets the frequency span to the maximum span: 6.2 GHz (RSA6106A) / 14 GHz (RSA6114A).

Conditions Measurement views: Spectrum

Group Sense commands

Syntax
```
[SENSe]:SPECtrum:MAX:SPAN
```

Arguments None

Examples SENSE:SPECTRUM:MAX:SPAN sets the frequency span to the maximum span.

[SENSe]:SPECtrum:POINts:COUNt

Sets or queries the number of sample points on the signal spectrum.

Conditions Measurement views: Spectrum

Group Sense commands

Syntax
```
[SENSe]:SPECtrum:POINts:COUNt { P801 | P2401 | P4001 | P8001
| P10401 }
[SENSe]:SPECtrum:POINts:COUNt?
```

Arguments

`P801` sets the number of sample points to 801.

`P2401` sets the number of sample points to 2401.

`P4001` sets the number of sample points to 4001.

`P8001` sets the number of sample points to 8001.

`P10401` sets the number of sample points to 10401.

Examples

`SENSE:SPECTRUM:POINTS:COUNT P801` sets the number of sample points to 801.

[SENSe]:SPECtrum:STARt

Sets or queries the spectrum offset time when [SENSe]:SPECtrum:TIMe:MODe is INDependent.

Conditions

Measurement views: All

Group

Sense commands

Syntax

```
[SENSe]:SPECtrum:STARt <value>
[SENSe]:SPECtrum:STARt?
```

Related Commands

[SENSe]:SPECtrum:LENGth

Arguments

`<value>::=<NRf>` specifies the spectrum offset time.
Range: 0 to the acquisition length.

If [(spectrum start) + (spectrum length)] > (acquisition length), the actual spectrum length is reduced to the acquisition length.

Examples

`SENSE:SPECTRUM:START 23.5us` sets the analysis offset to 23.5 µs.

[SENSe]:SPECtrum:TIMe:MODe

Determines whether to set the spectrum time parameters automatically or manually.

Conditions

Measurement views: All

Group Sense commands

Syntax
```
[SENSe]:SPECtrum:TIMe:MODe { INDependent | COMMon }
[SENSe]:SPECtrum:TIMe:MODe?
```

Related Commands [SENSe]:ANALysis commands

Arguments `INDependent` sets the spectrum time parameters manually, using the [SENSe]:SPECtrum:STARt and [SENSe]:SPECtrum:LENGth commands.

`COMMon` sets the spectrum time parameters automatically using the analysis time settings.

Examples `SENSE:SPECTRUM:TIME:MODE COMMon` sets the spectrum time parameters automatically.

[SENSe]:SPURious:CARRier:{BANDwidth|BWIDth}

Sets or queries the channel bandwidth for the carrier as power reference. This command is valid when [SENSe]:SPURious:REFerence is set to CARRier.

Conditions Measurement views: Spurious

Group Sense commands

Syntax
```
[SENSe]:SPURious:CARRier:{BANDwidth|BWIDth} <value>
[SENSe]:SPURious:CARRier:{BANDwidth|BWIDth}?
```

Arguments `<value>::=<NRf>` specifies the channel bandwidth.
Range: 100 Hz to 40 MHz (Standard) / 110 MHz (Option 110).

Examples `SENSE:SPURIOUS:CARRIER:BANDWIDTH 20MHz` sets the channel bandwidth to 20 MHz.

[SENSe]:SPURious:CARRier:{BANDwidth|BWIDth}:INTegration

Sets or queries the integration bandwidth to calculate the carrier power. This command is valid when [SENSe]:SPURious:REFerence is set to CARRier.

Conditions Measurement views: Spurious

Group Sense commands

Syntax
```
[SENSe]:SPURious:CARRier:{BANDwidth|BWIDth}:INTegration
<value>
[SENSe]:SPURious:CARRier:{BANDwidth|BWIDth}:INTegration?
```

Arguments `<value>::=<NRf>` specifies the integration bandwidth.
Range: 100 Hz to 40 MHz (Standard) / 110 MHz (Option 110).

Examples `SENSE:SPURIOUS:CARRIER:BANDWIDTH:INTEGRATION 2MHz` sets the integration bandwidth to 2 MHz.

[SENSe]:SPURious:CARRier:{BANDwidth|BWIDth}[:RESolution]

Sets or queries the resolution bandwidth (RBW) to measure the carrier power. This command is valid when [SENSe]:SPURious:REFerence is set to CARRier. Programming a specified RBW sets [SENSe]:SPURious:CARRier:BANDwidth [RESolution]:AUTO OFF.

Conditions Measurement views: Spurious

Group Sense commands

Syntax
```
[SENSe]:SPURious:CARRier:{BANDwidth|BWIDth}[:RESolution]
<value>
[SENSe]:SPURious:CARRier:{BANDwidth|BWIDth}[:RESolution]?
```

Related Commands [SENSe]:SPURious:CARRier:{BANDwidth|BWIDth}[:RESolution]:AUTO

Arguments `<value>::=<NRf>` specifies the RBW. Range: 1 Hz to 10 MHz.

Examples `SENSE:SPURIOUS:CARRIER:BANDWIDTH:RESOLUTION 200kHz` sets the RBW to 200 kHz.

[SENSe]:SPURious:CARRier:{BANDwidth|BWIDth}[:RESolution]:AUTO

Determines whether to set the resolution bandwidth (RBW) automatically or manually when the power reference is set to carrier ([SENSe]:SPURious:REFerence is set to CARRier) in the Spurious measurement.

Conditions Measurement views: Spurious

Group Sense commands

Syntax

```
[SENSe]:SPURious:CARRier:{BANDwidth|BWIDth}[:RESolution]:
AUTO { OFF | ON | 0 | 1 }
[SENSe]:SPURious:CARRier:{BANDwidth|BWIDth}[:RESolution]:
AUTO?
```

Arguments `OFF` or `0` specifies that the resolution bandwidth is set manually using the [SENSe]:SPURious:CARRier:{BANDwidth|BWIDth}[:RESolution] command.

`ON` or `1` specifies that the resolution bandwidth is set automatically.

Examples `SENSE:SPURIOUS:CARRIER:BANDWIDTH:RESOLUTION:AUTO ON` sets the resolution bandwidth automatically.

[SENSe]:SPURious:CARRier:DETection

Selects or queries the carrier detection method. This command is valid when [SENSe]:SPURious:REFerence is set to CARRier.

Conditions Measurement views: Spurious

Group Sense commands

Syntax

```
[SENSe]:SPURious:CARRier:DETection { AVERage | PEAK }
[SENSe]:SPURious:CARRier:DETection?
```

Arguments `AVERage` selects the average detection.

`PEAK` selects the peak detection.

Examples `SENSE:SPURIOUS:CARRIER:DETECTION PEAK` selects the peak detection.

[SENSe]:SPURious:CARRier:FREQuency

Sets or queries the carrier frequency in the Spurious measurement. This command is valid when [SENSe]:SPURious:REFerence is set to CARRier.

Conditions Measurement views: Spurious

Group Sense commands

Syntax

```
[SENSe]:SPURious:CARRier:FREQuency <value>
[SENSe]:SPURious:CARRier:FREQuency?
```

Arguments `<value>::=<NRf>` specifies the carrier frequency.
Range: 0 Hz to 6.2 GHz (RSA6106A) / 14 GHz (RSA6114A).

Examples `SENSE:SPURIOUS:CARRIER:FREQUENCY 7.5GHz` sets the carrier frequency to 7.5 GHz.

[SENSe]:SPURious:CARRier:THReshold

Sets or queries the threshold level to detect the carrier in the Spurious measurement. This command is valid when [SENSe]:SPURious:REFerence is set to CARRier.

Conditions Measurement views: Spurious

Group Sense commands

Syntax

```
[SENSe]:SPURious:CARRier:THReshold <value>
[SENSe]:SPURious:CARRier:THReshold?
```

Arguments `<value>::=<NRf>` specifies the threshold level above which the input signal is determined to be a carrier. Range: -170 to +50 dBm. The unit can be changed by the [SENSe]:POWer:UNITs or UNIT:POWer command.

Examples `SENSE:SPURIOUS:CARRIER:THRESHOLD -25` sets the carrier threshold level to -25 dBm.

[SENSe]:SPURious:CLEar:RESults (No Query Form)

Restarts multi-trace functions (Average and Max Hold).

Conditions Measurement views: Spurious

Group Sense commands

Syntax `[SENSe]:SPURious:CLEar:RESults`

Arguments None

Examples `SENSE:SPURIOUS:CLEAR:RESULTS` restarts multi-trace functions.

[SENSe]:SPURious[:FREQuency]:OVERlap? (Query Only)

Queries whether any of the frequency ranges (A to T) overlap, including the carrier when [SENSe]:SPURious:REFerence is set to CARRier.

NOTE. *If there are any overlaps between the ranges, the measurement will not run correctly.*

Conditions Measurement views: Spurious

Group Sense commands

Syntax `[SENSe]:SPURious[:FREQuency]:OVERlap?`

Arguments None

Returns `0` (no overlap) or `1` (overlap).

Examples `SENSE:SPURIOUS:FREQUENCY:OVERLAP?` might return `1`, indicating that some frequency ranges overlap.

[SENSe]:SPURious:LIST

Selects or queries how to list the spurious signals in the Spurious measurement.

Conditions Measurement views: Spurious

Group Sense commands

Syntax
```
[SENSe]:SPURious:LIST { ALL | OVERlimit }
[SENSe]:SPURious:LIST?
```

Arguments `ALL` lists all of the detected spurious signals.

`OVERlimit` lists the spurious signals exceeding the limits. Use the [SENSe]:SPURious:RANGe<x>:LIMit command group to set the limits.

Examples `SENSE:SPURIOUS:LIST ALL` lists all of the detected spurious signals.

[SENSe]:SPURious:MODE

Selects or queries the frequency range mode in the Spurious measurement.

Conditions Measurement views: Spurious

Group Sense commands

Syntax
```
[SENSe]:SPURious:MODE { MULTi | SINGle }
[SENSe]:SPURious:MODE?
```

Arguments `MULTi` displays all of the ranges that are enabled. Use the [SENSe]:SPURious:RANGe<x>:STATe command to enable the range.

`SINGle` displays only the range that the selected spurious signal is in. Use the DISPlay:SPURious:SELect:NUMBer command to select the spurious signal.

Examples `SENSE:SPURIOUS:MODE MULTi` displays all of the enabled ranges.

[SENSe]:SPURious:OPTimization

Selects or queries the method of optimizing the gain and input bandwidth in the Spurious measurement.

Conditions Measurement views: Spurious

Group Sense commands

Syntax

```
[SENSe]:SPURious:OPTimization { AUTO | MAXDynrange |
MINNoise }
[SENSe]:SPURious:OPTimization?
```

Arguments

`AUTO` optimizes automatically the gain and input bandwidth.

`MAXDynrange` optimizes the gain and input bandwidth to maximize the dynamic range.

`MINNoise` optimizes the gain and input bandwidth to minimize noise.

Examples

`SENSE:SPURIOUS:OPTIMIZATION AUTO` optimizes automatically the gain and input bandwidth.

[SENSe]:SPURious:POINts:COUNt

Sets or queries the number of sample points on the spectrum trace per range in the Spurious measurement.

Conditions Measurement views: Spurious

Group Sense commands

Syntax

```
[SENSe]:SPURious:POINts:COUNt { P801 | P2401 | P4001 | P8001
| P10401 }
[SENSe]:SPURious:POINts:COUNt?
```

Arguments

`P801` sets the trace points to 801 per range.

`P2401` sets the trace points to 2401 per range.

`P4001` sets the trace points to 4001 per range.

`P8001` sets the trace points to 8001 per range.

P10401 sets the trace points to 10401 per range.

Examples SENSE:SPURIOUS:POINTS:COUNT P801 sets the trace points to 801 per range.

[SENSe]:SPURious:RANGe<x>:BANDwidth:VIDeo

Sets or queries the video bandwidth (VBW) in the specified frequency range. Programming a specified VBW sets [SENSe]:SPURious:BANDwidth:VIDeo STATe OFF.

The parameter <x> = 1 to 20, representing Range A to T, respectively.

Conditions Measurement views: Spurious

Group Sense commands

Syntax

```
[SENSe]:SPURious:RANGe<x>:BANDwidth:VIDeo <value>
[SENSe]:SPURious:RANGe<x>:BANDwidth:VIDeo?
```

Related Commands [SENSe]:SPECtrum:{BANDwidth|BWIDth}:VIDeo:STATe

Arguments <value>::=<NRf> specifies the VBW.
Range: Current RBW/10^4 (1 Hz minimum) to Current RBW.

Examples SENSE:SPURIOUS:RANGE1:BANDWIDTH:VIDEO 200kHz sets the VBW to 200 kHz for Range A.

[SENSe]:SPURious:RANGe<x>:BANDwidth:VIDeo:STATe

Determines whether to enable or disable the video bandwidth (VBW) in the specified frequency range.

The parameter <x> = 1 to 20, representing Range A to T, respectively.

Conditions Measurement views: Spurious

Group Sense commands

Syntax

```
[SENSe]:SPURious:RANGe<x>:BANDwidth:VIDeo:STATe { OFF | ON
| 0 | 1 }
[SENSe]:SPURious:RANGe<x>:BANDwidth:VIDeo:STATe?
```

Arguments

`OFF` or `0` disables the VBW.

`ON` or `1` enables the VBW.

Examples

`SENSE:SPURIOUS:RANGE1:BANDWIDTH:VIDEO:STATE ON` enables the VBW for Range A.

[SENSe]:SPURious:RANGe<x>:DETection

Selects or queries the spurious detection method in the specified frequency range.

The parameter <x> = 1 to 20, representing Range A to T, respectively.

Conditions

Measurement views: Spurious

Group

Sense commands

Syntax

```
[SENSe]:SPURious:RANGe<x>:DETection { AVERage | PEAK |
QUASipeak | CAVerage | CPEak }
[SENSe]:SPURious:RANGe<x>:DETection?
```

Arguments

`AVERage` selects the average detection.

`PEAK` selects the peak detection.

`QUASipeak` selects the quasi-peak detection.

`CAVerage` selects the CISPR average detection.

`CPEak` selects the CISPR peak detection.

Examples

`SENSE:SPURIOUS:RANGE1:DETECTION PEAK` selects the peak detection for Range A.

[SENSe]:SPURious:RANGe<x>:EXCursion

Sets or queries the excursion level (how far down the signal must drop between spurious emissions) in the specified frequency range.

The parameter <x> = 1 to 20, representing Range A to T, respectively.

Conditions Measurement views: Spurious

Group Sense commands

Syntax

```
[SENSe]:SPURious:RANGe<x>:EXCursion <value>
[SENSe]:SPURious:RANGe<x>:EXCursion?
```

Arguments `<value>::=<NRf>` specifies the excursion level. A signal with amplitude less than the excursion level is considered to be a noise. Range: 1 to 50 dB.

Examples `SENSE:SPURIOUS:RANGE1:EXCURSION 8` sets the excursion level to 8 dB.

[SENSe]:SPURious:RANGe<x>:FILTer[:SHAPe]

Selects or queries the filter shape to search the specified frequency range for spurious signals.

The parameter <x> = 1 to 20, representing Range A to T, respectively.

Conditions Measurement views: Spurious

Group Sense commands

Syntax

```
[SENSe]:SPURious:RANGe<x>:FILTer[:SHAPe] { RBW | MIL6db |
CISPr }
[SENSe]:SPURious:RANGe<x>:FILTer[:SHAPe]?
```

Arguments `RBW` selects the RBW filter.

`MIL6db` selects the -6 dB RBW (MIL) filter.

`CISPr` selects the CISPR filter.

Examples `SENSE:SPURIOUS:RANGE1:FILTER:SHAPE CISPr` uses the CISPR filter in Range A.

[SENSe]:SPURious:RANGe<x>:FILTer[:SHAPe]:BANDwidth

Sets or queries the filter bandwidth to search the specified frequency range for spurious signals.

The parameter <x> = 1 to 20, representing Range A to T, respectively.

Conditions Measurement views: Spurious

Group Sense commands

Syntax
```
[SENSe]:SPURious:RANGe<x>:FILTer[:SHAPe]:BANDwidth <value>
[SENSe]:SPURious:RANGe<x>:FILTer[:SHAPe]:BANDwidth?
```

Arguments `<value>::=<NRf>` specifies the filter bandwidth.

The setting range varies depending on the center frequency and trace detector when using the CISPR filters and detectors. Otherwise, the bandwidth minimum is a function of the span (= (stop frequency) - (start frequency)) of the range and the maximum is fixed at 5 MHz.

By default, the value is set automatically ([SENSe]:SPURious:RANGe<x>:FILTer[:SHAPe]:BANDwidth:AUTO is ON). When the detection is set to CISPR QPk ([SENSe]:SPURious:RANGe<x>:DETection is QUASipeak), only one value is allowed, which is set automatically.

Examples `SENSE:SPURIOUS:RANGE1:FILTER:SHAPE:BANDWIDTH 200kHz` sets the filter bandwidth to 200 kHz for Range A.

[SENSe]:SPURious:RANGe<x>:FILTer[:SHAPe]:BANDwidth:AUTO

Determines whether to set the filter bandwidth automatically or manually for the specified frequency range in the Spurious measurement.

The parameter <x> = 1 to 20, representing Range A to T, respectively.

Conditions Measurement views: Spurious

Group Sense commands

Syntax
```
[SENSe]:SPURious:RANGe<x>:FILTer[:SHAPe]:BANDwidth:AUTO {
OFF | ON | 0 | 1 }
[SENSe]:SPURious:RANGe<x>:FILTer[:SHAPe]:BANDwidth:AUTO?
```

Arguments `OFF` or `0` specifies that the filter bandwidth is set manually using the [SENSe]:SPURious:RANGe<x>:FILTer[:SHAPe]:BANDwidth command.

`ON` or `1` specifies that the filter bandwidth is set automatically.

Examples `SENSE:SPURIOUS:RANGE1:FILTER:SHAPE:BANDWIDTH:AUTO ON` sets the filter bandwidth automatically for Range A.

[SENSe]:SPURious:RANGe<x>:FREQuency:STARt

Sets or queries the start frequency of the specified range in the Spurious measurement.

The parameter <x> = 1 to 20, representing Range A to T, respectively.

NOTE. *The frequency ranges must not be overlapped. Use the [SENSe]:SPURious[:FREQuency]:OVERlap? query to check whether there is any overlap.*

Conditions Measurement views: Spurious

Group Sense commands

Syntax

```
[SENSe]:SPURious:RANGe<x>:FREQuency:STARt <value>
[SENSe]:SPURious:RANGe<x>:FREQuency:STARt?
```

Arguments `<value>::=<NRf>` specifies the start frequency.
Range: 0 Hz to 6.2 GHz (RSA6106A) / 14 GHz (RSA6114A).

Examples `SENSE:SPURIOUS:RANGE1:FREQUENCY:START 1.615GHz` sets the start frequency of Range A to 1.615 GHz.

[SENSe]:SPURious:RANGe<x>:FREQuency:STOP

Sets or queries the stop frequency of the specified range in the Spurious measurement.

The parameter <x> = 1 to 20, representing Range A to T, respectively.

NOTE. *The frequency ranges must not be overlapped. Use the [SENSe]:SPURious[:FREQuency]:OVERlap? query to check whether there is any overlap.*

Conditions Measurement views: Spurious

Group Sense commands

Syntax
```
[SENSe]:SPURious:RANGe<x>:FREQuency:STOP <value>
[SENSe]:SPURious:RANGe<x>:FREQuency:STOP?
```

Arguments `<value>::=<NRf>` specifies the stop frequency.
Range: 0 Hz to 6.2 GHz (RSA6106A) / 14 GHz (RSA6114A).

Examples `SENSE:SPURIOUS:RANGE1:FREQUENCY:STOP 1.715GHz` sets the stop frequency of Range A to 1.715 GHz.

[SENSe]:SPURious:RANGe<x>:LIMit:ABSolute:STARt

Sets or queries the absolute amplitude of the limits at the start (left edge) of the specified range in the Spurious measurement. This command is valid when [SENSe]:SPURious:RANGe<x>:LIMit:MASK is set to ABS, AND, or OR.

The parameter <x> = 1 to 20, representing Range A to T, respectively.

Conditions Measurement views: Spurious

Group Sense commands

Syntax
```
[SENSe]:SPURious:RANGe<x>:LIMit:ABSolute:START <value>
[SENSe]:SPURious:RANGe<x>:LIMit:ABSolute:STARt?
```

Arguments `<value>::=<NRf>` specifies the absolute start amplitude of the limits.
Range: -170 to +50 dBm.
The unit can be changed by the [SENSe]:POWer:UNITs or UNIT:POWer command.

Examples `SENSE:SPURIOUS:RANGE1:LIMIT:ABSOLUTE:START -30` sets the absolute start amplitude of the limits for Range A to -30 dBm.

[SENSe]:SPURious:RANGe<x>:LIMit:ABSolute:STOP

Sets or queries the absolute amplitude of the limits at the stop (right edge) of the specified range in the Spurious measurement. This command is valid when [SENSe]:SPURious:RANGe<x>:LIMit:MASK is set to ABS, AND, or OR.

The parameter <x> = 1 to 20, representing Range A to T, respectively.

Conditions Measurement views: Spurious

Group Sense commands

Syntax

```
[SENSe]:SPURious:RANGe<x>:LIMit:ABSolute:STOP <value>
[SENSe]:SPURious:RANGe<x>:LIMit:ABSolute:STOP?
```

Related Commands [SENSe]:POWer:UNITs, UNIT:POWer

Arguments `<value>::=<NRf>` specifies the absolute stop amplitude of the limits.
Range: -170 to +50 dBm.
The unit can be changed by the [SENSe]:POWer:UNITs or UNIT:POWer command.

Examples `SENSE:SPURIOUS:RANGE1:LIMIT:ABSOLUTE:STOP -10` sets the absolute stop amplitude of the limits for Range A to -10 dBm.

[SENSe]:SPURious:RANGe<x>:LIMit:MASK

Selects or queries the limit mask function mode for the specified range in the Spurious measurement.

The parameter <x> = 1 to 20, representing Range A to T, respectively.

Conditions Measurement views: Spurious

Group Sense commands

Syntax

```
[SENSe]:SPURious:RANGe<x>:LIMit:MASK { ABS | REL | AND | OR
| OFF }
[SENSe]:SPURious:RANGe<x>:LIMit:MASK?
```

Arguments The following table lists the arguments.

Table 2-41: Limit mask mode

Argument	Description
ABS	Failure is detected when one of the spurious signals is larger than the absolute amplitude limit.
REL	Failure is detected when one of the spurious signals is larger than the relative amplitude limit.
AND	Failure is detected when one of the spurious signals is larger than the absolute AND relative amplitude limits.
OR	Failure is detected when one of the spurious signals is larger than the absolute OR relative amplitude limit.
OFF	Disables the mask.

To set the absolute amplitude limits, use the [SENSe]:SPURious:RANGe<x>:LIMit:ABSolute:STARt and [SENSe]:SPURious:RANGe<x>:LIMit:ABSolute:STOP commands.

To set the relative amplitude limits, use the [SENSe]:SPURious:RANGe<x>:LIMit:RELative:STARt and [SENSe]:SPURious:RANGe<x>:LIMit:RELative:STOP commands.

Examples `SENSE:SPURIOUS:RANGE1:LIMIT:MASK ABS` specifies that failure is detected when one of the spurious signals is larger than the absolute amplitude limit in Range A.

[SENSe]:SPURious:RANGe<x>:LIMit:RELative:STARt

Sets or queries the relative amplitude of the limits at the start (left edge) of the specified range in the Spurious measurement. This command is valid when [SENSe]:SPURious:RANGe<x>:LIMit:MASK is set to REL, AND, or OR.

The parameter <x> = 1 to 20, representing Range A to T, respectively.

Conditions Measurement views: Spurious

Group Sense commands

Syntax

```
[SENSe]:SPURious:RANGe<x>:LIMit:RELative:STARt <value>
[SENSe]:SPURious:RANGe<x>:LIMit:RELative:STARt?
```

Arguments `<value>::=<NRf>` specifies the relative start amplitude of the limits. Range: -100 to 0 dB.

Use the [SENSe]:SPURious:REFerence command to select the power reference.

Examples

`SENSE:SPURIOUS:RANGE1:LIMIT:RELATIVE:START -30` sets the relative start amplitude of the limits for Range A to -30 dB.

[SENSe]:SPURious:RANGe<x>:LIMit:RELative:STOP

Sets or queries the relative amplitude of the limits at the stop (right edge) of the specified range in the Spurious measurement. This command is valid when [SENSe]:SPURious:RANGe<x>:LIMit:MASK is set to REL, AND, or OR.

The parameter <x> = 1 to 20, representing Range A to T, respectively.

Conditions

Measurement views: Spurious

Group

Sense commands

Syntax

```
[SENSe]:SPURious:RANGe<x>:LIMit:RELative:STOP <value>
[SENSe]:SPURious:RANGe<x>:LIMit:RELative:STOP?
```

Arguments

`<value>::=<NRf>` specifies the relative stop amplitude of the limits.
Range: -100 to 0 dB.

Use the [SENSe]:SPURious:REFerence command to select the power reference.

Examples

`SENSE:SPURIOUS:RANGE1:LIMIT:RELATIVE:STOP -10` sets the relative stop amplitude of the limits for Range A to -10 dB.

[SENSe]:SPURious:RANGe<x>:STATe

Determines whether to enable or disable the frequency range in the Spurious measurement.

The parameter <x> = 1 to 20, representing Range A to T, respectively.

Conditions

Measurement views: Spurious

Group

Sense commands

Syntax
```
[SENSe]:SPURious:RANGe<x>:STATe { OFF | ON | 0 | 1 }
[SENSe]:SPURious:RANGe<x>:STATe?
```

Arguments
OFF or 0 disables the frequency range.

ON or 1 enables the frequency range.

Examples
SENSE:SPURIOUS:RANGE1:STATE ON enables Range A.

[SENSe]:SPURious:RANGe<x>:THReshold

Sets or queries the threshold level to detect spurious signals in the specified frequency range.

The parameter <x> = 1 to 20, representing Range A to T, respectively.

Conditions
Measurement views: Spurious

Group
Sense commands

Syntax
```
[SENSe]:SPURious:RANGe<x>:THReshold <value>
[SENSe]:SPURious:RANGe<x>:THReshold?
```

Arguments
<value>::=<NRf> specifies the threshold level above which the signal is determined to be spurious. Range: -50 to +30 dBm.

The unit can be changed by the [SENSe]:POWer:UNITs or UNIT:POWer command.

Examples
SENSE:SPURIOUS:RANGE1:THRESHOLD -25 sets the threshold level to -25 dBm in Range A.

[SENSe]:SPURious:REFerence

Selects or queries the power reference in the Spurious measurement.

Conditions
Measurement views: Spurious

Group
Sense commands

Syntax

```
[SENSe]:SPURious:REFerence { CARRier | MANual | NONE }
[SENSe]:SPURious:REFerence?
```

Arguments

`CARRier` uses the carrier as the power reference.

`MANual` sets the power reference using the [SENSe]:SPURious:REFerence:MANual:POWer command.

`NONE` uses no reference.

Examples

`SENSE:SPURIOUS:REFERENCE CARRier` uses the carrier as the power reference.

[SENSe]:SPURious:REFerence:MANual:POWer

Sets or queries the reference power level in the Spurious measurement. This command is valid when [SENSe]:SPURious:REFerence is set to MANual.

Conditions

Measurement views: Spurious

Group

Sense commands

Syntax

```
[SENSe]:SPURious:REFerence:MANual:POWer <value>
[SENSe]:SPURious:REFerence:MANual:POWer?
```

Arguments

`<value>::=<NRf>` specifies the reference power level. Range: -170 to +50 dBm. The unit can be changed by the [SENSe]:POWer:UNITs or UNIT:POWer command.

Examples

`SENSE:SPURIOUS:REFERENCE:MANUAL:POWER -25` sets the reference power level to -25 dBm.

[SENSe]:TOVerview:FREQuency:CENTer

Sets or queries the center frequency in the time overview.

Conditions

Measurement views: Time overview

Group

Sense commands

Syntax

```
[SENSe]:TOVerview:FREQuency:CENTer <value>
[SENSe]:TOVerview:FREQuency:CENTer?
```

Arguments

`<value>::=<NRf>` specifies the center frequency.
Range: 0 Hz to 6.2 GHz (RSA6106A) / 14 GHz (RSA6114A).

Examples

`SENSE:TOVERVIEW:FREQUENCY:CENTER 7.5GHz` sets the center frequency to 7.5 GHz.

[SENSe]:TOVerview:MAXTracepoints

Selects or queries the maximum trace points in the time overview.

Conditions

Measurement views: Time overview

Group

Sense commands

Syntax

```
[SENSe]:TOVerview:MAXTracepoints { ONEK | TENK | HUNDredk
| NEVerdecimate }
[SENSe]:TOVerview:MAXTracepoints?
```

Arguments

`ONEK` sets the maximum trace points to 1 k.

`TENK` sets the maximum trace points to 10 k.

`HUNDredk` sets the maximum trace points to 100 k.

`NEVerdecimate` never decimates the trace points.

Examples

`SENSE:TOVERVIEW:MAXTRACEPOINTS TENK` sets the maximum trace points to 10 k.

[SENSe]:USETtings (No Query Form)

Updates the analyzer settings. This command is useful when you need to set the analyzer including the RF attenuation before taking data acquisition. Unless this command is executed, the attenuation value is not set until acquisition is taken.

Conditions

Measurement views: All

Group Sense commands

Syntax `[SENSe]:USETtings`

Arguments None

Examples `SENSE:USETTINGS` updates settings.

*SRE

Sets or queries the value of the Service Request Enable Register (SRER). Refer to Section 3, *Status and Events*, for the register information.

Conditions Measurement views: All

Group IEEE common commands

Syntax
```
*SRE <value>
*SRE?
```

Related Commands *CLS, *ESE, *ESR?, *STB?

Arguments `<value>::=<NR1>` is a value in the range from 0 to 255.
The binary bits of the SRER are set according to this value.
Using an out-of-range value causes an execution error.

Examples `*SRE 48` sets binary 00110000 in the SRER's bits.

`*SRE?` might return `32`, indicating that binary value 00100000 has been set in the SRER's bits.

STATus:ACPower:EVENts? (Query Only)

Returns the current events for the Channel power and ACPR measurement.

Conditions Measurement views: Channel power and ACPR

Group Status commands

Syntax `STATus:ACPower:EVENts?`

Arguments None

Returns `<ecode>,"<edesc>[<einfo>]"{,<ecode>,"<edesc>[:<einfo>]"}`

Where
`<ecode>::=<NR1>` is the error/event code (-32768 to 32767).
`<edesc>::=<string>` is the description on the error/event.
`<einfo>::=<string>` is the additional information on the error/event.

If there is no error, the response is 0,"No error".

Examples `STATUS:ACPOWER:EVENTS?` might return `2026,"Acq Sampling Params: manual control"`, indicating that the sampling parameters are controlled manually.

STATus:AVTime:EVENts? (Query Only)

Returns the current events for the Amplitude versus Time measurement.

Conditions Measurement views: Amplitude versus Time

Group Status commands

Syntax `STATus:AVTime:EVENts?`

Arguments None

Returns `<ecode>,"<edesc>[<einfo>]"{,<ecode>,"<edesc>[:<einfo>]"}`

Where
`<ecode>::=<NR1>` is the error/event code (-32768 to 32767).
`<edesc>::=<string>` is the description on the error/event.
`<einfo>::=<string>` is the additional information on the error/event.

If there is no error, the response is 0,"No error".

Examples `STATUS:AVTIME:EVENTS?` might return `2026,"Acq Sampling Params: manual control"`, indicating that the sampling parameters are controlled manually.

STATus:CCDF:EVENts? (Query Only)

Returns the current events for the CCDF measurement.

Conditions Measurement views: CCDF

Group Status commands

Syntax

```
STATus:CCDF:EVENts?
```

Arguments None

Returns

```
<ecode>,"<edesc>[<einfo>]"{,<ecode>,"<edesc>[:<einfo>]"}
```

Where
<ecode>::=<NR1> is the error/event code (-32768 to 32767).
<edesc>::=<string> is the description on the error/event.
<einfo>::=<string> is the additional information on the error/event.

If there is no error, the response is 0,"No error".

Examples STATUS:CCDF:EVENTS? might return 2026,"Acq Sampling Params: manual control", indicating that the sampling parameters are controlled manually.

STATus:CONSte:EVENts? (Query Only)

Returns the current events for the constellation measurement.

Conditions Measurement views: Constellation

Group Status commands

Syntax

```
STATus:CONSte:EVENts?
```

Arguments None

Returns

```
<ecode>,"<edesc>[<einfo>]"{,<ecode>,"<edesc>[:<einfo>]"}
```

Where
`<ecode>::=<NR1>` is the error/event code (-32768 to 32767).
`<edesc>::=<string>` is the description on the error/event.
`<einfo>::=<string>` is the additional information on the error/event.

If there is no error, the response is 0,"No error".

Examples `STATUS:CONSTE:EVENTS?` might return `2026,"Acq Sampling Params: manual control"`, indicating that the sampling parameters are controlled manually.

STATus:DPSA:EVENts? (Query Only)

Returns the current events for the DPX spectrum measurement.

Conditions Measurement views: DPX spectrum

Group Status commands

Syntax `STATus:DPSA:EVENts?`

Arguments None

Returns `<ecode>,"<edesc>[<einfo>]"{,<ecode>,"<edesc>[:<einfo>]"}`

Where
`<ecode>::=<NR1>` is the error/event code (-32768 to 32767).
`<edesc>::=<string>` is the description on the error/event.
`<einfo>::=<string>` is the additional information on the error/event.

If there is no error, the response is 0,"No error".

Examples `STATUS:DPSA:EVENTS?` might return `2026,"Acq Sampling Params: manual control"`, indicating that the sampling parameters are controlled manually.

STATus:EVM:EVENts? (Query Only)

Returns the current events for the EVM versus Time measurement.

Conditions Measurement views: EVM versus Time

Group Status commands

Syntax `STATus:EVM:EVENts?`

Arguments None

Returns `<ecode>,"<edesc>[<einfo>]"{,<ecode>,"<edesc>[:<einfo>]"}`

Where
`<ecode>::=<NR1>` is the error/event code (-32768 to 32767).
`<edesc>::=<string>` is the description on the error/event.
`<einfo>::=<string>` is the additional information on the error/event.

If there is no error, the response is 0,"No error".

Examples `STATUS:EVM:EVENTS?` might return `2026,"Acq Sampling Params: manual control"`, indicating that the sampling parameters are controlled manually.

STATus:FVTime:EVENts? (Query Only)

Returns the current events for the Frequency versus Time measurement.

Conditions Measurement views: Frequency versus Time

Group Status commands

Syntax `STATus:FVTime:EVENts?`

Arguments None

Returns `<ecode>,"<edesc>[<einfo>]"{,<ecode>,"<edesc>[:<einfo>]"}`

Where
`<ecode>::=<NR1>` is the error/event code (-32768 to 32767).
`<edesc>::=<string>` is the description on the error/event.
`<einfo>::=<string>` is the additional information on the error/event.

If there is no error, the response is 0,"No error".

Examples `STATUS:FVTIME:EVENTS?` might return `2026,"Acq Sampling Params: manual control"`, indicating that the sampling parameters are controlled manually.

STATus:IQVTime:EVENts? (Query Only)

Returns the current events for the RF I&Q versus Time measurement.

Conditions Measurement views: RF I&Q versus Time

Group Status commands

Syntax `STATus:IQVTime:EVENts?`

Arguments None

Returns `<ecode>,"<edesc>[<einfo>]"{,<ecode>,"<edesc>[:<einfo>]"}`

Where
`<ecode>::=<NR1>` is the error/event code (-32768 to 32767).
`<edesc>::=<string>` is the description on the error/event.
`<einfo>::=<string>` is the additional information on the error/event.

If there is no error, the response is 0,"No error".

Examples `STATUS:IQVTIME:EVENTS?` might return `2026,"Acq Sampling Params: manual control"`, indicating that the sampling parameters are controlled manually.

STATus:MCPower:EVENts? (Query Only)

Returns the current events for the MCPR (multi-carrier ACPR) measurement.

Conditions Measurement views: MCPR

Group Status commands

Syntax `STATus:MCPower:EVENts?`

Arguments None

Returns `<ecode>,"<edesc>[<einfo>]"{,<ecode>,"<edesc>[:<einfo>]"}`

Where
`<ecode>::=<NR1>` is the error/event code (-32768 to 32767).
`<edesc>::=<string>` is the description on the error/event.
`<einfo>::=<string>` is the additional information on the error/event.

If there is no error, the response is 0,"No error".

Examples `STATUS:MCPOWER:EVENTS?` might return `2026,"Acq Sampling Params: manual control"`, indicating that the sampling parameters are controlled manually.

STATus:MERRor:EVENts? (Query Only)

Returns the current events for the Magnitude error versus Time measurement.

Conditions Measurement views: Magnitude error versus Time

Group Status commands

Syntax `STATus:MERRor:EVENts?`

Arguments None

Returns `<ecode>,"<edesc>[<einfo>]"{,<ecode>,"<edesc>[:<einfo>]"}`

Where
`<ecode>::=<NR1>` is the error/event code (-32768 to 32767).
`<edesc>::=<string>` is the description on the error/event.
`<einfo>::=<string>` is the additional information on the error/event.

If there is no error, the response is 0,"No error".

Examples `STATUS:MERROR:EVENTS?` might return `2026,"Acq Sampling Params: manual control"`, indicating that the sampling parameters are controlled manually.

STATus:OBWidth:EVENts? (Query Only)

Returns the current events for the Occupied Bandwidth (OBW) measurement.

Conditions Measurement views: Occupied Bandwidth

Group Status commands

Syntax `STATus:OBWidth:EVENts?`

Arguments None

Returns <ecode>,"<edesc>[<einfo>]"{,<ecode>,"<edesc>[:<einfo>]"}

Where
`<ecode>::=<NR1>` is the error/event code (-32768 to 32767).
`<edesc>::=<string>` is the description on the error/event.
`<einfo>::=<string>` is the additional information on the error/event.

If there is no error, the response is 0,"No error".

Examples `STATUS:OBWIDTH:EVENTS?` might return `2026,"Acq Sampling Params: manual control"`, indicating that the sampling parameters are controlled manually.

STATus:OPERation:CONDition? (Query Only)

Returns the contents of the Operation Condition Register (OCR).

Conditions Measurement views: All

Group Status commands

Syntax `STATus:OPERation:CONDition?`

Arguments None

Returns `<NR1>` is a decimal number showing the contents of the OCR.

Examples STATUS:OPERATION:CONDITION? might return 16, showing that the bits in the OCR have the binary value 0000000000010000, which means the analyzer is in measurement.

STATus:OPERation:ENABle

Sets or queries the enable mask of the Operation Enable Register (OENR) which allows true conditions in the Operation Event Register to be reported in the summary bit.

Conditions Measurement views: All

Group Status commands

Syntax
```
STATus:OPERation:ENABle <bit_value>
STATus:OPERation:ENABle?
```

Arguments <bit_value>::=<NR1> is the enable mask of the OENR. Range: 0 to 65535.

Returns <NR1> is a decimal number showing the contents of the OENR.
Range: 0 to 32767 (The most-significant bit cannot be set true.)

Examples STATUS:OPERATION:ENABLE 1 enables the ALIGning bit.

STATUS:OPERATION:ENABLE? might return 1, showing that the bits in the OENR have the binary value 00000000 00000001, which means that the ALIGning bit is valid.

STATus:OPERation[:EVENt]? (Query Only)

Returns the contents of the Operation Event Register (OEVR).
Reading the OEVR clears it.

Conditions Measurement views: All

Group Status commands

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

Arguments None

Returns `<NR1>` is a decimal number showing the contents of the OEVR.

Examples `STATUS:OPERATION:EVENT?` might return `1`, showing that the bits in the OEVR have the binary value 00000000 00000001, which means that the ALIGning bit is set.

STATus:OPERation:NTRansition

Sets or queries the negative transition filter value of the Operation Transition Register (OTR).

Conditions Measurement views: All

Group Status commands

Syntax

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

Arguments `<bit_value>::=<NR1>` is the negative transition filter value. Range: 0 to 65535.

Returns `<NR1>` is a decimal number showing the contents of the OTR.
Range: 0 to 32767 (The most-significant bit cannot be set true.)

Examples `STATUS:OPERATION:NTRANSITION #H0011` sets the negative transition filter value to #H0011.

`STATUS:OPERATION:NTRANSITION?` might return `17`.

STATus:OPERation:PTRansition

Sets or queries the positive transition filter value of the Operation Transition Register (OTR).

Conditions Measurement views: All

Group Status commands

Syntax

STATus:OPERation:PTRansition <bit_value>
STATus:OPERation:PTRansition?

Arguments

<bit_value>::=<NR1> is the positive transition filter value. Range: 0 to 65535.

Returns

<NR1> is a decimal number showing the contents of the OTR.
Range: 0 to 32767 (The most-significant bit cannot be set true.)

Examples

STATUS:OPERATION:PTRANSITION 0 sets the positive transition filter value to 0.

STATUS:OPERATION:PTRANSITION? might return 0.

STATus:PERRor:EVENts? (Query Only)

Returns the current events for the Phase error versus Time measurement.

Conditions

Measurement views: Phase error versus Time

Group

Status commands

Syntax

STATus:PERRor:EVENts?

Arguments

None

Returns

<ecode>,"<edesc>[<einfo>]"{,<ecode>,"<edesc>[:<einfo>]"}

Where
<ecode>::=<NR1> is the error/event code (-32768 to 32767).
<edesc>::=<string> is the description on the error/event.
<einfo>::=<string> is the additional information on the error/event.

If there is no error, the response is 0,"No error".

Examples

STATUS:PERROR:EVENTS? might return 2026,"Acq Sampling Params: manual control", indicating that the sampling parameters are controlled manually.

STATus:PHVTime:EVENts? (Query Only)

Returns the current events for the Phase versus Time measurement.

Conditions Measurement views: Phase versus Time

Group Status commands

Syntax `STATus:PHVTime:EVENts?`

Arguments None

Returns `<ecode>,"<edesc>[<einfo>]"{,<ecode>,"<edesc>[:<einfo>]"}`

Where
`<ecode>::=<NR1>` is the error/event code (-32768 to 32767).
`<edesc>::=<string>` is the description on the error/event.
`<einfo>::=<string>` is the additional information on the error/event.

If there is no error, the response is 0,"No error".

Examples `STATUS:PHVTIME:EVENTS?` might return `2026,"Acq Sampling Params: manual control"`, indicating that the sampling parameters are controlled manually.

STATus:PNOise:EVENts? (Query Only)

Returns the current events for the phase noise measurement.

Conditions Measurement views: Spurious

Group Status commands

Syntax `STATus:PNOise:EVENts?`

Arguments None

Returns `<ecode>,"<edesc>[<einfo>]"{,<ecode>,"<edesc>[:<einfo>]"}`

Where
`<ecode>::=<NR1>` is the error/event code (-32768 to 32767).
`<edesc>::=<string>` is the description on the error/event.
`<einfo>::=<string>` is the additional information on the error/event.

If there is no error, the response is 0,"No error".

Examples

`STATUS:PNOISE:EVENTS?` might return `2026,"Acq Sampling Params: manual control"`, indicating that the sampling parameters are controlled manually.

STATus:PRESet (No Query Form)

Presets the SCPI enable registers and transition registers.

Conditions

Measurement views: All

Group

Status commands

Syntax

`STATus:PRESet`

Arguments

None

Examples

`STATUS:PRESET` presets the SCPI enable registers and transition registers.

STATus:PULSe:RESult:EVENts? (Query Only)

Returns the current events for the pulse table measurement.

Conditions

Measurement views: Pulse table

Group

Status commands

Syntax

`STATus:PULSe:RESult:EVENts?`

Arguments

None

Returns `<ecode>,"<edesc>[<einfo>]"{,<ecode>,"<edesc>[:<einfo>]"}`

Where
`<ecode>::=<NR1>` is the error/event code (-32768 to 32767).
`<edesc>::=<string>` is the description on the error/event.
`<einfo>::=<string>` is the additional information on the error/event.

If there is no error, the response is 0,"No error".

Examples `STATUS:PULSE:RESULT:EVENTS?` might return `2026,"Acq Sampling Params:  manual control"`, indicating that the sampling parameters are controlled manually.

STATus:PULSe:STATistics:EVENts? (Query Only)

Returns the current events for the pulse statistics measurement.

Conditions Measurement views: Pulse statistics

Group Status commands

Syntax `STATus:PULSe:STATistics:EVENts?`

Arguments None

Returns `<ecode>,"<edesc>[<einfo>]"{,<ecode>,"<edesc>[:<einfo>]"}`

Where
`<ecode>::=<NR1>` is the error/event code (-32768 to 32767).
`<edesc>::=<string>` is the description on the error/event.
`<einfo>::=<string>` is the additional information on the error/event.

If there is no error, the response is 0,"No error".

Examples `STATUS:PULSE:STATISTICS:EVENTS?` might return `2026,"Acq Sampling Params:  manual control"`, indicating that the sampling parameters are controlled manually.

STATus:PULSe:TRACe:EVENts? (Query Only)

Returns the current events for the pulse trace measurement.

Conditions Measurement views: Pulse trace

Group Status commands

Syntax `STATus:PULSe:TRACe:EVENts?`

Arguments None

Returns `<ecode>,"<edesc>[<einfo>]"{,<ecode>,"<edesc>[:<einfo>]"}`

Where
`<ecode>::=<NR1>` is the error/event code (-32768 to 32767).
`<edesc>::=<string>` is the description on the error/event.
`<einfo>::=<string>` is the additional information on the error/event.

If there is no error, the response is 0,"No error".

Examples `STATUS:PULSE:TRACE:EVENTS?` might return `2026,"Acq Sampling Params:  manual control"`, indicating that the sampling parameters are controlled manually.

STATus:QUEStionable:CALibration:CONDition? (Query Only)

Returns the contents of the questionable calibration condition register.

Conditions Measurement views: All

Group Status commands

Syntax `STATus:QUEStionable:CALibration:CONDition?`

Arguments None

Returns `<NR1>` is a decimal number showing the contents of the questionable calibration condition register.

Examples `STATUS:QUESTIONABLE:CALIBRATION:CONDITION?` might return `16384`, showing that the bits in the questionable calibration condition register have the binary value 01000000 00000000, which means the Alignment Needed bit is set.

STATus:QUEStionable:CALibration:ENABle

Sets or queries the enable mask of the questionable calibration enable register which allows true conditions in the questionable calibration event register to be reported in the summary bit.

Conditions Measurement views: All

Group Status commands

Syntax

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

Arguments `<bit_value>::=<NR1>` is the enable mask of the questionable calibration enable register. Range: 0 to 65535.

Returns `<NR1>` is a decimal number showing the contents of the questionable calibration enable register. Range: 0 to 32767 (The most-significant bit cannot be set true.)

Examples `STATUS:QUESTIONABLE:CALIBRATION:ENABLE 16384` enables the Alignment Needed bit.

`STATUS:QUESTIONABLE:CALIBRATION:ENABLE?` might return `16384`, showing that the bits in the questionable calibration enable register have the binary value 01000000 00000000, which means that the Calibration Summary bit is valid.

STATus:QUEStionable:CALibration[:EVENt]? (Query Only)

Returns the contents of the questionable calibration event register.
Reading the register clears it.

Conditions Measurement views: All

Group Status commands

Syntax

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

Arguments None

Returns `<NR1>` is a decimal number showing the contents of the questionable calibration event register.

Examples `STATUS:QUESTIONABLE:CALIBRATION:EVENT?` might return `16384`, showing that the bits in the questionable calibration event register have the binary value 01000000 00000000, which means that the Calibration Summary bit is set.

STATus:QUEStionable:CALibration:NTRansition

Sets or queries the negative transition filter value of the questionable calibration transition register.

Conditions Measurement views: All

Group Status commands

Syntax

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

Arguments `<bit_value>::=<NR1>` is the negative transition filter value. Range: 0 to 65535.

Returns `<NR1>` is a decimal number showing the contents of the questionable calibration transition register. Range: 0 to 32767 (The most-significant bit cannot be set true.)

Examples `STATUS:QUESTIONABLE:CALIBRATION:NTRANSITION #H4000` sets the negative transition filter value to #H4000.

`STATUS:QUESTIONABLE:CALIBRATION:NTRANSITION?` might return `16384`.

STATus:QUEStionable:CALibration:PTRansition

Sets or queries the positive transition filter value of the questionable calibration transition register.

Conditions Measurement views: All

Group Status commands

Syntax
```
STATus:QUEStionable:CALibration:PTRansition <bit_value>
STATus:QUEStionable:CALibration:PTRansition?
```

Arguments `<bit_value>::=<NR1>` is the positive transition filter value. Range: 0 to 65535.

Returns `<NR1>` is a decimal number showing the contents of the questionable calibration transition register. Range: 0 to 32767 (The most-significant bit cannot be set true.)

Examples `STATUS:QUESTIONABLE:CALIBRATION:PTRANSITION 0` sets the positive transition filter value to 0.

`STATUS:QUESTIONABLE:CALIBRATION:PTRANSITION?` might return `0`.

STATus:QUEStionable:CONDition? (Query Only)

Returns the contents of the Questionable Condition Register (QCR).

Conditions Measurement views: All

Group Status commands

Syntax `STATus:QUEStionable:CONDition?`

Arguments None

Returns `<NR1>` is a decimal number showing the contents of the QCR.

Examples `STATUS:QUESTIONABLE:CONDITION?` might return `256`, showing that the bits in the QCR have the binary value 00000001 00000000, which means the Calibration Summary bit is set.

STATus:QUEStionable:ENABle

Sets or queries the enable mask of the Questionable Enable Register (QENR) which allows true conditions in the Questionable Event Register to be reported in the summary bit.

Conditions Measurement views: All

Group Status commands

Syntax
```
STATus:QUEStionable:ENABle <bit_value>
STATus:QUEStionable:ENABle?
```

Arguments `<bit_value>::=<NR1>` is the enable mask of QENR. Range: 0 to 65535.

Returns `<NR1>` is a decimal number showing the contents of the QENR.
Range: 0 to 32767 (The most-significant bit cannot be set true.)

Examples `STATUS:QUESTIONABLE:ENABLE 256` enables the Calibration Summary bit.

`STATUS:QUESTIONABLE:ENABLE?` might return `256`, showing that the bits in the QENR have the binary value 00000001 00000000, which means that the Calibration Summary bit is valid.

STATus:QUEStionable[:EVENt]? (Query Only)

Returns the contents of the Questionable Event Register (QEVR).
Reading the QEVR clears it.

Conditions Measurement views: All

Group Status commands

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

Arguments None

Returns `<NR1>` is a decimal number showing the contents of the QEVR.

Examples `STATUS:QUESTIONABLE:EVENT?` might return `256`, showing that the bits in the QEVR have the binary value 00000001 00000000, which means that the Calibration Summary bit is set.

STATus:QUEStionable:FREQuency:CONDition? (Query Only)

Returns the contents of the questionable frequency condition register.

Conditions Measurement views: All

Group Status commands

Syntax `STATus:QUEStionable:FREQuency:CONDition?`

Arguments None

Returns `<NR1>` is a decimal number showing the contents of the questionable frequency condition register.

Examples `STATUS:QUESTIONABLE:FREQUENCY:CONDITION?` might return `512`, showing that the bits in the questionable frequency condition register have the binary value 00000010 00000000, which means the Locked To External Ref bit is set.

STATus:QUEStionable:FREQuency:ENABle

Sets or queries the enable mask of the questionable frequency enable register which allows true conditions in the questionable frequency event register to be reported in the summary bit.

Conditions Measurement views: All

Group Status commands

Syntax

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

Arguments `<bit_value>::=<NR1>` is the enable mask of the questionable frequency enable register. Range: 0 to 65535.

Returns `<NR1>` is a decimal number showing the contents of the questionable frequency enable register. Range: 0 to 32767 (The most-significant bit cannot be set true.)

Examples `STATUS:QUESTIONABLE:FREQUENCY:ENABLE 512` enables the Locked To External Ref bit.

STATUS:QUESTIONABLE:FREQUENCY:ENABLE? might return 512, showing that the bits in the questionable calibration enable register have the binary value 00000010 00000000, which means that the Locked To External Ref bit is valid.

STATus:QUEStionable:FREQuency[:EVENt]? (Query Only)

Returns the contents of the questionable frequency event register. Reading the register clears it.

Conditions Measurement views: All

Group Status commands

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

Arguments None

Returns <NR1> is a decimal number showing the contents of the questionable frequency event register.

Examples STATUS:QUESTIONABLE:FREQUENCY:EVENT? might return 512, showing that the bits in the questionable frequency event register have the binary value 00000010 00000000, which means that the Locked To External Ref bit is set.

STATus:QUEStionable:FREQuency:NTRansition

Sets or queries the negative transition filter value of the questionable frequency transition register.

Conditions Measurement views: All

Group Status commands

Syntax

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

Arguments <bit_value>::=<NR1> is the negative transition filter value. Range: 0 to 65535.

Returns <NR1> is a decimal number showing the contents of the questionable frequency transition register. Range: 0 to 32767 (The most-significant bit cannot be set true.)

Examples STATUS:QUESTIONABLE:FREQUENCY:NTRANSITION #H0200 sets the negative transition filter value to #H0200.

STATUS:QUESTIONABLE:FREQUENCY:NTRANSITION? might return 512.

STATus:QUEStionable:FREQuency:PTRansition

Sets or queries the positive transition filter value of the questionable frequency transition register.

Conditions Measurement views: All

Group Status commands

Syntax
```
STATus:QUEStionable:FREQuency:PTRansition <bit_value>
STATus:QUEStionable:FREQuency:PTRansition?
```

Arguments <bit_value>::=<NR1> is the positive transition filter value. Range: 0 to 65535.

Returns <NR1> is a decimal number showing the contents of the questionable frequency transition register. Range: 0 to 32767 (The most-significant bit cannot be set true.)

Examples STATUS:QUESTIONABLE:FREQUENCY:PTRANSITION 0 sets the positive transition filter value to 0.

STATUS:QUESTIONABLE:FREQUENCY:PTRANSITION? might return 0.

STATus:QUEStionable:NTRansition

Sets or queries the negative transition filter value of the Questionable Transition Register (QTR).

Conditions Measurement views: All

Group Status commands

Syntax

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

Arguments `<bit_value>::=<NR1>` is the negative transition filter value. Range: 0 to 65535.

Returns `<NR1>` is a decimal number showing the contents of the QTR.
Range: 0 to 32767 (The most-significant bit cannot be set true.)

Examples `STATUS:QUESTIONABLE:NTRANSITION #H0020` sets the negative transition filter value to #H0020.

`STATUS:QUESTIONABLE:NTRANSITION?` might return `32`.

STATus:QUEStionable:PTRansition

Sets or queries the positive transition filter value of the Questionable Transition Register (QTR).

Conditions Measurement views: All

Group Status commands

Syntax

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

Arguments `<bit_value>::=<NR1>` is the positive transition filter value. Range: 0 to 65535.

Returns `<NR1>` is a decimal number showing the contents of the QTR.
Range: 0 to 32767 (The most-significant bit cannot be set true.)

Examples `STATUS:QUESTIONABLE:PTRANSITION 0` sets the positive transition filter value to 0.

`STATUS:QUESTIONABLE:PTRANSITION?` might return `0`.

STATus:SGRAM:EVENts? (Query Only)

Returns the current events for the spectrogram measurement.

Conditions Measurement views: Spectrogram

Group Status commands

Syntax `STATus:SGRAM:EVENts?`

Arguments None

Returns `<ecode>,"<edesc>[<einfo>]"{,<ecode>,"<edesc>[:<einfo>]"}`

Where
`<ecode>::=<NR1>` is the error/event code (-32768 to 32767).
`<edesc>::=<string>` is the description on the error/event.
`<einfo>::=<string>` is the additional information on the error/event.

If there is no error, the response is 0,"No error".

Examples `STATUS:SGRAM:EVENTS?` might return `2026,"Acq Sampling Params: manual control"`, indicating that the sampling parameters are controlled manually.

STATus:SPECtrum:EVENts? (Query Only)

Returns the current events for the spectrum measurement.

Conditions Measurement views: Spectrum

Group Status commands

Syntax `STATus:SPECtrum:EVENts?`

Arguments None

Returns `<ecode>,"<edesc>[<einfo>]"{,<ecode>,"<edesc>[:<einfo>]"}`

Where
`<ecode>::=<NR1>` is the error/event code (-32768 to 32767).
`<edesc>::=<string>` is the description on the error/event.
`<einfo>::=<string>` is the additional information on the error/event.

If there is no error, the response is 0,"No error".

Examples `STATUS:SPECTRUM:EVENTS?` might return `2026,"Acq Sampling Params: manual control"`, indicating that the sampling parameters are controlled manually.

STATus:SPURious:EVENts? (Query Only)

Returns the current events for the Spurious measurement.

Conditions Measurement views: Spurious

Group Status commands

Syntax `STATus:SPURious:EVENts?`

Arguments None

Returns `<ecode>,"<edesc>[<einfo>]"{,<ecode>,"<edesc>[:<einfo>]"}`

Where
`<ecode>::=<NR1>` is the error/event code (-32768 to 32767).
`<edesc>::=<string>` is the description on the error/event.
`<einfo>::=<string>` is the additional information on the error/event.

If there is no error, the response is 0,"No error".

Examples `STATUS:SPURIOUS:EVENTS?` might return `2026,"Acq Sampling Params: manual control"`, indicating that the sampling parameters are controlled manually.

STATus:SQUality:EVENts? (Query Only)

Returns the current events for the signal quality measurement.

Conditions Measurement views: Signal quality

Group Status commands

Syntax STATus:SQuality:EVENts?

Arguments None

Returns <ecode>,"<edesc>[<einfo>]"{,<ecode>,"<edesc>[:<einfo>]"}

Where
<ecode>::=<NR1> is the error/event code (-32768 to 32767).
<edesc>::=<string> is the description on the error/event.
<einfo>::=<string> is the additional information on the error/event.

If there is no error, the response is 0,"No error".

Examples STATUS:SQUALITY:EVENTS? might return 2026,"Acq Sampling Params: manual control", indicating that the sampling parameters are controlled manually.

*STB? (Query Only)

Returns the contents of the Status Byte Register (SBR) in the status/event reporting structure using the Master Summary Status (MSS) bit. Refer to Section3, *Status and Events*, for the register information.

Conditions Measurement views: All

Group IEEE common commands

Syntax *STB?

Related Commands *CLS, *ESE, *ESR?, *SRE

Arguments None

Returns <NR1> representing the contents of the SBR as a decimal number.

Examples *STB? might return 96, indicating that the SBR contains binary 0110 0000.

SYSTem:COMMunicate:GPIB[:SELF]:ADDRess

Sets or queries the GPIB address of the instrument.

Conditions Measurement views: All

Group System commands

Syntax

```
SYSTem:COMMunicate:GPIB[:SELF]:ADDRess <value>
SYSTem:COMMunicate:GPIB[:SELF]:ADDRess?
```

Arguments `<value>::=<NR1>` specifies the GPIB address of the instrument.
*RST has no effect on the value.

Examples `SYSTEM:COMMUNICATE:GPIB:SELF:ADDRESS 18` sets the GPIB address to 18.

SYSTem:DATE

Sets or queries the date (year, month, and day). This command is equivalent to the date setting through the Windows Control Panel.

Conditions Measurement views: All

Group System commands

Syntax

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

Related Commands SYSTem:TIME

Arguments `<year>::=<NRf>` specifies the year (4 digits). Range: 2000 to 2099.
`<month>::=<NRf>` specifies the month. Range: 1 (January) to 12 (December).
`<day>::=<NRf>` specifies the day. Range: 1 to 31.
These values are rounded to the nearest integer.

*RST has no effect on the settings.

Examples `SYSTEM:DATE 2008,3,19` sets the internal calendar to March 19, 2008.

SYSTem:ERRor:ALL? (Query Only)

Queries the error/event queue for all the unread items and removes them from the queue. The response is a comma separated list of number, string pairs in FIFO order. For details of the error messages, refer to (See Table 3-8.)

Conditions Measurement views: All

Group System commands

Syntax `SYSTem:ERRor:ALL?`

Arguments None

Returns `<ecode>,"<edesc>[;<einfo>]"{,<ecode>,"<edesc>[;<einfo>]"}`

Where
`<ecode>::=<NR1>` is the error/event code (-32768 to 32767).
`<edesc>::=<string>` is the description on the error/event.
`<einfo>::=<string>` is the detail of the error/event.

If the queue is empty, the response is 0, "No error; Queue empty - No events to report".

Examples `SYSTEM:ERROR:ALL?` might return `-130, "Suffix error; Unrecognized suffix, INPutMLEVel -10dB",` , indicating that the unit of the reference level is improper.

SYSTem:ERRor:CODE:ALL? (Query Only)

Queries the error/event queue for all the unread items and removes them from the queue. The response returns a comma separated list of only the error/event code numbers in FIFO order. For details of the error messages, refer to (See Table 3-8.)

Conditions Measurement views: All

Group System commands

Syntax `SYSTem:ERRor:CODE:ALL?`

Arguments None

Returns `<ecode>{,<ecode>}`

Where
`<ecode>::=<NR1>` is the error/event code, ranging from -32768 to 32767.

If the queue is empty, the response is 0.

Examples `SYSTEM:ERROR:CODE:ALL?` might return `-101,-108` of the error codes.

SYSTem:ERRor:CODE[:NEXT]? (Query Only)

Queries the error/event queue for the next item and removes it from the queue. The response returns only the error/event code number omitting the string. Except for the shortened response, the query operates identically to SYSTem:ERRor[:NEXT]?. For details of the error messages, refer to (See Table 3-8.)

Conditions Measurement views: All

Group System commands

Syntax `SYSTem:ERRor:CODE[:NEXT]?`

Arguments None

Returns `<ecode>::=<NR1>` is the error/event code, ranging from -32768 to 32767.

Examples `SYSTEM:ERROR:CODE:NEXT?` might return `-101` of the error code.

SYSTem:ERRor:COUNt? (Query Only)

Queries the error/event queue for the number of unread items. As errors and events may occur at any time, more items may be present in the queue at the time it is actually read.

Conditions Measurement views: All

Group System commands

Syntax `SYSTem:ERRor:COUNt?`

Arguments None

Returns `<enum>::=<NR1>` is the number of errors/events.

If the queue is empty, the response is 0.

Examples `SYSTEM:ERROR:COUNT?` might return `2`, indicating that the error/event queue contains two of unread errors/events.

SYSTem:ERRor[:NEXT]? (Query Only)

Queries the error/event queue for the next item and removes it from the queue. The response returns the full queue item consisting of an integer and a string. For details of the error messages, refer to (See Table 3-8.)

Conditions Measurement views: All

Group System commands

Syntax `SYSTem:ERRor[:NEXT]?`

Arguments None

Returns `<ecode>,"<edesc>[;<einfo>]"`

Where
`<ecode>::=<NR1>` is the error/event code, ranging from -32768 to 32767.
`<edesc>::=<string>` is the description on the error/event.
`<einfo>::=<string>` is the additional information on the error/event.

Examples `SYSTEM:ERROR:NEXT?` might return `-130, "Suffix error; Unrecognized suffix, INPutMLEVel -10dB"`, indicating that the unit is improper.

SYSTem:OPTions? (Query Only)

Queries the options installed in the analyzer. This command is equivalent to the IEEE common command *OPT?.

Conditions Measurement views: All

Group System commands

Syntax `SYSTem:OPTions?`

Arguments None

Returns `<option>::=<string>` contains the comma-separated option numbers.

Examples `SYSTEM:OPTIONS?` might return `"01,02,20"`, indicating that Option 01, 02, and 20 are currently installed in the analyzer.

SYSTem:PRESet (No Query Form)

Restores the analyzer to the defaults. This command is equivalent to the **Preset** key on the front panel.

Conditions Measurement views: All

Group System commands

Syntax `SYSTem:PRESet`

Arguments None

Examples `SYSTEM:PRESET` restores the analyzer to the defaults.

SYSTem:TIME

Sets or queries the time (hours, minutes, and seconds). This command is equivalent to the time setting through the Windows Control Panel.

Conditions Measurement views: All

Group System commands

Syntax

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

Related Commands SYSTem:DATE

Arguments <hour>::=<NRf> specifies the hours. Range: 0 to 23.
<minute>::=<NRf> specifies the minutes. Range: 0 to 59.
<second>::=<NRf> specifies the seconds. Range: 0 to 59.
These values are rounded to the nearest integer.

*RST has no effect on the settings.

Examples `SYSTEM:TIME 10,15,30` sets the time to 1015:30.

SYSTem:VERSion? (Query Only)

Returns the SCPI version number for which the analyzer complies.

Conditions Measurement views: All

Group System commands

Syntax

```
SYSTem:VERSion?
```

Arguments None

Returns <NR2> has the form YYYY.V where the Ys represent the year-version (for example, 1999) and the V represents an approved revision number for that year.

Examples `SYSTEM:VERSION?` might return `1999.0` for the SCPI version.

TRACe:CONSte:MODE

Selects or queries how to display the constellation trace.

Conditions Measurement views: Constellation

Group Trace commands

Syntax

```
TRACe:CONSte:MODE { VECTors | SYMBols }
TRACe:CONSte:MODE?
```

Arguments `VECTors` connects adjacent symbol points with the signal locus.

`SYMBols` displays individual symbol points.

Examples `TRACE:CONSTE:MODE VECTors` shows the constellation connecting adjacent symbol points with the signal locus.

TRACe:FVTime

Determines whether or not to show the trace in the Frequency versus Time view.

Conditions Measurement views: Frequency versus Time

Group Trace commands

Syntax

```
TRACe:FVTime { OFF | ON | 0 | 1 }
TRACe:FVTime?
```

Arguments `OFF` or `0` hides the trace in the Frequency versus Time view.

`ON` or `1` shows the trace in the Frequency versus Time view.

Examples `TRACE:FVTIME ON` shows the trace in the Frequency versus Time view.

TRACe:FVTime:AVERage:COUNt

Sets or queries the number of traces to combine. This command is effective when you select AVERage with the TRACe:FVTime:FUNCtion command.

Conditions Measurement views: Frequency versus Time

Group Trace commands

Syntax
```
TRACe:FVTime:AVERage:COUNt <number>
TRACe:FVTime:AVERage:COUNt?
```

Arguments `<number>::=<NR1>` specifies the number of traces to combine for averaging.
Range: 1 to 10000.

Examples `TRACE:FVTIME:AVERAGE:COUNT 64` sets the average count to 64.

TRACe:FVTime:COUNt

Sets or queries how many acquisitions run in the single acquisition mode for the Max or Min Hold trace in the Frequency versus Time measurement. This command is effective when TRACe:FVTime:FUNCtion is set to MAXHold or MINHold and INITiate:CONTinuous is set to OFF.

Conditions Measurement views: Frequency versus Time

Group Trace commands

Syntax
```
TRACe:FVTime:COUNt <number>
TRACe:FVTime:COUNt?
```

Arguments `<number>::=<NR1>` specifies the count for Max/Min Hold.
Range: 1 to 10000.

Examples `TRACe:FVTime:COUNt 32` sets the count to 32 for the Max/Min Hold trace.

TRACe:FVTime:COUNt:ENABle

Determines whether to enable or disable the count for the Max or Min Hold trace in the Frequency versus Time view. This command is effective when TRACe:FVTime:FUNCtion is set to MAXHold or MINHold.

Conditions Measurement views: Frequency versus Time

Group Trace commands

Syntax
```
TRACe:FVTime:COUNt:ENABle { OFF | ON | 0 | 1 }
TRACe:FVTime:COUNt:ENABle?
```

Arguments `OFF` or `0` disables the count for the Max/Min Hold trace.

`ON` or `1` enables the count for the Max/Min Hold trace.

Examples `TRACe:FVTime:COUNt:ENABle ON` enables the Max/Min Hold count.

TRACe:FVTime:COUNt:RESet (No Query Form)

Clears the Max or Min Hold data and counter, and restarts the process in the Frequency versus Time view. This command is effective when TRACe:FVTime:FUNCtion is set to MAXHold or MINHold.

Conditions Measurement views: Frequency versus Time

Group Trace commands

Syntax
```
TRACe:FVTime:COUNt:RESet
```

Arguments None

Examples `TRACe:FVTime:COUNt:RESet` clears the Max/Min Hold data and counter, and restarts the process.

TRACe:FVTime:FREeze

Determines whether or not to freeze the display of the trace in the Frequency versus Time measurement.

Conditions Measurement views: Frequency versus Time

Group Trace commands

Syntax
```
TRACe:FVTime:FREeze { OFF | ON | 0 | 1 }
TRACe:FVTime:FREeze?
```

Arguments

`OFF` or `0` updates the display of the trace normally.

`ON` or `1` stops updating the display of the trace.

Examples

`TRACE:FVTIME:FREEZE ON` stops updating the display of the trace.

TRACe:FVTime:FUNCtion

Selects or queries the trace function in the Frequency versus Time measurement.

Conditions

Measurement views: Frequency versus Time

Group

Trace commands

Syntax
```
TRACe:FVTime:FUNCtion { NORMal | AVERage | MAXHold | MINHold
}
TRACe:FVTime:FUNCtion?
```

Arguments

`NORMal` selects the normal waveform display.

`AVERage` selects the Average display that indicates the average frequency drift at each time point.

`MAXHold` selects the Max Hold display that indicates the maximum frequency drift at each time point.

`MINHold` selects the Min Hold display that indicates the minimum frequency drift at each time point.

Examples

`TRACE:FVTIME:FUNCTION MAXHold` displays the Max Hold trace in the Frequency versus Time measurement.

TRACe:IQVTime:AVERage:COUNt

Sets or queries the number of traces to combine. This command works for both I and Q traces when you select AVERage in the TRACe:IQVTime:FUNCtion command.

Conditions

Measurement views: RF I&Q versus Time

Group Trace commands

Syntax

```
TRACe:IQVTime:AVERage:COUNt <number>
TRACe:IQVTime:AVERage:COUNt?
```

Arguments `<number>::=<NR1>` specifies the number of traces to combine for averaging.
Range: 1 to 10000.

Examples `TRACE:IQVTIME:AVERAGE:COUNT 64` sets the average count to 64.

TRACe:IQVTime:COUNt

Sets or queries how many acquisitions run in the single acquisition mode for the Max or Min Hold trace in the RF I&Q versus Time measurement. This command is effective when TRACe:IQVTime:FUNCtion is set to MAXHold or MINHold and INITiate:CONTinuous is set to OFF.

Conditions Measurement views: RF I&Q versus Time

Group Trace commands

Syntax

```
TRACe:IQVTime:COUNt <number>
TRACe:IQVTime:COUNt?
```

Arguments `<number>::=<NR1>` specifies the count for Max/Min Hold.
Range: 1 to 10000.

Examples `TRACE:IQVTIME:COUNT 32` sets the count to 32 for the Max/Min Hold trace.

TRACe:IQVTime:COUNt:ENABle

Determines whether to enable or disable the count for the Max or Min Hold trace in the RF I&Q versus Time view. This command is effective when TRACe:IQVTime:FUNCtion is set to MAXHold or MINHold.

Conditions Measurement views: RF I&Q versus Time

Group Trace commands

Syntax
```
TRACe:IQVTime:COUNt:ENABle { OFF | ON | 0 | 1 }
TRACe:IQVTime:COUNt:ENABle?
```

Arguments `OFF` or `0` disables the count for the Max/Min Hold trace.

`ON` or `1` enables the count for the Max/Min Hold trace.

Examples `TRACE:IQVTIME:COUNT:ENABLE ON` enables the Max/Min Hold count.

TRACe:IQVTime:COUNt:RESet (No Query Form)

Clears the Max or Min Hold data and counter, and restarts the process in the RF I&Q versus Time view. This command is effective when TRACe:IQVTime:FUNCtion is set to MAXHold or MINHold.

Conditions Measurement views: RF I&Q versus Time

Group Trace commands

Syntax `TRACe:IQVTime:COUNt:RESet`

Arguments None

Examples `TRACE:IQVTIME:COUNT:RESET` clears the Max/Min Hold data and counter, and restarts the process.

TRACe:IQVTime:ENABle:I

Determines whether to show or hide the I trace in the RF I&Q versus Time measurement.

Conditions Measurement views: RF I&Q versus Time

Group Trace commands

Syntax
```
TRACe:IQVTime:ENABle:I { OFF | ON | 0 | 1 }
TRACe:IQVTime:ENABle:I?
```

Arguments `OFF` or `0` hides the I trace.

`ON` or `1` shows the I trace.

Examples `TRACE:IQVTIME:ENABLE:I ON` shows the I trace in the RF I&Q versus Time measurement.

TRACe:IQVTime:ENABle:Q

Determines whether to show or hide the Q trace in the RF I&Q versus Time measurement.

Conditions Measurement views: RF I&Q versus Time

Group Trace commands

Syntax
```
TRACe:IQVTime:ENABle:Q { OFF | ON | 0 | 1 }
TRACe:IQVTime:ENABle:Q?
```

Arguments `OFF` or `0` hides the Q trace.

`ON` or `1` shows the Q trace.

Examples `TRACE:IQVTIME:ENABLE:Q ON` shows the Q trace in the IQ level versus Time measurement.

TRACe:IQVTime:FREeze

Determines whether to freeze the IQ traces in the RF I&Q versus Time measurement.

Conditions Measurement views: RF I&Q versus Time

Group Trace commands

Syntax
```
TRACe:IQVTime:FREeze { OFF | ON | 0 | 1 }
TRACe:IQVTime:FREeze?
```

Arguments

`OFF` or `0` updates IQ trace display normally.

`ON` or `1` stops updating IQ trace display.

Examples

`TRACE:IQVTIME:FREEZE ON` freezes the IQ traces.

TRACe:IQVTime:FUNCtion

Selects or queries the trace function in the RF I&Q versus Time measurement.

Conditions

Measurement views: RF I&Q versus Time

Group

Trace commands

Syntax

```
TRACe:IQVTime:FUNCtion { NORMal | AVERage | MAXHold |
MINHold }
TRACe:IQVTime:FUNCtion?
```

Arguments

`NORMal` selects the normal waveform display.

`AVERage` selects the Average display that indicates the average signal level at each time point.

`MAXHold` selects the Max Hold display that indicates the maximum signal level at each time point.

`MINHold` selects the Min Hold display that indicates the minimum signal level at each time point.

Examples

`TRACE:IQVTIME:FUNCTION MAXHold` displays the Max Hold trace in the IQ level versus Time measurement.

TRACe:IQVTime:SELect:I

Determines whether or not to select the I trace to obtain the maximum and minimum measurement results.

Conditions

Measurement views: RF I&Q versus Time

Group

Trace commands

Syntax

```
TRACe:IQVTime:SELect:I { OFF | ON | 0 | 1 }
TRACe:IQVTime:SELect:I?
```

Related Commands TRACe:IQVTime:SELect:Q

Arguments `OFF` or `0` deselects the I trace.

`ON` or `1` selects the I trace.

Executing TRACe:IQVTime:SELect:I ON sets TRACe:IQVTime:SELect:Q OFF.

Examples `TRACE:IQVTIME:SELECT:I ON` selects the I trace in the RF I&Q versus Time measurement.

TRACe:IQVTime:SELect:Q

Determines whether or not to select the Q trace to obtain the maximum and minimum measurement results.

Conditions Measurement views: RF I&Q versus Time

Group Trace commands

Syntax

```
TRACe:IQVTime:SELect:Q { OFF | ON | 0 | 1 }
TRACe:IQVTime:SELect:Q?
```

Related Commands TRACe:IQVTime:SELect:I

Arguments `OFF` or `0` deselects the Q trace.

`ON` or `1` selects the Q trace.

Executing TRACe:IQVTime:SELect:Q ON sets TRACe:IQVTime:SELect:I OFF.

Examples `TRACE:IQVTIME:SELECT:Q ON` selects the Q trace in the RF I&Q versus Time measurement.

TRACe:OBW:MAXHold

Determines whether or not to perform a Max Hold on the spectrum data for the Occupied Bandwidth trace.

Conditions Measurement views: Occupied Bandwidth

Group Trace commands

Syntax
```
TRACe:OBW:MAXHold { OFF | ON | 0 | 1 }
TRACe:OBW:MAXHold?
```

Arguments `OFF` or `0` does not perform a Max Hold on the spectrum data.

`ON` or `1` performs a Max Hold on the spectrum data.

Examples `TRACE:OBW:MAXHOLD ON` performs a Max Hold on the spectrum data for the Occupied Bandwidth trace.

TRACe:PHVTime

Determines whether or not to show the trace in the Phase versus Time view.

Conditions Measurement views: Phase versus Time

Group Trace commands

Syntax
```
TRACe:PHVTime { OFF | ON | 0 | 1 }
TRACe:PHVTime?
```

Arguments `OFF` or `0` hides the trace in the Phase versus Time view.

`ON` or `1` shows the trace in the Phase versus Time view.

Examples `TRACE:PHVTIME ON` shows the trace in the Phase versus Time view.

TRACe:PHVTime:AVERage:COUNt

Sets or queries the number of traces to combine. This command is effective when you select AVERage with the TRACe:PHVTime:FUNCtion command.

Conditions Measurement views: Phase versus Time

Group Trace commands

Syntax
```
TRACe:PHVTime:AVERage:COUNt <number>
TRACe:PHVTime:AVERage:COUNt?
```

Arguments `<number>::=<NR1>` specifies the number of traces to combine for averaging.
Range: 1 to 10000.

Examples `TRACE:PHVTIME:AVERAGE:COUNT 64` sets the average count to 64.

TRACe:PHVTime:COUNt

Sets or queries how many acquisitions run in the single acquisition mode for the Max or Min Hold trace in the Phase versus Time measurement. This command is effective when TRACe:PHVTime:FUNCtion is set to MAXHold or MINHold and INITiate:CONTinuous is set to OFF.

Conditions Measurement views: Phase versus Time

Group Trace commands

Syntax
```
TRACe:PHVTime:COUNt <number>
TRACe:PHVTime:COUNt?
```

Arguments `<number>::=<NR1>` specifies the count for Max/Min Hold.
Range: 1 to 10000.

Examples `TRACE:PHVTIME:COUNT 32` sets the count to 32 for the Max/Min Hold trace.

TRACe:PHVTime:COUNt:ENABle

Determines whether to enable or disable the count for the Max or Min Hold trace in the Phase versus Time view. This command is effective when TRACe:PHVTime:FUNCtion is set to MAXHold or MINHold.

Conditions Measurement views: Phase versus Time

Group Trace commands

Syntax

```
TRACe:PHVTime:COUNt:ENABle { OFF | ON | 0 | 1 }
TRACe:PHVTime:COUNt:ENABle?
```

Related Commands

Arguments

`OFF` or `0` disables the count for the Max/Min Hold trace.

`ON` or `1` enables the count for the Max/Min Hold trace.

Examples

`TRACE:PHVTIME:COUNT:ENABLE ON` enables the Max/Min Hold count.

TRACe:PHVTime:COUNt:RESet (No Query Form)

Clears the Max or Min Hold data and counter, and restarts the process in the Phase versus Time view. This command is effective when TRACe:PHVTime:FUNCtion is set to MAXHold or MINHold.

Conditions

Measurement views: Phase versus Time

Group

Trace commands

Syntax

```
TRACe:PHVTime:COUNt:RESet
```

Arguments

None

Examples

`TRACE:PHVTIME:COUNT:RESET` clears the Max/Min Hold data and counter, and restarts the process.

TRACe:PHVTime:FREeze

Determines whether to freeze the trace display in the Phase versus Time measurement.

Conditions

Measurement views: Phase versus Time

Group

Trace commands

Syntax

```
TRACe:PHVTime:FREeze { OFF | ON | 0 | 1 }
TRACe:PHVTime:FREeze?
```

Arguments

`OFF` or `0` updates the trace display normally.

`ON` or `1` stops updating trace display.

Examples

`TRACE:PHVTIME:FREEZE ON` stops updating trace display.

TRACe:PHVTime:FUNCtion

Selects or queries the trace function in the Phase versus Time measurement.

Conditions

Measurement views: Phase versus Time

Group

Trace commands

Syntax

```
TRACe:PHVTime:FUNCtion { NORMal | AVERage | MAXHold |
MINHold }
TRACe:PHVTime:FUNCtion?
```

Arguments

`NORMal` selects the normal waveform display.

`AVERage` selects the Average display that indicates the average phase drift at each time point.

`MAXHold` selects the Max Hold display that indicates the maximum phase drift at each time point.

`MINHold` selects the Min Hold display that indicates the minimum phase drift at each time point.

Examples

`TRACE:PHVTIME:FUNCTION MAXHold` displays the Max Hold trace in the Phase versus Time measurement.

TRACe:SGRam:DETection

Selects or queries the display detector (method to be used for decimating traces to fit the available horizontal space on screen). The number of horizontal pixels on screen is generally smaller than that of waveform data points. When actually displayed, the waveform data is therefore thinned out, according to the number of pixels, for being compressed.

Conditions Measurement views: Spectrogram

Group Trace commands

Syntax

```
TRACe:SGRam:DETection { AVERage | POSitive | NEGative |
CAVerage | CPEak | QUASipeak }
TRACe:SGRam:DETection?
```

Arguments

`AVERage` displays the average data value for each pixel.

`POSitive` displays the maximum data value for each pixel.

`NEGative` displays the minimum data value for each pixel.

`CAVerage` displays the CISPR average value for each pixel.

`CPEak` displays the CISPR peak value for each pixel.

`QUASipeak` displays the quasi-peak value for each pixel.

Examples `TRACE:SGRAM:DETECTION POSitive` displays the maximum data value for each pixel.

TRACe:SGRam:FREeze

Determines whether or not to freeze the spectrogram display.

Conditions Measurement views: Spectrogram

Group Trace commands

Syntax

```
TRACe:SGRam:FREeze { OFF | ON | 0 | 1 }
TRACe:SGRam:FREeze?
```

Arguments

`OFF` or `0` updates the display of the spectrogram normally.

`ON` or `1` stops updating the display of the spectrogram.

Examples `TRACE:SGRAM:FREEZE ON` freezes the spectrogram display.

TRACe:SGRam:FUNCtion

Selects or queries the trace function for the specified trace in the spectrogram.

Conditions Measurement views: Spectrogram

Group Trace commands

Syntax

```
TRACe:SGRam:FUNCtion { NONE | AVERage | MAXHold | MINHold }
TRACe:SGRam:FUNCtion?
```

Arguments

`NONE` selects the normal spectrogram display.

`AVERage` selects the Average display that indicates the average signal level at each frequency point.

`MAXHold` selects the Max Hold display that indicates the maximum signal level at each frequency point.

`MINHold` selects the Min Hold display that indicates the minimum signal level at each frequency point.

Examples `TRACE:SGRAM:FUNCTION MAXHold` selects the Max Hold display for the spectrogram.

TRACe:SGRam:FUNCtion:TIME

Sets or queries the time length to combine traces for averaging in the spectrogram. This command is effective when TRACe:SGRam:FUNCtion is set to AVERage, MAXHold or MINHold.

Conditions Measurement views: Spectrogram

Group Trace commands

Syntax

```
TRACe:SGRam:FUNCtion:TIME <value>
TRACe:SGRam:FUNCtion:TIME?
```

Arguments `<value>::=<NR1>` specifies the time length to combine traces for averaging. Range: 0.02 to 60 minutes.

Examples

`TRACE:SGRAM:FUNCTION:TIME 1.5` sets the time length to 1.5 minutes to combine traces for averaging.

TRACe:SGRam:SELect:LINE

Selects or queries the number of line to send to the spectrum display.

Conditions

Measurement views: Spectrogram

Group

Trace commands

Syntax

```
TRACe:SGRam:SELect:LINE <number>
TRACe:SGRam:SELect:LINE?
```

Related Commands

TRACe<x>:SPECtrum

Arguments

`<number>::=<NR1>` specifies the number of line to send to the spectrum display.
Range: 0 to the maximum line number of the spectrogram displayed on screen.

Examples

`TRACE:SGRAM:SELECT:LINE 75` selects Line #75 in the spectrogram to send to the spectrum display.

TRACe:SPURious:COUNt

Sets or queries how many acquisitions run in the single acquisition mode for multi-trace functions (Max Hold and Average) in the Spurious measurement. This command is effective when TRACe:SPURious:FUNCtion is set to MAXHold or AVERage and INITiate:CONTinuous is set to OFF.

Conditions

Measurement views: Spurious

Group

Trace commands

Syntax

```
TRACe:SPURious:COUNt <number>
TRACe:SPURious:COUNt?
```

Arguments

`<number>::=<NR1>` specifies the count for multi-trace functions.
Range: 1 to 10000.

Examples `TRACE:SPURIOUS:COUNT 32` sets the count to 32 for multi-trace functions.

TRACe:SPURious:COUNt:ENABle

Determines whether to enable or disable the count for multi-trace functions (Max Hold and Average) in the Spurious view. This command is effective when TRACe:SPURious:FUNCtion is set to MAXHold or AVERage.

Conditions Measurement views: Spurious

Group Trace commands

Syntax

```
TRACe:SPURious:COUNt:ENABle { OFF | ON | 0 | 1 }
TRACe:SPURious:COUNt:ENABle?
```

Arguments `OFF` or `0` disables the count for multi-trace functions.

`ON` or `1` enables the count for multi-trace functions.

Examples `TRACE:SPURIOUS:COUNT:ENABLE ON` enables the count for multi-trace functions.

TRACe:SPURious:COUNt:RESet (No Query Form)

Clears the multi-function (Max Hold or Average) data and counter, and restarts the process in the Spurious view. This command is effective when TRACe:SPURious:FUNCtion is set to MAXHold or AVERage.

Conditions Measurement views: Spurious

Group Trace commands

Syntax

```
TRACe:SPURious:COUNt:RESet
```

Arguments None

Examples `TRACE:SPURIOUS:COUNT:RESET` clears the multi-function data and counter, and restarts the process.

TRACe:SPURious:FREeze

Determines whether or not to freeze the display of the trace in the Spurious view.

Conditions Measurement views: Spurious

Group Trace commands

Syntax

```
TRACe:SPURious:FREeze { OFF | ON | 0 | 1 }
TRACe:SPURious:FREeze?
```

Arguments

`OFF` or `0` updates the display of the trace normally.

`ON` or `1` stops updating the display of the trace.

Examples `TRACE:SPURIOUS:FREEZE ON` stops updating the display of the trace.

TRACe:SPURious:FUNCtion

Selects or queries the trace function in the Spurious view.

Conditions Measurement views: Spurious

Group Trace commands

Syntax

```
TRACe:SPURious:FUNCtion { NONE | MAXHold | AVERage }
TRACe:SPURious:FUNCtion?
```

Arguments

`NONE` selects normal display.

`MAXHold` selects the Max Hold display that indicates the maximum amplitude drift at each frequency point.

`AVERage` selects the Average display that indicates the average amplitude drift at each frequency point.

Examples `TRACE:SPURIOUS:FUNCTION MAXHold` displays the Max Hold trace in the Spurious measurement.

TRACe<x>:AVTime

Determines whether or not to show the specified trace in the Amplitude versus Time view.

The parameter <x> = 1 to 4; All traces are valid.

Conditions Measurement views: Amplitude versus Time

Group Trace commands

Syntax

```
TRACe<x>:AVTime { OFF | ON | 0 | 1 }
TRACe<x>:AVTime?
```

Arguments `OFF` or `0` hides the specified trace in the Amplitude versus Time view.

`ON` or `1` shows the specified trace in the Amplitude versus Time view.

Examples `TRACE1:AVTIME ON` shows Trace 1 in the Amplitude versus Time view.

TRACe<x>:AVTime:AVERage:COUNt

Sets or queries the number of traces to combine. This command is effective when you select AVERage with the TRACe<x>:AVTime:FUNCtion command.

The parameter <x> = 1 to 3; Trace 4 (math trace) is invalid.

Conditions Measurement views: Amplitude versus Time

Group Trace commands

Syntax

```
TRACe<x>:AVTime:AVERage:COUNt <number>
TRACe<x>:AVTime:AVERage:COUNt?
```

Arguments `<number>::=<NR1>` specifies the number of traces to combine for averaging. Range: 1 to 10000.

Examples `TRACE1:AVTIME:AVERAGE:COUNT 64` sets the average count to 64 for Trace 1.

TRACe<x>:AVTime:AVERage:RESet (No Query Form)

Restarts acquisition and display of waveforms for the specified trace. For an Average, Max Hold, or Min Hold trace, it restarts the sequence, discarding accumulated data and resetting the counter.

The parameter <x> = 1 to 3; Trace 4 (math trace) is invalid.

Conditions Measurement views: Amplitude versus Time

Group Trace commands

Syntax

```
TRACe<x>:AVTime:AVERage:RESet
```

Related Commands TRACe<x>:AVTime:FUNCtion

Arguments None

Examples `TRACE1:AVTIME:AVERAGE:RESET` restarts acquisition and display of waveforms for Trace 1.

TRACe<x>:AVTime:COUNt

Sets or queries how many acquisitions run in the single acquisition mode for the Max or Min Hold trace in the Amplitude versus Time measurement. This command is effective when TRACe<x>:AVTime:FUNCtion is set to MAXHold or MINHold and INITiate:CONTinuous is set to OFF.

The parameter <x> = 1 to 3; Trace 4 (math trace) is invalid.

Conditions Measurement views: Amplitude versus Time

Group Trace commands

Syntax

```
TRACe<x>:AVTime:COUNt <number>
TRACe<x>:AVTime:COUNt?
```

Arguments `<number>::=<NR1>` specifies the count for Max/Min Hold.
Range: 1 to 10000.

Examples `TRACE1:AVTIME:COUNT 32` sets the count to 32 for Trace 1.

TRACe<x>:AVTime:COUNt:ENABle

Determines whether to enable or disable the count for the Max or Min Hold trace in the Amplitude versus Time view. This command is effective when TRACe<x>:AVTime:FUNCtion is set to MAXHold or MINHold.

The parameter <x> = 1 to 3; Trace 4 (math trace) is invalid.

Conditions Measurement views: Amplitude versus Time

Group Trace commands

Syntax

```
TRACe<x>:AVTime:COUNt:ENABle { OFF | ON | 0 | 1 }
TRACe<x>:AVTime:COUNt:ENABle?
```

Arguments `OFF` or `0` disables the count for the Max/Min Hold trace.

`ON` or `1` enables the count for the Max/Min Hold trace.

Examples `TRACE1:AVTIME:COUNT:ENABLE ON` enables the Max/Min Hold count for Trace 1.

TRACe<x>:AVTime:COUNt:RESet (No Query Form)

Clears the Max or Min Hold data and counter, and restarts the process for the specified trace in the Amplitude versus Time view. This command is effective when TRACe<x>:AVTime:FUNCtion is set to MAXHold or MINHold.

The parameter <x> = 1 to 3; Trace 4 (math trace) is invalid.

Conditions Measurement views: Amplitude versus Time

Group Trace commands

Syntax

```
TRACe<x>:AVTime:COUNt:RESet
```

Arguments None

Examples `TRACE1:AVTIME:COUNT:RESET` clears the Max/Min Hold data and counter, and restarts the process for Trace 1.

TRACe<x>:AVTime:FREeze

Determines whether or not to freeze the display of the specified trace in the Amplitude versus Time view.

The parameter <x> = 1 to 4; All traces are valid.

Conditions Measurement views: Amplitude versus Time

Group Trace commands

Syntax

```
TRACe<x>:AVTime:FREeze { OFF | ON | 0 | 1 }
TRACe<x>:AVTime:FREeze?
```

Arguments `OFF` or `0` updates the display of the specified trace normally.

`ON` or `1` stops updating the display of the specified trace.

Examples `TRACE1:AVTIME:FREEZE ON` freezes the display for Trace 1.

TRACe<x>:AVTime:FUNCtion

Selects or queries the function for the specified trace in the Amplitude versus Time view.

The parameter <x> = 1 to 3; Trace 4 (math trace) is invalid.

Conditions Measurement views: Amplitude versus Time

Group Trace commands

Syntax

```
TRACe<x>:AVTime:FUNCtion { NORMal | AVERage | MAXHold |
MINHold }
TRACe<x>:AVTime:FUNCtion?
```

Arguments

`NORMal` selects the normal display.

`AVERage` selects the Average display that indicates the average amplitude at each time point.

`MAXHold` selects the Max Hold display that indicates the maximum amplitude at each time point.

`MINHold` selects the Min Hold display that indicates the minimum amplitude at each time point.

Examples

`TRACE1:AVTIME:FUNCTION MAXHold` selects Max Hold for Trace 1 in the Amplitude versus Time view.

TRACe<x>:AVTime:LEFToperand

Selects or queries the left operand for the math trace (Trace 4) in the Amplitude versus Time view.

The parameter <x> = 4; Only Trace 4 (math trace) is valid.

Conditions

Measurement views: Amplitude versus Time

Group

Trace commands

Syntax

```
TRACe<x>:AVTime:LEFToperand { TRACE1 | TRACE2 | TRACE3 }
TRACe<x>:AVTime:LEFToperand?
```

Related Commands

TRACe<x>:AVTime:RIGHtoperand

Arguments

`TRACE1` selects Trace 1 as the left operand for the math trace.

`TRACE2` selects Trace 2 as the left operand for the math trace.

`TRACE3` selects Trace 3 as the left operand for the math trace.

Examples

`TRACE4:AVTIME:LEFTOPERAND TRACE2` selects Trace 2 as the left operand for the math trace.

TRACe<x>:AVTime:RIGHtoperand

Selects or queries the right operand for the math trace (Trace 4) in the Amplitude versus Time view.

The parameter <x> = 4; Only Trace 4 (math trace) is valid.

Conditions Measurement views: Amplitude versus Time

Group Trace commands

Syntax

```
TRACe<x>:AVTime:RIGHtoperand { TRACE1 | TRACE2 | TRACE3 }
TRACe<x>:AVTime:RIGHtoperand?
```

Related Commands TRACe<x>:AVTime:LEFToperand

Arguments `TRACE1` selects Trace 1 as the right operand for the math trace.

`TRACE2` selects Trace 2 as the right operand for the math trace.

`TRACE3` selects Trace 3 as the right operand for the math trace.

Examples `TRACE4:AVTIME:RIGHTOPERAND TRACE1` selects Trace 1 as the right operand for the math trace.

TRACe<x>:AVTime:SELect

Selects or queries the trace whose measurement results are being displayed in the readout on the top and bottom of the view.

The parameter <x> = 1 to 4; All traces are valid.

Conditions Measurement views: Amplitude versus Time

Group Trace commands

Syntax

```
TRACe<x>:AVTime:SELect
TRACe<x>:AVTime:SELect?
```

Arguments None

Returns `{ 0 | 1 }`

`0` indicates that the results are not being displayed in the readout for the trace.

`1` indicates that the results are being displayed in the readout for the trace.

Examples `TRACE1:AVTIME:SELECT` selects Trace 1 to display the measurement results in the readout.

`TRACE1:AVTIME:SELECT?` might return `1`, indicating that the results are being displayed in the readout for Trace 1.

TRACe<x>:CCDF:FREeze

Determines whether or not to freeze the display of the specified trace (Trace 1 or 2) in the CCDF view.

The parameter <x> = 1 or 2; Trace 3 (Gaussian curve) is invalid.

Conditions Measurement views: CCDF

Group Trace commands

Syntax

```
TRACe<x>:CCDF:FREeze { OFF | ON | 0 | 1 }
TRACe<x>:CCDF:FREeze?
```

Arguments `OFF` or `0` updates the display of the specified trace normally.

`ON` or `1` stops updating the display of the specified trace.

Examples `TRACE1:CCDF:FREEZE ON` freezes the display for Trace 1.

TRACe<x>:CCDF:SELect

Selects or queries the trace whose measurement results are being displayed in the readout on the top of the view. The selected trace is indicated by the measurement pointer (pink triangle) on the waveform.

The parameter <x> = 1 to 3; All traces are valid.

Conditions Measurement views: CCDF

Group Trace commands

Syntax

```
TRACe<x>:CCDF:SELect
TRACe<x>:CCDF:SELect?
```

Arguments None

Returns `{ 0 | 1 }`

`0` indicates that the results are not being displayed in the readout for the trace.

`1` indicates that the results are being displayed in the readout for the trace.

Examples `TRACE1:CCDF:SELECT` selects Trace 1 to display the measurement results in the readout.

`TRACE1:CCDF:SELECT?` might return `1`, indicating that the results are being displayed in the readout for Trace 1.

TRACe<x>:CCDF:SHOW

Determines whether to show or hide the specified trace in the CCDF view.

The parameter <x> = 1 to 3; All traces are valid.

Conditions Measurement views: CCDF

Group Trace commands

Syntax

```
TRACe<x>:CCDF:SHOW { OFF | ON | 0 | 1 }
TRACe<x>:CCDF:SHOW?
```

Arguments `OFF` or `0` hides the specified trace.

`ON` or `1` shows the specified trace.

Examples `TRACE1:CCDF:SHOW ON` shows Trace 1 in the CCDF view.

TRACe<x>:CCDF:X

Sets or queries the horizontal position of the measurement pointer (pink triangle) to measure the CCDF. Use the TRACe<x>:CCDF:Y? query to read the value.

The parameter <x> = 1 to 3; All traces are valid.

NOTE. Use the TRACe<x>:CCDF:SHOW command to show the specified trace and the TRACe<x>:CCDF:SELect command to select the trace before running the TRACe<x>:CCDF:X command.

Conditions Measurement views: CCDF

Group Trace commands

Syntax
```
TRACe<x>:CCDF:X <value>
TRACe<x>:CCDF:X?
```

Arguments `<value>::=<NRf>` specifies the horizontal position of the measurement pointer. Range: 0 to 20 dB.

Examples `TRACE1:CCDF:X 5` puts the measurement pointer at 5 dB on Trace 1.

TRACe<x>:CCDF:Y? (Query Only)

Queries the vertical position (CCDF value) of the measurement pointer (displayed as a pink triangle). Use the TRACe<x>:CCDF:X command to set the horizontal position of the pointer.

The parameter <x> = 1 to 3; All traces are valid.

NOTE. Use the TRACe<x>:CCDF:SHOW command to show the specified trace and the TRACe<x>:CCDF:SELect command to select the trace before running the TRACe<x>:CCDF:Y? query.

Conditions Measurement views: CCDF

Group Trace commands

Syntax
```
TRACe<x>:CCDF:Y?
```

Arguments None

Returns `<value>::=<NRf>` is the vertical position (CCDF) of the measurement pointer. Range: 0 to 100%.

The value of 99.0999953003E+36 is returned if the trace is not available.

Examples

`TRACE1:CCDF:Y?` might return `14.72`, indicating the CCDF is 14.72% at the measurement pointer on Trace 1.

TRACe<x>:DPSA

Determines whether or not to show the specified trace in the DPX spectrum view.

The parameter <x> = 1 to 5; All traces are valid.

Conditions

Measurement views: DPX spectrum

Group

Trace commands

Syntax

```
TRACe<x>:DPSA { OFF | ON | 0 | 1 }
TRACe<x>:DPSA?
```

Arguments

`OFF` or `0` hides the specified trace in the DPX spectrum view.

`ON` or `1` shows the specified trace in the DPX spectrum view.

Examples

`TRACE1:DPSA ON` shows Trace 1 (the maximum trace) in the DPX spectrum view.

TRACe<x>:DPSA:AVERage:COUNt

Sets or queries the number of traces to combine for averaging in the DPX spectrum view.

The parameter <x> = 3; Only Trace 3 (average trace) is valid.

Conditions

Measurement views: DPX spectrum

Group

Trace commands

Syntax

```
TRACe<x>:DPSA:AVERage:COUNt <number>
TRACe<x>:DPSA:AVERage:COUNt?
```

Arguments

`<number>::=<NR1>` specifies the number of traces to combine for averaging.
Range: 1 to 10000.

Examples

`TRACE3:DPSA:AVERAGE:COUNT 32` sets the average count to 32.

TRACe<x>:DPSA:COLor:INTensity

Sets or queries the color intensity in the DPX spectrum view.
The value is common to all traces.

The parameter <x> = 1 to 5; All traces are valid.

Conditions

Measurement views: DPX spectrum

Group

Trace commands

Syntax

```
TRACe<x>:DPSA:COLor:INTensity <value>
TRACe<x>:DPSA:COLor:INTensity?
```

Arguments

`<value>::=<NRf>` specifies color intensity. Range: 1 to 100%.

Examples

`TRACE1:DPSA:COLOR:INTENSITY 30` sets the color intensity to 30%.

TRACe<x>:DPSA:DOT:PERSistent

Determines whether to enable or disable the dot persistence for the bitmap trace (Trace 5) in the DPX spectrum view.

The parameter <x> = 5; Only Trace 5 (bitmap trace) is valid.

Conditions

Measurement views: DPX spectrum

Group

Trace commands

Syntax

```
TRACe<x>:DPSA:DOT:PERSistent { OFF | ON | 0 | 1 }
TRACe<x>:DPSA:DOT:PERSistent?
```

Arguments OFF or 0 disables the dot persistence.

ON or 1 enables the dot persistence.

Examples TRACE5:DPSA:DOT:PERSISTENT ON enables the dot persistence in the DPX spectrum view.

TRACe<x>:DPSA:DOT:PERSistent:TYPE

Selects or queries the persistence type for the bitmap trace (Trace 5) in the DPX spectrum view.

The parameter <x> = 5; Only Trace 5 (bitmap trace) is valid.

Conditions Measurement views: DPX spectrum

Group Trace commands

Syntax

```
TRACe<x>:DPSA:DOT:PERSistent:TYPE { VARiable | INFinite }
TRACe<x>:DPSA:DOT:PERSistent:TYPE?
```

Arguments VARiable selects the variable persistence display which leaves acquired data points on the display for a period of time specified by the TRACe<x>:DPSA:DOT:PERSistent:VARiable command.

INFinite selects the infinite persistence display which accumulates data points on the display indefinitely.

Examples TRACE5:DPSA:DOT:PERSISTENT:TYPE VARiable selects the variable persistence display.

TRACe<x>:DPSA:DOT:PERSistent:VARiable

Sets or queries how long data points are displayed. This command is effective when TRACe<x>:DPSA:DOT:PERSistent:TYPE is set to VARiable. This affects the display only.

The parameter <x> = 5; Only Trace 5 (bitmap trace) is valid.

Conditions Measurement views: DPX spectrum

Group Trace commands

Syntax

```
TRACe<x>:DPSA:DOT:PERSistent:VARiable <number>
TRACe<x>:DPSA:DOT:PERSistent:VARiable?
```

Arguments `<number>::=<NR1>` specifies the number that the waveform points are displayed on the screen. Range: 1 to 1000 (unitless; the default value is 10).

Examples `TRACE5:DPSA:DOT:PERSISTENT:VARIABLE 20` specifies that the waveform points are displayed on the screen for a period of 20 before they disappear.

TRACe<x>:DPSA:FREeze

Determines whether or not to freeze the display of the specified trace in the DPX spectrum view.

The parameter <x> = 1 to 5; All traces are valid.

Conditions Measurement views: DPX spectrum

Group Trace commands

Syntax

```
TRACe<x>:DPSA:FREeze { OFF | ON | 0 | 1 }
TRACe<x>:DPSA:FREeze?
```

Arguments `OFF` or `0` updates the display of the specified trace normally.

`ON` or `1` stops updating the display of the specified trace.

Examples `TRACE1:DPSA:FREEZE ON` freezes the display for the +peak trace.

TRACe<x>:DPSA:FUNCtion

Selects or queries the trace function for the +Peak, -Peak, or Average trace (Trace 1, 2, or 3, respectively) in the DPX spectrum view.

The parameter <x> = 1 to 3; Trace 4 (math trace) and Trace 5 (bitmap trace) are invalid.

Conditions Measurement views: DPX spectrum

Group Trace commands

Syntax

```
TRACe<x>:DPSA:FUNCtion { NORMal | HOLD | AVERage }
TRACe<x>:DPSA:FUNCtion?
```

Arguments The following table shows the trace function and display. For the average trace, use the TRACe<x>:DPSA:AVERage:COUNt command to set the average count.

Function	Trace1 (+Peak trace)	Trace2 (-Peak trace)	Trace3 (Average trace)
NORMal	Normal spectrum (Detection: +Peak)	Normal spectrum (Detection: -Peak)	Normal spectrum (Detection: Average)
HOLD	Max-hold spectrum	Min-hold spectrum	NA
AVERage	NA	NA	Average spectrum

Examples `TRACE1:DPSA:FUNCTION HOLD` selects the max hold waveform for Trace 1.

TRACe<x>:DPSA:LEFToperand

Selects or queries the left operand for the math trace (Trace 4) in the DPX spectrum view.

The parameter <x> = 4; Only Trace 4 (math trace) is valid.

Conditions Measurement views: DPX spectrum

Group Trace commands

Syntax

```
TRACe<x>:DPSA:LEFToperand { TRACE1 | TRACE2 | TRACE3 }
TRACe<x>:DPSA:LEFToperand?
```

Related Commands TRACe<x>:DPSA:RIGHtoperand

Arguments `TRACE1` selects Trace 1 as the left operand for the math trace.

`TRACE2` selects Trace 2 as the left operand for the math trace.

`TRACE3` selects Trace 3 as the left operand for the math trace.

Examples `TRACE4:DPSA:LEFTOPERAND TRACE2` selects Trace 2 as the left operand for the math trace.

TRACe<x>:DPSA:RIGHtoperand

Selects or queries the right operand for the math trace (Trace 4) in the DPX spectrum view.

The parameter <x> = 4; Only Trace 4 (math trace) is valid.

Conditions Measurement views: DPX spectrum

Group Trace commands

Syntax

```
TRACe<x>:DPSA:RIGHtoperand { TRACE1 | TRACE2 | TRACE3 }
TRACe<x>:DPSA:RIGHtoperand?
```

Related Commands TRACe<x>:DPSA:LEFToperand

Arguments

`TRACE1` selects Trace 1 as the right operand for the math trace.

`TRACE2` selects Trace 2 as the right operand for the math trace.

`TRACE3` selects Trace 3 as the right operand for the math trace.

Examples `TRACE4:DPSA:RIGHTOPERAND TRACE1` selects Trace 1 as the right operand for the math trace.

TRACe<x>:DPSA:SELect

Selects or queries the trace to display the readout at the upper left of the DPX spectrum view.

The parameter <x> = 1 to 5; All traces are valid.

Conditions Measurement views: DPX spectrum

Group Trace commands

Syntax

```
TRACe<x>:DPSA:SELect
TRACe<x>:DPSA:SELect?
```

Arguments None

Returns `{ 0 | 1 }`

`0` indicates that the readout is not being displayed for the specified trace.

`1` indicates that the readout is being displayed for the specified trace.

Examples `TRACE1:DPSA:SELECT` selects Trace 1 (+Peak trace) to display the readout.

`TRACE1:DPSA:SELECT?` might return 1, indicating that the readout is being displayed for Trace 1.

TRACe<x>:PNOise:DETection

Selects or queries the display detector (method to be used for decimating traces to fit the available horizontal space on screen) for the specified trace. The number of horizontal pixels on screen is generally smaller than that of waveform data points. When actually displayed, the waveform data is therefore thinned out, according to the number of pixels, for being compressed.

The parameter <x> = 1 and 2.

Conditions Measurement views: Phase noise

Group Trace commands

Syntax

```
TRACe<x>:PNOise:DETection { AVERage | POSNegative }
TRACe<x>:PNOise:DETection?
```

Arguments `AVERage` displays the average data value for each pixel.

`POSNegative` displays the maximum and minimum data values for each pixel.

Examples `TRACE1:PNOISE:DETECTION AVERage` specifies that Trace 1 displays the average data value for each pixel.

TRACe<x>:PNOise:FREeze

Determines whether or not to freeze the trace display in the phase noise measurement.

The parameter <x> = 1 and 2.

Conditions Measurement views: Phase noise

Group Trace commands

Syntax
```
TRACe<x>:PNOise:FREeze { OFF | ON | 0 | 1 }
TRACe<x>:PNOise:FREeze?
```

Arguments `OFF` or `0` updates the trace display normally.

`ON` or `1` stops updating the trace display.

Examples `TRACE1:PNOISE:FREEZE ON` freezes the Trace 1 display.

TRACe<x>:PNOise:SELect

Selects the trace in the phase noise measurement. The query returns the currently selected trace.

The parameter <x> = 1 and 2.

Conditions Measurement views: Phase noise

Group Trace commands

Syntax
```
TRACe<x>:PNOise:SELect
TRACe<x>:PNOise:SELect?
```

Arguments None

Returns `0` (not selected) or `1` (selected).

Examples `TRACE2:PNOISE:SELECT` selects Trace 2.

TRACe<x>:PNOise:SHOW

Determines whether to show or hide the specified trace in the phase noise view.

The parameter <x> = 1 and 2.

Conditions Measurement views: Phase noise

Group Trace commands

Syntax
```
TRACe<x>:PNOise:SHOW { OFF | ON | 0 | 1 }
TRACe<x>:PNOise:SHOW?
```

Arguments OFF or 0 hides the specified trace.

ON or 1 shows the specified trace.

Examples TRACE1:PNOISE:SHOW ON shows Trace 1 in the phase noise view.

TRACe<x>:PNOise:SMOothing:COUNt

Sets or queries the number of data points to take the moving average for smoothing the trace. This command is effective when TRACe<x>:PNOise:SMOothing:ENABle is set to ON.

The parameter <x> = 1 and 2.

Conditions Measurement views: Phase noise

Group Trace commands

Syntax
```
TRACe<x>:PNOise:SMOothing:COUNT <number>
TRACe<x>:PNOise:SMOothing:COUNT?
```

Arguments <number>::=<NR1> specifies the number of data points to take the moving average for smoothing. Range: 3 to 50.

Examples TRACE1:PNOISE:SMOOTHING:COUNT 16 sets the smoothing count to 16 for Trace 1.

TRACe<x>:PNOise:SMOothing:ENABle

Determines whether to enable or disable smoothing the specified trace in the phase noise view.

The parameter <x> = 1 and 2.

Conditions Measurement views: Phase noise

Group Trace commands

Syntax
```
TRACe<x>:PNOise:SMOothing:ENABle { OFF | ON | 0 | 1 }
TRACe<x>:PNOise:SMOothing:ENABle?
```

Arguments OFF or 0 disables smoothing.

ON or 1 enables smoothing.

Examples TRACE1:PNOISE:SMOOTHING:ENABLE ON enables smoothing Trace 1 in the phase noise view.

TRACe<x>:PNOise:SMOothing:RESet (No Query Form)

Restarts the smoothing process, discarding accumulated data and resetting the counter. This command is effective when TRACe<x>:PNOise:SMOothing:ENABle is set to ON.

The parameter <x> = 1 and 2.

Conditions Measurement views: Phase noise

Group Trace commands

Syntax
```
TRACe<x>:PNOise:SMOothing:RESet
```

Arguments None

Examples TRACE1:PNOISE:SMOOTHING:RESET restarts the smoothing process for Trace 1.

TRACe<x>:SPECtrum

Determines whether to show or hide the specified trace in the Spectrum view.

The parameter <x> = 1 to 5; All traces are valid.

Conditions Measurement views: Spectrum

Group Trace commands

Syntax

```
TRACe<x>:SPECtrum { OFF | ON | 0 | 1 }
TRACe<x>:SPECtrum?
```

Arguments

OFF or 0 hides the specified trace.

ON or 1 shows the specified trace.

For Trace 5 (spectrogram), use the TRACe:SGRam:SELect:LINE command to select the number of line to send to the spectrum display.

Examples

TRACE1:SPECTRUM ON shows Trace 1 in the Spectrum Analyzer view.

TRACe<x>:SPECtrum:AVERage:COUNt

Sets or queries the number of traces to combine. This command is effective when TRACe<x>:SPECtrum:FUNCtion is AVERage, MAXHold or MINHold.

The parameter <x> = 1 to 3; Trace 4 (math trace) and Trace 5 (spectrogram) are invalid.

Conditions

Measurement views: Spectrum

Group

Trace commands

Syntax

```
TRACe<x>:SPECtrum:AVERage:COUNT <number>
TRACe<x>:SPECtrum:AVERage:COUNt?
```

Arguments

<number>::=<NR1> specifies the number of traces to combine for averaging. Range: 1 to 10000.

Examples

TRACE1:SPECTRUM:AVERAGE:COUNT 64 sets the average count to 64 for Trace 1.

TRACe<x>:SPECtrum:AVERage:RESet (No Query Form)

Clears average data and counter, and restarts the average process for the specified trace in the Spectrum view. This command is effective when TRACe<x>:SPECtrum:FUNCtion is set to AVERage, MAXHold or MINHold.

The parameter <x> = 1 to 3; Trace 4 (math trace) and Trace 5 (spectrogram) are invalid.

Conditions Measurement views: Spectrum

Group Trace commands

Syntax
```
TRACe<x>:SPECtrum:AVERage:RESet
```

Arguments None

Examples `TRACE1:SPECTRUM:AVERAGE:RESET` clears average data and counter, and restarts the average process for Trace 1.

TRACe<x>:SPECtrum:COUNt

Sets or queries how many acquisitions run in the single acquisition mode for the Max or Min Hold trace in the Spectrum measurement. This command is effective when TRACe<x>:SPECtrum:FUNCtion is set to MAXHold or MINHold and INITiate:CONTinuous is set to OFF.

The parameter <x> = 1 to 3; Trace 4 (math trace) and Trace 5 (spectrogram) are invalid.

Conditions Measurement views: Spectrum

Group Trace commands

Syntax
```
TRACe<x>:SPECtrum:COUNt <number>
TRACe<x>:SPECtrum:COUNt?
```

Arguments `<number>::=<NR1>` specifies the count for Max/Min Hold. Range: 1 to 10000.

Examples `TRACE1:SPECTRUM:COUNT 32` sets the count to 32 for Trace 1.

TRACe<x>:SPECtrum:COUNt:ENABle

Determines whether to enable or disable the count for the Max or Min Hold trace in the Spectrum view. This command is effective when TRACe<x>:SPECtrum:FUNCtion is set to MAXHold or MINHold.

The parameter <x> = 1 to 3; Trace 4 (math trace) and Trace 5 (spectrogram) are invalid.

Conditions Measurement views: Spectrum

Group Trace commands

Syntax

```
TRACe<x>:SPECtrum:COUNt:ENABle { OFF | ON | 0 | 1 }
TRACe<x>:SPECtrum:COUNt:ENABle?
```

Arguments `OFF` or `0` disables the count for the Max/Min Hold trace.

`ON` or `1` enables the count for the Max/Min Hold trace.

Examples `TRACE1:SPECTRUM:COUNT:ENABLE ON` enables the Max/Min Hold count for Trace 1.

TRACe<x>:SPECtrum:COUNt:RESet (No Query Form)

Clears the Max or Min Hold data and counter, and restarts the process for the specified trace in the Spectrum view. This command is effective when TRACe<x>:SPECtrum:FUNCtion is set to MAXHold or MINHold.

The parameter <x> = 1 to 3; Trace 4 (math trace) and Trace 5 (spectrogram) are invalid.

Conditions Measurement views: Spectrum

Group Trace commands

Syntax

```
TRACe<x>:SPECtrum:COUNt:RESet
```

Arguments None

Examples `TRACE1:SPECTRUM:COUNT:RESET` clears the Max/Min Hold data and counter, and restarts the process for Trace 1.

TRACe<x>:SPECtrum:DETection

Selects or queries the display detector (method to be used for decimating traces to fit the available horizontal space on screen). The number of horizontal pixels on screen is generally smaller than that of waveform data points. When actually displayed, the waveform data is therefore thinned out, according to the number of pixels, for being compressed.

The parameter <x> = 1 to 3; Trace 4 (math trace) and Trace 5 (spectrogram) are invalid.

Conditions

Measurement views: Spectrum

Group

Trace commands

Syntax

```
TRACe<x>:SPECtrum:DETection { AVERage | POSitive | NEGative
| CAVerage | CPEak | QUASipeak }
TRACe<x>:SPECtrum:DETection?
```

Arguments

`AVERage` displays the average data value for each pixel.

`POSitive` displays the maximum data value for each pixel.

`NEGative` displays the minimum data value for each pixel.

`CAVerage` displays the CISPR average value for each pixel.

`CPEak` displays the CISPR peak value for each pixel.

`QUASipeak` displays the quasi-peak value for each pixel.

Examples

`TRACE1:SPECTRUM:DETECTION POSitive` displays the maximum data value for each pixel on Trace 1.

TRACe<x>:SPECtrum:FREeze

Determines whether or not to freeze the display of the specified trace in the Spectrum view.

The parameter <x> = 1 to 5; All traces are valid.

Conditions

Measurement views: Spectrum

Group

Trace commands

Syntax
```
TRACe<x>:SPECtrum:FREeze { OFF | ON | 0 | 1 }
TRACe<x>:SPECtrum:FREeze?
```

Arguments

`OFF` or `0` updates the display of the specified trace normally.

`ON` or `1` stops updating the display of the specified trace.

Examples

`TRACE1:SPECTRUM:FREEZE ON` freezes Trace 1 in the Spectrum Analysis display.

TRACe<x>:SPECtrum:FUNCtion

Selects or queries the function for the specified trace in the Spectrum view.

The parameter <x> = 1 to 3; Trace 4 (math trace) and Trace 5 (spectrogram) are invalid.

Conditions

Measurement views: Spectrum

Group

Trace commands

Syntax
```
TRACe<x>:SPECtrum:FUNCtion { NONE | AVERage | MAXHold |
MINHold }
TRACe<x>:SPECtrum:FUNCtion?
```

Arguments

`NONE` selects the normal spectrum display.

`AVERage` selects the Average display that indicates the average signal level at each frequency point.

`MAXHold` selects the Max Hold display that indicates the maximum signal level at each frequency point.

`MINHold` selects the Min Hold display that indicates the minimum signal level at each frequency point.

Examples

`TRACE1:SPECTRUM:FUNCTION MAXHold` selects Max Hold for Trace 1 in the Spectrum view.

TRACe<x>:SPECtrum:LEFToperand

Selects or queries the left operand for the math trace (Trace 4) in the Spectrum view.

The parameter <x> = 4; Only Trace 4 (math trace) is valid.

Conditions Measurement views: Spectrum

Group Trace commands

Syntax

```
TRACe<x>:SPECtrum:LEFToperand { TRACE1 | TRACE2 | TRACE3 }
TRACe<x>:SPECtrum:LEFToperand?
```

Related Commands TRACe<x>:SPECtrum:RIGHtoperand

Arguments TRACE1 selects Trace 1 as the left operand for the math trace.

TRACE2 selects Trace 2 as the left operand for the math trace.

TRACE3 selects Trace 3 as the left operand for the math trace.

Examples TRACE4:SPECTRUM:LEFTOPERAND TRACE1 selects Trace 1 as the left operand for the math trace.

TRACe<x>:SPECtrum:RIGHtoperand

Selects or queries the right operand for the math trace (Trace 4) in the Spectrum view.

The parameter <x> = 4; Only Trace 4 (math trace) is valid.

Conditions Measurement views: Spectrum

Group Trace commands

Syntax

```
TRACe<x>:SPECtrum:RIGHtoperand { TRACE1 | TRACE2 | TRACE3 }
TRACe<x>:SPECtrum:RIGHtoperand?
```

Related Commands TRACe<x>:SPECtrum:LEFToperand

Arguments TRACE1 selects Trace 1 as the right operand for the math trace.

TRACE2 selects Trace 2 as the right operand for the math trace.

TRACE3 selects Trace 3 as the right operand for the math trace.

Examples `TRACE4:SPECTRUM:RIGHTOPERAND TRACE1` selects Trace 1 as the right operand for the math trace.

TRACe<x>:SPECtrum:SELect

Selects or queries the trace to display the readout at the upper left of the Spectrum view.

The parameter <x> = 1 to 5; All traces are valid.

NOTE. *TRACe5 (spectrogram) is valid when the spectrum and spectrogram measurements are running.*

Conditions Measurement views: Spectrum

Group Trace commands

Syntax

```
TRACe<x>:SPECtrum:SELect
TRACe<x>:SPECtrum:SELect?
```

Arguments None

Returns `{ 0 | 1 }`

`0` indicates that the readout is not being displayed for the specified trace.

`1` indicates that the readout is being displayed for the specified trace.

Examples `TRACE1:SPECTRUM:SELECT` selects Trace 1 to display the readout.

`TRACE1:SPECTRUM:SELECT?` might return `1`, indicating that the readout is being displayed for Trace 1.

*TRG (No Query Form)

Generates a trigger. It produces the same effect as the Force Trigger button on the Trigger control panel. This command is valid when the trigger mode is Triggered.

Conditions Measurement views: All

Group IEEE common commands

Syntax `*TRG`

Related Commands TRIGger[:SEQuence]:STATus

Arguments None

Examples `*TRG` generates a trigger.

TRIGger:MASK:NEW (No Query Form)

Loads a new frequency mask.

Conditions Measurement views: All

Group Trigger commands

Syntax

```
TRIGger:MASK:NEW <freq(1)>,<ampl(1)>,<freq(2)>,<ampl(2)>,
...,<freq(n)>,<ampl(n)>     (n = 500 maximum)
```

Arguments `<freq(n)>,<ampl(n)>` is a frequency (Hz) and amplitude (dBm) pair to specify a point of the mask. Up to 500 pairs can be specified with zero frequency being the center screen. The mask is visible in the spectrum view with the following trigger conditions

- Trigger mode: Triggered (TRIGger[:SEQuence]:STATus is set to ON or 1.)
- Trigger type: Frequency Mask (TRIGger[:SEQuence]:EVENt:INPut:TYPE is set to FMASK.)

Examples `TRIGGER:MASK:NEW -8E6,-80,0,-10,8E6,-80` loads the mask with the points A (-8 MHz, -80 dBm), B (0 Hz, -10 dBm), and C (8 MHz, -80 dBm), as shown in the following figure.

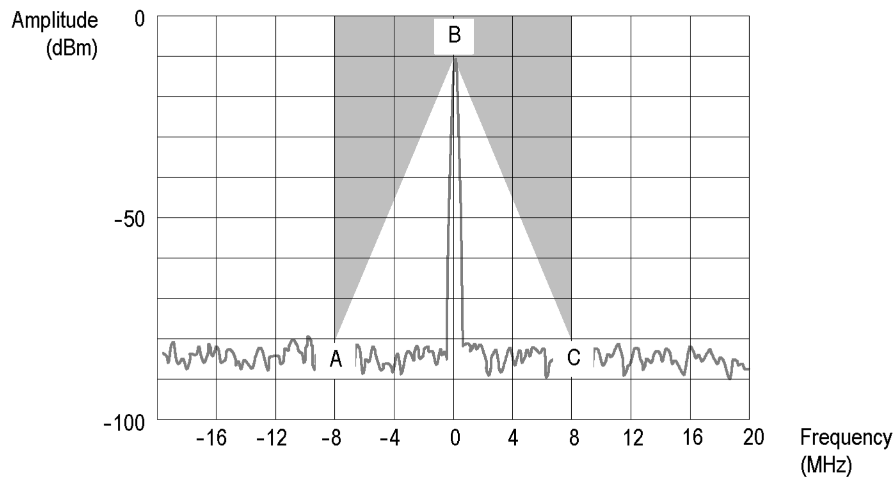


Figure 2-6: Trigger mask setting example

TRIGger:MASK:NEW:AUTO (No Query Form)

Draws a new frequency mask automatically based on a reference trace.

Conditions Measurement views: All

Group Trigger commands

Syntax

```
TRIGger:MASK:NEW:AUTO
<meas_ID>,<trace_ID>,<x_margin>,<y_margin>
```

Arguments `<meas_ID>::=<string>` specifies the measurement view.

`<trace_ID>` specifies the reference trace.

The values of `<meas_ID>` and `<trace_ID>` are listed in the following table for each possible measurement view.

`<x_margin>::=<NRf>` specifies the X margin (horizontal offset from the reference trace) in Hz.

`<y_margin>::=<NRf>` specifies the Y margin (vertical offset from the reference trace) in dB.

Measurement view	<meas_ID>	<trace_ID>
Spectrum	"specan"	TRACE1 (Trace 1), TRACE2 (Trace 2), TRACE3 (Trace 3), TRACE4 (Math trace), TRACE5 (Spectrogram trace)
DPX spectrum	"dpsa"	TRACE1 (+Peak trace), TRACE2 (-Peak trace), TRACE3 (Avg trace), TRACE4 (Math trace)
Channel power and ACPR	"acpr"	TRACE1 (Trace 1)
MCPR	"mcpr"	TRACE1 (Trace 1)
OBW	"obw"	TRACE1 (Trace 1)

Examples

`TRIGGER:MASK:NEW:AUTO "specan",TRACE1,2E+6,15` draws a new frequency mask automatically in the Spectrum view based on Trace 1 with the horizontal margin of 2 MHz and the vertical margin of 15 dB.

TRIGger:MASK:OPEN (No Query Form)

Opens a trigger mask with a specified file. To save a trigger mask, use the TRIGger:MASK:SAVE command.

Conditions

Measurement views: All

Group

Trigger commands

Syntax

`TRIGger:MASK:OPEN <file_name>`

Arguments

`<file_name>::=<string>` specifies the trigger mask file to open. The file extension is .msk. You can omit the extension.

For the directory of file, refer to *Specifying the File* (See page 2-40.)

Examples

`TRIGGER:MASK:OPEN "C:\My Documents\Mask1"` opens the trigger mask with the *Mask1* file in the *My Documents* directory.

TRIGger:MASK:SAVE (No Query Form)

Saves the current trigger mask to a specified file. To open the trigger mask, use the TRIGger:MASK:OPEN command.

Conditions Measurement views: All

Group Trigger commands

Syntax `TRIGger:MASK:SAVE <file_name>`

Arguments `<file_name>::=<string>` specifies the file to save the trigger mask. The file extension is .msk. You can omit the extension.

For the directory of file, refer to *Specifying the File* (See page 2-40.)

Examples `TRIGGER:MASK:SAVE "C:\My Documents\Mask1“` saves the trigger mask to the *Mask1* file in the *My Documents* directory.

TRIGger[:SEQuence]:ADVanced:SWEep:MODE

Determines whether or not to trigger each segment in the swept acquisition mode.

Conditions Measurement views: All

Group Trigger commands

Syntax

```
TRIGger[:SEQuence]:ADVanced:SWEep:MODE { OFF | ON | 0 | 1 }
TRIGger[:SEQuence]:ADVanced:SWEep:MODE?
```

Arguments `OFF` or `0` does not trigger each segment in the swept acquisition mode.

`ON` or `1` triggers each segment in the swept acquisition mode.

Examples `TRIGGER:SEQUENCE:ADVANCED:SWEEP:MODE ON` triggers each segment in the swept acquisition mode.

TRIGger[:SEQuence]:EVENt:EXTFront:IMPedance

Selects or queries the impedance of the external trigger input on the front panel.

Conditions Measurement views: All

Group Trigger commands

Syntax

```
TRIGger[:SEQuence]:EVENt:EXTFront:IMPedance { OHM50 | OHM5K
}
TRIGger[:SEQuence]:EVENt:EXTFront:IMPedance?
```

Arguments `OHM50` selects 50Ω impedance.

`OHM5K` selects 5 kΩ impedance.

Examples `TRIGGER:SEQUENCE:EVENT:EXTFRONT:IMPEDANCE OHM50` selects 50Ω impedance for the external trigger input on the front panel.

TRIGger[:SEQuence]:EVENt:EXTFront:LEVel

Sets or queries the trigger level at the external trigger input on the front panel.

Conditions Measurement views: All

Group Trigger commands

Syntax

```
TRIGger[:SEQuence]:EVENt:EXTFront:LEVel <value>
TRIGger[:SEQuence]:EVENt:EXTFront:LEVel?
```

Arguments `<value>::=<NRf>` specifies the trigger level. Range: -2.5 to +2.5 V.

Examples `TRIGGER:SEQUENCE:EVENT:EXTFRONT:LEVEL 1.5` sets the trigger level to 1.5 V at the external trigger input on the front panel.

TRIGger[:SEQuence]:EVENt:EXTFront:SLOPe

Selects or queries the trigger slope of the external trigger input on the front panel.

Conditions Measurement views: All

Group Trigger commands

Syntax
```
TRIGger[:SEQuence]:EVENt:EXTFront:SLOPe { RISe | FALL }
TRIGger[:SEQuence]:EVENt:EXTFront:SLOPe?
```

Arguments `RISe` causes the trigger event on the rising edge.

`FALL` causes the trigger event on the falling edge.

Examples `TRIGGER:SEQUENCE:EVENT:EXTFRONT:SLOPE RISe` causes the trigger event on the rising edge of the external input signal.

TRIGger[:SEQuence]:EVENt:EXTRear:SLOPe

Selects or queries the trigger slope of the external trigger input on the rear panel. The trigger level is fixed to the TTL threshold.

Conditions Measurement views: All

Group Trigger commands

Syntax
```
TRIGger[:SEQuence]:EVENt:EXTRear:SLOPe { RISe | FALL }
TRIGger[:SEQuence]:EVENt:EXTRear:SLOPe?
```

Arguments `RISe` causes the trigger event on the rising edge.

`FALL` causes the trigger event on the falling edge.

Examples `TRIGGER:SEQUENCE:EVENT:EXTREAR:SLOPE RISe` causes the trigger event on the rising edge of the external input signal.

TRIGger[:SEQuence]:EVENt:GATed

Selects or queries the positive or negative logic for the gated trigger input on the rear panel.

Conditions Measurement views: All

Group Trigger commands

Syntax

```
TRIGger[:SEQuence]:EVENt:GATed { HIGh | LOW }
TRIGger[:SEQuence]:EVENt:GATed?
```

Arguments `HIGh` specifies that the gated trigger input is high active.

`LOW` specifies that the gated trigger input is low active.

Examples `TRIGGER:SEQUENCE:EVENT:GATED HIGh` specifies that the gated trigger input is high active.

TRIGger[:SEQuence]:EVENt:INPut:FMASk:VIOLation

Selects or queries when the analyzer triggers in the frequency mask trigger.

Conditions Measurement views: All

Group Trigger commands

Syntax

```
TRIGger[:SEQuence]:EVENt:INPut:FMASk:VIOLation { T | F | TF
| FT | TFT | FTF }
TRIGger[:SEQuence]:EVENt:INPut:FMASk:VIOLation?
```

Arguments The following table lists the arguments which represent the trigger violations.

Table 2-42: Trigger violations

Violation	Description
T	Only one state is required to initiate a trigger event. The signal has at least one data point inside the mask. The trigger event occurs at the first point that appears inside the mask. A trigger event could occur on the first acquisition.
F	Only one state change is required to initiate a trigger event. The signal has at least one data point outside the mask. The trigger event occurs at the first point that appears outside the mask. A trigger event could occur on the first acquisition.
TF	Two states are required to initiate a trigger event. The signal must be inside the mask and then passes out of the mask. The trigger event occurs at the first transition where the signal passes out of the mask.

Table 2-42: Trigger violations (cont.)

Violation	Description
FT	Two states are required to initiate a trigger event. The signal must be outside the mask and then passes into the mask. The trigger event occurs at the first transition where the signal passes into the mask.
TFT	Three states are required to initiate a trigger event. The signal starts inside the mask and then passes out of the mask. Next, the signal must pass into the mask. The trigger event occurs at the second transition where the signal passes back into the mask.
FTF	Three states are required to initiate a trigger event. The signal starts outside the mask and then passes into the mask. Next, the signal must pass back outside the mask. The trigger event occurs at the second transition where the signal passes back out of the mask.

Examples

`TRIGGER:SEQUENCE:EVENT:INPUT:FMASK:VIOLATION TF` specifies that the analyzer will trigger when the signal has crossed into the mask and then outside of the mask.

TRIGger[:SEQuence]:EVENt:INPut:LEVel

Sets or queries the trigger level for the RF input level trigger.

Conditions

Measurement views: All

Group

Trigger commands

Syntax

```
TRIGger[:SEQuence]:EVENt:INPut:LEVel <value>
TRIGger[:SEQuence]:EVENt:INPut:LEVel?
```

Arguments

`<value>::=<NRf>` specifies the trigger level. Range: -170 to +50 dBm.

Examples

`TRIGGER:SEQUENCE:EVENT:INPUT:LEVEL -10` sets the trigger level to -10 dBm for the RF input level trigger.

TRIGger[:SEQuence]:EVENt:INPut:SLOPe

Selects or queries the trigger slope for the RF input level trigger.

Conditions

Measurement views: All

Group Trigger commands

Syntax

```
TRIGger[:SEQuence]:EVENt:INPut:SLOPe { RISe | FALL }
TRIGger[:SEQuence]:EVENt:INPut:SLOPe?
```

Arguments `RISe` causes the trigger event on the rising edge.

`FALL` causes the trigger event on the falling edge.

Examples `TRIGGER:SEQUENCE:EVENT:INPUT:SLOPE RISe` causes the trigger event on the rising edge of the RF input signal.

TRIGger[:SEQuence]:EVENt:INPut:TDBWidth

Sets or queries the time-domain bandwidth for the RF input power trigger. This command is effective when TRIGger[:SEQuence]:EVENt:INPut:TDBWidth:STATe is ON.

Conditions Measurement views: All

Group Trigger commands

Syntax

```
TRIGger[:SEQuence]:EVENt:INPut:TDBWidth <value>
TRIGger[:SEQuence]:EVENt:INPut:TDBWidth?
```

Arguments `<value>::=<NRf>` specifies the time-domain bandwidth.
Range: 1 Hz to 60 MHz.

Examples `TRIGGER:SEQUENCE:EVENT:INPUT:TDBWIDTH 5MHz` sets the time-domain bandwidth to 5 MHz for the RF input power trigger.

TRIGger[:SEQuence]:EVENt:INPut:TDBWidth:ACTual? (Query Only)

Queries the actual time-domain bandwidth for the RF input power trigger.

Conditions Measurement views: All

Group Trigger commands

Syntax `TRIGger[:SEQuence]:EVENt:INPut:TDBWidth:ACTual?`

Related Commands TRIGger[:SEQuence]:EVENt:INPut:TDBWidth

Arguments None

Returns `<NRf>` Actual time-domain bandwidth.

Examples `TRIGGER:SEQUENCE:EVENT:INPUT:TDBWIDTH:ACTUAL?` might return `1.000E+6`, indicating that the actual time-domain bandwidth is 1 MHz.

TRIGger[:SEQuence]:EVENt:INPut:TDBWidth:STATe

Determines whether to set the time-domain bandwidth automatically or manually for the RF input power trigger.

Conditions Measurement views: All

Group Trigger commands

Syntax
```
TRIGger[:SEQuence]:EVENt:INPut:TDBWidth:STATe { OFF | ON |
0 | 1 }
TRIGger[:SEQuence]:EVENt:INPut:TDBWidth:STATe?
```

Arguments `OFF` or `0` sets the time-domain bandwidth automatically.

`ON` or `1` sets the time-domain bandwidth manually using the TRIGger[:SEQuence]:EVENt:INPut:TDBWidth command.

Examples `TRIGGER:SEQUENCE:EVENT:INPUT:TDBWIDTH:STATE OFF` sets the time-domain bandwidth automatically.

TRIGger[:SEQuence]:EVENt:INPut:TYPE

Selects or queries the trigger type for the source of RF input.

Conditions Measurement views: All

Group Trigger commands

Syntax

```
TRIGger[:SEQuence]:EVENt:INPut:TYPE { POWer | FMASK }
TRIGger[:SEQuence]:EVENt:INPut:TYPE?
```

Arguments `POWer` uses the power level for triggering.

`FMASK` uses the frequency mask for triggering.

Examples `TRIGGER:SEQUENCE:EVENT:INPUT:TYPE FMASK` uses the frequency mask for triggering.

TRIGger[:SEQuence]:EVENt:SOURce

Selects or queries the trigger event source.

Conditions Measurement views: All

Group Trigger commands

Syntax

```
TRIGger[:SEQuence]:EVENt:SOURce { INPut | EXTFront | EXTRear
| EXTGated | LINe }
TRIGger[:SEQuence]:EVENt:SOURce?
```

Arguments The following table lists the arguments.

Table 2-43: Trigger event source

Argument	Source
INPut	RF input
EXTFront	Trigger in (front)
EXTRear	Trigger in (rear)
EXTGated	Gate in
LINe	AC line

Examples `TRIGGER:SEQUENCE:EVENT:SOURCE INPut` specifies the trigger event source as the RF input.

TRIGger[:SEQuence]:FORCed

Determines whether or not to cause a manual trigger if the acquisition is armed, ready and waiting for a trigger. This command is valid when TRIGger[:SEQuence]:STATus is set to On (the trigger mode is Triggered).

Conditions Measurement views: All

Group Trigger commands

Syntax

```
TRIGger[:SEQuence]:FORCed { OFF | ON | 0 | 1 }
TRIGger[:SEQuence]:FORCed?
```

Arguments `OFF` or `0` does not cause a manual trigger.

`ON` or `1` causes a manual trigger.

Examples `TRIGGER:SEQUENCE:FORCED ON` causes a manual trigger if the acquisition is armed, ready and waiting for a trigger.

TRIGger[:SEQuence]:IMMediate (No Query Form)

Causes a trigger immediately, skipping the event detection and delay. This command is valid when TRIGger[:SEQuence]:STATus is set to On (the trigger mode is Triggered).

Conditions Measurement views: All

Group Trigger commands

Syntax

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

Arguments None

Examples `TRIGGER:SEQUENCE:IMMEDIATE` causes a trigger immediately, skipping the event detection and delay.

TRIGger[:SEQuence]:STATus

Selects or queries the trigger mode (Free Run or Triggered).

Conditions Measurement views: All

Group Trigger commands

Syntax

```
TRIGger[:SEQuence]:STATus { OFF | ON | 0 | 1 }
TRIGger[:SEQuence]:STATus?
```

Arguments `OFF` or `0` selects the free-run mode.

`ON` or `1` selects the triggered mode.

Examples `TRIGGER:SEQUENCE:STATUS ON` selects the triggered mode.

TRIGger[:SEQuence]:TIME:DELay

Sets or queries the trigger delay time (after recognizing the event before actually declaring the trigger).

Conditions Measurement views: All

Group Trigger commands

Syntax

```
TRIGger[:SEQuence]:TIME:DELay <value>
TRIGger[:SEQuence]:TIME:DELay?
```

Arguments `<value>::=<NRf>` specifies the trigger delay time. Range: 0 to 60 s.

Examples `TRIGGER:SEQUENCE:TIME:DELAY 1.5` sets the trigger delay time to 1.5 s.

TRIGger[:SEQuence]:TIME:POSition

Sets or queries the trigger position (location of the trigger event within the acquisition record).

Conditions Measurement views: All

Group Trigger commands

Syntax

```
TRIGger[:SEQuence]:TIME:POSition <value>
TRIGger[:SEQuence]:TIME:POSition?
```

Arguments `<value>::=<NRf>` specifies the trigger position. Range: 0 to 100%.

Examples `TRIGGER:SEQUENCE:TIME:POSITION 20` sets the trigger position to 20% of the acquisition record.

UNIT:POWer

Selects or queries the fundamental unit of power. This command is equivalent to [SENSe]:POWer:UNITs.

Conditions Measurement views: All

Group Unit commands

Syntax

```
UNIT:POWer { DBM | DBV | VOLTs | WATTs | DBUW | DBW| DBUV |
DBMV | DBUA | DBUV_M | DBUA_M | AMPS }
UNIT:POWer?
```

Arguments The following table lists the arguments.

Table 2-44: Power units

Argument	Power unit
DBM	dBm
DBV	dBV
VOLTs	Volts
WATTs	Watts
DBUW	dBμW
DBW	dBW
DBUV	dBμV
DBMV	dBmV
DBUA	dBμA

Table 2-44: Power units (cont.)

Argument	Power unit
DBUV_M	dBμV/m
DBUA_M	dBμA/m
AMPS	Amps

NOTE. *Select dBμV/m or dBμA/m unit when using an antenna table.*

Examples

`UNIT:POWER DBM` specifies the fundamental unit of power as dBm.

*WAI (No Query Form)

Prevents the analyzer from executing further commands or queries until all pending operations finish. This command allows you to synchronize the operation of the analyzer with your application program. For the details, refer to *Synchronizing Execution* (See page 3-12.).

Conditions

Measurement views: All

Group

IEEE common commands

Syntax

`*WAI`

Related Commands

*OPC

Arguments

None

Status and Events

Status and Events

The SCPI interface in the analyzer includes a status and event reporting system that enables the user to monitor crucial events that occur in the instrument. The analyzer is equipped with four registers and one queue that conform to IEEE Std 488.2-1987. This section will discuss these registers and queues along with status and event processing.

Status and Event Reporting System

The following figure outlines the status and event reporting mechanism offered in the RSA6100A Series analyzers. It contains three major blocks

- Standard Event Status
- Operation Status
- Questionable Status (fan-out structure)

The processes performed in these blocks are summarized in the Status Byte. The three blocks contain four types of registers as shown in the following table.

Table 3-1: Register type

Register	Description
Condition register	Records event occurrence in the instrument. Read only.
Transition register (positive/negative)	A positive transition filter allows an event to be reported when a condition changes from false to true. A negative filter allows an event to be reported when a condition changes from true to false. Setting both positive and negative filters true allows an event to be reported anytime the condition changes. Clearing both filters disables event reporting.
Event register	Records events filtered by the transition register. Read only.
Enable register	Masks the event register to report in the summary bit. User-definable.

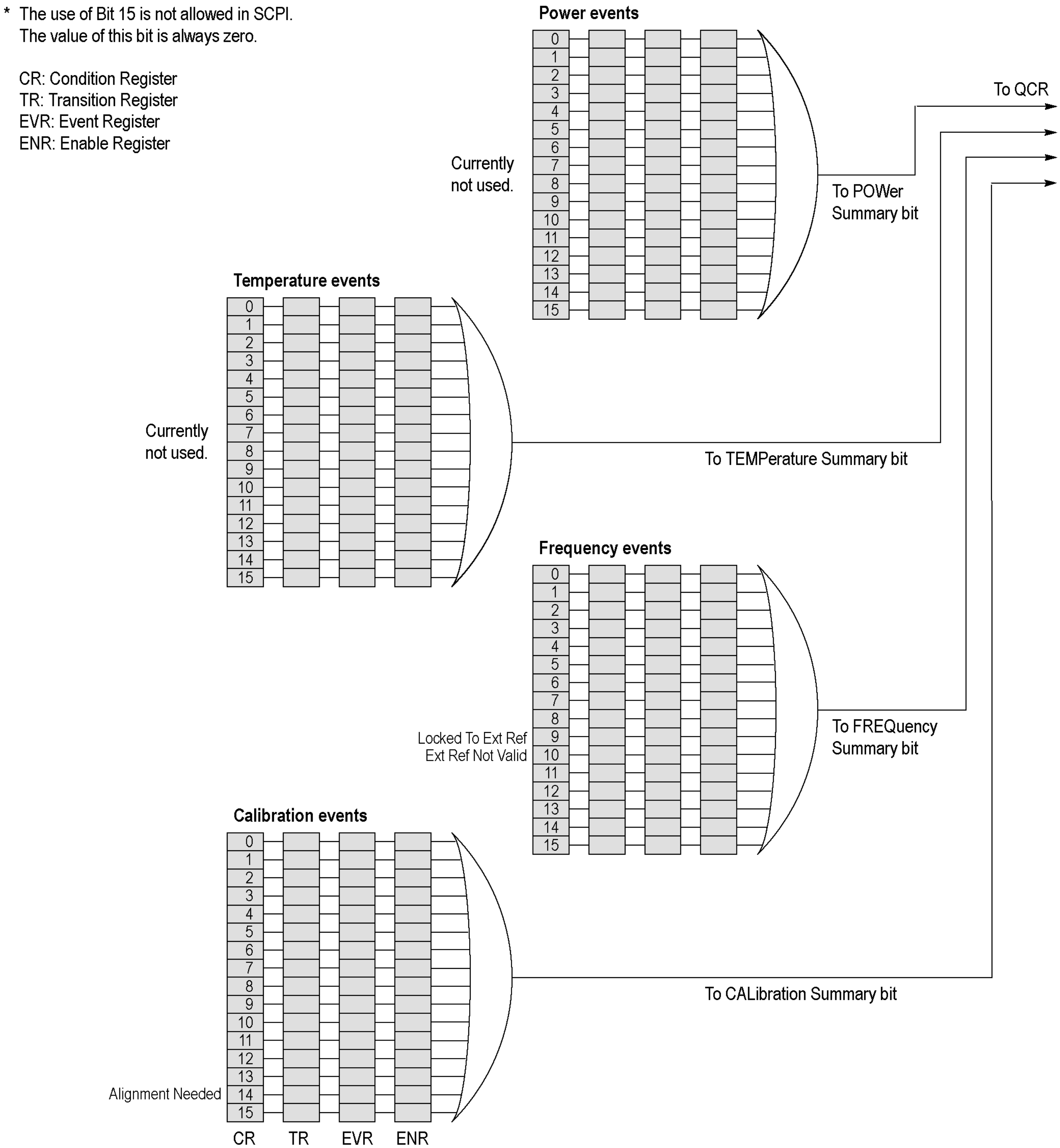


Figure 3-1: Status/Event reporting mechanism

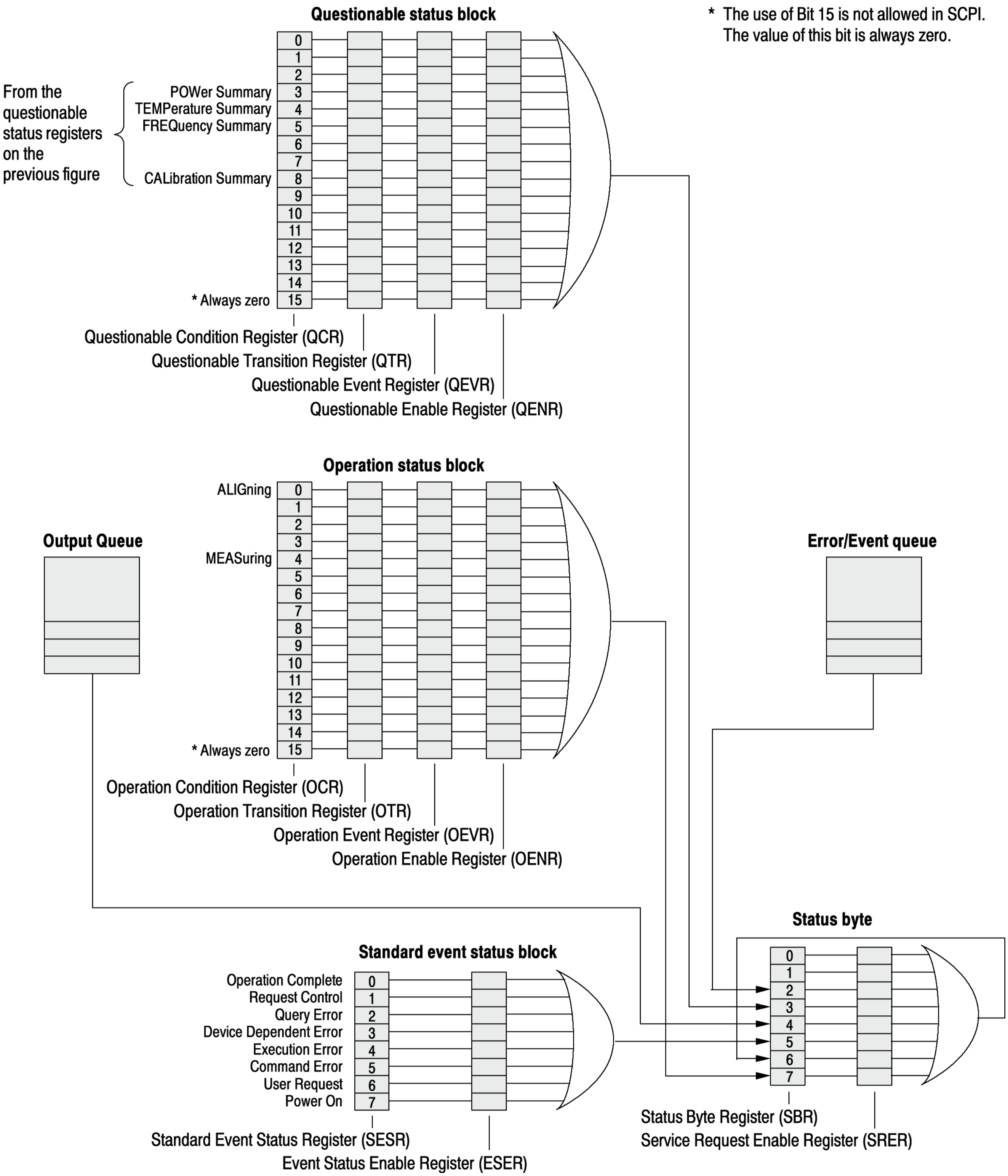


Figure 3-2: Status/Event reporting mechanism (Cont.)

Status Byte

The Status Byte contains the following two registers

- Status Byte Register (SBR)
- Service Request Enable Register (SRER)

Status Byte Register (SBR)

The SBR is made up of 8 bits. Bits 4, 5 and 6 are defined in accordance with IEEE Std 488.2-1987. These bits are used to monitor the output queue, SESR and service requests, respectively. The contents of this register are returned when the *STB? query is used.

7	6	5	4	3	2	1	0
OSS	6 RQS / 6 MSS	ESB	MAV	QSS	EAV	—	—

Figure 3-3: Status Byte Register (SBR)

Table 3-2: SBR bit functions

Bit	Description
7	Operation Summary Status (OSS). Summary of the operation status register.
6	Request Service (RQS)/Master Status Summary (MSS). When the instrument is accessed using the GPIB serial poll command, this bit is called the Request Service (RQS) bit and indicates to the controller that a service request has occurred (in other words, that the GPIB bus SRQ line is LOW). The RQS bit is cleared when serial poll ends. When the instrument is accessed using the *STB? query, this bit is called the Master Status Summary (MSS) bit and indicates that the instrument has issued a service request for one or more reasons. The MSS bit is never cleared to 0 by the *STB? query.
5	Event Status Bit (ESB). This bit indicates whether or not a new event has occurred after the previous Standard Event Status Register (SESR) has been cleared or after an event readout has been performed.
4	Message Available Bit (MAV). This bit indicates that a message has been placed in the output queue and can be retrieved.
3	Questionable Summary Status (QSS). Summary of the Questionable Status Byte register.
2	Event Quantity Available (EAV). Summary of the Error Event Queue.
1-0	Not used

Service Request Enable Register (SRER)

The SRER is made up of bits defined exactly the same as bits 0 through 7 in the SBR as shown in the following figure. This register is used by the user to determine what events will generate service requests.

The SRER bit 6 cannot be set. Also, the RQS is not maskable.

The generation of a service request with the GPIB interface involves changing the SRQ line to LOW and making a service request to the controller. The result is that a status byte for which an RQS has been set is returned in response to serial polling by the controller.

Use the *SRE command to set the bits of the SRER. Use the *SRE? query to read the contents of the SRER. Bit 6 must normally be set to 0.

7	6	5	4	3	2	1	0
OSB	—	ESB	MAV	QSB	—	—	—

Figure 3-4: Service Request Enable Register (SRER)

Standard Event Status Block

Reports the power on/off state, command errors, and the running state. It consists of the following registers

- Standard Event Status Register (SESR)
- Event Status Enable Register (ESER)

These registers are made up of the same bits defined in the following figure and table. Use the *ESR? query to read the contents of the SESR. Use the *ESE() command to access the ESER.

7	6	5	4	3	2	1	0
PON	—	CME	EXE	DDE	QYE	—	OPC

Figure 3-5: Standard event status register

Table 3-3: Standard event status register bit definition

Bit	Description
7	Power On (PON). Indicates that the power to the instrument is on.
6	Not used.
5	Command Error (CME). Indicates that a command error has occurred while parsing by the command parser was in progress.
4	Execution Error (EXE). Indicates that an error occurred during the execution of a command. Execution errors occur for one of the following reasons ■ When a value designated in the argument is outside the allowable range of the instrument, or is in conflict with the capabilities of the instrument. ■ When the command could not be executed properly because the conditions for execution differed from those essentially required.
3	Device-Dependent Error (DDE). An instrument error has been detected.
2	Query Error (QYE). Indicates that a query error has been detected by the output queue controller. Query errors occur for one of the following reasons ■ An attempt was made to retrieve messages from the output queue, despite the fact that the output queue is empty or in pending status. ■ The output queue messages have been cleared despite the fact that they have not been retrieved.
1	Not used.
0	Operation Complete (OPC). This bit is set with the results of the execution of the *OPC command. It indicates that all pending operations have been completed.

When an event occurs, the SESR bit corresponding to the event is set, resulting in the event being stacked in the Error/Event Queue. The SBR OAV bit is also set. If the bit corresponding to the event has also been set in the ESER, the SBR ESB bit is also set. When a message is sent to the Output Queue, the SBR MAV bit is set.

Operation Status Block

The operation status block contains conditions that are part of the instrument's normal operation. It consists of the following registers

- Operation Condition Register (OCR)
- Operation Positive/ Negative Transition Register (OPTR/ONTR)
- Operation Event Register (OEVR)
- Operation Enable Register (OENR)

These registers are made up of the same bits defined in the following table and figure. Use the STATus:OPERation commands to access the operation status register set.

15	14	13	12	11	10	9	8	7	6	5	4 MEAS	3	2	1	0 ALIG

Figure 3-6: Operation status register

Table 3-4: Operation status register bit definition

Bit	Description
15	Always zero (0).
14 - 5	Not used.
4	Measuring (MEAS). Indicates that the instrument is actively measuring. When the measurement ends after this bit is set in measurement, it is reset. "In measurement" means that one of the following commands is in execution: ■ INITiate commands ■ READ commands
3 - 1	Not used.
0	Aligning (ALIG). Indicates that the instrument is currently performing an alignment. When the alignment ends after this bit is set in alignment, it is reset.

When the specified state changes in the OCR, its bit is set or reset. This change is filtered with a transition register, and the corresponding bit of the OEVR is set. If the bit corresponding to the event has also been set in the OENR, the SBR OSS bit is also set.

Questionable Status Block

The questionable status register set contains bits which give an indication of the quality of various aspects of the signal together with the fanned out registers as described in the next subsections. It consists of the following registers

- Questionable Condition Register (QCR)
- Questionable Positive/Negative Transition Register (QPTR/QNTR)
- Questionable Event Register (QEVR)
- Questionable Enable Register (QENR)

These registers are made up of the same bits defined in the following table and figure. Use the STATus:QUEStionable commands to access the questionable status register set.

15	14 CW	13	12	11	10	9	8 CAL	7	6	5 FREQ	4 TEMP	3 POW	2	1	0

Figure 3-7: Questionable status register

Table 3-5: Questionable status register bit definition

Bit	Description
15	Always zero (0).
14	Command Warning (CW). Indicates a non-fatal warning that relates to the instrument's interpretation of a command, query, or one or more parameters of a specific command or query.
13 - 9	Not used.
8	CALibration Summary (CAL). Summary of the Questionable Calibration register.
7, 6	Not used.
5	FREQuency Summary (FREQ). Summary of the Questionable Frequency register.
4	TEMPerature Summary (TEMP). Summary of the Questionable Temperature register.
3	POWer Summary (POW). Summary of the Questionable Power register.
2 - 0	Not used.

When the specified state changes in the QCR, its bit is set or reset. This change is filtered with a transition register, and the corresponding bit of the QEVR is set. If the bit corresponding to the event has also been set in the QENR, the SBR QSS bit is also set.

Questionable Power Register Set

Refines the power error for the POWer bit in the QCR.

Currently not used.

Questionable Temperature Register Set

Refines the temperature error for the TEMPerature bit in the QCR.

Currently not used.

Questionable Frequency Register Set

The questionable frequency register set is made up of bits defined in the following table and figure. It refines the frequency error for the FREQuency bit in the QCR. Use the STATus:QUEStionable:FREQuency commands to access the questionable frequency register set.

15	14	13	12	11	10 ERNV	9 LTER	8	7	6	5	4	3	2	1	0

Figure 3-8: Questionable frequency status register

Table 3-6: Questionable frequency status register bit definition

Bit	Description
15	Always zero (0).
14 - 11	Not used.
10	External Ref Not Valid (ERNV). Indicates that the external reference signal is not valid so the instrument is no longer locked to it.
9	Locked To External Ref (LTEF). Indicates that the instrument is locked to the external reference signal.
8 - 0	Not used.

Questionable Calibration Register Set

The questionable calibration register set is made up of bits defined in the following table and figure. It refines the calibration error for the CALibration bit in the QCR. Use the STATus:QUEStionable:CALibration commands to access the questionable calibration register set.

15	14 ALN	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Figure 3-9: Questionable calibration status register

Table 3-7: Questionable calibration status register bit definition

Bit	Description
15	Always zero.
14	Alignment Needed (ALN). Indicates the instrument needs the alignment.
13 - 0	Not used.

Queues

There are two types of queues in the status reporting system used in the analyzer: output queues and event queues.

Output Queue

The output queue is a FIFO (first in, first out) queue and holds response messages to queries, where they await retrieval. When there are messages in the queue, the SBR MAV bit is set.

The output queue will be emptied each time a command or query is received, so the controller must read the output queue before the next command or query is issued. If this is not done, an error will occur and the output queue will be emptied; however, the operation will proceed even if an error occurs.

Event Queue

The event queue is a FIFO queue and stores events as they occur in the analyzer. If more than 32 events occur, event 32 will be replaced with event code -350 ("Queue Overflow"). The error code and text are retrieved using the SYSTem:ERRor queries.

Status and Event Processing Sequence

The following figure shows an outline of the sequence for status and event processing.

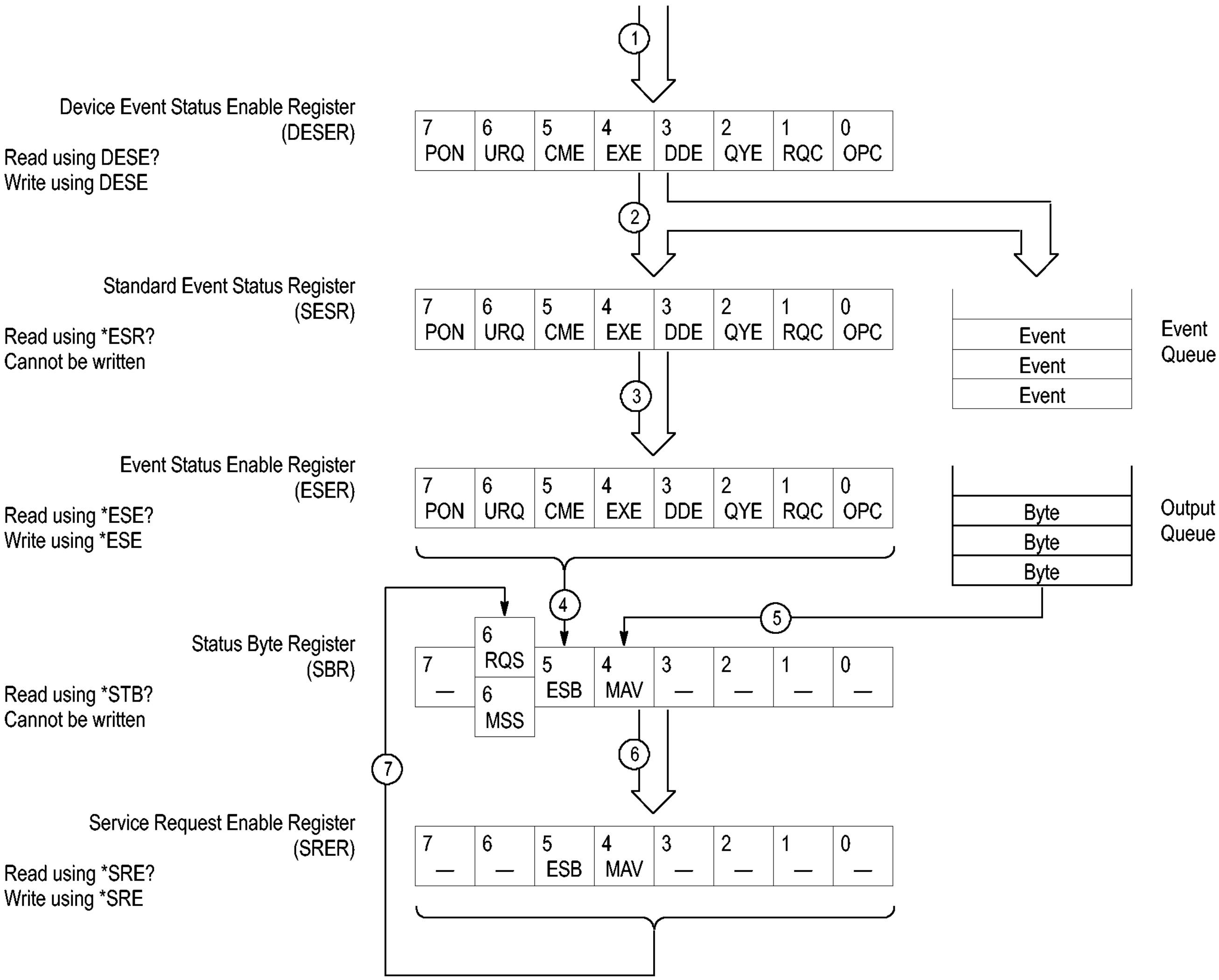


Figure 3-10: Status and event processing sequence

1. If an event has occurred, the SESR bit corresponding to that event is set and the event is placed in the event queue.
2. A bit corresponding to that event in the ESER has is set.
3. The SBR ESB bit is set to reflect the status of the ESER.
4. When a message is sent to the output queue, the SBR MAV bit is set.

5. Setting either the ESB or MAV bits in the SBR sets the respective bit in the SRER.

6. When the SRER bit is set, the SBR MSS bit is set and a service request is generated when using the GPIB interface.

Synchronizing Execution

Almost all commands are executed in the order in which they are sent from the controller, and the execution of each command is completed in a short period of time. However, the following commands perform data analysis in another thread, and another command can thus be executed concurrently

- INITiate commands
- READ commands
- [SENSe]:REANalyze

These commands are designed so that the next command to be sent is executed without waiting for the previous command to be completed. In some cases, a process executed by another command must first be completed before these commands can be executed; in other cases, these commands must be completed before the next command is executed.

You have two options to achieve command synchronization

- Using the status and event reporting function
- Using synchronizing commands

Using the Status and Event Reporting Function

In the following example, a READ command is used to obtain the measurement results while the Operation Condition Register (OCR) is being used to provide synchronization.

```
STATus:OPERation:NTRansition 16
    // Set the filter of the OCR MEASuring bit
STATus:OPERation:ENABle 16
    // Enable the filter of the OCR MEASuring bit
*SRE 128
    // Set the SRER OSS bit
READ:SPECtrum:TRACe
    // Obtain the measurement results
```

The command waits for generation of SRQ.

Using Synchronizing Commands

The IEEE-488.2 common commands include the following synchronizing commands

- *OPC
- *OPC?
- *WAI

Using the *OPC command. The *OPC command sets the SESR OPC bit when all the operations for which it is waiting are completed. If the GPIB interface is in use, you can synchronize the execution by using this command together with the serial polling or service request function.

The following is a command sequence example:

```
*ESE 1
    // Enable the ESER OPC bit
*SRE 32
    // Enable the SRER ESB bit
ABORt;INITiate:IMMediate;*OPC
    // Wait for SRQ to provide synchronization
```

Using the *OPC? query. The query *OPC? writes ASCII code "1" into the Output Queue when all operations for which it is waiting are completed. You can provide synchronization using the command string as the following example:

```
ABORt;INITiate:IMMediate;*OPC
```

The command waits until "1" is written into the Output Queue. When the command goes to the Output Queue to read the data, a time-out may occur before the data is written into the queue.

Using the *WAI Command. After the process of the preceding command is completed, the *WAI command begins to execute the process of the next command as the following example:

```
ABORt;INITiate:IMMediate;*WAI
    // Wait for the *WAI process to provide synchronization
```

Error Messages and Codes

Error codes with a negative value are SCPI standard error codes; errors with a positive value are unique to the RSA6100A Series Real-Time Spectrum Analyzers.

Event codes and messages can be obtained by using the queries `SYSTem:ERRor?` and `SYSTem:ERRor:ALL?` These are returned in the following format

```
<event_code>,"<event_message>"
```

Command Errors

Command errors are returned when there is a syntax error in the command.

Table 3-8: Command errors

Error code	Error message
-100	Command error
-101	Invalid character
-102	Syntax error
-103	Invalid separator
-104	Data type error
-105	GET not allowed
-108	Parameter not allowed
-109	Missing parameter
-110	Command header error
-111	Header separator error
-112	Program mnemonic too long
-113	Undefined header
-114	Header suffix out of range
-120	Numeric data error
-121	Character
-123	Exponent too large
-124	Too many digits
-128	Numeric data not allowed
-130	Suffix error
-131	Invalid suffix
-134	Suffix too long
-138	Suffix not allowed
-140	Character data error
-141	Invalid character data
-144	Character data too long

Table 3-8: Command errors (cont.)

Error code	Error message
-148	Character data not allowed
-150	String data error
-151	Invalid string data
-158	String data not allowed
-160	Block data error
-161	Invalid block data
-168	Block data not allowed
-170	Command expression error
-171	Invalid expression
-178	Expression data not allowed
-180	Macro error
-181	Invalid outside macro definition
-183	Invalid inside macro definition
-184	Macro parameter error

Execution Errors

These error codes are returned when an error is detected while a command is being executed.

Table 3-9: Execution errors

Error code	Error message
-200	Execution error
-201	Invalid while in local
-202	Settings lost due to RTL
-210	Trigger error
-211	Trigger ignored
-212	Arm ignored
-213	Init ignored
-214	Trigger deadlock
-215	Arm deadlock
-220	Parameter error
-221	Settings conflict
-222	Data out of range
-223	Too much data
-224	Illegal parameter value
-225	Out of memory

Table 3-9: Execution errors (cont.)

Error code	Error message
-226	Lists not same length
-230	Data corrupt or stale
-231	Data questionable
-240	Hardware error
-241	Hardware missing
-250	Mass storage error
-251	Missing mass storage
-252	Missing media
-253	Corrupt media
-254	Media full
-255	Directory full
-256	Filename not found
-257	Filename error
-258	Media protected
-260	Execution expression error
-261	Math error in expression
-270	Execution macro error
-271	Macro syntax error
-272	Macro execution error
-273	Illegal macro label
-274	Execution macro parameter error
-275	Macro definition too long
-276	Macro recursion error
-277	Macro redefinition not allowed
-278	Macro header not found
-280	Program error
-281	Cannot create program
-282	Illegal program name
-283	Illegal variable name
-284	Program currently running
-285	Program syntax error
-286	Program runtime error

Device Specific Errors

These error codes are returned when an internal instrument error is detected. This type of error may indicate a hardware problem.

Table 3-10: Device specific errors

Error code	Error message
-300	Device specific error
-310	System error
-311	Memory error
-312	PUD memory lost
-313	Calibration memory lost
-314	Save/Recall memory lost
-315	Configuration memory lost
-330	Self test failed
-350	Queue overflow

Query Errors

These error codes are returned in response to an unanswered query.

Table 3-11: Query errors

Error code	Error message
-400	Query error
-410	Query interrupted
-420	Query unterminated
-430	Query deadlocked
-440	Query unterminated after indefinite period

Device Errors

These error codes are unique to the RSA6100A Series. They are classified into three groups: global, measurement, and source conditions, as shown in the following tables.

Table 3-12: Device errors, global condition

Error code	Error message
100	Setup error
101	Disabled: data is from swept acquisition
102	Disabled: swept settings; Acquire data while display is selected
103	Acquisition bandwidth too small for current setup
104	Can’t get acquisition data record
105	Can’t open the requested display
106	Analysis failure
107	Analysis length was limited
108	Analysis length too small for current setup
109	No math trace: unmatched trace lengths
110	Analysis time was adjusted
111	Not enough samples for current setup
112	Can’t replay. Data is from swept acquisition.
113	Can’t replay. Live data needed for swept settings.
114	Recall error: setup not completely restored
115	Recall failure: problem with file or file contents
116	Save failure: file not saved
117	Unexpected software error. Please cycle power and try again.
118	Export failure: file not saved
119	Export failure: unable to open results file for export. File not saved.
120	Search condition for this result is already defined.
121	Search condition for this result was not found.
124	Load failed: <filename>
125	Store error: file not saved.
126	No Math trace: unmatched trace X range
127	Not enough memory for measurement
128	Incomplete analysis
129	Not enough samples for current setup
130	Mask creation error: <reason message>

Table 3-13: Device errors, measurement condition

Error code	Error message
1000	TDBW actual (TDBW: Time Domain Bandwidth)
1001	Average transmit not available in volts units
1002	RBW increased to
1003	RBW limited by acquisition bandwidth to
1004	RBW conflict. Increase span or analysis length
1005	Analysis stopped: ambiguous pulse shape
1006	Setup error: Phase measurement location.
1007	No pulses found
1008	No FFT (not all pulses have results)
1009	No burst detected
1011	Audio disabled: configuration problem
1012	Audio Demod disabled: swept acquisition
1013	Audio Demod disabled: trigger in use
1014	Audio disabled: IF band outside Acq BW
1015	Calibration error. See Windows Event Viewer for error detail.
OBW errors	
1016	Analysis failure: AcqBW < MeasBW + (5 x RBW)
1017	Analysis failure: AcqBW must be 10 kHz or more
1018	x dB BW > Meas BW
Pulse errors	
1019	AcqBW too low for current Chirp BW setting
1023	Not enough memory for measurement
Other measurements	
1024	BW actual (limited by Acq BW)
1025	CISPR not available in FastFrame. Uncheck FastFrame in the Acquire panel.
1026	Analysis length must be in auto.
1027	Carrier not found
1029	CISPR accuracy limited by acq memory. Adjust RBW or freq range.
1030	CISPR: Acq BW too small for RBW. Try increasing span or freq range.
1031	Insufficient data for CISPR. Acquire while display is selected.
1032	VBW increased - Analysis Length too short
1033	VBW does not use full Analysis Length.

Table 3-14: Device errors, source condition

Error code	Error message
2000	Data acquired during RF ADC overrange
2001	(internal use only)
2002	(internal use only)
2003	Preamp not specified for frequencies > 3 GHz
2004	Data acquired using preamp
2005	Aligning
2006	Not aligned
2007	Data from unaligned instrument
2008	Not calibrated
2009	Data from uncalibrated instrument
2010	Dither: manual control
2011	Hardware failure - see Windows Event Viewer
2012	Data acquired during hardware failure
2013	Hardware failure detected by diagnostics
2014	Data acquired during RF digital gain overflow
2015	Locking to external frequency reference signal
2016	Locked to external reference
2017	No_RF_Deck mode
2018	RF attenuator: manual control
2019	Saving acquisition data
2020	Restoring acquisition data
2021	Simulated data
2022	Disabled: data is from swept acquisition
2023	Disabled: frequency mask trigger in use
2024	Span > RF acquisition bandwidth
2025	RBW not valid for current acquisition bandwidth
2026	Acquisition sampling parameters: manual control
2027	Swept: RF trigger invalid for most signals
2028	External frequency reference signal not valid. Using internal reference.
2029	Unable to lock to external frequency reference. Using internal reference.
2030	Data acquired during RF ADC overrange
2031	Data acquired during RF digital gain overflow
2032	Source factory error
2033	Alignment error. Please run alignment again. If the problem persists, contact your Tektronix Service Center.
2034	Operational error: unable to complete operation. Please try again. If the problem persists, contact your Tektronix Service Center. See Windows Event Viewer for error detail.

Table 3-14: Device errors, source condition (cont.)

Error code	Error message
2035	Hardware error: unable to configure hardware. Please try again. If the problem persists, contact your Tektronix Service Center. See Windows Event Viewer for error detail.
2036	Shutting down - internal temperature is too high. Check fans and airflow. If the problem persists, contact your Tektronix Service Center.
2037	Hardware error detected. To clear error, exit and restart the application. If the problem persists, contact your Tektronix Service Center.
2038	Disabled: RefLev too low\nfor Volts/Watts units
2042	Ext Corr > 20 dB pk-pk in acq segment
2044	Combined External Correction tables exceed the 60 dB peak-to-peak limit. External Correction tables were disabled. Please check table values and try again.
2045	Disabled: FastFrame doesn’t support swept settings.
2046	Attenuator use currently exceeds the cautionary limit of 1200 changes per hour. Operation was stopped to protect against premature wear-out. The monitor function can be temporarily disabled in the Amplitude control panel or over the programmatic interface. Specified lifetime for an attenuator is 10,000,000 changes.
2047	Disabled: settings conflict with selected measurement

Appendices

Appendix A: Character Charts

B7 B6 B5 BITS	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
B4 B3 B2 B1	CONTROL		NUMBERS SYMBOLS		UPPER CASE		LOWER CASE	
0 0 0 0	0 **NUL** 0 0	20 **DLE** 10 16	40 LA0 **SP** 20 32	60 LA16 **0** 30 48	100 TA0 **@** 40 64	120 TA16 **P** 50 80	140 SA0 **`** 60 96	160 SA16 **p** 70 112
0 0 0 1	1 GTL **SOH** 1 1	21 LL0 **DC1** 11 17	41 LA1 **!** 21 33	61 LA17 **1** 31 49	101 TA1 **A** 41 65	121 TA17 **Q** 51 81	141 SA1 **a** 61 97	161 SA17 **q** 71 113
0 0 1 0	2 **STX** 2 2	22 **DC2** 12 18	42 LA2 **"** 22 34	62 LA18 **2** 32 50	102 TA2 **B** 42 66	122 TA18 **R** 52 82	142 SA2 **b** 62 98	162 SA18 **r** 72 114
0 0 1 1	3 **ETX** 3 3	23 **DC3** 13 19	43 LA3 **#** 23 35	63 LA19 **3** 33 51	103 TA3 **C** 43 67	123 TA19 **S** 53 83	143 SA3 **c** 63 99	163 SA19 **s** 73 115
0 1 0 0	4 SDC **EOT** 4 4	24 DCL **DC4** 14 20	44 LA4 **$** 24 36	64 LA20 **4** 34 52	104 TA4 **D** 44 68	124 TA20 **T** 54 84	144 SA4 **d** 64 100	164 SA20 **t** 74 116
0 1 0 1	5 PPC **ENQ** 5 5	25 PPU **NAK** 15 21	45 LA5 **%** 25 37	65 LA21 **5** 35 53	105 TA5 **E** 45 69	125 TA21 **U** 55 85	145 SA5 **e** 65 101	165 SA21 **u** 75 117
0 1 1 0	6 **ACK** 6 6	26 **SYN** 16 22	46 LA6 **&** 26 38	66 LA22 **6** 36 54	106 TA6 **F** 46 70	126 TA22 **V** 56 86	146 SA6 **f** 66 102	166 SA22 **v** 76 118
0 1 1 1	7 **BEL** 7 7	27 **ETB** 17 23	47 LA7 **'** 27 39	67 LA23 **7** 37 55	107 TA7 **G** 47 71	127 TA23 **W** 57 87	147 SA7 **g** 67 103	167 SA23 **w** 77 119
1 0 0 0	10 GET **BS** 8 8	30 SPE **CAN** 18 24	50 LA8 **(** 28 40	70 LA24 **8** 38 56	110 TA8 **H** 48 72	130 TA24 **X** 58 88	150 SA8 **h** 68 104	170 SA24 **x** 78 120
1 0 0 1	11 TCT **HT** 9 9	31 SPD **EM** 19 25	51 LA9 **)** 29 41	71 LA25 **9** 39 57	111 TA9 **I** 49 73	131 TA25 **Y** 59 89	151 SA9 **i** 69 105	171 SA25 **y** 79 121
1 0 1 0	12 **LF** A 10	32 **SUB** 1A 26	52 LA10 ***** 2A 42	72 LA26 **:** 3A 58	112 TA10 **J** 4A 74	132 TA26 **Z** 5A 90	152 SA10 **j** 6A 106	172 SA26 **z** 7A 122
1 0 1 1	13 **VT** B 11	33 **ESC** 1B 27	53 LA11 **+** 2B 43	73 LA27 **;** 3B 59	113 TA11 **K** 4B 75	133 TA27 **[** 5B 91	153 SA11 **k** 6B 107	173 SA27 **{** 7B 123
1 1 0 0	14 **FF** C 12	34 **FS** 1C 28	54 LA12 **,** 2C 44	74 LA28 **<** 3C 60	114 TA12 **L** 4C 76	134 TA28 **** 5C 92	154 SA12 **l** 6C 108	174 SA28 **¦** 7C 124
1 1 0 1	15 **CR** D 13	35 **GS** 1D 29	55 LA13 **–** 2D 45	75 LA29 **=** 3D 61	115 TA13 **M** 4D 77	135 TA29 **]** 5D 93	155 SA13 **m** 6D 109	175 SA29 **}** 7D 125
1 1 1 0	16 **SO** E 14	36 **RS** 1E 30	56 LA14 **.** 2E 46	76 LA30 **>** 3E 62	116 TA14 **N** 4E 78	136 TA30 **^** 5E 94	156 SA14 **n** 6E 110	176 SA30 **~** 7E 126
1 1 1 1	17 **SI** F 15	37 **US** 1F 31	57 LA15 **/** 2F 47	77 UNL **?** 3F 63	117 TA15 **O** 4F 79	137 UNT **_** 5F 95	157 SA15 **o** 6F 111	177 **RUBOUT (DEL)** 7F 127
	ADDRESSED COMMANDS	UNIVERSAL COMMANDS	LISTEN ADDRESSES		TALK ADDRESSES		SECONDARY ADDRESSES OR COMMANDS	

KEY

octal → 5 PPC ← GPIB code (with ATN asserted)
ENQ ← ASCII character
hex → 5 5 ← decimal

Tektronix
REF: ANSI STD X3.4-1977
IEEE STD 488.1-1987
ISO STD 646-2973

Appendix B: GPIB Interface Specification

This appendix lists and describes the GPIB functions and messages the instrument can implement.

Interface Functions

The following table lists the GPIB interface functions this instrument implements. Each function is briefly described.

Table B-1: GPIB interface function implementation

Interface function	Implemented subset	Capability
Source Handshake (SH)	SH1	Complete
Acceptor Handshake (AH)	AH1	Complete
Talker (T)	T6	Basic Talker, Serial Poll Unaddress if my-listen-address (MLA) No Talk Only mode
Listener (L)	L4	Basic Listener Unaddress if my talk address (MTA) No Listen Only mode
Service Request (SR)	SR1	Complete
Remote/Local (RL)	RL0	None
Parallel Poll (PP)	PP0	None
Device Clear (DC)	DC1	Complete
Device Trigger (DT)	DT0	None
Controller (C)	C0	None
Electrical Interface	E2	Three-state driver

- Source Handshake (SH). Enables a talking device to support the coordination of data transfer. The SH function controls the initiation and termination of data byte transfers.
- Acceptor Handshake (AH). Enables a listening device to coordinate data reception. The AH function delays data transfer initiation or termination until the listening device is ready to receive the next data byte.
- Talker (T). Enables a device to send device-dependent data over the interface. This capability is available only when the device is addressed to talk, and uses a one-byte address.
- Listener (L). Enables a device to receive device-dependent data over the interface. This capability is available only when the device is addressed to listen, and uses a one-byte address.
- Service Request (SR). Enables a device to assert an SRQ (Service Request) line to notify the controller when it requires service.
- Remote/Local (RL). Enables a device to respond to both the GTL (Go To Local) and LLO (Local Lock Out) interface messages.
- Parallel Poll (PP). Enables a device to respond to the following interface messages: PPC, PPD, PPE, and PPU, as well as to send out a status message when the ATN (Attention) and EOI (End or Identify) lines are asserted simultaneously.
- Device Clear (DC). Enables a device to be cleared or initialized, either individually, or as part of a group of devices.
- Device Trigger (DT). Enables a device to respond to the GET (Group Execute Trigger) interface message when acting as a listener.
- Controller (C). Enables a device that has this capability to send its address, universal commands, and addressed commands to other devices over the interface.
- Electrical Interface (E). Identifies the electrical interface driver type. The notation E1 means the electrical interface uses open collector drivers, E2 means the electrical interface uses three-state drivers.

Interface Messages

Table B-2: Standard interface messages

Message	Type [1]	Implemented
Device Clear (DCL)	UC	Yes
Local Lockout (LLO)	UC	No
Serial Poll Disable (SPD)	UC	Yes
Serial Poll Enable (SPE)	UC	Yes
Parallel Poll Unconfigure (PPU)	UC	No
Go To Local (GTL)	AC	Yes
Selected Device Clear (SDC)	AC	Yes
Group Execute Trigger (GET)	AC	No
Take Control (TCT)	AC	No
Parallel Poll Configure (PPC)	AC	No

1 UC: Universal command; AC: Address command

- Device Clear (DCL). Will clear (initialize) all devices on the bus that have a device clear function, whether or not the controller has addressed them.
- Local Lockout (LLO). Disables the return to local function.
- Serial Poll Disable (SPD). Changes all devices on the bus from the serial poll state to the normal operating state.
- Serial Poll Enable (SPE). Puts all bus devices that have a service request function into the serial poll enabled state. In this state, each device sends the controller its status byte, instead of its normal output, after the device receives its talk address on the data lines. This function may be used to determine which device sent a service request.
- Go To Local (GTL). Causes the listen-addressed device to switch from remote to local (front-panel) control.
- Select Device Clear (SDC). Clears or initializes all listen-addressed devices.
- Group Execute Trigger (GET). Triggers all applicable devices and causes them to initiate their programmed actions.
- Take Control (TCT). Allows the controller in charge to pass control of the bus to another controller on the bus.
- Parallel Poll Configure (PPC). Causes the listen-addressed device to respond to the secondary commands Parallel Poll Enable (PPE) and Parallel Poll Disable (PPD), which are placed on the bus following the PPC command. PPE enables a device with parallel poll capability to respond on a particular data line. PPD disables the device from responding to the parallel poll.

Appendix C: Factory Initialization Settings

The factory initialization settings provide a known state for the analyzer. The *RST command returns the instrument settings to the factory defaults. Factory initialization sets values as shown in the following tables in this section.

Table C-1: Factory initialization settings, IEEE common commands

Header	Default value
*ESE	0
*OPC	0
*SRE	0

Table C-2: Factory initialization settings, Calibration commands

Header	Default value
CALibration:AUTO	ON
CALibration:CORRection:EXTernal:EDIT<x>:LABel	Ext Gain Table n
CALibration:CORRection:EXTernal:EDIT<x>:STATe	OFF
CALibration:CORRection:EXTernal:GAIN[:MAGNitude]	30 dB
CALibration:CORRection:EXTernal:GAIN:STATe	OFF
CALibration:CORRection:EXTernal:PROBe:STATe	OFF

Table C-3: Factory initialization settings, Calculate commands

Header	Default value
CALCulate basic command subgroup	
CALCulate:MARKer:DENSity:EXCursion	50
CALCulate:MARKer:DENSity:SMOothing	5
CALCulate:MARKer:DENSity:THReshold	500
CALCulate:MARKer:MODE	ABSolute
CALCulate:MARKer:PEAK:EXCursion	6 dB
CALCulate:MARKer:PEAK:THReshold	-150 dBm
CALCulate:SEARch:LIMit:MATCh:BEEP[:STATe]	OFF
CALCulate:SEARch:LIMit:MATCh:SACQuire[:STATe]	OFF
CALCulate:SEARch:LIMit:MATCh:SDATa[:STATe]	OFF
CALCulate:SEARch:LIMit:MATCh:SPICture[:STATe]	OFF
CALCulate:SEARch:LIMit:MATCh:STRace[:STATe]	OFF
CALCulate:SEARch:LIMit:OPERation	GT

Table C-3: Factory initialization settings, Calculate commands (cont.)

Header	Default value
CALCulate:SEARch:LIMit:OPERation:FEED	"Spectrum", "Trace 1"
CALCulate:SEARch:LIMit:OPERation:SLIMit	-20 dBm
CALCulate:SEARch:LIMit:STATe	OFF
CALCulate:DPSA subgroup	
CALCulate:DPSA:MARKer<x>:TRACe	TRACE1
CALCulate:DPSA:MARKer<x>:X:AMPLitude	0 dBm
CALCulate:DPSA:MARKer<x>:X[:FREQuency]	1.5 GHz
CALCulate:SPECtrum subgroup	
CALCulate:SPECtrum:MARKer<x>:TRACe	TRACE1

Table C-4: Factory initialization settings, Display commands

Header	Default value
DISPlay basic command subgroup	
DISPlay:WINDow:COLor:SCHeme	CLASsic
DISPlay:ACPower subgroup	
DISPlay:ACPower:MARKer:SHOW:STATe	ON
DISPlay:ACPower:PLEVel:SHOW:STATe	ON
DISPlay:ACPower:WINDow:TRACe:GRATicule:GRID:STATe	ON
DISPlay:ACPower:X[:SCALe]	13.84 MHz
DISPlay:ACPower:X[:SCALe]:OFFSet	1.5 GHz
DISPlay:ACPower:Y[:SCALe]	100 dB
DISPlay:ACPower:Y[:SCALe]:OFFSet	0
DISPlay:AVTime subgroup	
DISPlay:AVTime:MARKer:SHOW:STATe	ON
DISPlay:AVTime:TRIGger:LEVel:STATe	ON
DISPlay:AVTime:WINDow:TRACe:GRATicule:GRID:STATe	ON
DISPlay:AVTime:X[:SCALe]:AUTO:STATe	ON
DISPlay:AVTime:X[:SCALe]:OFFSet	0
DISPlay:AVTime:Y[:SCALe]:FULL	100 dB
DISPlay:AVTime:Y[:SCALe]:OFFSet	0
DISPlay:CONSte subgroup	
DISPlay:CONSte:WINDow:TRACe:GRATicule:GRID:STATe	ON
DISPlay:DDEMod subgroup	
DISPlay:DDEMod:RADix	BINary

Table C-4: Factory initialization settings, Display commands (cont.)

Header	Default value
DISPlay:DDEMod:X[:SCALe]	128 symbols
DISPlay:DDEMod:X[:SCALe]:AUTO:STATe	ON
DISPlay:DDEMod:X[:SCALe]:OFFSet	-1 symbol
DISPlay:DPSA subgroup	
DISPlay:DPSA:WINDow:TRACe:GRATicule:GRID:STATe	ON
DISPlay:DPSA:Y[:SCALe]:PDIVision	10 dB/div
DISPlay:EVM subgroup	
DISPlay:EVM:Y[:SCALe]	1%
DISPlay:EVM:Y[:SCALe]:OFFSet	0
DISPlay:FVTime subgroup	
DISPlay:FVTime:WINDow:TRACe:GRATicule:GRID:STATe	ON
DISPlay:FVTime:X[:SCALe]:AUTO:STATe	ON
DISPlay:FVTime:X[:SCALe]:OFFSet	0
DISPlay:FVTime:Y[:SCALe]	10 Hz
DISPlay:FVTime:Y[:SCALe]:OFFSet	0
DISPlay:GENeral subgroup	
DISPlay:GENeral:MEASview:SELect	SPECtrum
DISPlay:IQVTime subgroup	
DISPlay:IQVTime:WINDow:TRACe:GRATicule:GRID:STATe	ON
DISPlay:IQVTime:X[:SCALe]:AUTO:STATe	ON
DISPlay:IQVTime:X[:SCALe]:OFFSet	0
DISPlay:IQVTime:Y[:SCALe]	1 μV
DISPlay:IQVTime:Y[:SCALe]:OFFSet	0
DISPlay:MCPower subgroup	
DISPlay:MCPower:MARKer:SHOW:STATe	ON
DISPlay:MCPower:PLEVel:SHOW:STATe	ON
DISPlay:MCPower:WINDow:TRACe:GRATicule:GRID:STATe	ON
DISPlay:MCPower:X[:SCALe]	38.84 MHz
DISPlay:MCPower:X[:SCALe]:OFFSet	1.5 GHz
DISPlay:MCPower:Y[:SCALe]	100 dB
DISPlay:MCPower:Y[:SCALe]:OFFSet	0
DISPlay:MERRor subgroup	
DISPlay:MERRor:Y[:SCALe]	1%
DISPlay:MERRor:Y[:SCALe]:OFFSet	0
DISPlay:OBWidth subgroup	
DISPlay:OBWidth:MARKer:SHOW:STATe	ON
DISPlay:OBWidth:SELected:BANDwidth	OBWidth

Table C-4: Factory initialization settings, Display commands (cont.)

Header	Default value
DISPlay:OBWidth:WINDow:TRACe:GRATicule:GRID:STATe	ON
DISPlay:OBWidth:X[:SCALe]:OFFSet	0
DISPlay:OBWidth:Y[:SCALe]	100 dB
DISPlay:OBWidth:Y[:SCALe]:OFFSet	0
DISPlay:PERRor subgroup	
DISPlay:PERRor:Y[:SCALe]	1(degree)
DISPlay:PERRor:Y[:SCALe]:OFFSet	0
DISPlay:PHVTime subgroup	
DISPlay:PHVTime:WINDow:TRACe:GRATicule:GRID:STATe	ON
DISPlay:PHVTime:X[:SCALe]:AUTO:STATe	ON
DISPlay:PHVTime:X[:SCALe]:OFFSet	0
DISPlay:PHVTime:Y[:SCALe]	1°
DISPlay:PHVTime:Y[:SCALe]:AXIS	MODulopi
DISPlay:PHVTime:Y[:SCALe]:AXIS:REFerence	0
DISPlay:PHVTime:Y[:SCALe]:OFFSet	0
DISPlay:PNOise subgroup	
DISPlay:PNOise:MARKer:SHOW:STATe	ON
DISPlay:PNOise:WINDow:TRACe:GRATicule:GRID:STATe	ON
DISPlay:PNOise:X[:SCALe]:STARt	10 Hz
DISPlay:PNOise:X[:SCALe]:STOP	1 GHz
DISPlay:PNOise:Y[:SCALe]	100 dB
DISPlay:PNOise:Y[:SCALe]:OFFSet	-50 dBc/Hz
DISPlay:PNOise:Y[:SCALe]:PDIVision	10 dB
DISPlay:PULSe subgroup	
DISPlay:PULSe:RESult:ATX	OFF
DISPlay:PULSe:RESult:AVERage	ON
DISPlay:PULSe:RESult:DROop	OFF
DISPlay:PULSe:RESult:DUTPct	OFF
DISPlay:PULSe:RESult:DUTRatio	OFF
DISPlay:PULSe:RESult:FALL	OFF
DISPlay:PULSe:RESult:FRDeviation	OFF
DISPlay:PULSe:RESult:MFReqerror	OFF
DISPlay:PULSe:RESult:MPHerror	OFF
DISPlay:PULSe:RESult:PHDeviation	OFF
DISPlay:PULSe:RESult:PPFRequency	OFF
DISPlay:PULSe:RESult:PPOWer	OFF
DISPlay:PULSe:RESult:PPPHase	OFF

Table C-4: Factory initialization settings, Display commands (cont.)

Header	Default value
DISPlay:PULSe:RESult:RIPPle	OFF
DISPlay:PULSe:RESult:RMSFreqerror	OFF
DISPlay:PULSe:RESult:RMSPherror	OFF
DISPlay:PULSe:RESult:RRATe	OFF
DISPlay:PULSe:RESult:RINTerval	OFF
DISPlay:PULSe:RESult:RISE	OFF
DISPlay:PULSe:RESult:TIME	OFF
DISPlay:PULSe:RESult:WIDTh	OFF
DISPlay:PULSe:SELect:NUMBer	0
DISPlay:PULSe:SELect:RESult	AVERage
DISPlay:PULSe:STATistics:MARKer:SHOW:STATe	ON
DISPlay:PULSe:STATistics:PLOT	TRENd
DISPlay:PULSe:STATistics:WINDow:TRACe:GRATicule:GRID:STATe	ON
DISPlay:PULSe:STATistics:X[:SCALe]:NUMBer	1
DISPlay:PULSe:STATistics:X[:SCALe]:OFFSet	0
DISPlay:PULSe:STATistics:Y[:SCALe]:OFFSet	0
DISPlay:PULSe:STATistics:Y[:SCALe]:FULL	100 dB
DISPlay:PULSe:STATistics:Y[:SCALe]:OFFSet	0
DISPlay:PULSe:TRACe:MARKer:SHOW:STATe	ON
DISPlay:PULSe:TRACe:POINt:SHOW	ON
DISPlay:PULSe:TRACe:WINDow:TRACe:GRATicule:GRID:STATe	ON
DISPlay:PULSe:TRACe:X[:SCALe]:FULL	SELected
DISPlay:PULSe:TRACe:X[:SCALe]:OFFSet	0
DISPlay:PULSe:TRACe:X[:SCALe]:PDIVision	10 μs
DISPlay:PULSe:TRACe:Y[:SCALe]:FULL	100 dB
DISPlay:PULSe:TRACe:Y[:SCALe]:OFFSet	0
DISPlay:SGRam subgroup	
DISPlay:SGRam:TIME:OFFSet	0
DISPlay:SGRam:TIME:OVERlap	ON
DISPlay:SGRam:TIME:SCAle	0
DISPlay:SPECtrum subgroup	
DISPlay:SPECtrum:FREQuency:OFFSet	1.5 GHz
DISPlay:SPECtrum:FREQuency[:SCAle]	40 MHz
DISPlay:SPECtrum:MARKer:NOISe:MODE	OFF
DISPlay:SPECtrum:SCALe:LOG:STATe	OFF
DISPlay:SPECtrum:WINDow:TRACe:GRATicule:GRID:STATe	ON
DISPlay:SPECtrum:WINDow:TRACe:LEGend:STATe	OFF

Table C-4: Factory initialization settings, Display commands (cont.)

Header	Default value
DISPlay:SPECtrum:X:LABel	CFSPan
DISPlay:SPECtrum:Y[:SCALe]:PDIVision	10 dB/div
DISPlay:SPURious subgroup	
DISPlay:SPURious:MARKer:SHOW:STATe	ON
DISPlay:SPURious:SCALe:LOG:STATe	OFF
DISPlay:SPURious:SELect:NUMBer	1
DISPlay:SPURious:SHOW:LIMit	SHADed
DISPlay:SPURious:WINDow:TRACe:GRATicule:GRID:STATe	ON
DISPlay:SPURious:X[:SCALe]:STARt	1.61 GHz
DISPlay:SPURious:X[:SCALe]:STOP	2.16 GHz
DISPlay:SPECtrum:Y[:SCALe]	100 dB
DISPlay:SPECtrum:Y[:SCALe]:OFFSet	0
DISPlay:TOVerview subgroup	
DISPlay:TOVerview:WINDow:TRACe:GRATicule:GRID:STATe	ON
DISPlay:TOVerview:X[:SCALe]:OFFSet	0
DISPlay:TOVerview:Y[:SCALe]	100 dB
DISPlay:TOVerview:Y[:SCALe]:OFFSet	0

Table C-5: Factory initialization settings, Initiate commands

Header	Default value
INITiate:CONTinuous	ON

Table C-6: Factory initialization settings, Input commands

Header	Default value
INPut[:RF]:ATTenuation	20 dB
INPut[:RF]:ATTenuation:AUTO	ON
INPut[:RF]:ATTenuation:MONitor:STATe	ON
INPut[:RF]:GAIN:STATe	OFF
INPut:{MLEVel|RLEVel}	0 dBm

Table C-7: Factory initialization settings, Output commands

Header		Default value
OUTPut:IF:{BANDwidth|BWIDth}	(Option 05 only)	FLATtop
OUTPut:IF[:STATe]	(Option 05 only)	OFF
OUTPut:IQ[:STATe]	(Option 05 only)	OFF
OUTPut:NOISe[:STATe]		OFF

Table C-8: Factory initialization settings, Sense commands

Header	Default value
[SENSe] basic command subgroup	
[SENSe]:ACQuisition:FFRame:LIMit	1
[SENSe]:ACQuisition:FFRame:STATe	OFF
[SENSe]:ACQuisition:MODE	AUTO
[SENSe]:ANALysis:ADVanced:DITHer	AUTO
[SENSe]:ANALysis:LENGth	7.44 μs
[SENSe]:ANALysis:LENGth:AUTO	ON
[SENSe]:ANALysis:REFerence	ACQSTART
[SENSe]:ANALysis:STARt	0
[SENSe]:ANALysis:STARt:AUTO	ON
[SENSe]:POWer:UNITs	DBM
[SENSe]:ROSCillator:SOURce	INTernal
[SENSe]:SPECtrum:LENGth	7.44 μs
[SENSe]:SPECtrum:LENGth:AUTO	ON
[SENSe]:SPECtrum:STARt	0
[SENSe]:SPECtrum:TIMe:MODe	COMMon
[SENSe]:ACPower subgroup	
[SENSe]:ACPower:AVERage	OFF
[SENSe]:ACPower:AVERage:COUNt	10
[SENSe]:ACPower:{BANDwidth|BWIDth}[:RESolution]	30 kHz
[SENSe]:ACPower:{BANDwidth|BWIDth}[:RESolution]:AUTO	ON
[SENSe]:ACPower:{BANDwidth|BWIDth}:VIDeo	300 kHz
[SENSe]:ACPower:{BANDwidth|BWIDth}:VIDeo:STATe	OFF
[SENSe]:ACPower:CHANnel:{BANDwidth|BWIDth}	3.84 MHz
[SENSe]:ACPower:CHANnel:FILTer	RRCosine
[SENSe]:ACPower:CHANnel:PAIRs	1
[SENSe]:ACPower:CHANnel:SPACing	5 MHz
[SENSe]:ACPower:CHIPrate	3.84 MHz

Table C-8: Factory initialization settings, Sense commands (cont.)

Header	Default value
[SENSe]:ACPower:FREQuency	1.5 GHz
[SENSe]:ACPower:FREQuency:STEP	1 MHz
[SENSe]:ACPower:FREQuency:STEP:AUTO	ON
[SENSe]:ACPower:NFLoor:STATe	OFF
[SENSe]:ACPower:OPTimize:SPAN	DRANge
[SENSe]:ACPower:RRCRolloff	0.22
[SENSe]:AVTime subgroup	
[SENSe]:AVTime:{BANDwidth|BWIDth}	1 MHz
[SENSe]:AVTime:MAXTracepoints	HUNDredk
[SENSe]:AVTime:METHod	SPAN
[SENSe]:AVTime:SPAN	40 MHz
[SENSe]:CCDF subgroup	
[SENSe]:CCDF:{BANDwidth|BWIDth}	40 MHz
[SENSe]:CCDF:TIME:TYPE	SINGLe
[SENSe]:DDEMod subgroup	
[SENSe]:DDEMod:ANALysis:LENGth	128 symbols
[SENSe]:DDEMod:ANALysis:LENGth:AUTO	ON
[SENSe]:DDEMod:BURSt:DETect	OFF
[SENSe]:DDEMod:BURSt:THReshold	-20 dBc
[SENSe]:DDEMod:CARRier:OFFSet	0
[SENSe]:DDEMod:FILTer:ALPHa	0.22
[SENSe]:DDEMod:FILTer:MEASurement	RRCosine
[SENSe]:DDEMod:FILTer:REFerence	RCOSine
[SENSe]:DDEMod:MAGNitude:NORMalize	RSYMbol
[SENSe]:DDEMod:MODulation:TYPE	QPSK
[SENSe]:DDEMod:SRATe	3.84 MHz
[SENSe]:DDEMod:SWAP:IQ	OFF
[SENSe]:DDEMod:SYMBol:POINts	FOUR
[SENSe]:DDEMod:SYNCh:WORD	OFF
[SENSe]:DDEMod:SYNCh:WORD:SYMBol	#10
[SENSe]:DDEMod:TIMe:UNITs	SYMBols
[SENSe]:DPSA subgroup	
[SENSe]:DPSA:AUDio:DEMod:GAIN	3
[SENSe]:DPSA:AUDio:DEMod:RXBWidth	30 kHz
[SENSe]:DPSA:AUDio:DEMod:STATe	OFF
[SENSe]:DPSA:AUDio:DEMod:TUNE	SMARker
[SENSe]:DPSA:AUDio:DEMod:TYPE	FM

Table C-8: Factory initialization settings, Sense commands (cont.)

Header	Default value
[SENSe]:DPSA:{BANDwidth|BWIDth}[:RESolution]	400 kHz
[SENSe]:DPSA:{BANDwidth|BWIDth}[:RESolution]:AUTO	ON
[SENSe]:DPSA:COLor	TEMPerature
[SENSe]:DPSA:COLor:MAXimum	100%
[SENSe]:DPSA:COLor:MINimum	0%
[SENSe]:DPSA:FREQuency:CENTer	1.5 GHz
[SENSe]:DPSA:FREQuency:SPAN	40 MHz
[SENSe]:DPSA:FREQuency:STARt	1.48 GHz
[SENSe]:DPSA:FREQuency:STEP	2 MHz
[SENSe]:DPSA:FREQuency:STEP:AUTO	ON
[SENSe]:DPSA:FREQuency:STOP	1.52 GHz
[SENSe]:FVTime subgroup	
[SENSe]:FVTime:FREQuency:CENTer	1.5 GHz
[SENSe]:FVTime:FREQuency:SPAN	40 MHz
[SENSe]:FVTime:FREQuency:STARt	1.48 GHz
[SENSe]:FVTime:FREQuency:STEP	2 MHz
[SENSe]:FVTime:FREQuency:STEP:AUTO	ON
[SENSe]:FVTime:FREQuency:STOP	1.52 GHz
[SENSe]:FVTime:MAXTracepoints	HUNDredk
[SENSe]:IQVTime subgroup	
[SENSe]:IQVTime:FREQuency:CENTer	1.5 GHz
[SENSe]:IQVTime:FREQuency:SPAN	40 MHz
[SENSe]:IQVTime:FREQuency:STARt	1.48 GHz
[SENSe]:IQVTime:FREQuency:STEP	2 MHz
[SENSe]:IQVTime:FREQuency:STEP:AUTO	ON
[SENSe]:IQVTime:FREQuency:STOP	1.52 GHz
[SENSe]:IQVTime:MAXTracepoints	HUNDredk
[SENSe]:MCPower subgroup	
[SENSe]:MCPower:AVERage	OFF
[SENSe]:MCPower:AVERage:COUNt	10
[SENSe]:MCPower:{BANDwidth|BWIDth}[:RESolution]	30 kHz
[SENSe]:MCPower:{BANDwidth|BWIDth}[:RESolution]:ACTual?	ON
[SENSe]:MCPower:{BANDwidth|BWIDth}[:RESolution]:AUTO	300 kHz
[SENSe]:MCPower:{BANDwidth|BWIDth}:VIDeo	OFF
[SENSe]:MCPower:{BANDwidth|BWIDth}:VIDeo:STATe	3.84 MHz
[SENSe]:MCPower:CHANnel:FILTer	RRCosine
[SENSe]:MCPower:CHANnel:MAIN:{BANDwidth|BWIDth}	3.84 MHz

Table C-8: Factory initialization settings, Sense commands (cont.)

Header	Default value
[SENSe]:MCPower:CHANnel:MAIN:COUNt	4
[SENSe]:MCPower:CHANnel:MAIN:INACtive	NONE
[SENSe]:MCPower:CHANnel:MAIN:SPACing	5 MHz
[SENSe]:MCPower:CHIPrate	3.84 MHz
[SENSe]:MCPower:FREQuency	1.5 GHz
[SENSe]:MCPower:FREQuency:STEP	2 MHz
[SENSe]:MCPower:FREQuency:STEP:AUTO	ON
[SENSe]:MCPower:NFLoor:STATe	OFF
[SENSe]:MCPower:OPTimize:SPAN	DRANge
[SENSe]:MCPower:RRCRolloff	0.22
[SENSe]:OBWidth subgroup	
[SENSe]:OBWidth:AVERage	OFF
[SENSe]:OBWidth:AVERage:COUNt	10
[SENSe]:OBWidth:{BANDwidth|BWIDth}:MEASurement	10 MHz
[SENSe]:OBWidth:{BANDwidth|BWIDth}[:RESolution]	33 kHz
[SENSe]:OBWidth:{BANDwidth|BWIDth}[:RESolution]:AUTO	ON
[SENSe]:OBWidth:{BANDwidth|BWIDth}:VIDeo	300 kHz
[SENSe]:OBWidth:{BANDwidth|BWIDth}:VIDeo:STATe	OFF
[SENSe]:OBWidth:FREQuency:CENTer	1.5 GHz
[SENSe]:OBWidth:FREQuency:STEP	2 MHz
[SENSe]:OBWidth:FREQuency:STEP:AUTO	ON
[SENSe]:OBWidth:PERCent	99%
[SENSe]:OBWidth:XDBLevel	-26 dB
[SENSe]:PHVTime subgroup	
[SENSe]:PHVTime:FREQuency:CENTer	1.5 GHz
[SENSe]:PHVTime:FREQuency:SPAN	40 MHz
[SENSe]:PHVTime:FREQuency:STARt	1.48 GHz
[SENSe]:PHVTime:FREQuency:STEP	2 MHz
[SENSe]:PHVTime:FREQuency:STEP:AUTO	ON
[SENSe]:PHVTime:FREQuency:STOP	1.52 GHz
[SENSe]:PHVTime:MAXTracepoints	HUNDredk
[SENSe]:PNOise subgroup	
[SENSe]:PNOise:AVERage:COUNt	10
[SENSe]:PNOise:AVERage:ENABle	OFF
[SENSe]:PNOise:CARRier:FREQuency:TRACk	ON
[SENSe]:PNOise:CARRier:THReshold	-26 dBm
[SENSe]:PNOise:FREQuency:INTegration:OFFSet:STARt	100 Hz

Table C-8: Factory initialization settings, Sense commands (cont.)

Header	Default value
[SENSe]:PNOise:FREQuency:INTegration:OFFSet:STOP	10 MHz
[SENSe]:PNOise:FREQuency:PLOT:OFFSet:STARt	10 Hz
[SENSe]:PNOise:FREQuency:PLOT:OFFSet:STOP	100 MHz
[SENSe]:PNOise:OPTimization	SPEed
[SENSe]:PULSe subgroup	
[SENSe]:PULSe:ANALyze:LEVel	VOLTage
[SENSe]:PULSe:ANALyze:LEVel:FIFTy	VOLTage
[SENSe]:PULSe:ANALyze:LEVel:HUNDred	AVERage
[SENSe]:PULSe:ANALyze:MEASurement:TIME:AUTO	ON
[SENSe]:PULSe:ANALyze:MEASurement:TIME:STARt	0
[SENSe]:PULSe:ANALyze:MEASurement:TIME:STOP	0
[SENSe]:PULSe:ANALyze:PMLocation	500 ns
[SENSe]:PULSe:ANALyze:POINt:LOCation	AVERage
[SENSe]:PULSe:ANALyze:RFALl	WIDE
[SENSe]:PULSe:ANALyze:RIPPle	50%
[SENSe]:PULSe:CARRier:OFFSet	0
[SENSe]:PULSe:CARRier:SEARch	AUTO
[SENSe]:PULSe:DETect:MEASurement	OFF
[SENSe]:PULSe:DETect:NUMBer	100
[SENSe]:PULSe:DETect:POWer[:THReshold]	-10 dBc
[SENSe]:PULSe:DETect:TIMe[:THReshold]	1 ns
[SENSe]:PULSe:FILTer:{BANDwidth|BWIDth}	20 MHz (Standard), 60 MHz (Option 110)
[SENSe]:PULSe:FILTer:MEASurement	GAUSsian
[SENSe]:PULSe:FREFerence:AUTO	ON
[SENSe]:PULSe:FREFerence:CHIRpbw	1 MHz
[SENSe]:PULSe:FREFerence:OFFSet	0
[SENSe]:PULSe:MODulation:TYPE	CWConst
[SENSe]:PULSe:SIGNal:TYPE	CWConst
[SENSe]:SGRam subgroup	
[SENSe]:SGRam:{BANDwidth|BWIDth}:OPTimization	AUTO
[SENSe]:SGRam:{BANDwidth|BWIDth}:RESolution	300 kHz
[SENSe]:SGRam:{BANDwidth|BWIDth}[:RESolution]:AUTO	ON
[SENSe]:SGRam:{BANDwidth|BWIDth}[:RESolution]:MODE	ON
[SENSe]:SGRam:{BANDwidth|BWIDth}:VIDeo	300 kHz

Table C-8: Factory initialization settings, Sense commands (cont.)

Header	Default value
[SENSe]:SGRam:{BANDwidth|BWIDth}:VIDeo:STATe	OFF
[SENSe]:SGRam:COLor	TEMPerature
[SENSe]:SGRam:COLor:MAXimum	0 dBm
[SENSe]:SGRam:COLor:MINimum	-100 dBm
[SENSe]:SGRam:FFT:WINDow	KAISer
[SENSe]:SGRam:FILTer[:SHAPe]	KAISer
[SENSe]:SGRam:FREQuency:CENTer	1.5 GHz
[SENSe]:SGRam:FREQuency:SPAN	40 MHz
[SENSe]:SGRam:FREQuency:SPAN:BANDwidth[:RESolution]:RATio	100
[SENSe]:SGRam:FREQuency:STARt	1.48 GHz
[SENSe]:SGRam:FREQuency:STEP	2 MHz
[SENSe]:SGRam:FREQuency:STEP:AUTO	ON
[SENSe]:SGRam:FREQuency:STOP	1.52 GHz
[SENSe]:SPECtrum subgroup	
[SENSe]:SPECtrum:{BANDwidth|BWIDth}:OPTimization	AUTO
[SENSe]:SPECtrum:{BANDwidth|BWIDth}[:RESolution]	300 kHz
[SENSe]:SPECtrum:{BANDwidth|BWIDth}[:RESolution]:AUTO	ON
[SENSe]:SPECtrum:{BANDwidth|BWIDth}[:RESolution]:MODE	ON
[SENSe]:SPECtrum:{BANDwidth|BWIDth}:VIDeo	300 kHz
[SENSe]:SPECtrum:{BANDwidth|BWIDth}:VIDeo:STATe	OFF
[SENSe]:SPECtrum:FFT:WINDow	KAISer
[SENSe]:SPECtrum:FILTer[:SHAPe]	KAISer
[SENSe]:SPECtrum:FREQuency:CENTer	1.5 GHz
[SENSe]:SPECtrum:FREQuency:SPAN	40 MHz
[SENSe]:SPECtrum:FREQuency:SPAN:BANDwidth[:RESolution]: RATio	100
[SENSe]:SPECtrum:FREQuency:STARt	1.48 GHz
[SENSe]:SPECtrum:FREQuency:STEP	2 MHz
[SENSe]:SPECtrum:FREQuency:STEP:AUTO	ON
[SENSe]:SPECtrum:FREQuency:STOP	1.52 GHz
[SENSe]:SPECtrum:POINts:COUNt	P801
[SENSe]:SPURious subgroup	
[SENSe]:SPURious:CARRier:{BANDwidth|BWIDth}	4 MHz
[SENSe]:SPURious:CARRier:{BANDwidth|BWIDth}:INTegration	1 MHz
[SENSe]:SPURious:CARRier:{BANDwidth|BWIDth}[:RESolution]: AUTO	ON
[SENSe]:SPURious:CARRier:DETection	PEAK

Table C-8: Factory initialization settings, Sense commands (cont.)

Header	Default value
[SENSe]:SPURious:CARRier:FREQuency	1.484 GHz
[SENSe]:SPURious:CARRier:THReshold	-10 dBm
[SENSe]:SPURious:LIST	OVERlimit
[SENSe]:SPURious:MODE	MULTi
[SENSe]:SPURious:OPTimization	AUTO
[SENSe]:SPURious:POINts:COUNt	P801
[SENSe]:SPURious:RANGe<x>:BANDwidth:VIDeo	300 kHz
[SENSe]:SPURious:RANGe<x>:BANDwidth:VIDeo:STATe	OFF
[SENSe]:SPURious:RANGe<x>:DETection	PEAK
[SENSe]:SPURious:RANGe<x>:EXCursion	6 dB
[SENSe]:SPURious:RANGe<x>:FILTer[:SHAPe]	RBW
[SENSe]:SPURious:RANGe<x>:FILTer[:SHAPe]:BANDwidth	200 kHz
[SENSe]:SPURious:RANGe<x>:FILTer[:SHAPe]:BANDwidth:AUTO	ON
[SENSe]:SPURious:RANGe<x>:LIMit:ABSolute:STARt	-50 dBm
[SENSe]:SPURious:RANGe<x>:LIMit:ABSolute:STOP	-50 dBm
[SENSe]:SPURious:RANGe<x>:LIMit:MASK	ABS
[SENSe]:SPURious:RANGe<x>:LIMit:RELative:STARt	-30 dB
[SENSe]:SPURious:RANGe<x>:LIMit:RELative:STOP	-30 dB
[SENSe]:SPURious:RANGe<x>:STATe	<x>=1: ON <x>=2 to 20: OFF
[SENSe]:SPURious:REFerence	NONE
[SENSe]:SPURious:REFerence:MANual:POWer	-10 dBm
[SENSe]:TOVerview subgroup	
[SENSe]:TOVerview:FREQuency:CENTer	1.5 GHz
[SENSe]:TOVerview:MAXTracepoints	HUNDredk

Table C-9: Factory initialization settings, Trace commands

Header	Default value
TRACe<x>:AVTime subgroup	
TRACe<x>:AVTime	<x>=1: ON <x>=2 to 4: OFF
TRACe<x>:AVTime:AVERage:COUNt	10
TRACe<x>:AVTime:COUNt	10
TRACe<x>:AVTime:COUNt:ENABle	ON
TRACe<x>:AVTime:FREeze	OFF
TRACe<x>:AVTime:FUNCtion	NORMal

Table C-9: Factory initialization settings, Trace commands (cont.)

Header		Default value
TRACe<x>:AVTime:LEFToperand		TRACE2
TRACe<x>:AVTime:RIGHtoperand		TRACE1
TRACe<x>:CCDF subgroup		
TRACe<x>:CCDF:FREeze		OFF
TRACe<x>:CCDF:SHOW		<x>=1: ON <x>=2 and 3: OFF
TRACe<x>:CCDF:X		3 dB
TRACe:CONSte subgroup		
TRACe:CONSte:MODE		VECTors
TRACe<x>:DPSA subgroup		
TRACe<x>:DPSA		<x>=1 and 5: ON <x>=2 to 4: OFF
TRACe<x>:DPSA:AVERage:COUNt		10
TRACe<x>:DPSA:COLor:INTensity		25 %
TRACe<x>:DPSA:DOT:PERSistent	(<x>=5 only)	ON
TRACe<x>:DPSA:DOT:PERSistent:TYPE	(<x>=5 only)	VARiable
TRACe<x>:DPSA:DOT:PERSistent:VARiable	(<x>=5 only)	10
TRACe<x>:DPSA:FREeze		OFF
TRACe<x>:DPSA:FUNCtion		NORMal
TRACe<x>:DPSA:LEFToperand		TRACE1
TRACe<x>:DPSA:RIGHtoperand		TRACE3
TRACe:FVTime subgroup		
TRACe:FVTime:COUNt		10
TRACe:FVTime:COUNt:ENABle		ON
TRACe:FVTime:FREeze		OFF
TRACe:FVTime:FUNCtion		NORMal
TRACe:IQVTime subgroup		
TRACe:IQVTime:COUNt		10
TRACe:IQVTime:COUNt:ENABle		ON
TRACe:IQVTime:ENABle:I		ON
TRACe:IQVTime:ENABle:Q		ON
TRACe:IQVTime:FREeze		OFF
TRACe:IQVTime:FUNCtion		NORMal
TRACe:IQVTime:SELect:I		ON
TRACe:IQVTime:SELect:Q		OFF
TRACe:OBWidth subgroup		
TRACe:OBW:MAXHold		OFF

Table C-9: Factory initialization settings, Trace commands (cont.)

Header	Default value
TRACe:PHVTime subgroup	
TRACe:PHVTime:COUNt	10
TRACe:PHVTime:COUNt:ENABle	ON
TRACe:PHVTime:FREeze	OFF
TRACe:PHVTime:FUNCtion	NORMal
TRACe<x>:PNOise subgroup	
TRACe<x>:PNOise:DETection	AVERage
TRACe<x>:PNOise:FREeze	OFF
TRACe<x>:PNOise:SHOW	<x>=1: ON <x>=2: OFF
TRACe<x>:PNOise:SMOothing:COUNt	5
TRACe<x>:PNOise:SMOothing:ENABle	ON
TRACe:SGRam subgroup	
TRACe:SGRam:DETection	POSitive
TRACe:SGRam:FREeze	OFF
TRACe:SGRam:FUNCtion	NONE
TRACe:SGRam:FUNCtion:TIME	0.02 minutes
TRACe:SGRam:SELect:LINE	0
TRACe<x>:SPECtrum subgroup	
TRACe<x>:SPECtrum	<x>=1: ON <x>=2 to 4: OFF
TRACe<x>:SPECtrum:AVERage:COUNt	10
TRACe<x>:SPECtrum:COUNt	10
TRACe<x>:SPECtrum:COUNt:ENABle	ON
TRACe<x>:SPECtrum:DETection	POSitive
TRACe<x>:SPECtrum:FREeze	OFF
TRACe<x>:SPECtrum:FUNCtion	NONE
TRACe<x>:SPECtrum:LEFToperand	TRACE2
TRACe<x>:SPECtrum:RIGHtoperand	TRACE1
TRACe:SPURious subgroup	
TRACe:SPURious:COUNt	10
TRACe:SPURious:COUNt:ENABle	ON
TRACe:SPURious:FREeze	OFF
TRACe:SPURious:FUNCtion	NONE

Table C-10: Factory initialization settings, Trigger commands

Header	Default value
TRIGger[:SEQuence]:ADVanced:SWEep:MODE	OFF
TRIGger[:SEQuence]:EVENt:EXTFront:IMPedance	5 kΩ
TRIGger[:SEQuence]:EVENt:EXTFront:LEVel	1.6 V
TRIGger[:SEQuence]:EVENt:EXTFront:SLOPe	RISe
TRIGger[:SEQuence]:EVENt:GATed	HIGh
TRIGger[:SEQuence]:EVENt:INPut:FMASk:VIOLation	FT
TRIGger[:SEQuence]:EVENt:INPut:LEVel	-10 dBm
TRIGger[:SEQuence]:EVENt:INPut:SLOPe	RISe
TRIGger[:SEQuence]:EVENt:INPut:TDBWidth	1 MHz
TRIGger[:SEQuence]:EVENt:INPut:TDBWidth:STATe	ON
TRIGger[:SEQuence]:EVENt:INPut:TYPE	POWer
TRIGger[:SEQuence]:EVENt:SOURce	INPut
TRIGger[:SEQuence]:FORCed	OFF
TRIGger[:SEQuence]:STATus	OFF
TRIGger[:SEQuence]:TIME:DELay	0 s
TRIGger[:SEQuence]:TIME:POSition	25%

Table C-11: Factory initialization settings, UNIT commands

Header	Default value
UNIT:POWer	dBm

Appendix D: SCPI Conformance Information

All commands for the RSA6100A Series analyzers are based on SCPI Version 1999.0. The following table lists the commands that are defined in the SCPI 1999.0 Standard. The other commands not listed in the table are not defined in the SCPI 1999.0 Standard.

Table D-1: SCPI 1999.0-defined commands

Command group	Command		
IEEE common	*CAL		
	*CLS		
	*ESE		
	*ESR		
	*IDN		
	*OPC		
	*OPT		
	*RST		
	*SRE		
	*STB		
	*TRG		
	*WAI		
ABORt	:ABORt		
INITiate	:INITiate	:CONTinuous	
		[:IMMediate]	
STATus	:STATus	:OPERation	:CONDition?
			:ENABle
			[:EVENt]?
			:NTRansition
			:PTRansition
		:PRESet	
		:QUEStionable	:CONDition?
			:ENABle
			[:EVENt]?
			:NTRansition
			:PTRansition

Table D-1: SCPI 1999.0-defined commands (cont.)

Command group	Command				
SYSTem	:SYSTem	:COMMunicate	:GPIB	[:SELF]	:ADDRess
		:DATE			
		:ERRor	:ALL?		
			:CODE	:ALL?	
				[:NEXT]?	
			:COUNt?		
			[:NEXT]?		
		:PRESet			
		:TIME			
		:VERSion?			
UNIT	:UNIT	:POWer			

Glossary

Glossary

ASCII
Acronym for the American Standard Code for Information Interchange. Controllers transmit commands to the instrument using ASCII character encoding.

ASK
Acronym for Amplitude Shift Keying. The process, or result of a process, in which the amplitude of the carrier is varied in accordance with the state of a digital input signal.

BNF (Backus-Naur Form)
A standard notation system for command syntax diagrams. The syntax diagrams in this manual use BNF notation.

Controller
A computer or other device that sends commands to and accepts responses from the analyzer.

EVM (Error Vector Magnitude)
The magnitude of an error of an actual signal relative to an ideal signal in a constellation display.

GPIB
Acronym for General Purpose Interface Bus, the common name for the communications interface system defined in IEEE Std 488.

IEEE
Acronym for the Institute for Electrical and Electronic Engineers.

IS95
Acronym for Interim Standard-95. The standards name for first-generation CDMA cellphone technology.

Modulation
The process of varying some characteristic of a signal with a second signal.

PSK
Acronym for Phase Shift Keying. The process, or result of a process, in which the carrier phase is varied discretely in accordance with a digital code.

QAM
Acronym for Quadrature Amplitude Modulation. The process, or result of a process, in which the amplitude and phase of the carrier are varied concurrently by synthesizing two orthogonal ASK waves (see ASK).

Index

A

C

D

E

F

S

T

U

W