

# Programmer Manual



## **RSA6100A Series Real-Time Spectrum Analyzers**

**071-1913-03**

This document applies to firmware version 1.3  
and above.

**[www.tektronix.com](http://www.tektronix.com)**

Copyright © Tektronix. All rights reserved. Licensed software products are owned by Tektronix or its subsidiaries or suppliers, and are protected by national copyright laws and international treaty provisions.

Tektronix products are covered by U.S. and foreign patents, issued and pending. Information in this publication supercedes that in all previously published material. Specifications and price change privileges reserved.

TEKTRONIX and TEK are registered trademarks of Tektronix, Inc.

## **Contacting Tektronix**

Tektronix, Inc.  
14200 SW Karl Braun Drive  
P.O. Box 500  
Beaverton, OR 97077  
USA

For product information, sales, service, and technical support:

- In North America, call 1-800-833-9200.
- Worldwide, visit [www.tektronix.com](http://www.tektronix.com) to find contacts in your area.

# Table of Contents

<b>Preface</b> .....	<b>vii</b>
Related Documentation .....	vii

## Getting Started

Overview of the Manual .....	1-1
Connecting the Interface .....	1-3
Using the GPIB Port .....	1-4
Setting the GPIB Address .....	1-5
Using TekVISA .....	1-6

## Syntax and Commands

<b>Command Syntax</b> .....	<b>2-1</b>
Backus-Naur Form Definition .....	2-1
SCPI Commands and Queries .....	2-2
IEEE 488.2 Common Commands .....	2-9
Constructed Mnemonics .....	2-9
<b>Command Groups</b> .....	<b>2-11</b>
Measurement Views .....	2-11
Functional Groups .....	2-13
IEEE Common Commands .....	2-14
ABORt Commands .....	2-14
CALCulate Commands .....	2-15
CALibration Commands .....	2-21
DISPlay Commands .....	2-22
FETCh Commands .....	2-29
INITiate Commands .....	2-35
INPut Commands .....	2-35
MMEMory Commands .....	2-36
OUTPut Commands .....	2-37
READ Commands .....	2-38
SENSe Commands .....	2-44
STATus Commands .....	2-53
SYSTem Commands .....	2-54
TRACe Commands .....	2-55
TRIGger Commands .....	2-58
UNIT Commands .....	2-59
Programming Hints .....	2-60
<b>IEEE Common Commands</b> .....	<b>2-61</b>
<b>ABORt Commands</b> .....	<b>2-73</b>
<b>CALCulate Commands</b> .....	<b>2-75</b>
<b>CALibration Commands</b> .....	<b>2-201</b>
<b>DISPlay Commands</b> .....	<b>2-207</b>
<b>FETCh Commands</b> .....	<b>2-349</b>

<b>INITiate Commands</b> .....	<b>2-489</b>
<b>INPut Commands</b> .....	<b>2-491</b>
<b>MMEMory Commands</b> .....	<b>2-495</b>
<b>OUTPut Commands</b> .....	<b>2-521</b>
<b>READ Commands</b> .....	<b>2-525</b>
<b>SENSE Commands</b> .....	<b>2-659</b>
<b>STATus Commands</b> .....	<b>2-849</b>
<b>SYSTEM Commands</b> .....	<b>2-887</b>
<b>TRACe Commands</b> .....	<b>2-895</b>
<b>TRIGger Commands</b> .....	<b>2-961</b>
<b>UNIT Commands</b> .....	<b>2-977</b>
<b>Retrieving Response Message</b> .....	<b>2-979</b>

## Status and Events

Status and Event Reporting System .....	3-1
Status Byte .....	3-4
Standard Event Status Block .....	3-5
Operation Status Block .....	3-7
Questionable Status Block .....	3-8
Queues .....	3-10
Status and Event Processing Sequence .....	3-11
Synchronizing Execution .....	3-12
<b>Error Messages and Codes</b> .....	<b>3-15</b>
Command Errors .....	3-15
Execution Errors .....	3-16
Device Specific Errors .....	3-18
Query Errors .....	3-19
Device Errors .....	3-19

## Appendices

<b>Appendix A: Character Charts</b> .....	<b>A-1</b>
<b>Appendix B: GPIB Interface Specification</b> .....	<b>B-1</b>
Interface Functions .....	B-1
Interface Messages .....	B-3
<b>Appendix C: Factory Initialization Settings</b> .....	<b>C-1</b>
<b>Appendix D: SCPI Conformance Information</b> .....	<b>D-1</b>

## Glossary

## Index

# List of Figures

<b>Figure 1-1: Command parts</b> .....	<b>1-1</b>
<b>Figure 1-2: Functional groupings and an alphabetical list of commands</b> .....	<b>1-2</b>
<b>Figure 1-3: Event-driven program</b> .....	<b>1-2</b>
<b>Figure 1-4: GPIB connector (rear panel)</b> .....	<b>1-3</b>
<b>Figure 1-5: GPIB connection</b> .....	<b>1-4</b>
<b>Figure 1-6: Typical GPIB network configurations</b> .....	<b>1-4</b>
<b>Figure 2-1: Example of SCPI subsystem hierarchy tree</b> .....	<b>2-2</b>
<b>Figure 2-2: Example of abbreviating a command</b> .....	<b>2-5</b>
<b>Figure 2-3: Example of chaining commands and queries</b> .....	<b>2-6</b>
<b>Figure 2-4: Example of omitting root and lower-level nodes in a chained message</b> .....	<b>2-6</b>
<b>Figure 2-5: Trigger mask setting example</b> .....	<b>2-963</b>
<b>Figure 2-6: Retrieving response message</b> .....	<b>2-979</b>
<b>Figure 3-1: Status/Event reporting mechanism</b> .....	<b>3-2</b>
<b>Figure 3-2: Status Byte Register (SBR)</b> .....	<b>3-4</b>
<b>Figure 3-3: Service Request Enable Register (SRER)</b> .....	<b>3-5</b>
<b>Figure 3-4: Standard event status register</b> .....	<b>3-5</b>
<b>Figure 3-5: Operation status register</b> .....	<b>3-7</b>
<b>Figure 3-6: Questionable status register</b> .....	<b>3-8</b>
<b>Figure 3-7: Questionable frequency status register</b> .....	<b>3-9</b>
<b>Figure 3-8: Questionable calibration status register</b> .....	<b>3-10</b>
<b>Figure 3-9: Status and event processing sequence</b> .....	<b>3-11</b>

## List of Tables

<b>Table 2-1: BNF symbols and meanings</b> .....	<b>2-1</b>
<b>Table 2-2: Query response examples</b> .....	<b>2-3</b>
<b>Table 2-3: Parameter types used in syntax descriptions</b> .....	<b>2-4</b>
<b>Table 2-4: Available units</b> .....	<b>2-7</b>
<b>Table 2-5: Available SI prefixes</b> .....	<b>2-7</b>
<b>Table 2-6: Constructed mnemonics</b> .....	<b>2-10</b>
<b>Table 2-7: Measurement views</b> .....	<b>2-12</b>
<b>Table 2-8: List of command groups</b> .....	<b>2-13</b>
<b>Table 2-9: Status and error commands</b> .....	<b>2-14</b>
<b>Table 2-10: ABORt commands</b> .....	<b>2-14</b>
<b>Table 2-11: CALCulate commands</b> .....	<b>2-15</b>
<b>Table 2-12: CALibration commands</b> .....	<b>2-21</b>
<b>Table 2-13: DISPlay commands</b> .....	<b>2-22</b>
<b>Table 2-14: FETCh commands</b> .....	<b>2-29</b>
<b>Table 2-15: INITiate commands</b> .....	<b>2-35</b>
<b>Table 2-16: INPut commands</b> .....	<b>2-35</b>
<b>Table 2-17: :MMEMory commands</b> .....	<b>2-36</b>
<b>Table 2-18: OUTPut commands</b> .....	<b>2-37</b>
<b>Table 2-19: READ commands</b> .....	<b>2-38</b>
<b>Table 2-20: SENSE commands</b> .....	<b>2-44</b>
<b>Table 2-21: STATus commands</b> .....	<b>2-53</b>
<b>Table 2-22: SYSTem commands</b> .....	<b>2-54</b>
<b>Table 2-23: TRACe commands</b> .....	<b>2-55</b>
<b>Table 2-24: TRIGger commands</b> .....	<b>2-58</b>
<b>Table 2-25: UNIT commands</b> .....	<b>2-59</b>
<b>Table 2-26: Instrument options</b> .....	<b>2-67</b>
<b>Table 2-27: CALCulate command subgroups</b> .....	<b>2-75</b>
<b>Table 2-28: Marker mnemonics</b> .....	<b>2-76</b>
<b>Table 2-29: Search condition</b> .....	<b>2-86</b>
<b>Table 2-30: Source data of the search operation</b> .....	<b>2-87</b>
<b>Table 2-31: DISPlay command subgroups</b> .....	<b>2-207</b>
<b>Table 2-32: Measurement views</b> .....	<b>2-209</b>
<b>Table 2-33: Modulation measurement views</b> .....	<b>2-230</b>
<b>Table 2-34: General signal viewing views</b> .....	<b>2-253</b>
<b>Table 2-35: Pulse results</b> .....	<b>2-315</b>

<b>Table 2-36: FETCh command subgroups</b> .....	<b>2-349</b>
<b>Table 2-37: MMEMory command subgroups</b> .....	<b>2-495</b>
<b>Table 2-38: Trace file extension and trace selection command</b> ....	<b>2-499</b>
<b>Table 2-39: READ command subgroups</b> .....	<b>2-525</b>
<b>Table 2-40: SENSE command subgroups</b> .....	<b>2-659</b>
<b>Table 2-41: Dithering status</b> .....	<b>2-668</b>
<b>Table 2-42: Power units</b> .....	<b>2-673</b>
<b>Table 2-43: Digital modulation measurement filter</b> .....	<b>2-706</b>
<b>Table 2-44: Modulation type</b> .....	<b>2-708</b>
<b>Table 2-45: Frequency tuning</b> .....	<b>2-716</b>
<b>Table 2-46: Color palette for DPX spectrum</b> .....	<b>2-719</b>
<b>Table 2-47: Color palette for spectrogram</b> .....	<b>2-794</b>
<b>Table 2-48: Limit mask mode</b> .....	<b>2-839</b>
<b>Table 2-49: TRACe command subgroups</b> .....	<b>2-895</b>
<b>Table 2-50: Trigger violations</b> .....	<b>2-969</b>
<b>Table 2-51: Trigger event source</b> .....	<b>2-973</b>
<b>Table 2-52: Power units</b> .....	<b>2-978</b>
<b>Table 3-1: Register type</b> .....	<b>3-1</b>
<b>Table 3-2: SBR bit functions</b> .....	<b>3-4</b>
<b>Table 3-3: Standard event status register bit definition</b> .....	<b>3-6</b>
<b>Table 3-4: Operation status register bit definition</b> .....	<b>3-7</b>
<b>Table 3-5: Questionable status register bit definition</b> .....	<b>3-8</b>
<b>Table 3-6: Questionable frequency status register bit definition</b> ...	<b>3-9</b>
<b>Table 3-7: Questionable calibration status register bit definition</b> ..	<b>3-10</b>
<b>Table 3-8: Command errors</b> .....	<b>3-15</b>
<b>Table 3-9: Execution errors</b> .....	<b>3-16</b>
<b>Table 3-10: Device specific errors</b> .....	<b>3-18</b>
<b>Table 3-11: Query errors</b> .....	<b>3-19</b>
<b>Table 3-12: Device errors, global condition</b> .....	<b>3-19</b>
<b>Table 3-13: Device errors, measurement condition</b> .....	<b>3-20</b>
<b>Table 3-14: Device errors, source condition</b> .....	<b>3-21</b>
<b>Table A-1: ASCII &amp; GPIB code chart</b> .....	<b>A-2</b>
<b>Table B-1: GPIB interface function implementation</b> .....	<b>B-1</b>
<b>Table B-2: Standard interface messages</b> .....	<b>B-3</b>

<b>Table C-1: Factory initialization settings, IEEE common commands</b> .....	<b>C-1</b>
<b>Table C-2: Factory initialization settings, CALibration commands</b>	<b>C-1</b>
<b>Table C-3: Factory initialization settings, CALCulate commands</b> .	<b>C-1</b>
<b>Table C-4: Factory initialization settings, DISPlay commands</b> ....	<b>C-2</b>
<b>Table C-5: Factory initialization settings, INITiate commands</b> ...	<b>C-6</b>
<b>Table C-6: Factory initialization settings, INPut commands</b> .....	<b>C-6</b>
<b>Table C-7: Factory initialization settings, OUTPut commands</b> ...	<b>C-7</b>
<b>Table C-8: Factory initialization settings, SENSE commands</b> .....	<b>C-7</b>
<b>Table C-9: Factory initialization settings, TRACe commands</b> ....	<b>C-13</b>
<b>Table C-10: Factory initialization settings, TRIGger commands</b> ..	<b>C-15</b>
<b>Table C-11: Factory initialization settings, UNIT commands</b> .....	<b>C-16</b>
<b>Table D-1: SCPI 1999.0-defined commands</b> .....	<b>D-1</b>



# 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](http://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:

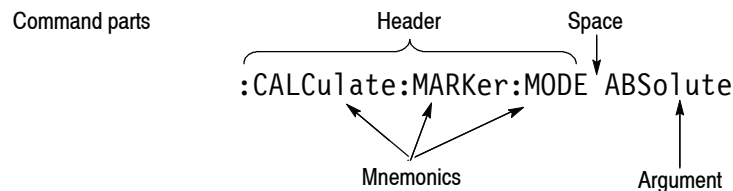
- *Overview of the Manual*  
Summarizes each major section of this manual.
- *Connecting the Interface*  
Describes how to physically connect the analyzer to a controller.
- *Using GPIB Ports*  
Describes how to use the GPIB port.
- *Setting the GPIB Address*  
Describes how to set the GPIB parameters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

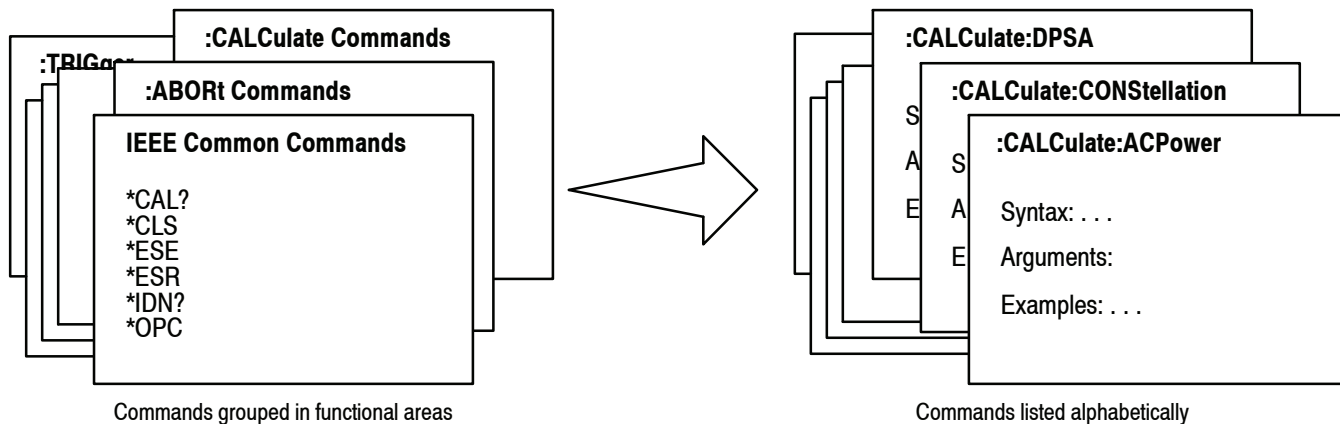
### Syntax and Commands

Section 2, *Syntax and Commands*, describes the structure and content of the messages your program sends to the analyzer. Figure 1-1 shows command parts as described in the *Command Syntax* subsection.



**Figure 1-1: Command parts**

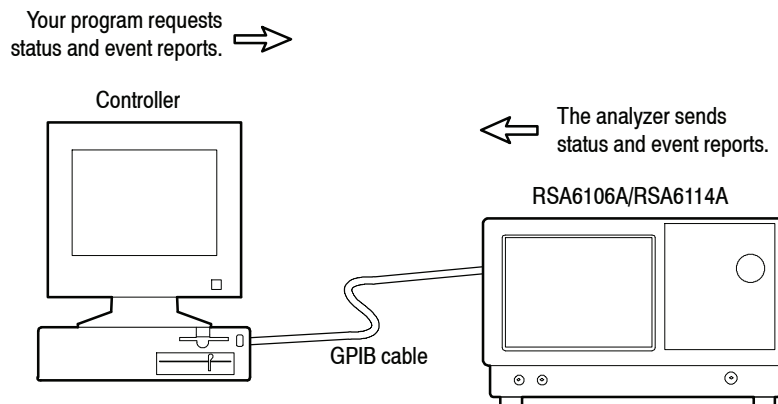
Section 2 also describes the effect of each command and provides examples of how you might use it. The *Command Groups* subsection provides lists by functional areas. The *IEEE Common Commands* and the subsequent subsections arrange commands alphabetically (Figure 1-2).



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

**Status and Events**

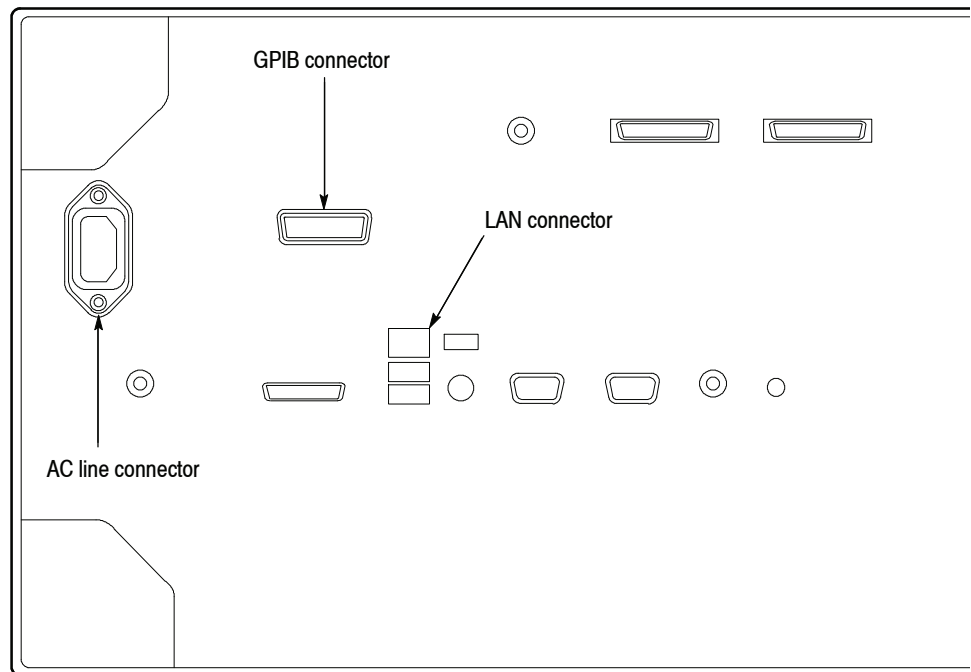
The program may request information from the instrument. The instrument provides information in the form of status and error messages. Figure 1-3 illustrates the basic operation of this system. Section 3, *Status and Events*, describes how to get status or event information from the program and details 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 Figure 1-4. 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.

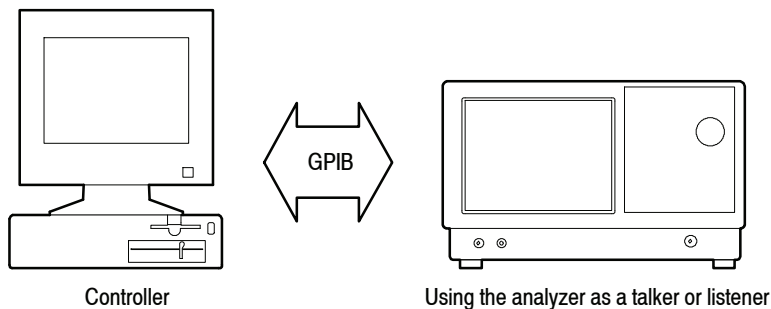


**Figure 1-4: GPIB connector (rear panel)**

*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.

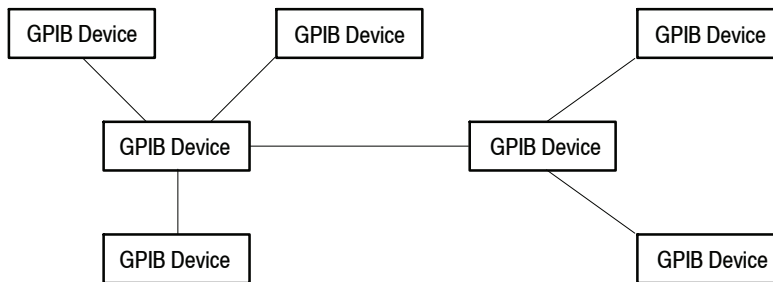


**Figure 1-5: GPIB connection**

### 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 Figure 1-6. Do not use loop or parallel configurations.



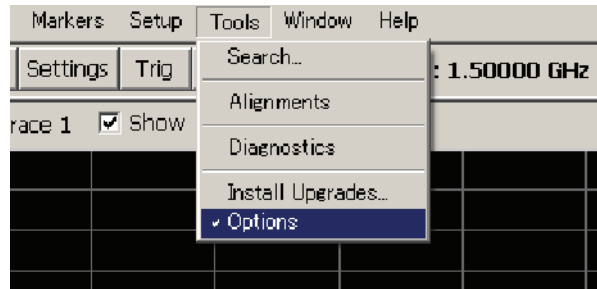
**Figure 1-6: Typical GPIB network configurations**



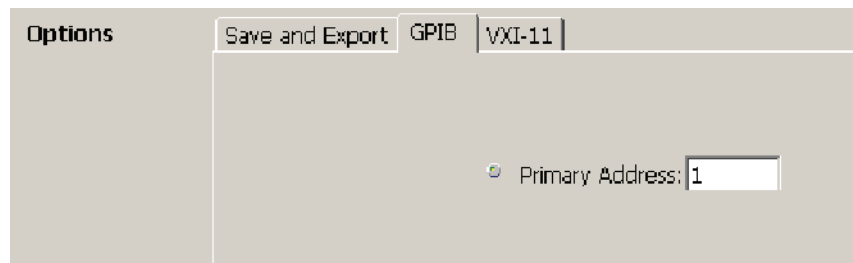
## Setting the GPIB Address

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](http://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. Table 2-1 defines the standard BNF symbols.

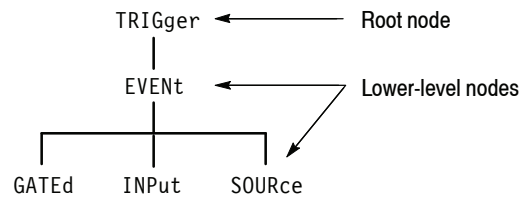
**Table 2- 1: BNF symbols and meanings**

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 (see Figure 2-1) 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 Figure 2-1, 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, which start on page 2-61, list the valid values for all parameters.

For example, TRIGger:EVENT:SOURce EXTRear is a valid SCPI command created from the hierarchy tree in 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. TRIGger:EVENT:SOURce? is an example of a valid SCPI query using the hierarchy tree in 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 Table 2-2.

**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 (refer to Table 2-3).

**Table 2-3: Parameter types used in syntax descriptions**

Parameter type	Description	Example
arbitrary block <sup>1</sup>	A specified length of arbitrary data	#512234xxxx . . . 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 <sup>2</sup>	Hexadecimal numbers (0-9, A, B, C, D, E, F)	#HAA, #H1
NR1 <sup>2,3</sup> numeric	Integers	0, 1, 15, -1
NR2 <sup>2</sup> numeric	Decimal numbers	1.2, 3.141516, -6.5
NR3 <sup>2</sup> numeric	Floating point numbers	3.1415E-9, -16.1E5
NRf <sup>2</sup> numeric	Flexible decimal number that may be type NR1, NR2 or NR3	See NR1, NR2, and NR3 examples
string <sup>4</sup>	Alphanumeric characters (must be within quotation marks)	"Testing 1, 2, 3"

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

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

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

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

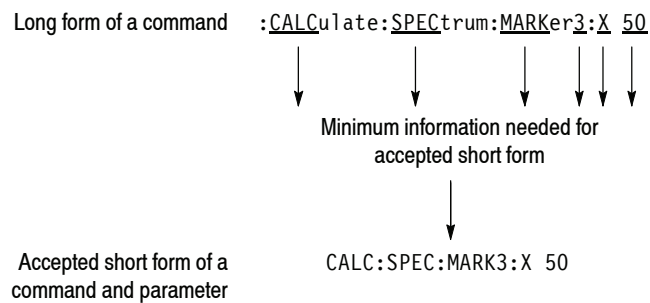


**Special Characters**

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 Figure 2-2, 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.

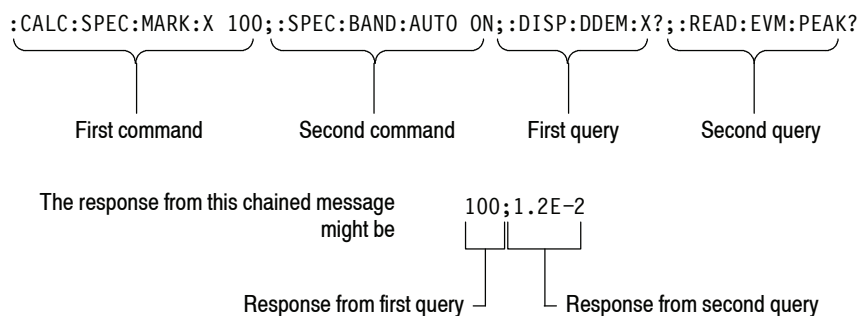


**Figure 2-2: Example of abbreviating a command**

**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 Figure 2-2, the “3” of “MARKer3” indicates that the command is directed to Marker 3.

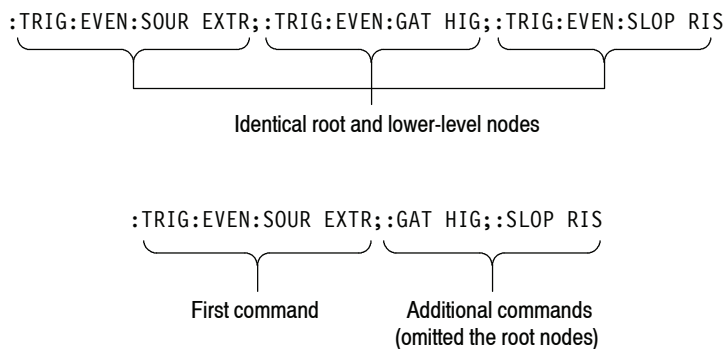
### Chaining Commands and Queries

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 (:). Figure 2-3 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.



**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 Figure 2-4, the second command has the same root node (`TRIGger:EVENT`) as the first command, so these nodes can be omitted.



**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.

Table 2-4 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 Table 2-5 below.

**Table 2-5: Available SI prefixes**

SI prefix	Z	A	F	P	N	U	M	K	MA <sup>1</sup>	G	T	PE	EX
Corresponding power	10 <sup>-21</sup>	10 <sup>-18</sup>	10 <sup>-15</sup>	10 <sup>-12</sup>	10 <sup>-9</sup>	10 <sup>-6</sup>	10 <sup>-3</sup>	10 <sup>+3</sup>	10 <sup>+6</sup>	10 <sup>+9</sup>	10 <sup>+12</sup>	10 <sup>+15</sup>	10 <sup>+18</sup>

<sup>1</sup> 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 channel 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 <i>Marker Mnemonics</i> on page 2-76.
RANGe<x>	A range specifier where <x> = 1 to 20. Refer to <i>[:SENSe]:SPURious Subgroup</i> on page 2-818 for details.
SPUR<x>	A spurious specifier where <x> = 1 to the number of spurious signals. Refer to <i>:FETCh :READ:SPURious Subgroup</i> on page 2-464 or 2-634 for details, respectively.
TRACe<x>	A trace specifier where <x> = 0, 1, 2, 3, 4, or 5. Refer to <i>TRACe Commands</i> on page 2-895 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 and starts on page 2-61.

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* on page 2-1 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 Table 2-7. Each command works in particular measurement view(s) which are specified in the **Measurement Views** item 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**

<b>Display group</b>	<b>Measurement view</b>
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 show in Table 2-8.

**Table 2-8: List of command groups**

<b>Command group</b>	<b>Function</b>
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.

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
<b>:CALCulate basic command subgroup</b>	<b>General marker control</b>
:CALCulate:MARKer:ADD	Adds a marker.
:CALCulate:MARKer:AOff	Turns off all markers.
:CALCulate:MARKer:DELeTe	Deletes the last marker added.
:CALCulate:MARKer:MODE(?)	Selects or queries the marker mode.
:CALCulate:MARKer:PEAK:EXCURsion(?)	Sets or queries the 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.
<b>:CALCulate:ACPower subgroup</b>	<b>Channel power and ACPR measurement</b>
: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
<b>:CALCulate:AVTime subgroup</b>	<b>Frequency versus Time measurement</b>
: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.
<b>:CALCulate:CONSte subgroup (Option 21 only)</b>	<b>Constellation measurement</b>
: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.
<b>:CALCulate:DPSA subgroup</b>	<b>DPX spectrum measurement</b>
: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(?)	Sets or queries the horizontal position of the marker.
:CALCulate:DPSA:MARKer<x>:Y?	Queries the vertical position of the marker.

Table 2- 11: CALCulate commands (Cont.)

Header	Description
<b>:CALCulate:EVM subgroup (Option 21 only)</b>	<b>EVM versus Time measurement</b>
: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.
: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.
<b>:CALCulate:FVTime subgroup</b>	<b>Frequency versus Time measurement</b>
: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.
<b>:CALCulate:IQVTime subgroup</b>	<b>RF I&amp;Q versus Time measurement</b>
: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.
<b>:CALCulate:MCPower subgroup</b>	<b>MCPR measurement</b>
: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.

**Table 2- 11: CALCulate commands (Cont.)**

<b>Header</b>	<b>Description</b>
: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.
<b>:CALCulate:MERRor subgroup (Option 21 only)</b>	<b>Magnitude error versus Time measurement</b>
: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.
<b>:CALCulate:OBWidth subgroup</b>	<b>Occupied Bandwidth measurement</b>
: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.
<b>:CALCulate:PERRor subgroup (Option 21 only)</b>	<b>Phase error versus Time measurement</b>
: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.

Table 2- 11: CALCulate commands (Cont.)

Header	Description
<b>:CALCulate:PHVTime subgroup</b>	<b>Phase versus Time measurement</b>
: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.
: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.
<b>:CALCulate:PULSe subgroup (Option 20 only)</b>	<b>Pulsed RF measurements</b>
: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.
<b>:CALCulate:SGRam subgroup</b>	<b>Spectrogram measurement</b>
: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.



**Table 2- 11: CALCulate commands (Cont.)**

<b>Header</b>	<b>Description</b>
: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.
<b>:CALCulate:SPECTrum subgroup</b>	<b>Spectrum measurement</b>
: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.
<b>:CALCulate:SPURious subgroup</b>	<b>Spurious measurement</b>
: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.
<b>:CALCulate:TOVerview subgroup</b>	<b>Time overview</b>
: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.



**Table 2- 11: CALCulate commands (Cont.)**

Header	Description
: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.

## CALibration Commands

Use the CALibration commands to control the external correction.

**Table 2- 12: 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- 13: DISPlay commands**

Header	Description
<b>:DISPlay basic command subgroup</b>	<b>General window control</b>
:DISPlay:WINDow:ACTive:MEASurement?	Queries the active measurement views.
:DISPlay:WINDow:OPTimized:MEASurement?	Queries the measurement views that are optimized.
<b>:DISPlay:ACPower subgroup</b>	<b>Channel power and ACPR measurement</b>
: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.
<b>:DISPlay:AVTime subgroup</b>	<b>Amplitude versus Time measurement</b>
: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.

Table 2-13: DISPlay commands (Cont.)

Header	Description
:DISPlay:AVTime:Y:RSCale	Rescales the vertical axis automatically.
:DISPlay:AVTime:Y[:SCALE]:FULL(?)	Sets or queries the vertical scale.
:DISPlay:AVTime:Y[:SCALE]:OFFSet(?)	Sets or queries the vertical offset.
<b>:DISPlay:CONSte subgroup (Option 21 only)</b>	<b>Constellation measurement</b>
:DISPlay:CONSte:WINDow:TRACe:GRATICule:GRID:STATe(?)	Selects or queries whether to show the graticule grid on the screen.
<b>:DISPlay:DDEMod subgroup (Option 21 only)</b>	<b>General purpose digital modulation measurements</b>
: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.
<b>:DISPlay:DPSA subgroup</b>	<b>DPX spectrum measurement</b>
: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).
<b>:DISPlay:EVM subgroup (Option 21 only)</b>	<b>EVM versus Time measurement</b>
: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).
<b>:DISPlay:FVTime subgroup</b>	<b>Frequency versus Time measurement</b>
: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.

**Table 2- 13: DISPLAY commands (Cont.)**

<b>Header</b>	<b>Description</b>
: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.
: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.
<b>:DISPlay:GENeral subgroup</b>	<b>General signal viewing</b>
: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.
<b>:DISPlay:GPRF subgroup</b>	<b>General purpose RF measurements</b>
: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.
<b>:DISPlay:IQVTime subgroup</b>	<b>RF I&amp;Q versus Time measurement</b>
: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.
<b>:DISPlay:MCPower subgroup</b>	<b>MCPR measurement</b>
: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.

Table 2-13: DISPLAY commands (Cont.)

Header	Description
: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.
:DISPlay:MCPower:Y[:SCALe]:AUTO	Rescales the vertical axis automatically.
:DISPlay:MCPower:Y[:SCALe]:OFFSet(?)	Sets or queries the vertical offset.
<b>:DISPlay:MERRor subgroup (Option 21 only)</b>	<b>Magnitude error versus Time measurement</b>
: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).
<b>:DISPlay:OBWidth subgroup</b>	<b>Occupied Bandwidth measurement</b>
: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.
<b>:DISPlay:PERRor subgroup (Option 21 only)</b>	<b>Phase error versus Time measurement</b>
: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).
<b>:DISPlay:PHVTime subgroup</b>	<b>Phase versus Time measurement</b>
: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.

**Table 2- 13: DISPLAY commands (Cont.)**

Header	Description
: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.
:DISPlay:PHVTime:Y[:SCALe]:OFFSet(?)	Sets or queries the vertical offset.
:DISPlay:PHVTime:Y[:SCALe]:REScale	Rescales the vertical scale.
<b>:DISPlay:PNOise subgroup (Option 11 only)</b>	<b>Phase noise measurements</b>
: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).
<b>:DISPlay:PULSe subgroup (Option 20 only)</b>	<b>Pulsed RF measurements</b>
: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.

**Table 2-13: DISPLAY commands (Cont.)**

Header	Description
:DISPlay:PULSe:RESult:MFRReqerror(?)	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:PPFRrequency(?)	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:RIPPIe(?)	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.
: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:PULSeSElect:NUMBer(?)	Selects or queries the pulse to measure.
:DISPlay:PULSeSElect: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.



**Table 2- 13: DISPlay commands (Cont.)**

<b>Header</b>	<b>Description</b>
: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.
<b>:DISPlay:SGRam subgroup</b>	<b>Spectrogram measurement</b>
: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).
<b>:DISPlay:SPECTrum subgroup</b>	<b>Spectrum measurement</b>
:DISPlay:SPECTrum:MARKer:NOISe:MODE(?)	Selects or queries whether to enable the marker noise mode.
:DISPlay:SPECTrum:WINDow:TRACe:GRATICule:GRID:STATE(?)	Selects or queries whether to show the graticule grid in the spectrum view.
:DISPlay:SPECTrum:Y[:SCALe]:PDIVision(?)	Sets or queries the vertical scale (per division).
<b>:DISPlay:SPURious subgroup</b>	<b>Spurious measurement</b>
: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: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](?)	Sets or queries the horizontal range.
: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.



**Table 2-13: DISPlay commands (Cont.)**

Header	Description
<b>:DISPlay:TOVerview subgroup</b>	<b>Time overview</b>
: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.
: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 2-14: FETCh commands**

Header	Description
<b>:FETCh basic command subgroup</b>	<b>General fetch control</b>
: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.
<b>:FETCh:ACPower subgroup</b>	<b>Channel power and ACPR measurement</b>
: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.
<b>:FETCh:AVTime subgroup</b>	<b>Amplitude versus Time measurement</b>
: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.

**Table 2- 14: FETCh commands (Cont.)**

<b>Header</b>	<b>Description</b>
:FETCh:AVTime:MINimum?	Returns the minimum value.
:FETCh:AVTime:MINLocation?	Returns the time at the minimum.
:FETCh:AVTime:RESult?	Returns the measurement results.
<b>:FETCh:CCDF subgroup</b>	<b>CCDF measurement</b>
: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.
<b>:FETCh:CONSte subgroup (Option 21 only)</b>	<b>Constellation measurement</b>
:FETCh:CONSte:RESults?	Returns the constellation measurement results.
:FETCh:CONSte:TRACe?	Returns the constellation trace data.
<b>:FETCh:DDEMod subgroup (Option 21 only)</b>	<b>General purpose digital modulation measurements</b>
: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.
<b>:FETCh:DPSA subgroup</b>	<b>DPX spectrum measurement</b>
: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.
<b>:FETCh:EVM subgroup (Option 21 only)</b>	<b>EVM versus Time measurement</b>
: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.
<b>:FETCh:FVTime subgroup</b>	<b>Frequency versus Time measurement</b>
: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.

Table 2-14: FETCh commands (Cont.)

Header	Description
<b>:FETCh:IQVTime subgroup</b>	<b>RF I&amp;Q versus Time measurement</b>
: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.
<b>:FETCh:MCPower subgroup</b>	<b>MCPR measurement</b>
: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.
<b>:FETCh:MERRor subgroup (Option 21 only)</b>	<b>Magnitude error versus Time measurement</b>
: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.
<b>:FETCh:OBWidth subgroup</b>	<b>Occupied Bandwidth measurement</b>
: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.

**Table 2- 14: FETCh commands (Cont.)**

<b>Header</b>	<b>Description</b>
<b>:FETCh:PERRor subgroup (Option 21 only)</b>	<b>Phase error versus Time measurement</b>
: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.
<b>:FETCh:PHVTime subgroup</b>	<b>Phase versus Time measurement</b>
: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.
<b>:FETCh:PNOise subgroup (Option 11 only)</b>	<b>Phase noise measurements</b>
: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.
:FETCh:PNOise:JITTer?	Returns the jitter.
<b>:FETCh:PULSe subgroup (Option 20 only)</b>	<b>Pulsed RF measurements</b>
: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]:MFRReqerror?	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]:PPFRrequency?	Returns the pulse-pulse carrier frequency in the results table.

**Table 2-14: FETCh commands (Cont.)**

Header	Description
: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:MFRReqerror?	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:PPFFrequency?	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.
: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.

Table 2- 14: FETCh commands (Cont.)

Header	Description
<b>:FETCh:SGRam subgroup</b>	<b>Spectrogram measurement</b>
:FETCh:SGRam?	Returns the spectrogram trace data.
<b>:FETCh:SPEctrum subgroup</b>	<b>Spectrum measurement</b>
:FETCh:SPEctrum:TRACe<x>?	Returns the trace data in the Spectrum Analyzer measurement.
<b>:FETCh:SPURious subgroup</b>	<b>Spurious measurement</b>
: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.
<b>:FETCh:SQUality subgroup (Option 21 only)</b>	<b>Signal quality measurement</b>
: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 (%).
: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 $\rho$ (waveform quality).
:FETCh:SQUality:RMS:EVM?	Returns the RMS EVM (%).
:FETCh:SQUality:RMS:EVM:DB?	Returns the RMS EVM (dB).

**Table 2-14: FETCh commands (Cont.)**

Header	Description
: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.
<b>:FETCh:TOVerview subgroup</b>	<b>Time overview</b>
:FETCh:TOVerview?	Returns the trace data.

## INITiate Commands

Use the INITiate commands to control the acquisition of data.

**Table 2-15: 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 2-16: 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.

## MMEMory Commands

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

**Table 2-17: :MMEMory commands**

Header	Description
<b>:MMEMory basic command subgroup</b>	<b>General file control</b>
: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.
<b>:MMEMory:AVTime subgroup</b>	<b>Amplitude versus Time measurement</b>
:MMEMory:AVTime:LOAD:TRACe<x>	Loads trace data from the specified file.
:MMEMory:AVTime:STORE:TRACe<x>	Stores trace data in the specified file.
<b>:MMEMory:CCDF subgroup</b>	<b>CCDF measurement</b>
:MMEMory:CCDF:LOAD:TRACe<x>	Loads trace data from the specified file.
:MMEMory:CCDF:STORE:TRACe<x>	Stores trace data in the specified file.
<b>:MMEMory:DPSA subgroup</b>	<b>DPX spectrum measurement</b>
:MMEMory:DPSA:LOAD:TRACe<x>	Loads trace data from the specified file.
:MMEMory:DPSA:STORE:TRACe<x>	Stores trace data in the specified file.
<b>:MMEMory:FVTime subgroup</b>	<b>Frequency versus Time measurement</b>
:MMEMory:FVTime:LOAD:TRACe	Loads trace data from the specified file.
:MMEMory:FVTime:STORE:TRACe	Stores trace data in the specified file.
<b>:MMEMory:IQVTime subgroup</b>	<b>RF I&amp;Q versus Time measurement</b>
: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 2-17: :MMEMory commands (Cont.)**

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

## OUTPut Commands

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

**Table 2-18: 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 2-19: READ commands**

Header	Description
<b>:READ:ACPower subgroup</b>	<b>Channel power and ACPR measurement</b>
: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.
<b>:READ:AVTime subgroup</b>	<b>Amplitude versus Time measurement</b>
: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.
<b>:READ:CCDF subgroup</b>	<b>CCDF measurement</b>
: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.
<b>:READ:CONSte subgroup (Option 21 only)</b>	<b>Constellation measurement</b>
:READ:CONSte:RESults?	Returns the constellation measurement results.
:READ:CONSte:TRACe?	Returns the constellation trace data.
<b>:READ:DDEMod subgroup (Option 21 only)</b>	<b>General purpose digital modulation measurements</b>
:READ:DDEMod:STABle?	Returns the symbol table data.
<b>:READ:DPSA subgroup</b>	<b>DPX spectrum measurement</b>
: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.

Table 2-19: READ commands (Cont.)

Header	Description
<b>:READ:EVM subgroup (Option 21 only)</b>	<b>EVM versus Time measurement</b>
:READ:EVM:FERRor?	Returns the frequency error.
:READ:EVM:PEAK?	Returns the peak value.
: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.
<b>:READ:FVTime subgroup</b>	<b>Frequency versus Time measurement</b>
: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.
<b>:READ:IQVTime subgroup</b>	<b>RF I&amp;Q versus Time measurement</b>
: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.
<b>:READ:MCPower subgroup</b>	<b>MCPR measurement</b>
: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.
<b>:READ:MERRor subgroup (Option 21 only)</b>	<b>Magnitude error versus Time measurement</b>
: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.

**Table 2- 19: READ commands (Cont.)**

<b>Header</b>	<b>Description</b>
<b>:READ:PERRor subgroup (Option 21 only)</b>	<b>Phase error versus Time measurement</b>
: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.
<b>:READ:OBWidth subgroup</b>	<b>Occupied Bandwidth measurement</b>
:READ:OBWidth:FREQuency:ERRor?	Returns the frequency error.
: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.
<b>:READ:PHVTime subgroup</b>	<b>Phase versus Time measurement</b>
: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.
<b>:READ:PNOise subgroup (Option 11 only)</b>	<b>Phase noise measurements</b>
: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.

Table 2-19: READ commands (Cont.)

Header	Description
: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.
<b>:READ:PULSe subgroup (Option 20 only)</b>	<b>Pulsed RF measurements</b>
: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.
:READ:PULSe[:RESult]:FRDeviatiOn?	Returns the frequency deviation in the results table.
:READ:PULSe[:RESult]:MFRreqerror?	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.

**Table 2- 19: READ commands (Cont.)**

<b>Header</b>	<b>Description</b>
:READ:PULSe:STATistics:MFRerror?	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:PPPPhase?	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.
: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.
<b>:READ:SGRam subgroup</b>	<b>Spectrogram measurement</b>
:READ:SGRam?	Returns the spectrogram trace data.
<b>:READ:SPECTrum subgroup</b>	<b>Spectrum measurement</b>
:READ:SPECTrum:TRACe<x>?	Returns the trace data in the Spectrum Analyzer measurement.
<b>:READ:SPURious subgroup</b>	<b>Spurious measurement</b>
: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.

Table 2-19: READ commands (Cont.)

Header	Description
<b>:READ:SQUality subgroup (Option 21 only)</b>	<b>Signal quality measurement</b>
: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 $\rho$ (waveform quality).
:READ:SQUality:RMS:EVM?	Returns the RMS EVM (%).
:READ:SQUality:RMS:EVM:DB?	Returns the RMS EVM (dB).
:READ:SQUality:RMS:MERRor: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.
<b>:READ:TOVerview subgroup</b>	<b>Time overview</b>
:READ:TOVerview?	Returns the trace data.

## SENSe Commands

Use the SENSE commands to set up detailed measurement conditions.

**Table 2-20: SENSe commands**

Header	Description
<b>[[:SENSe] basic command subgroup</b>	<b>General analysis parameter control</b>
[[:SENSe]:ACQuisition:BAWIDth]: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.



Table 2-20: SENSE commands (Cont.)

Header	Description
<b>[[:SENSE]:ACPower subgroup</b>	<b>Channel power and ACPR measurement</b>
[[: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).
[[: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.
<b>[[:SENSE]:AVTime subgroup</b>	<b>Amplitude versus Time measurement</b>
[[:SENSE]:AVTime:BANDwith[:BWIDth(?)]	Sets or queries the time-domain bandwidth filter.
[[:SENSE]:AVTime:BANDwidth[:BWIDth[:RESolution]: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.
<b>[[:SENSE]:CCDF subgroup</b>	<b>CCDF measurement</b>
[[: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.

**Table 2-20: SENSE commands (Cont.)**

Header	Description
<b>[[:SENSE]:DDEMod subgroup (Option 21 only)</b>	<b>General purpose digital modulation measurements</b>
[[: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.
[[:SENSE]:DDEMod:FILTer:ALPHa(?)]	Sets or queries the filter factor ( $\alpha/BT$ ).
[[: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.
<b>[[:SENSE]:DPSA subgroup</b>	<b>DPX spectrum measurement</b>
[[: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:RXFRrequency?]	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.

Table 2-20: SENSE commands (Cont.)

Header	Description
[: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.
<b>[:SENSe]:FVTime subgroup</b>	<b>Frequency versus Time measurement</b>
[: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.
[: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.
<b>[:SENSe]:IQVTime subgroup</b>	<b>RF I&amp;Q versus Time measurement</b>
[: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.
<b>[:SENSe]:MCPower subgroup</b>	<b>MCPR measurement</b>
[: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.

**Table 2-20: SENSE commands (Cont.)**

<b>Header</b>	<b>Description</b>
[[: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.
[[: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.
<b>[[:SENSe]:OBWidth subgroup</b>	<b>Occupied Bandwidth measurement</b>
[[: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.

Table 2-20: SENSE commands (Cont.)

Header	Description
<b>[:SENSe]:PHVTime subgroup</b>	<b>Phase versus Time measurement</b>
[: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.
<b>[:SENSe]:PNOise subgroup (Option 11 only)</b>	<b>Phase noise measurement</b>
[:SENSe]:PNOise:AVERAge:COUNT(?)	Sets or queries the number of traces to combine for averaging.
[:SENSe]:PNOise:AVERAge(?)	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.
[: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.
<b>[:SENSe]:PULSe subgroup (Option 20 only)</b>	<b>Pulsed RF measurements</b>
[: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:RFAlI(?)	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.

**Table 2-20: SENSE commands (Cont.)**

<b>Header</b>	<b>Description</b>
[[: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.
<b>[[:SENSe]:SGRAM subgroup</b>	<b>Spectrogram measurement</b>
[[: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.
[[: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.



Table 2-20: SENSE commands (Cont.)

Header	Description
<b>[:SENSe]:SPECTrum subgroup</b>	<b>Spectrum measurement</b>
[: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.
[: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 real-time span.
[:SENSe]:SPECTrum:POINts:COUNt(?)	Sets or queries the number of sample points on the signal spectrum.
<b>[:SENSe]:SPURious subgroup</b>	<b>Spurious measurement</b>
[:SENSe]:SPURious:CARRier:BANDwidth(?)	Sets or queries the channel width for the carrier as power reference.
[:SENSe]:SPURious:CARRier:BANDwidth:INTegration(?)	Sets or queries the integration bandwidth to calculate the carrier power.
[:SENSe]:SPURious:CARRier:BANDwidth[:RESolution](?)	Sets or queries the resolution bandwidth to measure the carrier power.
[:SENSe]:SPURious:CARRier:BANDwidth[: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.

**Table 2-20: SENSE commands (Cont.)**

<b>Header</b>	<b>Description</b>
[[: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.
[[: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.
<b>[[:SENSe]:TOVerview subgroup</b>	<b>Time overview</b>
[[: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 2-21: 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.

**Table 2-21: STATus commands (Cont.)**

Header	Description
: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.
: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 2-22: 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 2-23: TRACe commands**

Header	Description
<b>:TRACe&lt;x&gt;:AVTime subgroup</b>	<b>Amplitude versus Time measurement</b>
: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.
<b>:TRACe&lt;x&gt;:CCDF subgroup</b>	<b>CCDF measurement</b>
: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.
<b>:TRACe:CONStE subgroup (Option 21 only)</b>	<b>Constellation measurement</b>
:TRACe:CONStE:MODE(?)	Selects or queries how to display the constellation trace.
<b>:TRACe&lt;x&gt;:DPSA subgroup</b>	<b>DPX spectrum measurement</b>
: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.

**Table 2-23: TRACe commands (Cont.)**

<b>Header</b>	<b>Description</b>
: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.
<b>:TRACe:FVTime subgroup</b>	<b>Frequency versus Time measurement</b>
: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.
<b>:TRACe:IQVTime subgroup</b>	<b>RF I&amp;Q versus Time measurement</b>
: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.
<b>:TRACe:OBWidth subgroup</b>	<b>Occupied Bandwidth measurement</b>
:TRACe:OBWidth:MAXHold(?)	Determines whether to enable or disable the Max Hold trace.
<b>:TRACe:PHVTime subgroup</b>	<b>Phase versus Time measurement</b>
: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.

Table 2-23: TRACe commands (Cont.)

Header	Description
<b>:TRACe:PNOise subgroup (Option 11 only)</b>	<b>Phase noise measurement</b>
: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.
<b>:TRACe:SGRam subgroup</b>	<b>Spectrogram measurement</b>
: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.
<b>:TRACe&lt;x&gt;:SPECTrum subgroup</b>	<b>Spectrum measurement</b>
: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.
<b>:TRACe:SPURious subgroup</b>	<b>Spurious measurement</b>
: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.

## TRIGger Commands

Use the TRIGger commands to set up the trigger system.

**Table 2-24: 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 2-25: UNIT commands**

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

## 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.



# IEEE Common Commands

This section details the IEEE common commands.

## Command Tree

Header	Parameter
*CAL	
*CLS	
*ESE	
*ESR?	
*IDN?	
*OPC	
*OPT?	
*RST	
*SRE	
*STB?	
*TRG	
*WAI	

## **\*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.*

---

**Syntax**     \*CAL?

**Returns**     <NR1> = 1 indicates that the alignment was successful.  
                 <NR1> = 0 indicates that the alignment was unsuccessful.

**Measurement Views**     All

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

## \*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 <E0I>, 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, begins on page 3-1.)

\*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.

<b>Syntax</b>	*CLS
<b>Arguments</b>	None
<b>Measurement Views</b>	All
<b>Examples</b>	*CLS clears the analyzer status data structures.
<b>Related Commands</b>	*ESE, *ESR, *SRE, *STB?

**\*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.

**Syntax** \*ESE <value>

\*ESE?

**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.

**Measurement Views** All

**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.

**Related Commands** \*CLS, \*ESR, \*SRE, \*STB?

**\*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.

**Syntax** \*ESR?

**Arguments** None

**Returns** <NR1> representing the contents of the SESR by a 0 to 255 decimal number.

**Measurement Views** All

**Examples** \*ESR?  
might return the value 213, showing that the SESR contains binary 11010101.

**Related Commands** \*CLS, \*ESE?, \*SRE, \*STB?

**\*IDN? (Query Only)**

Returns the analyzer identification code.

**Syntax** \*IDN?

**Arguments** None

**Measurement Views** All

**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,FV:1.0.1500.

**\*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* on page 3-12 for the details.

**Syntax** \*OPC

\*OPC?

**Arguments** None

**Measurement Views** All

**\*OPT? (Query Only)**

Returns a list of options installed in your analyzer.

**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.

Table 2-26 lists the options for the RSA6100A Series analyzers.

**Table 2-26: 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 <sup>1</sup>	Removable HDD, 80 GB
07 <sup>1</sup>	DVD-RW
20	Advanced signal analysis (including pulse measurements)
21	General purpose modulation analysis
110	110 MHz real-time capture bandwidth
1R <sup>1</sup>	Rackmount

<sup>1</sup> These options are not returned from this query.

**Measurement Views** All

**Examples** \*OPT?  
might return 02,05,21, indicating that Option 02, 05, and 21 are currently installed in the analyzer.

## \*RST (No Query Form)

Returns the instrument settings to the factory defaults (see *Appendix C: 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.

---

<b>Syntax</b>	*RST
<b>Arguments</b>	None
<b>Measurement Views</b>	All
<b>Examples</b>	*RST returns the instrument settings to the factory defaults.
<b>Related Commands</b>	*CLS



**\*SRE(?)**

Sets or queries the value of the Service Request Enable Register (SRER). Refer to Section 3, *Status and Events*, for the register information.

**Syntax** \*SRE <value>

\*SRE?

**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.

**Measurement Views** All

**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.

**Related Commands** \*CLS, \*ESE, \*ESR?, \*STB?

**\*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.

**Syntax** \*STB?

**Arguments** None

**Returns** <NR1> representing the contents of the SBR as a decimal number.

**Measurement Views** All

**Examples** \*STB?  
might return 96, indicating that the SBR contains binary 0110 0000.

**Related Commands** \*CLS, \*ESE, \*ESR?, \*SRE

## \*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.

<b>Syntax</b>	*TRG
<b>Arguments</b>	None
<b>Measurement Views</b>	All
<b>Examples</b>	*TRG generates a trigger.
<b>Related Commands</b>	:TRIGger[:SEQuence]:STATus

## \*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* on page 3-12.

<b>Syntax</b>	*WAI
<b>Arguments</b>	None
<b>Measurement Views</b>	All
<b>Related Commands</b>	*OPC



# ABORt Commands

The ABORt command resets the trigger system and places all trigger sequences in the idle state.

## Command Tree

Header	Parameter
:ABORt	

### :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.

**Syntax** :ABORt

**Arguments** None

**Measurement Views** All

**Examples** :ABORt  
resets the trigger system and stops data acquisition.

**Related Commands** INITiate commands



# CALCulate Commands

The CALCulate commands control the markers and the search operations. These commands are divided by measurement views into the subgroups listed in Table 2-27.

**Table 2-27: CALCulate command subgroups**

Command header	Measurement view	Refer to:
:CALCulate (basic)	All views, general marker control	page 2-77
:CALCulate:ACPower	Channel power and ACPR	page 2-92
:CALCulate:AVTime	Amplitude versus Time	page 2-97
:CALCulate:CONSte	Constellation (Option 21 only)	page 2-104
:CALCulate:DPsA	DPX (Digital Phosphor) spectrum	page 2-109
:CALCulate:EVM	EVM versus Time (Option 21 only)	page 2-116
:CALCulate:FVTime	Frequency versus Time	page 2-122
:CALCulate:IQVTime	RF I&Q versus Time	page 2-128
:CALCulate:MCPower	MCPR (Multiple Carrier Power Ratio)	page 2-135
:CALCulate:MERRor	Magnitude error versus Time (Option 21 only)	page 2-140
:CALCulate:OBWidth	OBW (Occupied Bandwidth)	page 2-146
:CALCulate:PERRor	Phase error versus Time (Option 21 only)	page 2-152
:CALCulate:PHVTime	Phase versus Time	page 2-158
:CALCulate:PULSe	Pulsed RF measurements (Option 20 only)	page 2-164
:CALCulate:SGRam	Spectrogram	page 2-175
:CALCulate:SPEctrum	Spectrum	page 2-182
:CALCulate:SPURious	Spurious	page 2-189
:CALCulate:TOVerview	Time overview	page 2-195

## 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 Table 2-28.

**Table 2-28: 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 (refer to page 2-77).*

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

---



## :CALCulate Basic Command Subgroup

The :CALCulate basic commands globally control the markers and the search operations.

Command Tree	Header	Parameter
	:CALCulate	
	:MARKer	
	:ADD	
	:AOFF	
	:DELeTe	
	:MODE	ABSolute   DELTa
	:PEAK	
	:EXCursion	<numeric_value>
	:THReshold	<numeric_value>
	:SEARCh	
	:LIMit	
	:FAIL?	
	:MATCH	
	:BEEP	
	[:STATE]	<boolean>
	:SACQuire	
	[:STATE]	<boolean>
	:SDATa	
	[:STATE]	<boolean>
	:SPICture	
	[:STATE]	<boolean>
	:STRace	
	[:STATE]	<boolean>
	:OPERation	LT   GT   IMASK   OMASK
	:FEED	<string>,<string>
	:MASK	
	:LOAD	<string>
	:STORE	<string>
	:SLIMit	<numeric_value>
	:REPort	
	:DATA?	
	:POINts?	
	:STATe	<boolean>

**: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.*

---

<b>Syntax</b>	:CALCulate:MARKer:ADD
<b>Arguments</b>	None
<b>Measurement Views</b>	All
<b>Examples</b>	:CALCulate:MARKer:ADD adds a marker.
<b>Related Commands</b>	:CALCulate:MARKer:AOff, :CALCulate:MARKer:DELeTe

**:CALCulate:MARKer:AOff (No Query Form)**

Turns off all markers.

<b>Syntax</b>	:CALCulate:MARKer:AOff
<b>Arguments</b>	None
<b>Measurement Views</b>	All
<b>Examples</b>	:CALCulate:MARKer:AOff turns off all markers.
<b>Related Commands</b>	:CALCulate:MARKer:ADD, :CALCulate:MARKer:DELeTe

**: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.*

---

<b>Syntax</b>	:CALCulate:MARKer:DELeTe
<b>Arguments</b>	None
<b>Measurement Views</b>	All
<b>Examples</b>	:CALCulate:MARKer:DELeTe deletes the last marker added.
<b>Related Commands</b>	:CALCulate:MARKer:ADD, :CALCulate:MARKer:AOff

**:CALCulate:MARKer:MODE(?)**

Selects or queries the marker mode.

<b>Syntax</b>	:CALCulate:MARKer:MODE { ABSolute   DELTa } :CALCulate:MARKer:MODE?
<b>Arguments</b>	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.
<b>Measurement Views</b>	All
<b>Examples</b>	:CALCulate:MARKer:MODE DELTA selects the delta marker mode.

**:CALCulate:MARKer:PEAK:EXCursion(?)**

Sets or queries the 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 excursion level and the threshold level set by the :CALCulate:MARKer:PEAK:THReshold command.

**Syntax** :CALCulate:MARKer:PEAK:EXCursion <value>  
:CALCulate:MARKer:PEAK:EXCursion?

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

**Measurement Views** All

**Examples** :CALCulate:MARKer:PEAK:EXCursion 10  
sets the excursion level to 10 dB.

**Related Commands** :CALCulate:MARKer:PEAK:THReshold

**:CALCulate:MARKer:PEAK:THReshold(?)**

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

**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.

**Measurement Views** All

**Examples** :CALCulate:MARKer:PEAK:THReshold -50  
sets the threshold level to -50 dBm.

**:CALCulate:SEARch:LIMit:FAIL? (Query Only)**

Queries whether the waveform cuts across the limit in the search operation.

**Syntax** :CALCulate:SEARch:LIMit:FAIL?

**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.

**Measurement Views** All

**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.

**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.

**Measurement Views** All

**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.

**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.

**Measurement Views** All

**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.

**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* on page 2-495 in the MMEMory command section.

**Measurement Views** All

**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.

**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* on page 2-495 in the MMEMory command section.

**Measurement Views**   All

**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.

**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* on page 2-495 in the MMEMory command section.

**Measurement Views** All

**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.

**Syntax** :CALCulate:SEARch:LIMit:OPERation { LT | GT | IMASK | OMASK }  
:CALCulate:SEARch:LIMit:OPERation?

**Arguments** Table 2-29 lists the arguments.

**Table 2-29: 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:FEED command.

You can set the limit value using the :CALCulate:SEARch:LIMit:OPERation:SLIMit command.

You can store or load the limit mask using the :MMEMory:LOAD|:STORE:LIMit command.

**Measurement Views** All

**Examples** :CALCulate:SEARch:LIMit:OPERation GT  
selects “the data is greater than the limit value” for the search condition.

**Related Commands** :CALCulate:SEARch:LIMit:FEED,  
:CALCulate:SEARch:LIMit:OPERation:SLIMit,  
:MMEMory:LOAD|:STORE:LIMit

**:CALCulate:SEARch:LIMit:OPERation:FEED(?)**

Sets or queries the data flow to be fed in the search operation.

**Syntax** :CALCulate:SEARch:LIMit:OPERation:FEED <view>,<trace>  
:CALCulate:SEARch:LIMit:OPERation:FEED?

**Arguments** <view>::=<string> and <trace>::=<string>are listed in Table 2-30.

**Table 2-30: Source data of the search operation**

<view>	<trace>	Meaning
"Spectrum"	"Trace 1" <sup>1</sup>	Trace 1 in the Spectrum view.
	"Trace 2" <sup>1</sup>	Trace 2 in the Spectrum view.
	"Trace 3" <sup>1</sup>	Trace 3 in the Spectrum view.
	"Math Trace"	Math trace in the Spectrum view.
	"Spectrogram Trace"	Spectrogram trace in the Spectrum view.

<sup>1</sup> **There is a space character between Trace and the number.**

**Measurement Views** All

**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.

**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* on page 2-495 in the MMEMory command section.

**Measurement Views** All

**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.

**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* on page 2-495 in the MMEMory command section.

**Measurement Views** All

**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.

**Syntax** :CALCulate:SEARch:LIMit:OPERation:SLIMit <value>  
:CALCulate:SEARch:LIMit:OPERation:SLIMit?

**Arguments** <value>::=<NRf> specifies the limit value in the search operation.  
Range: -100 to +100 dBm.

**Measurement Views** All

**Examples** :CALCulate:SEARch:LIMit:OPERation:SLIMit -20  
sets the limit value to -20 dBm.

**Related Commands** :CALCulate:SEARch:LIMit:OPERation

**:CALCulate:SEARch:LIMit:REPort:DATA? (Query Only)**

Returns the frequency range(s) that satisfy the search condition.

**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<sup>th</sup> 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.

**Measurement Views** All

**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.

**Syntax** :CALCulate:SEARch:LIMit:REPort:POINTs?

**Arguments** None

**Returns** <number>::=<NR1> represents the number of frequency range(s) that satisfy the search condition.

**Measurement Views** All

**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.

**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.

**Measurement Views** All

**Examples** :CALCulate:SEARch:LIMit:STATe ON  
enables the search function.

## :CALCulate:ACPower Subgroup

The :CALCulate:ACPower commands control the markers in the Channel power and ACPR (Adjacent Channel Power Ratio) measurement.

Command Tree	Header	Parameter
	:CALCulate	
	:ACPower	
	:MARKer<x>	
	:DELTA	
	:X?	
	:Y?	
	:MAXimum	
	:PEAK	
	:LEFT	
	:RIGHT	
	:X	<numeric_value>
	:Y?	



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

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

**Syntax** :CALCulate:ACPower:MARKer<x>:DELTA:X?

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

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

**Measurement Views** Channel power and ACPR

**Examples** :CALCulate:ACPower:MARKer1:DELTA:X?  
might return 1.28E+6, indicating that the delta marker frequency is 1.28 MHz.

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

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

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

**Syntax** :CALCulate:ACPower:MARKer<x>:DELTA:Y?

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

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

**Measurement Views** Channel power and ACPR

**Examples** :CALCulate:ACPower:MARKer1:DELTA:Y?  
might return 23.45, indicating that the delta marker amplitude is 23.45 dB.

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

## **: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.

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

**Arguments** None

**Measurement Views** Channel power and ACPR

**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.

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

**Arguments** None

**Measurement Views** Channel power and ACPR

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

**Related Commands** :CALCulate:ACPower:MARKer<x>:PEAK:RIGHT

**: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.

**Syntax** :CALCulate:ACPower:MARKer<x>:PEAK:RIGHT

**Arguments** None

**Measurement Views** Channel power and ACPR

**Examples** :CALCulate:ACPower:MARKer1:PEAK:RIGHT  
moves Marker 1 (M1) to the next peak to the right on the trace.

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

**:CALCulate:ACPower:MARKer<x>:X(?)**

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

**Syntax** :CALCulate:ACPower:MARKer<x>:X <value>

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

**Arguments** <value> ::= <NRf> specifies the horizontal position of the marker.  
Range: (center frequency)  $\pm$  (span)/2.

**Measurement Views** Channel power and ACPR

**Examples** :CALCulate:ACPower:MARKer1:X 800MHz  
places Marker 1 (M1) at 800 MHz on the trace.

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

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

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

**Syntax** :CALCulate:ACPower:MARKer<x>:Y?

**Arguments** None

**Measurement Views** Channel power and ACPR

**Examples** :CALCulate:ACPower:MARKer1:Y?  
might return -34.28 indicating Marker 1 (M1) is at -34.28 dBm.

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

## :CALCulate:AVTime Subgroup

The :CALCulate:AVTime commands control the markers in the Amplitude versus Time measurement.

Command Tree	Header	Parameter
	:CALCulate	
	:AVTime	
	:MARKer<x>	
	:DELTA	
	:X?	
	:Y?	
	:MAXimum	
	:PEAK	
	:HIGHer	
	:LEFT	
	:LOWer	
	:RIGHT	
	:TRACe	TRACE1   TRACE2   TRACE3   TRACE4
	:X	<numeric_value>
	:Y?	

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

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

**Syntax** :CALCulate:AVTime:MARKer<x>:DELTA:X?

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

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

**Measurement Views** Amplitude versus Time

**Examples** :CALCulate:AVTime:MARKer1:DELTA:X?  
might return 38.0E-9, indicating that the delta marker time is 38.0 ns.

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

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

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

**Syntax** :CALCulate:AVTime:MARKer<x>:DELTA:Y?

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

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

**Measurement Views** Amplitude versus Time

**Examples** :CALCulate:AVTime:MARKer1:DELTA:Y?  
might return 23.45, indicating that the delta marker amplitude is 23.45 dB.

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

**: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.

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

**Arguments** None

**Measurement Views** Amplitude versus Time

**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.

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

**Arguments** None

**Measurement Views** Amplitude versus Time

**Examples** :CALCulate:AVTime:MARKer1:PEAK:HIGHer  
moves Marker 1 (M1) to the next peak higher in amplitude on the trace.

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

**: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.

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

**Arguments** None

**Measurement Views** Amplitude versus Time

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

**Related Commands** :CALCulate:AVTime:MARKer<x>:PEAK:RIGHT



**: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.

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

**Arguments** None

**Measurement Views** Amplitude versus Time

**Examples** :CALCulate:AVTime:MARKer1:PEAK:LOWer  
moves Marker 1 (M1) to the next peak lower in amplitude on the trace.

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

**: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.

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

**Arguments** None

**Measurement Views** Amplitude versus Time

**Examples** :CALCulate:AVTime:MARKer1:PEAK:RIGHt  
moves Marker 1 (M1) to the next peak to the right on the trace.

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

**:CALCulate:AVTime:MARKer<x>:TRACe(?)**

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

**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).

**Measurement Views**   Amplitude versus Time

**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.

**Syntax** :CALCulate:AVTime:MARKer<x>:X <value>

:CALCulate:AVTime:MARKer<x>:X?

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

**Measurement Views** Amplitude versus Time

**Examples** :CALCulate:AVTime:MARKer1:X 1.5u  
places Marker 1 (M1) at 1.5  $\mu$ s on the trace.

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

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

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

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

**Arguments** None

**Measurement Views** Amplitude versus Time

**Examples** :CALCulate:AVTime:MARKer1:Y?  
might return -2.73 indicating Marker 1 (M1) is at -2.73 dBm.

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

## :CALCulate:CONSte Subgroup (Option 21 Only)

The :CALCulate:CONSte commands control the markers in the constellation measurement.

Command Tree	Header	Parameter
	:CALCulate	
	:CONSte	
	:MARKer<x>	
	:DELTA	
	:X	
		[:TIME]?
	:MAGNitude?	
	:MAXimum	
	:PEAK	
	:LEFT	
	:RIGHT	
	:PHASe?	
	:SYMBol?	
	:VALue?	
	:X	<numeric_value>

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

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

**Syntax**     :CALCulate:CONSte:MARKer<x>:DELTA:X[:TIME]?

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

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

**Measurement Views**    Constellation

**Examples**     :CALCulate:CONSte:MARKer1:DELTA:X:TIME?  
might return -62.75, indicating that the delta marker time is -62.75 symbols.

**Related Commands**    :CALCulate:MARKer:ADD, [:SENSe]:DDEMod:TIME:UNITs

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

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

**Syntax** :CALCulate:CONSt:MARKer<x>:MAGNitude?

**Arguments** None

**Returns** <Nrf> The magnitude readout for the selected marker.

**Measurement Views** Constellation

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

**Related Commands** :CALCulate:CONSt:MARKer<x>:X

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

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

**Syntax** :CALCulate:CONSt:MARKer<x>:MAXimum

**Arguments** None

**Measurement Views** Constellation

**Examples** :CALCulate:CONSt: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.

**Syntax** :CALCulate:CONStE:MARKer<x>:PEAK:LEFT

**Arguments** None

**Measurement Views** Constellation

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

**Related Commands** :CALCulate:CONStE:MARKer<x>:PEAK:RIGHT

**: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.

**Syntax** :CALCulate:CONStE:MARKer<x>:PEAK:RIGHT

**Arguments** None

**Measurement Views** Constellation

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

**Related Commands** :CALCulate:CONStE:MARKer<x>:PEAK:LEFT

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

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

**Syntax** :CALCulate:CONSt:MARKer<x>:PHASe?

**Arguments** None

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

**Measurement Views** Constellation

**Examples** :CALCulate:CONSt:MARKer1:PHASe?  
might return 35.74 indicating the phase readout of Marker 1 (M1) is 35.74°.

**Related Commands** :CALCulate:CONSt:MARKer<x>:X

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

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

**Syntax** :CALCulate:CONSt:MARKer<x>:SYMBol?

**Arguments** None

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

**Measurement Views** Constellation

**Examples** :CALCulate:CONSt:MARKer1:SYMBol?  
might return 62.00 indicating the symbol readout of Marker 1 (M1) is 62.

**Related Commands** :CALCulate:CONSt:MARKer<x>:X

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

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

**Syntax** :CALCulate:CONSt:MARKer<x>:VALue?

**Arguments** None

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

**Measurement Views** Constellation

**Examples** :CALCulate:CONSt:MARKer1:VALue?  
might return 2.00 indicating the value readout of Marker 1 (M1) is 2.

**Related Commands** :CALCulate:CONSt:MARKer<x>:X

**:CALCulate:CONSt:MARKer<x>:X(?)**

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

**Syntax** :CALCulate:CONSt:MARKer<x>:X <value>  
:CALCulate:CONSt:MARKer<x>:X?

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

**Measurement Views** Constellation

**Examples** :CALCulate:CONSt:MARKer1:X -1.63875m  
places the Marker 1 (M1) at -1.63875 ms on the constellation trace.

**Related Commands** :CALCulate:CONSt:MARKer<x>:MAGNitude[:PHASe[:SYMBOL]:VALue?



## :CALCulate:DPSA Subgroup

The :CALCulate:DPSA commands control the markers in the DPX (Digital Phosphor) spectrum measurement.

Command Tree	Header	Parameter
	:CALCulate	
	:DPSA	
	:MARKer<x>	
	:DELTA	
	:X?	
	:Y?	
	:MAXimum	
	:PEAK	
	:HIGHer	
	:LEFT	
	:LOWer	
	:RIGHT	
	[:SET]	
	:CENTer	
	:TRACe	BITMAP   TRACE1   TRACE2   TRACE3   TRACE4
	:X	<numeric_value>
	:Y?	

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

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

**Syntax** :CALCulate:DPSA:MARKer<x>:DELTA:X?

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

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

**Measurement Views** DPX spectrum

**Examples** :CALCulate:DPSA:MARKer1:DELTA:X?  
might return 1.28E+6, indicating that the delta marker frequency is 1.28 MHz.

**Related Commands** :CALCulate:MARKer:ADD, :CALCulate:DPSA:MARKer<x>:DELTA:Y?

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

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

**Syntax** :CALCulate:DPSA:MARKer<x>:DELTA:Y?

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

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

**Measurement Views** DPX spectrum

**Examples** :CALCulate:DPSA:MARKer1:DELTA:Y?  
might return 23.45, indicating that the delta marker amplitude is 23.45 dB.

**Related Commands** :CALCulate:MARKer:ADD, :CALCulate:DPSA:MARKer<x>:DELTA:X?

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

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

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

**Arguments** None

**Measurement Views** DPX spectrum

**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.

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

**Arguments** None

**Measurement Views** DPX spectrum

**Examples** :CALCulate:DPSA:MARKer1:PEAK:HIGHer  
moves Marker 1 (M1) to the next peak higher in amplitude on the trace.

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

**: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.

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

**Arguments** None

**Measurement Views** DPX spectrum

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

**Related Commands** :CALCulate:DPSA:MARKer<x>:PEAK:RIGHT

**: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.

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

**Arguments** None

**Measurement Views** DPX spectrum

**Examples** :CALCulate:DPSA:MARKer1:PEAK:LOWer  
moves Marker 1 (M1) to the next peak lower in amplitude on the trace.

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

**: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.

**Syntax** :CALCulate:DPSA:MARKer<x>:PEAK:RIGHT

**Arguments** None

**Measurement Views** DPX spectrum

**Examples** :CALCulate:DPSA:MARKer1:PEAK:RIGHT  
moves Marker 1 (M1) to the next peak to the right on the trace.

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

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

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

**Syntax** :CALCulate:DPSA:MARKer<x>[:SET]:CENTER

**Arguments** None

**Measurement Views** DPX spectrum

**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.

**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 maximum trace.  
TRACE2 places the specified marker on the minimum trace.  
TRACE3 places the specified marker on the average trace.  
TRACE4 places the specified marker on the math trace.

**Measurement Views**   DPX spectrum

**Examples**     :CALCulate:DPSA:MARKer1:TRACe TRACE1  
                  places Marker 1 (M1) on the maximum trace.

**:CALCulate:DPSA:MARKer<x>:X(?)**

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

**Syntax** :CALCulate:DPSA:MARKer<x>:X <value>

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

**Arguments** <value>::=<NRf> specifies the horizontal position of the marker.  
Range: (center frequency)  $\pm$  (span)/2.

**Measurement Views** DPX spectrum

**Examples** :CALCulate:DPSA:MARKer1:X 800MHz  
places Marker 1 (M1) at 800 MHz on the trace.

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

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

Queries the vertical position of the selected marker in the DPX spectrum view.

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

**Arguments** None

**Measurement Views** DPX spectrum

**Examples** :CALCulate:DPSA:MARKer1:Y?  
might return -34.28 indicating Marker 1 (M1) is at -34.28 dBm.

**Related Commands** :CALCulate:DPSA:MARKer<x>:X

## :CALCulate:EVM Subgroup (Option 21 Only)

The :CALCulate:EVM commands control the markers in the EVM versus Time measurement.

Command Tree	Header	Parameter
	:CALCulate	
	:EVM	
	:MARKer<x>	
	:DELTA	
	:X?	
	:Y?	
	:MAXimum	
	:PEAK	
	:HIGHer	
	:LEFT	
	:LOWer	
	:RIGHT	
	:X	<numeric_value>
	:Y?	



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

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

**Syntax** :CALCulate:EVM:MARKer<x>:DELTA:X?

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

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

**Measurement Views** EVM versus Time

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

**Related Commands** :CALCulate:MARKer:ADD, :CALCulate:EVM:MARKer<x>:DELTA:Y?, [:SENSe]:DDEMod:TIME:UNITs

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

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

**Syntax** :CALCulate:EVM:MARKer<x>:DELTA:Y?

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

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

**Measurement Views** EVM versus Time

**Examples** :CALCulate:EVM:MARKer1:DELTA:Y?  
might return 1.62, indicating that the delta marker amplitude is 1.62%.

**Related Commands** :CALCulate:MARKer:ADD, :CALCulate:EVM:MARKer<x>:DELTA:X?

## **: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.

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

**Arguments** None

**Measurement Views** EVM versus Time

**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.

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

**Arguments** None

**Measurement Views** EVM versus Time

**Examples** :CALCulate:EVM:MARKer1:PEAK:HIGHer  
moves Marker 1 (M1) to the next peak higher in amplitude on the trace.

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

**: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.

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

**Arguments** None

**Measurement Views** EVM versus Time

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

**Related Commands** :CALCulate:EVM:MARKer<x>:PEAK:RIGHT

**: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.

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

**Arguments** None

**Measurement Views** EVM versus Time

**Examples** :CALCulate:EVM:MARKer1:PEAK:LOWer  
moves Marker 1 (M1) to the next peak lower in amplitude on the trace.

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

**: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.

**Syntax** :CALCulate:EVM:MARKer<x>:PEAK:RIGHT

**Arguments** None

**Measurement Views** EVM versus Time

**Examples** :CALCulate:EVM:MARKer1:PEAK:RIGHT  
moves Marker 1 (M1) to the next peak to the right on the trace.

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

**:CALCulate:EVM:MARKer<x>:X(?)**

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

**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.

**Measurement Views** EVM versus Time

**Examples** :CALCulate:EVM:MARKer1:X 1.5us  
places Marker 1 (M1) at 1.5  $\mu$ s on the trace.

**Related Commands** :CALCulate:EVM:MARKer<x>:Y?, [:SENSe]:DDEMod:TIME:UNITs

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

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

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

**Arguments** None

**Measurement Views** EVM versus Time

**Examples** :CALCulate:EVM:MARKer1:Y?  
might return -15.34E+6 indicating Marker 1 (M1) is at -15.34 MHz.

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

## :CALCulate:FVTime Subgroup

The :CALCulate:FVTime commands control the markers in the Frequency versus Time measurement.

Command Tree	Header	Parameter
	:CALCulate	
	:FVTime	
	:MARKer<x>	
	:DELTA	
	:X?	
	:Y?	
	:MAXimum	
	:PEAK	
	:HIGHer	
	:LEFT	
	:LOWer	
	:RIGHT	
	:X	<numeric_value>
	:Y?	

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

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

**Syntax** :CALCulate:FVTime:MARKer<x>:DELTA:X?

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

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

**Measurement Views** Frequency versus Time

**Examples** :CALCulate:FVTime:MARKer1:DELTA:X?  
might return 120.0E-9, indicating that the delta marker time is 120 ns.

**Related Commands** :CALCulate:MARKer:ADD, :CALCulate:FVTime:MARKer<x>:DELTA:Y?

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

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

**Syntax** :CALCulate:FVTime:MARKer<x>:DELTA:Y?

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

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

**Measurement Views** Frequency versus Time

**Examples** :CALCulate:FVTime:MARKer1:DELTA:Y?  
might return 27.05E+3, indicating that the delta marker frequency is 27.05 kHz.

**Related Commands** :CALCulate:MARKer:ADD, :CALCulate:FVTime:MARKer<x>:DELTA:X?

## **: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.

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

**Arguments**    None

**Measurement Views**    Frequency versus Time

**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.

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

**Arguments** None

**Measurement Views** Frequency versus Time

**Examples** :CALCulate:FVTime:MARKer1:PEAK:HIGHer  
moves Marker 1 (M1) to the next peak higher in amplitude on the trace.

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

**: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.

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

**Arguments** None

**Measurement Views** Frequency versus Time

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

**Related Commands** :CALCulate:FVTime:MARKer<x>:PEAK:RIGHT

**: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.

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

**Arguments** None

**Measurement Views** Frequency versus Time

**Examples** :CALCulate:FVTime:MARKer1:PEAK:LOWer  
moves Marker 1 (M1) to the next peak lower in amplitude on the trace.

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

**: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.

**Syntax** :CALCulate:FVTime:MARKer<x>:PEAK:RIGHT

**Arguments** None

**Measurement Views** Frequency versus Time

**Examples** :CALCulate:FVTime:MARKer1:PEAK:RIGHT  
moves Marker 1 (M1) to the next peak to the right on the trace.

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

**:CALCulate:FVTime:MARKer<x>:X(?)**

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

**Syntax** :CALCulate:FVTime:MARKer<x>:X <value>

:CALCulate:FVTime:MARKer<x>:X?

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

**Measurement Views** Frequency versus Time

**Examples** :CALCulate:FVTime:MARKer1:X 1.5u  
places Marker 1 (M1) at 1.5  $\mu$ s on the trace.

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

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

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

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

**Arguments** None

**Measurement Views** Frequency versus Time

**Examples** :CALCulate:FVTime:MARKer1:Y?  
might return -15.34E+6 indicating Marker 1 (M1) is at -15.34 MHz.

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

## :CALCulate:IQVTime Subgroup

The :CALCulate:IQVTime commands control the markers in the RF I&Q versus Time measurements.

Command Tree	Header	Parameter
	:CALCulate	
	:IQVTime	
	:MARKer<x>	
	:DELTA	
	:X?	
	:Y?	
	:MAXimum	
	:PEAK	
	:HIGHer	
	:LEFT	
	:LOWer	
	:RIGHT	
	:TRACe	TRACE1   TRACE2
	:X	<numeric_value>
	:Y?	

**: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.

**Syntax** :CALCulate:IQVTime:MARKer<x>:DELTA:X?

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

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

**Measurement Views** RF I&Q versus Time

**Examples** :CALCulate:IQVTime:MARKer1:DELTA:X?  
might return 120.0E-9, indicating that the delta marker time is 120 ns.

**Related Commands** :CALCulate:MARKer:ADD, :CALCulate:IQVTime:MARKer<x>:DELTA:Y?

**: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.

**Syntax** :CALCulate:IQVTime:MARKer<x>:DELTA:Y?

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

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

**Measurement Views** RF I&Q versus Time

**Examples** :CALCulate:IQVTime:MARKer1:DELTA:Y?  
might return -3.45E-3, indicating that the delta marker amplitude is -3.45 mV.

**Related Commands** :CALCulate:MARKer:ADD, :CALCulate:IQVTime:MARKer<x>:DELTA:X?

## **: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.

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

**Arguments** None

**Measurement Views** RF I&Q versus Time

**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.

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

**Arguments** None

**Measurement Views** RF I&Q versus Time

**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.

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

**Arguments** None

**Measurement Views** RF I&Q versus Time

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

**Related Commands** :CALCulate:IQVTime:MARKer<x>:PEAK:RIGHT

**: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.

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

**Arguments** None

**Measurement Views** RF I&Q versus Time

**Examples** :CALCulate:IQVTime:MARKer1:PEAK:LOWer  
moves Marker 1 (M1) to the next peak lower in amplitude on the trace.

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

**: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.

**Syntax** :CALCulate:IQVTime:MARKer<x>:PEAK:RIGHT

**Arguments** None

**Measurement Views** RF I&Q versus Time

**Examples** :CALCulate:IQVTime:MARKer1:PEAK:RIGHT  
moves Marker 1 (M1) to the next peak to the right on the trace.

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



**: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.

**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.

**Measurement Views** RF I&Q versus Time

**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.

**Syntax** :CALCulate:IQVTime:MARKer<x>:X <value>

:CALCulate:IQVTime:MARKer<x>:X?

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

**Measurement Views** RF I&Q versus Time

**Examples** :CALCulate:IQVTime:MARKer1:X 1.5us  
places Marker 1 (M1) at 1.5  $\mu$ s on the trace.

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

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

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

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

**Arguments** None

**Measurement Views** RF I&Q versus Time

**Examples** :CALCulate:IQVTime:MARKer1:Y?  
might return 25.803E-3 indicating Marker 1 (M1) is at 25.803 mV.

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

## :CALCulate:MCPower Subgroup

The :CALCulate:MCPower commands control the markers in the MCPR (Multiple Carrier Power Ratio) measurement.

Command Tree	Header	Parameter
	:CALCulate	
	:MCPower	
	:MARKer<x>	
	:DELTA	
	:X?	
	:Y?	
	:MAXimum	
	:PEAK	
	:LEFT	
	:RIGHT	
	:X	<numeric_value>
	:Y?	

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

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

**Syntax** :CALCulate:MCPower:MARKer<x>:DELTA:X?

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

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

**Measurement Views** MCPR

**Examples** :CALCulate:MCPower:MARKer1:DELTA:X?  
might return 1.28E+6, indicating that the delta marker frequency is 1.28 MHz.

**Related Commands** :CALCulate:MARKer:ADD, :CALCulate:MCPower:MARKer<x>:DELTA:Y?

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

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

**Syntax** :CALCulate:MCPower:MARKer<x>:DELTA:Y?

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

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

**Measurement Views** MCPR

**Examples** :CALCulate:MCPower:MARKer1:DELTA:Y?  
might return 23.45, indicating that the delta marker amplitude is 23.45 dB.

**Related Commands** :CALCulate:MARKer:ADD, :CALCulate:MCPower:MARKer<x>:DELTA:X?

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

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

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

**Arguments** None

**Measurement Views** MCPR

**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.

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

**Arguments** None

**Measurement Views** MCPR

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

**Related Commands** :CALCulate:MCPower:MARKer<x>:PEAK:RIGHT

**: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.

**Syntax** :CALCulate:MCPower:MARKer<x>:PEAK:RIGHT

**Arguments** None

**Measurement Views** MCPR

**Examples** :CALCulate:MCPower:MARKer1:PEAK:RIGHT  
moves Marker 1 (M1) to the next peak to the right on the trace.

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

**:CALCulate:MCPower:MARKer<x>:X(?)**

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

**Syntax** :CALCulate:MCPower:MARKer<x>:X <value>

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

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

**Measurement Views** MCPR

**Examples** :CALCulate:MCPower:MARKer1:X 800MHz  
places Marker 1 (M1) at 800 MHz on the trace.

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

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

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

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

**Arguments** None

**Measurement Views** MCPR

**Examples** :CALCulate:MCPower:MARKer1:Y?  
might return -34.28 indicating Marker 1 (M1) is at -34.28 dBm.

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

## :CALCulate:MERRor Subgroup (Option 21 Only)

The :CALCulate:MERRor commands control the markers in the Magnitude error versus Time measurement.

Command Tree	Header	Parameter
	:CALCulate	
	:MERRor	
	:MARKer<x>	
	:DELTA	
	:X?	
	:Y?	
	:MAXimum	
	:PEAK	
	:HIGHer	
	:LEFT	
	:LOWer	
	:RIGHT	
	:X	<numeric_value>
	:Y?	



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

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

**Syntax** :CALCulate:MERRor:MARKer<x>:DELTA:X?

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

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

**Measurement Views** Magnitude error versus Time

**Examples** :CALCulate:MERRor:MARKer1:DELTA:X?  
might return 9.52, indicating that the delta marker time is 9.52 symbols.

**Related Commands** :CALCulate:MARKer:ADD, :CALCulate:MERRor:MARKer<x>:DELTA:Y?, [:SENSe]:DDEMod:TIME:UNITs

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

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

**Syntax** :CALCulate:MERRor:MARKer<x>:DELTA:Y?

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

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

**Measurement Views** Magnitude error versus Time

**Examples** :CALCulate:MERRor:MARKer1:DELTA:Y?  
might return 3.84, indicating that the delta marker amplitude is 3.84%.

**Related Commands** :CALCulate:MARKer:ADD, :CALCulate:MERRor:MARKer<x>:DELTA:X?

## **: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.

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

**Arguments** None

**Measurement Views** Magnitude error versus Time

**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.

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

**Arguments** None

**Measurement Views** Magnitude error versus Time

**Examples** :CALCulate:MERRor:MARKer1:PEAK:HIGHer  
moves Marker 1 (M1) to the next peak higher in amplitude on the trace.

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

**: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.

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

**Arguments** None

**Measurement Views** Magnitude error versus Time

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

**Related Commands** :CALCulate:MERRor:MARKer<x>:PEAK:RIGHT

**: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.

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

**Arguments** None

**Measurement Views** Magnitude error versus Time

**Examples** :CALCulate:MERRor:MARKer1:PEAK:LOWer  
moves Marker 1 (M1) to the next peak lower in amplitude on the trace.

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

**: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.

**Syntax** :CALCulate:MERRor:MARKer<x>:PEAK:RIGHT

**Arguments** None

**Measurement Views** Magnitude error versus Time

**Examples** :CALCulate:MERRor:MARKer1:PEAK:RIGHT  
moves Marker 1 (M1) to the next peak to the right on the trace.

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

**:CALCulate:MERRor:MARKer<x>:X(?)**

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

**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.

**Measurement Views** Magnitude error versus Time

**Examples** :CALCulate:MERRor:MARKer1:X 1.5u  
places Marker 1 (M1) at 1.5  $\mu$ s on the trace.

**Related Commands** :CALCulate:MERRor:MARKer<x>:Y?, [:SENSe]:DDEMod:TIME:UNITs

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

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

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

**Arguments** None

**Measurement Views** Magnitude error versus Time

**Examples** :CALCulate:MERRor:MARKer1:Y?  
might return -15.34E+6 indicating Marker 1 (M1) is at -15.34 MHz.

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

## :CALCulate:OBWidth Subgroup

The :CALCulate:OBWidth commands control the markers in the Occupied Bandwidth (OBW) measurement.

Command Tree	Header	Parameter
	:CALCulate	
	:OBWidth	
	:MARKer<x>	
	:DELTA	
	:X?	
	:Y?	
	:MAXimum	
	:PEAK	
	:HIGHer	
	:LEFT	
	:LOWer	
	:RIGHT	
	[:SET]	
	:CENTer	
	:X	<numeric_value>
	:Y?	

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

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

**Syntax** :CALCulate:OBWidth:MARKer<x>:DELTA:X?

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

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

**Measurement Views** Occupied Bandwidth

**Examples** :CALCulate:OBWidth:MARKer1:DELTA:X?  
might return 1.28E+6, indicating that the delta marker frequency is 1.28 MHz.

**Related Commands** :CALCulate:MARKer:ADD, :CALCulate:OBWidth:MARKer<x>:DELTA:Y?

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

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

**Syntax** :CALCulate:OBWidth:MARKer<x>:DELTA:Y?

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

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

**Measurement Views** Occupied Bandwidth

**Examples** :CALCulate:OBWidth:MARKer1:DELTA:Y?  
might return 23.45, indicating that the delta marker amplitude is 23.45 dB.

**Related Commands** :CALCulate:MARKer:ADD, :CALCulate:OBWidth:MARKer<x>:DELTA:X?

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

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

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

**Arguments** None

**Measurement Views** Occupied Bandwidth

**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.

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

**Arguments** None

**Measurement Views** Occupied Bandwidth

**Examples** :CALCulate:OBWidth:MARKer1:PEAK:HIGHer  
moves Marker 1 (M1) to the next peak higher in amplitude on the trace.

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



**: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.

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

**Arguments** None

**Measurement Views** Occupied Bandwidth

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

**Related Commands** :CALCulate:OBWidth:MARKer<x>:PEAK:RIGHT

**: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.

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

**Arguments** None

**Measurement Views** Occupied Bandwidth

**Examples** :CALCulate:OBWidth:MARKer1:PEAK:LOWer  
moves Marker 1 (M1) to the next peak lower in amplitude on the trace.

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

**: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.

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

**Arguments** None

**Measurement Views** Occupied Bandwidth

**Examples** :CALCulate:OBWidth:MARKer1:PEAK:RIGHt  
moves Marker 1 (M1) to the next peak to the right on the trace.

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

**: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.

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

**Arguments** None

**Measurement Views** Occupied Bandwidth

**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.

**Syntax** :CALCulate:OBWidth:MARKer<x>:X <value>

:CALCulate:OBWidth:MARKer<x>:X?

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

**Measurement Views** Occupied Bandwidth

**Examples** :CALCulate:OBWidth:MARKer1:X 800MHz  
places Marker 1 (M1) at 800 MHz on the trace.

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

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

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

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

**Arguments** None

**Measurement Views** Occupied Bandwidth

**Examples** :CALCulate:OBWidth:MARKer1:Y?  
might return -34.28 indicating Marker 1 (M1) is at -34.28 dBm.

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

## :CALCulate:PERRor Subgroup (Option 21 Only)

The :CALCulate:PERRor commands control the markers in the Phase error versus Time measurement.

Command Tree	Header	Parameter
	:CALCulate	
	:PERRor	
	:MARKer<x>	
	:DELTA	
	:X?	
	:Y?	
	:MAXimum	
	:PEAK	
	:HIGHer	
	:LEFT	
	:LOWer	
	:RIGHT	
	:X	<numeric_value>
	:Y?	

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

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

**Syntax** :CALCulate:PERRor:MARKer<x>:DELTA:X?

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

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

**Measurement Views** Phase error versus Time

**Examples** :CALCulate:PERRor:MARKer1:DELTA:X?  
might return 9.52, indicating that the delta marker time is 9.52 symbols.

**Related Commands** :CALCulate:MARKer:ADD, :CALCulate:PERRor:MARKer<x>:DELTA:Y?, [:SENSe]:DDEMod:TIME:UNITs

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

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

**Syntax** :CALCulate:PERRor:MARKer<x>:DELTA:Y?

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

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

**Measurement Views** Phase error versus Time

**Examples** :CALCulate:PERRor:MARKer1:DELTA:Y?  
might return -7.93, indicating that the delta marker phase is  $-7.93^\circ$ .

**Related Commands** :CALCulate:MARKer:ADD, :CALCulate:PERRor:MARKer<x>:DELTA:X?

## **: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.

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

**Arguments**    None

**Measurement Views**    Phase error versus Time

**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.

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

**Arguments** None

**Measurement Views** Phase error versus Time

**Examples** :CALCulate:PERRor:MARKer1:PEAK:HIGHer  
moves Marker 1 (M1) to the next peak higher in amplitude on the trace.

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

**: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.

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

**Arguments** None

**Measurement Views** Phase error versus Time

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

**Related Commands** :CALCulate:PERRor:MARKer<x>:PEAK:RIGHT

**: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.

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

**Arguments** None

**Measurement Views** Phase error versus Time

**Examples** :CALCulate:PERRor:MARKer1:PEAK:LOWer  
moves Marker 1 (M1) lower in amplitude to the next peak on the trace.

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

**: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.

**Syntax** :CALCulate:PERRor:MARKer<x>:PEAK:RIGHT

**Arguments** None

**Measurement Views** Phase error versus Time

**Examples** :CALCulate:PERRor:MARKer1:PEAK:RIGHT  
moves Marker 1 (M1) to the next peak to the right on the trace.

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



**:CALCulate:PERRor:MARKer<x>:X(?)**

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

**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.

**Measurement Views** Phase error versus Time

**Examples** :CALCulate:PERRor:MARKer1:X 1.5u  
places Marker 1 (M1) at 1.5  $\mu$ s on the trace.

**Related Commands** :CALCulate:PERRor:MARKer<x>:Y?, [:SENSe]:DDEMod:TIME:UNITs

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

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

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

**Arguments** None

**Measurement Views** Phase error versus Time

**Examples** :CALCulate:PERRor:MARKer1:Y?  
might return -15.34E+6 indicating Marker 1 (M1) is at -15.34 MHz.

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

## :CALCulate:PHVTime Subgroup

The :CALCulate:PHVTime commands control the markers in the Phase versus Time measurement.

Command Tree	Header	Parameter
	:CALCulate	
	:PHVTime	
	:MARKer<x>	
	:DELTA	
	:X?	
	:Y?	
	:MAXimum	
	:PEAK	
	:HIGHer	
	:LEFT	
	:LOWer	
	:RIGHT	
	:X	<numeric_value>
	:Y?	

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

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

**Syntax** :CALCulate:PHVTime:MARKer<x>:DELTA:X?

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

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

**Measurement Views** Phase versus Time

**Examples** :CALCulate:PHVTime:MARKer1:DELTA:X?  
might return 38.0E-9, indicating that the delta marker time is 38.0 ns.

**Related Commands** :CALCulate:MARKer:ADD, :CALCulate:PHVTime:MARKer<x>:DELTA:Y?

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

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

**Syntax** :CALCulate:PHVTime:MARKer<x>:DELTA:Y?

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

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

**Measurement Views** Phase versus Time

**Examples** :CALCulate:PHVTime:MARKer1:DELTA:Y?  
might return 162.38, indicating that the delta marker phase is 162.38°.

**Related Commands** :CALCulate:MARKer:ADD, :CALCulate:PHVTime:MARKer<x>:DELTA:X?

## **: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.

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

**Arguments** None

**Measurement Views** Phase versus Time

**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.

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

**Arguments** None

**Measurement Views** Phase versus Time

**Examples** :CALCulate:PHVTime:MARKer1:PEAK:HIGHer  
moves Marker 1 (M1) to the next peak higher in amplitude on the trace.

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

**: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.

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

**Arguments** None

**Measurement Views** Phase versus Time

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

**Related Commands** :CALCulate:PHVTime:MARKer<x>:PEAK:RIGHT

**: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.

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

**Arguments** None

**Measurement Views** Phase versus Time

**Examples** :CALCulate:PHVTime:MARKer1:PEAK:LOWer  
moves Marker 1 (M1) to the next peak lower in amplitude on the trace.

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

**: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.

**Syntax** :CALCulate:PHVTime:MARKer<x>:PEAK:RIGHT

**Arguments** None

**Measurement Views** Phase versus Time

**Examples** :CALCulate:PHVTime:MARKer1:PEAK:RIGHT  
moves Marker 1 (M1) to the next peak to the right on the trace.

**Related Commands** :CALCulate:PHVTime:MARKer<x>:PEAK:LEFT

**:CALCulate:PHVTime:MARKer<x>:X(?)**

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

**Syntax** :CALCulate:PHVTime:MARKer<x>:X <value>

:CALCulate:PHVTime:MARKer<x>:X?

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

**Measurement Views** Phase versus Time

**Examples** :CALCulate:PHVTime:MARKer1:X 1.5us  
places Marker 1 (M1) at 1.5  $\mu$ s on the trace.

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

**:CALCulate:PHVTime:MARKer<x>:Y?**

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

**Syntax** :CALCulate:PHVTime:MARKer<x>:Y?

**Arguments** None

**Measurement Views** Phase versus Time

**Examples** :CALCulate:PHVTime:MARKer1:Y?  
might return -18.435 indicating Marker 1 (M1) is at -18.435°.

**Related Commands** :CALCulate:PHVTime:MARKer<x>:X

## :CALCulate:PULSE Subgroup (Option 20 Only)

The :CALCulate:PULSE commands control the markers in the Pulsed RF measurements.

Command Tree	Header	Parameter
	:CALCulate	
	:PULSE	
	:STATistics	
	:MARKer<x>	
	:DELTA	
	:X?	
	:Y?	
	:MAXimum	
	:PEAK	
	:HIGHer	
	:LEFT	
	:LOWer	
	:RIGHT	
	:X	<numeric_value>
	:Y?	
	:TRACe	
	:MARKer<x>	
	:DELTA	
	:X?	
	:Y?	
	:MAXimum	
	:PEAK	
	:HIGHer	
	:LEFT	
	:LOWer	
	:RIGHT	
	:X	<numeric_value>
	:Y?	

---

**NOTE.** The :CALCulate:PULSE:STATistics commands are valid when :DISPlay:PULSE:STATistics:PLOT is set to FFT.

---

For the marker mnemonics, refer to page 2-76.



**: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.

**Syntax** :CALCulate:PULSE:STATistics:MARKer<x>:DELTA:X?

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

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

**Measurement Views** Pulse statistics

**Examples** :CALCulate:PULSE:STATistics:MARKer1:DELTA:X?  
might return 614.2, indicating that the delta marker frequency is 614.2 Hz.

**Related Commands** :CALCulate:MARKer:ADD, :DISPlay:PULSE:STATistics:PLOT,  
:CALCulate:PULSE:STATistics:MARKer<x>:DELTA:Y?

**: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.

**Syntax** :CALCulate:PULSE:STATistics:MARKer<x>:DELTA:Y?

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

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

**Measurement Views** Pulse statistics

**Examples** :CALCulate:PULSE:STATistics:MARKer1:DELTA:Y?  
might return 2.345, indicating that the delta marker amplitude is 2.345 dB.

**Related Commands** :CALCulate:MARKer:ADD, :DISPlay:PULSE:STATistics:PLOT,  
:CALCulate:PULSE:STATistics:MARKer<x>:DELTA:X?

**: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.

**Syntax** :CALCulate:PULSE:STATistics:MARKer<x>:MAXimum

**Arguments** None

**Measurement Views** Pulse statistics

**Examples** :CALCulate:PULSE:STATistics:MARKer1:MAXimum  
moves Marker 1 (M1) to the highest peak on the trace.

**Related Commands** :DISPlay:PULSE:STATistics:PLOT

**: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.

**Syntax** :CALCulate:PULSE:STATistics:MARKer<x>:PEAK:HIGHer

**Arguments** None

**Measurement Views** Pulse statistics

**Examples** :CALCulate:PULSE:STATistics:MARKer1:PEAK:HIGHer  
moves Marker 1 (M1) to the next peak higher in amplitude on the trace.

**Related Commands** :CALCulate:PULSE:STATistics:MARKer<x>:PEAK:LOWer,  
:DISPlay:PULSE:STATistics:PLOT

**: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.

**Syntax** :CALCulate:PULSE:STATistics:MARKer<x>:PEAK:LEFT

**Arguments** None

**Measurement Views** Pulse statistics

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

**Related Commands** :CALCulate:PULSE:STATistics:MARKer<x>:PEAK:RIGHT,  
:DISPlay:PULSE:STATistics:PLOT

**: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.

**Syntax** :CALCulate:PULSE:STATistics:MARKer<x>:PEAK:LOWer

**Arguments** None

**Measurement Views** Pulse statistics

**Examples** :CALCulate:PULSE:STATistics:MARKer1:PEAK:LOWer  
moves Marker 1 (M1) to the next peak lower in amplitude on the trace.

**Related Commands** :CALCulate:PULSE:STATistics:MARKer<x>:PEAK:HIGHer,  
:DISPlay:PULSE:STATistics:PLOT

**: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.

**Syntax** :CALCulate:PULSE:STATistics:MARKer<x>:PEAK:RIGHT

**Arguments** None

**Measurement Views** Pulse statistics

**Examples** :CALCulate:PULSE:STATistics:MARKer1:PEAK:RIGHT  
moves Marker 1 (M1) to the next peak to the right on the trace.

**Related Commands** :CALCulate:PULSE:STATistics:MARKer<x>:PEAK:LEFT,  
:DISPlay:PULSE:STATistics:PLOT

**: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.

**Syntax** :CALCulate:PULSE:STATistics:MARKer<x>:X <value>  
:CALCulate:PULSE:STATistics:MARKer<x>:X?

**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.

**Measurement Views** Pulse statistics

**Examples** :CALCulate:PULSE:STATistics:MARKer1:X 12.5kHz  
places Marker 1 (M1) at 12.5 kHz on the trace.

**Related Commands** :CALCulate:PULSE:STATistics:MARKer<x>:Y?,  
:DISPlay:PULSE:STATistics:PLOT

**:CALCulate:PULSE:STATistics:MARKer<x>:Y?**

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.

**Syntax** :CALCulate:PULSE:STATistics:MARKer<x>:Y?

**Arguments** None

**Measurement Views** Pulse statistics

**Examples** :CALCulate:PULSE:STATistics:MARKer1:Y?  
might return -28.173 indicating Marker 1 (M1) is at -28.173 dB.

**Related Commands** :CALCulate:PULSE:STATistics:MARKer<x>:X,  
:DISPlay:PULSE:STATistics:PLOT

**:CALCulate:PULSE:TRACE:MARKer<x>:DELTA:X? (Query Only)**

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

**Syntax** :CALCulate:PULSE:TRACE:MARKer<x>:DELTA:X?

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

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

**Measurement Views** Pulse trace

**Examples** :CALCulate:PULSE:TRACE:MARKer1:DELTA:X?  
might return 38.0E-9, indicating that the delta marker time is 38.0 ns.

**Related Commands** :CALCulate:MARKer:ADD,  
:CALCulate:PULSE:TRACE:MARKer<x>:DELTA:Y?

**:CALCulate:PULSE:TRACE:MARKer<x>:DELTA:Y? (Query Only)**

Returns the delta marker amplitude for the selected marker on the pulse trace.

**Syntax** :CALCulate:PULSE:TRACE:MARKer<x>:DELTA:Y?

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

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

**Measurement Views** Pulse trace

**Examples** :CALCulate:PULSE:TRACE:MARKer1:DELTA:Y?  
might return 23.45, indicating that the delta marker amplitude is 23.45 dB.

**Related Commands** :CALCulate:MARKer:ADD,  
:CALCulate:PULSE:TRACE:MARKer<x>:DELTA:X?

**:CALCulate:PULSE:TRACE:MARKer<x>:MAXimum (No Query Form)**

Moves the selected marker to the highest peak on the pulse trace.

**Syntax** :CALCulate:PULSE:TRACE:MARKer<x>:MAXimum

**Arguments** None

**Measurement Views** Pulse trace

**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.

**Syntax** :CALCulate:PULSE:TRACE:MARKer<x>:PEAK:HIGHer

**Arguments** None

**Measurement Views** Pulse trace

**Examples** :CALCulate:PULSE:TRACE:MARKer1:PEAK:HIGHer  
moves Marker 1 (M1) to the next peak higher in amplitude on the trace.

**Related Commands** :CALCulate:PULSE:TRACE:MARKer<x>:PEAK:LOWer

**: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.

**Syntax** :CALCulate:PULSE:TRACe:MARKer<x>:PEAK:LEFT

**Arguments** None

**Measurement Views** Pulse trace

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

**Related Commands** :CALCulate:PULSE:TRACe:MARKer<x>:PEAK:RIGHT

**: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.

**Syntax** :CALCulate:PULSE:TRACe:MARKer<x>:PEAK:LOWer

**Arguments** None

**Measurement Views** Pulse trace

**Examples** :CALCulate:PULSE:TRACe:MARKer1:PEAK:LOWer  
moves Marker 1 (M1) to the next peak lower in amplitude on the trace.

**Related Commands** :CALCulate:PULSE:TRACe:MARKer<x>:PEAK:HIGHer



**: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.

**Syntax** :CALCulate:PULSE:TRACE:MARKer<x>:PEAK:RIGHT

**Arguments** None

**Measurement Views** Pulse trace

**Examples** :CALCulate:PULSE:TRACE:MARKer1:PEAK:RIGHT  
moves Marker 1 (M1) to the next peak to the right on the trace.

**Related Commands** :CALCulate:PULSE:TRACE:MARKer<x>:PEAK:LEFT

**:CALCulate:PULSE:TRACe:MARKer<x>:X(?)**

Sets or queries the horizontal position of the selected marker in the pulse trace view.

**Syntax** :CALCulate:PULSE:TRACe:MARKer<x>:X <value>

:CALCulate:PULSE:TRACe:MARKer<x>:X?

**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.

---

**Measurement Views** Pulse trace

**Examples** :CALCulate:PULSE:TRACe:MARKer1:X 1.5us  
places Marker 1 (M1) at 1.5  $\mu$ s on the trace.

**Related Commands** :CALCulate:PULSE:TRACe:MARKer<x>:Y?,  
:DISPlay:PULSE:SElect:NUMBER

**:CALCulate:PULSE:TRACe:MARKer<x>:Y?**

Queries the vertical position of the selected marker in the pulse trace view.

**Syntax** :CALCulate:PULSE:TRACe:MARKer<x>:Y?

**Arguments** None

**Measurement Views** Pulse trace

**Examples** :CALCulate:PULSE:TRACe:MARKer1:Y?  
might return 228.858E-3 indicating Marker 1 (M1) is at 228.858 mV.

**Related Commands** :CALCulate:PULSE:TRACe:MARKer<x>:X

## :CALCulate:SGRam Subgroup

The :CALCulate:SGRam commands control the markers in the Spectrogram.

Command Tree	Header	Parameter
	:CALCulate	
	:SGRam	
	:MARKer<x>	
	:DELTA	
	:X	
	:FREQuency?	
	[:TIME]?	
	:Y?	
	:MAXimum	
	:PEAK	
	:HIGHer	
	:LEFT	
	:LOWer	
	:RIGHT	
	[:SET]	
	:CENTer	
	:X	
	:FREQuency	<numeric_value>
	[:TIME]	<numeric_value>
	:Y?	

**:CALCulate:SGRam:MARKer<x>:DELTA:X:FREQUENCY? (Query Only)**

Returns the delta marker frequency for the selected marker in the spectrogram.

**Syntax** :CALCulate:SGRam:MARKer<x>:DELTA:X:FREQUENCY?

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

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

**Measurement Views** Spectrogram

**Examples** :CALCulate:SGRam:MARKer1:DELTA:X:FREQUENCY?  
might return 5.95E+6, indicating that the delta marker frequency is 5.95 MHz.

**Related Commands** :CALCulate:MARKer:ADD, :CALCulate:SGRam:MARKer<x>:DELTA:Y?

**:CALCulate:SGRam:MARKer<x>:DELTA:X[:TIME]? (Query Only)**

Returns the delta marker time for the selected marker in the spectrogram.

**Syntax** :CALCulate:SGRam:MARKer<x>:DELTA:X[:TIME]?

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

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

**Measurement Views** Spectrogram

**Examples** :CALCulate:SGRam:MARKer1:DELTA:X:TIME?  
might return -1.84E-3, indicating that the delta marker time is -1.84 ms.

**Related Commands** :CALCulate:MARKer:ADD, :CALCulate:SGRam:MARKer<x>:DELTA:Y?

**:CALCulate:SGRam:MARKer<x>:DELTA:Y? (Query Only)**

Returns the delta marker amplitude for the selected marker in the spectrogram.

**Syntax** :CALCulate:SGRam:MARKer<x>:DELTA:Y?

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

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

**Measurement Views** Spectrogram

**Examples** :CALCulate:SGRam:MARKer1:DELTA:Y?  
might return -8.45, indicating that the delta marker amplitude is -8.45 dB.

**Related Commands** :CALCulate:MARKer:ADD, :CALCulate:SGRam:MARKer<x>:DELTA:X?

**: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.

**Syntax** :CALCulate:SGRam:MARKer<x>:MAXimum

**Arguments** None

**Measurement Views** Spectrogram

**Examples** :CALCulate:SGRam:MARKer1:MAXimum  
moves Marker 1 (M1) to the highest peak on the line.

**Related Commands** :TRACe:SGRam:SELEct: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.

**Syntax** :CALCulate:SGRam:MARKer<x>:PEAK:HIGHer

**Arguments** None

**Measurement Views** Spectrogram

**Examples** :CALCulate:SGRam:MARKer1:PEAK:HIGHer  
moves Marker 1 (M1) to the next peak higher in amplitude on the line.

**Related Commands** :CALCulate:SGRam:MARKer<x>:PEAK:LOWer,  
TRACe:SGRam:SElect: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.

**Syntax** :CALCulate:SGRam:MARKer<x>:PEAK:LEFT

**Arguments** None

**Measurement Views** Spectrogram

**Examples** :CALCulate:SGRam:MARKer1:PEAK:LEFT  
moves Marker 1 (M1) to the next peak to the left on the line.

**Related Commands** :CALCulate:SGRam:MARKer<x>:PEAK:RIGHT,  
:TRACe:SGRam:SElect: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.

**Syntax** :CALCulate:SGRam:MARKer<x>:PEAK:LOWer

**Arguments** None

**Measurement Views** Spectrogram

**Examples** :CALCulate:SGRam:MARKer1:PEAK:LOWer  
moves Marker 1 (M1) to the next peak lower in amplitude on the line.

**Related Commands** :CALCulate:SGRam:MARKer<x>:PEAK:HIGHer,  
:TRACe:SGRam:SElect: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.

**Syntax** :CALCulate:SGRam:MARKer<x>:PEAK:RIGHT

**Arguments** None

**Measurement Views** Spectrogram

**Examples** :CALCulate:SGRam:MARKer1:PEAK:RIGHT  
moves Marker 1 (M1) to the next peak to the right on the trace.

**Related Commands** :CALCulate:SGRam:MARKer<x>:PEAK:LEFT,  
:TRACe:SGRam:SElect:LINE

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

Sets the center frequency to the marker frequency in the spectrogram.

**Syntax** :CALCulate:SGRam:MARKer<x>[:SET]:CENTer

**Arguments** None

**Measurement Views** Spectrogram

**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.

**Syntax** :CALCulate:SGRam:MARKer<x>:X <value>  
:CALCulate:SGRam:MARKer<x>:X?

**Arguments** <value>::=<NRf> specifies the marker frequency.  
Range: (center frequency)  $\pm$  (span)/2.

**Measurement Views** Spectrogram

**Examples** :CALCulate:SGRam:MARKer1:X 800MHz  
places Marker 1 (M1) at 800 MHz on the trace.

**Related Commands** :CALCulate:SGRam:MARKer<x>:Y?



**:CALCulate:SGRam:MARKer<x>:X[:TIME](?)**

Sets or queries the marker time in the spectrogram.

**Syntax** :CALCulate:SGRam:MARKer<x>:X[:TIME] <value>  
:CALCulate:SGRam:MARKer<x>:X[:TIME]?

**Arguments** <value>::=<NRf> specifies the marker time.

**Measurement Views** Spectrogram

**Examples** :CALCulate:SGRam:MARKer1:X:TIME -234.5us  
places Marker 1 (M1) at -234.5  $\mu$ s on the trace.

**Related Commands** :CALCulate:SGRam:MARKer<x>:Y?

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

Queries the marker amplitude in the spectrogram.

**Syntax** :CALCulate:SGRam:MARKer<x>:Y?

**Arguments** None

**Measurement Views** Spectrogram

**Examples** :CALCulate:SGRam:MARKer1:Y?  
might return -34.28 indicating Marker 1 (M1) is at -34.28 dBm.

**Related Commands** :CALCulate:SGRam:MARKer<x>:X

## :CALCulate:SPECTrum Subgroup

The :CALCulate:SPECTrum commands control the markers in the Spectrum measurement.

Command Tree	Header	Parameter
	:CALCulate	
	:SPECTrum	
	:MARKer<x>	
	:DELTA	
	:X?	
	:Y?	
	:MAXimum	
	:PEAK	
	:HIGHer	
	:LEFT	
	:LOWer	
	:RIGHT	
	[:SET]	
	:CENTer	
	:TRACe	TRACE1   TRACE2   TRACE3   TRACE4   TRACE5
	:X	<numeric_value>
	:Y?	

**:CALCulate:SPECTrum:MARKer<x>:DELTA:X? (Query Only)**

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

**Syntax** :CALCulate:SPECTrum:MARKer<x>:DELTA:X?

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

**Returns** <Nrf> Delta marker frequency for the selected marker.

**Measurement Views** Spectrum

**Examples** :CALCulate:SPECTrum:MARKer1:DELTA:X?  
might return 1.28E+6, indicating that the delta marker frequency is 1.28 MHz.

**Related Commands** :CALCulate:MARKer:ADD, :CALCulate:SPECTrum:MARKer<x>:DELTA:Y?

**:CALCulate:SPECTrum:MARKer<x>:DELTA:Y? (Query Only)**

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

**Syntax** :CALCulate:SPECTrum:MARKer<x>:DELTA:Y?

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

**Returns** <Nrf> Delta marker amplitude for the selected marker.

**Measurement Views** Spectrum

**Examples** :CALCulate:SPECTrum:MARKer1:DELTA:Y?  
might return 23.45, indicating that the delta marker amplitude is 23.45 dB.

**Related Commands** :CALCulate:MARKer:ADD, :CALCulate:SPECTrum:MARKer<x>:DELTA:X?

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

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

**Syntax** :CALCulate:SPECTrum:MARKer<x>:MAXimum

**Arguments** None

**Measurement Views** Spectrum

**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.

**Syntax** :CALCulate:SPECTrum:MARKer<x>:PEAK:HIGHer

**Arguments** None

**Measurement Views** Spectrum

**Examples** :CALCulate:SPECTrum:MARKer1:PEAK:HIGHer  
moves Marker 1 (M1) to the next peak higher in amplitude on the trace.

**Related Commands** :CALCulate:SPECTrum:MARKer<x>:PEAK:LOWer

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

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

**Syntax** :CALCulate:SPECTrum:MARKer<x>:PEAK:LEFT

**Arguments** None

**Measurement Views** Spectrum

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

**Related Commands** :CALCulate:SPECTrum:MARKer<x>:PEAK:RIGHT

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

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

**Syntax** :CALCulate:SPECTrum:MARKer<x>:PEAK:LOWer

**Arguments** None

**Measurement Views** Spectrum

**Examples** :CALCulate:SPECTrum:MARKer1:PEAK:LOWer  
moves Marker 1 (M1) to the next peak lower in amplitude on the trace.

**Related Commands** :CALCulate:SPECTrum:MARKer<x>:PEAK:HIGHer

**:CALCulate:SPECTrum:MARKer<x>:PEAK:RIGHT (No Query Form)**

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

**Syntax** :CALCulate:SPECTrum:MARKer<x>:PEAK:RIGHT

**Arguments** None

**Measurement Views** Spectrum

**Examples** :CALCulate:SPECTrum:MARKer1:PEAK:RIGHT  
moves Marker 1 (M1) to the next peak to the right on the trace.

**Related Commands** :CALCulate:SPECTrum:MARKer<x>:PEAK:LEFT

**:CALCulate:SPECTrum:MARKer<x>[:SET]:CENTER (No Query Form)**

Sets the center frequency to the marker frequency in the spectrum measurement.

**Syntax** :CALCulate:SPECTrum:MARKer<x>[:SET]:CENTER

**Arguments** None

**Measurement Views** Spectrum

**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.

**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.

**Measurement Views** Spectrum

**Examples** :CALCulate:SPECTrum:MARKer1:TRACe TRACE1  
 places Marker 1 (M1) on Trace 1.

**Related Commands** :TRACe<x>:SPECTrum:FUNCTion

**:CALCulate:SPECTrum:MARKer<x>:X(?)**

Sets or queries the horizontal position of the selected marker on the spectrum trace.

**Syntax** :CALCulate:SPECTrum:MARKer<x>:X <value>

:CALCulate:SPECTrum:MARKer<x>:X?

**Arguments** <value> ::= <NRf> specifies the horizontal position of the marker.  
Range: (center frequency)  $\pm$  (span)/2.

**Measurement Views** Spectrum

**Examples** :CALCulate:SPECTrum:MARKer1:X 800MHz  
places Marker 1 (M1) at 800 MHz on the spectrum trace.

**Related Commands** :CALCulate:SPECTrum:MARKer<x>:Y?

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

Queries the vertical position of the selected marker on the spectrum trace.

**Syntax** :CALCulate:SPECTrum:MARKer<x>:Y?

**Arguments** None

**Measurement Views** Spectrum

**Examples** :CALCulate:SPECTrum:MARKer1:Y?  
might return -34.28 indicating Marker 1 (M1) is at -34.28 dBm.

**Related Commands** :CALCulate:SPECTrum:MARKer<x>:X



## :CALCulate:SPURious Subgroup

The :CALCulate:SPURious commands control the markers in the Spurious measurement.

Command Tree	Header	Parameter
	:CALCulate	
	:SPURious	
	:MARKer<x>	
	:DELTA	
	:X?	
	:Y?	
	:MAXimum	
	:PEAK	
	:HIGHer	
	:LEFT	
	:LOWer	
	:RIGHT	
	[:SET]	
	:CENTer	
	:X	<numeric_value>
	:Y?	

**:CALCulate:SPURious:MARKer<x>:DELTA:X? (Query Only)**

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

**Syntax** :CALCulate:SPURious:MARKer<x>:DELTA:X?

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

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

**Measurement Views** Spurious

**Examples** :CALCulate:SPURious:MARKer1:DELTA:X?  
might return 1.28E+6, indicating that the delta marker frequency is 1.28 MHz.

**Related Commands** :CALCulate:MARKer:ADD, :CALCulate:SPURious:MARKer<x>:DELTA:Y?

**:CALCulate:SPURious:MARKer<x>:DELTA:Y? (Query Only)**

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

**Syntax** :CALCulate:SPURious:MARKer<x>:DELTA:Y?

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

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

**Measurement Views** Spurious

**Examples** :CALCulate:SPURious:MARKer1:DELTA:Y?  
might return 23.45, indicating that the delta marker amplitude is 23.45 dB.

**Related Commands** :CALCulate:MARKer:ADD, :CALCulate:SPURious:MARKer<x>:DELTA:X?

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

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

**Syntax** :CALCulate:SPURious:MARKer<x>:MAXimum

**Arguments** None

**Measurement Views** Spurious

**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.

**Syntax** :CALCulate:SPURious:MARKer<x>:PEAK:HIGHer

**Arguments** None

**Measurement Views** Spurious

**Examples** :CALCulate:SPURious:MARKer1:PEAK:HIGHer  
moves Marker 1 (M1) to the next peak higher in amplitude on the trace.

**Related Commands** :CALCulate:SPURious:MARKer<x>:PEAK:LOWer

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

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

**Syntax** :CALCulate:SPURious:MARKer<x>:PEAK:LEFT

**Arguments** None

**Measurement Views** Spurious

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

**Related Commands** :CALCulate:SPURious:MARKer<x>:PEAK:RIGHT

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

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

**Syntax** :CALCulate:SPURious:MARKer<x>:PEAK:LOWer

**Arguments** None

**Measurement Views** Spurious

**Examples** :CALCulate:SPURious:MARKer1:PEAK:LOWer  
moves Marker 1 (M1) to the next peak lower in amplitude on the trace.

**Related Commands** :CALCulate:SPURious:MARKer<x>:PEAK:HIGHer

**:CALCulate:SPURious:MARKer<x>:PEAK:RIGHT (No Query Form)**

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

**Syntax** :CALCulate:SPURious:MARKer<x>:PEAK:RIGHT

**Arguments** None

**Measurement Views** Spurious

**Examples** :CALCulate:SPURious:MARKer1:PEAK:RIGHT  
moves Marker 1 (M1) to the next peak to the right on the trace.

**Related Commands** :CALCulate:SPURious:MARKer<x>:PEAK:LEFT

**:CALCulate:SPURious:MARKer<x>[:SET]:CENTER (No Query Form)**

Sets the center frequency to the marker frequency in the Spurious measurement.

**Syntax** :CALCulate:SPURious:MARKer<x>[:SET]:CENTER

**Arguments** None

**Measurement Views** Spurious

**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.

**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 :STOP commands.

**Measurement Views** Spurious

**Examples** :CALCulate:SPURious:MARKer1:X 800MHz  
places Marker 1 (M1) at 800 MHz on the spectrum trace.

**Related Commands** :CALCulate:SPURious:MARKer<x>:Y?,  
:DISPlay:SPURious:X[:SCALe]:START, :DISPlay:SPURious:X[:SCALe]:STOP

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

Queries the vertical position of the selected marker on the spectrum trace.

**Syntax** :CALCulate:SPURious:MARKer<x>:Y?

**Arguments** None

**Measurement Views** Spurious

**Examples** :CALCulate:SPURious:MARKer1:Y?  
might return -34.28, indicating Marker 1 (M1) is at -34.28 dBm.

**Related Commands** :CALCulate:SPURious:MARKer<x>:X

## :CALCulate:TOVerview Subgroup

The :CALCulate:TOVerview commands control the markers in the Time overview.

Command Tree	Header	Parameter
	:CALCulate	
	:TOVerview	
	:MARKer<x>	
	:DELTA	
	:X?	
	:Y?	
	:MAXimum	
	:PEAK	
	:HIGHer	
	:LEFT	
	:LOWer	
	:RIGHT	
	:X	<numeric_value>
	:Y?	

**:CALCulate:TOVerview:MARKer<x>:DELTA:X? (Query Only)**

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

**Syntax** :CALCulate:TOVerview:MARKer<x>:DELTA:X?

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

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

**Measurement Views** Time overview

**Examples** :CALCulate:TOVerview:MARKer1:DELTA:X?  
might return 38.0E-9, indicating that the delta marker time is 38.0 ns.

**Related Commands** :CALCulate:MARKer:ADD, :CALCulate:TOVerview:MARKer<x>:DELTA:Y?

**:CALCulate:TOVerview:MARKer<x>:DELTA:Y? (Query Only)**

Returns the delta marker amplitude for the selected marker on the time overview trace.

**Syntax** :CALCulate:TOVerview:MARKer<x>:DELTA:Y?

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

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

**Measurement Views** Time overview

**Examples** :CALCulate:TOVerview:MARKer1:DELTA:Y?  
might return 23.45, indicating that the delta marker amplitude is 23.45 dB.

**Related Commands** :CALCulate:MARKer:ADD, :CALCulate:TOVerview:MARKer<x>:DELTA:X?



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

Moves the selected marker to the highest peak on the time overview trace.

**Syntax** :CALCulate:TOVerview:MARKer<x>:MAXimum

**Arguments** None

**Measurement Views** Time overview

**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.

**Syntax** :CALCulate:TOVerview:MARKer<x>:PEAK:HIGHer

**Arguments** None

**Measurement Views** Time overview

**Examples** :CALCulate:TOVerview:MARKer1:PEAK:HIGHer  
moves Marker 1 (M1) to the next peak higher in amplitude on the trace.

**Related Commands** :CALCulate:TOVerview:MARKer<x>:PEAK:LOWer

**: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.

**Syntax** :CALCulate:TOVerview:MARKer<x>:PEAK:LEFT

**Arguments** None

**Measurement Views** Time overview

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

**Related Commands** :CALCulate:TOVerview:MARKer<x>:PEAK:RIGHT

**: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.

**Syntax** :CALCulate:TOVerview:MARKer<x>:PEAK:LOWer

**Arguments** None

**Measurement Views** Time overview

**Examples** :CALCulate:TOVerview:MARKer1:PEAK:LOWer  
moves Marker 1 (M1) to the next peak lower in amplitude on the trace.

**Related Commands** :CALCulate:TOVerview:MARKer<x>:PEAK:HIGHer

**: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.

**Syntax** :CALCulate:TOVerview:MARKer<x>:PEAK:RIGHT

**Arguments** None

**Measurement Views** Time overview

**Examples** :CALCulate:TOVerview:MARKer1:PEAK:RIGHT  
moves Marker 1 (M1) to the next peak to the right on the trace.

**Related Commands** :CALCulate:TOVerview:MARKer<x>:PEAK:LEFT

**:CALCulate:TOVerview:MARKer<x>:X(?)**

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

**Syntax** :CALCulate:TOVerview:MARKer<x>:X <value>

:CALCulate:TOVerview:MARKer<x>:X?

**Arguments** <value>::=<NRf> specifies the horizontal position of the marker.  
Range: (center frequency)  $\pm$  (span)/2.

**Measurement Views** Time overview

**Examples** :CALCulate:TOVerview:MARKer1:X 800MHz  
places Marker 1 (M1) at 800 MHz on the trace.

**Related Commands** :CALCulate:TOVerview:MARKer<x>:Y?

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

Queries the vertical position of the selected marker in the time overview.

**Syntax** :CALCulate:TOVerview:MARKer<x>:Y?

**Arguments** None

**Measurement Views** Time overview

**Examples** :CALCulate:TOVerview:MARKer1:Y?  
might return -34.28 indicating Marker 1 (M1) is at -34.28 dBm.

**Related Commands** :CALCulate:TOVerview:MARKer<x>:X

# CALibration Commands

The CALibration commands perform correction for the external gain/loss and probe.

## Command Tree

Header	Parameter
:CALibration	
:ABORt	
:AUTO	<boolean>
:CORRection	
:EXTeRnal	
:EDIT<x>	
:LABe1	<string>
:STATe	<boolean>
:GAIN	
[:MAGNitude]	<numeric_value>
:STATe	<boolean>
:PROBe	
:CONNect?	
[:MAGNitude]?	
:STATe	<boolean>

Where <x> = 1 to 3 indicates the following:

EDIT1: External Loss Table 1

EDIT2: External Loss Table 2

EDIT3: External Loss Table 3

**:CALibration:ABORt (No Query Form)**

Aborts any actions related to the alignments in progress.

**Syntax** :CALibration:ABORt

**Arguments** None

**Measurement Views** All

**Examples** :CALibration:ABORt  
aborts any actions related to the alignments in progress.

**:CALibration:AUTO(?)**

Selects or queries whether or not to run alignments automatically.

**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.

**Measurement Views** All

**Examples** :CALibration:AUTO ON  
runs alignments automatically as needed.

**Related Commands** \*CAL?

**:CALibration:CORRection:EXTErnal:EDIT<x>:LABel(?)**

Sets or queries the name of the external loss table.

**Syntax** :CALibration:CORRection:EXTErnal:EDIT<x>:LABel <name>  
:CALibration:CORRection:EXTErnal:EDIT<x>:LABel?

Where <x> = 1 to 3 represent the External Loss Table 1 to 3, respectively.

**Arguments** <name>::=<string> specifies the name of the external loss table.

**Measurement Views** All

**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.

**Syntax** :CALibration:CORRection:EXTErnal:EDIT<x>:STATe  
{ OFF | ON | 0 | 1 }  
:CALibration:CORRection:EXTErnal:EDIT<x>:STATe?

Where <x> = 1 to 3 represent the External Loss Table 1 to 3, respectively.

**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.

---

**Measurement Views** All

**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.

**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.

**Measurement Views** All

**Examples** :CALibration:CORRection:EXTeRnal:GAIN:MAGNitude -10  
sets the external gain to -10 dB.

**Related Commands** :CALibration:CORRection:EXTeRnal:GAIN:STATe

**:CALibration:CORRection:EXTeRnal:GAIN:STATe(?)**

Determines whether to enable or disable the external gain value.

**Syntax** :CALibration:CORRection:EXTeRnal:GAIN:STATe { OFF | ON | 0 | 1 }  
:CALibration:CORRection:EXTeRnal:GAIN:STATe?

**Arguments** OFF or 0 disables the external gain value.  
ON or 1 enables the external gain value.

**Measurement Views** All

**Examples** :CALibration:CORRection:EXTeRnal:GAIN:STATe ON  
enables the external gain value.

**Related Commands** :CALibration:CORRection:EXTeRnal:GAIN[:MAGNitude]



**:CALibration:CORRection:EXTeRnal:PROBe:CONNeCt? (Query Only)**

Queries whether the external probe is connected to the analyzer or not.

**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.

**Measurement Views** All

**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.

**Syntax** :CALibration:CORRection:EXTeRnal:PROBe[:MAGNitude]?

**Arguments** None

**Returns** <attenuation>::=<NRf>      The probe attenuation value in dB.

**Measurement Views** All

**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.

**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.

**Measurement Views** All

**Examples** :CALibration:CORRection:EXTernal:PROBe:STAtE ON  
corrects data for the external probe attenuation.

# DISPlay Commands

The DISPlay commands control the display of measurement data on the screen. These commands are divided by measurement views into the subgroups listed in Table 2-31.

**NOTE.** The :DISPlay commands concern the measurement display only, and do not affect analyzer hardware settings.

**Table 2-31: DISPlay command subgroups**

Command header	Measurement view	Refer to:
:DISPlay (basic)	All views, general window control	page 2-208
:DISPlay:ACPower	Channel power and ACPR	page 2-211
:DISPlay:AVTime	Amplitude versus Time	page 2-217
:DISPlay:CONSte	Constellation (Option 21 only)	page 2-228
:DISPlay:DDEMod	General purpose digital modulation measurements (Option 21 only)	page 2-229
:DISPlay:DPSA	DPX (Digital Phosphor) spectrum	page 2-239
:DISPlay:EVM	EVM versus Time (Option 21 only)	page 2-241
:DISPlay:FVTime	Frequency versus Time	page 2-243
:DISPlay:GENeral	General signal viewing	page 2-252
:DISPlay:GPRF	RF measurements	page 2-256
:DISPlay:IQVTime	RF I&Q versus Time	page 2-259
:DISPlay:MCPower	MCPR (Multiple Carrier Power Ratio)	page 2-268
:DISPlay:MERRor	Magnitude error versus Time (Option 21 only)	page 2-274
:DISPlay:OBWidth	OBW (Occupied Bandwidth)	page 2-276
:DISPlay:PERRor	Phase error versus Time (Option 21 only)	page 2-282
:DISPlay:PHVTime	Phase versus Time	page 2-284
:DISPlay:PNOise	Phase noise (Option 11 only)	page 2-294
:DISPlay:PULSe	Pulsed RF measurements (Option 20 only)	page 2-300
:DISPlay:SGRam	Spectrogram	page 2-329
:DISPlay:SPECTrum	Spectrum	page 2-333
:DISPlay:SPURious	Spurious	page 2-336
:DISPlay:TOVerview	Time overview	page 2-343

## :DISPlay Basic Command Subgroup

The :DISPlay basic commands control the measurement window display.

Command Tree	Header	Parameter
	:DISPlay	
	:WINDow	
	:ACTive	
	:MEASurement?	
	:OPTimized	
	:MEASurement?	

**:DISPlay:WINDow:ACTive:MEASurement? (Query Only)**

Queries the active measurement views.

**Syntax** :DISPlay:WINDow:ACTive:MEASurement?

**Arguments** None

**Returns** <view1>,<view2>,...,<view(n)>  
Where <view(n)>::=<string> is the view name as shown in Table 2-32.

**Table 2-32: Measurement views**

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	

**Measurement Views** All

**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: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

**Syntax** :DISPlay:WINDow:OPTimized:MEASurement?

**Arguments** None

**Returns** <view1>,<view2>,...,<view(n)>  
 Where <view(n)>::=<string> is the view name as shown in Table 2-32.

**Measurement Views** All

**Examples** :DISPlay:WINDow:OPTimized:MEASurement?  
 might return "SPEC", "MCP" indicating that the views of spectrum and MCPR are optimized.

**Related Commands** :DISPlay:DDEMod:MEASview:SElect, :DISPlay:GENeral:MEASview:SElect, :DISPlay:GPRF:MEASview:SElect, :DISPlay:PULSe:MEASview:SElect

## :DISPlay:ACPower Subgroup

The :DISPlay:ACPower commands control the Channel power and ACPR (Adjacent Channel Power Ratio) view.

Command Tree	Header	Parameter
	:DISPlay	
	:ACPower	
	:MARKer	
	:SHOW	
	:STATE	<boolean>
	:PLEVl	
	:SHOW	
	:STATE	<boolean>
	:RESet	
	:SCALe	
	:WINDow	
	:TRACe	
	:GRATicule	
	:GRID	
	:STATE	<boolean>
	:X	
	[:SCALe]	<numeric_value>
	:AUTO	
	:OFFSet	<numeric_value>
	:Y	
	[:SCALe]	<numeric_value>
	:AUTO	
	:OFFSet	<numeric_value>

**:DISPlay:ACPower:MARKer:SHOW:STATe(?)**

Determines whether to show or hide the readout for the selected marker in the Channel power and ACPR view.

**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.

**Measurement Views** Channel power and ACPR

**Examples** :DISPlay:ACPower:MARKer:SHOW:STATe ON  
shows the readout for the selected marker in the graph.

**:DISPlay:ACPower:PLEVe1:SHOW:STATe(?)**

Determines whether to show or hide the power levels in the Channel power and ACPR view.

**Syntax** :DISPlay:ACPower:PLEVe1:SHOW:STATe { OFF | ON | 0 | 1 }  
:DISPlay:ACPower:PLEVe1:SHOW:STATe?

**Arguments** OFF or 0 hides the power levels in the graph.  
ON or 1 shows the power levels in the graph.

**Measurement Views** Channel power and ACPR

**Examples** :DISPlay:ACPower:PLEVe1: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  
 Horizontal scale = Default span

**Syntax** :DISPlay:ACPower:RESet:SCALE

**Arguments** None

**Measurement Views** Channel power and ACPR

**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.

**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.

**Measurement Views** Channel power and ACPR

**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.

**Syntax**     :DISPlay:ACPower:X[:SCALE] <value>  
              :DISPlay:ACPower:X[:SCALE]?

**Arguments**   <value>::=<NRf> specifies the horizontal range.  
                  Range: 10 Hz to 6.2 GHz (RSA6106A) / 14 GHz (RSA6114A).

**Measurement Views**   Channel power and ACPR

**Examples**       :DISPlay:ACPower:X:SCALE 10MHz  
                  sets the horizontal range to 10 MHz.

**Related Commands**   :DISPlay:ACPower:X[:SCALE]:OFFSet

**: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.

**Syntax**       :DISPlay:ACPower:X[:SCALE]:AUTO

**Arguments**   None

**Measurement Views**   Channel power and ACPR

**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.

**Syntax** :DISPlay:ACPower:X[:SCALe]:OFFSet <value>  
:DISPlay:ACPower:X[:SCALe]:OFFSet?

**Arguments** <value>: :=<NRf> specifies the minimum horizontal value.  
Range: [(center frequency) - (X scale) × 0.9] to  
[(center frequency) + (X scale) × 0.9]

**Measurement Views** Channel power and ACPR

**Examples** :DISPlay:ACPower:X:SCALe:OFFSet 1.45GHz  
sets the minimum horizontal value to 1.45 GHz in the Channel power and ACPR graph.

**Related Commands** :DISPlay:ACPower:X[:SCALe]

**:DISPlay:ACPower:Y[:SCALe](?)**

Sets or queries the vertical range of the channel power and ACPR graph.

**Syntax** :DISPlay:ACPower:Y[:SCALe] <value>  
:DISPlay:ACPower:Y[:SCALe]?

**Arguments** <value>: :=<NRf> specifies the vertical range.  
Range: 0.1 to 200 dB.

**Measurement Views** Channel power and ACPR

**Examples** :DISPlay:ACPower:Y:SCALe 100  
sets the vertical range to 100 dB in the Channel power and ACPR graph.

**Related Commands** :DISPlay:ACPower:Y[:SCALe]:OFFSet

**: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.

**Syntax** :DISPlay:ACPower:Y[:SCALE]:AUTO

**Arguments** None

**Measurement Views** Channel power and ACPR

**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.

**Syntax** :DISPlay:ACPower:Y[:SCALE]:OFFSet <value>  
:DISPlay:ACPower:Y[:SCALE]:OFFSet?

**Arguments** <value>::=<NRf> specifies the vertical offset.  
Range: -170 to +50 dBm.

**Measurement Views** Channel power and ACPR

**Examples** :DISPlay:ACPower:Y:SCALE:OFFSet -12.5  
sets the vertical offset to -12.5 dBm in the Channel power and ACPR graph.

**Related Commands** :DISPlay:ACPower:Y[:SCALE]

## :DISPlay:AVTime Subgroup

The :DISPlay:AVTime commands control the Amplitude versus Time view.

Command Tree	Header	Parameter
	:DISPlay	
	:AVTime	
	:MARKer	
	:SHOW	
	:STATe	<boolean>
	:RESet	
	:TRIGger	
	:LEVe1	
	:STATe	<boolean>
	:WINDow	
	:TRACe	
	:GRATicule	
	:GRID	
	:STATe	<boolean>
	:X	
	:RSCale	
	[:SCALe]	
	:AUTO	
	:STATe	<boolean>
	:FULL	<numeric_value>
	:MAXimum?	
	:MINimum?	
	:OFFSet	<numeric_value>
	:MAXimum?	
	:MINimum?	
	:Y	
	:RSCale	
	[:SCALe]	
	:FULL	<numeric_value>
	:OFFSet	<numeric_value>

**: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.

**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.

**Measurement Views**   Amplitude versus Time

**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  
Horizontal scale = Analysis length

**Syntax**     :DISPlay:AVTime:RESet

**Arguments**   None

**Measurement Views**   Amplitude versus Time

**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.

**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.

**Measurement Views** Amplitude versus Time

**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.

**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.

**Measurement Views** Amplitude versus Time

**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.

**Syntax**       :DISPlay:AVTime:X:RSCale

**Arguments**   None

**Measurement Views**   Amplitude versus Time

**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.

**Syntax** :DISPlay:AVTime:X[:SCALe]:AUTO

**Arguments** None

**Measurement Views** Amplitude versus Time

**Examples** :DISPlay:AVTime:X:SCALe:AUTO  
sets the horizontal scale automatically to fit the waveform to the screen.

**Related Commands** :DISPlay:AVTime:X[:SCALe]:AUTO:STATe

**:DISPlay:AVTime:X[:SCALe]:AUTO:STATe(?)**

Determines whether to set the horizontal scale automatically or manually.

**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.

**Measurement Views** Amplitude versus Time

**Examples** :DISPlay:AVTime:X:SCALe:AUTO:STATe ON  
specifies that the horizontal scale is set automatically.

**Related Commands** :DISPlay:AVTime:X[:SCALe]:FULL, :DISPlay:AVTime:X[:SCALe]:OFFSet

**: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.

**Syntax**       :DISPlay:AVTime:X[:SCALe]:FULL <value>  
                  :DISPlay:AVTime:X[:SCALe]:FULL?

**Arguments**    <value>::=<NRf> specifies the horizontal scale in full-scale time.  
                  Use the :DISPlay:AVTime:X[:SCALe]:MAXimum|:MINimum? query to get the upper and lower limits of the setting range.

**Measurement Views**    Amplitude versus Time

**Examples**        :DISPlay:AVTime:X:SCALe:FULL 25.6us  
                  sets the horizontal scale to 25.6  $\mu$ s.

**Related Commands**    :DISPlay:AVTime:X[:SCALe]:AUTO:STATe,  
                          :DISPlay:AVTime:X[:SCALe]:OFFSet,  
                          :DISPlay:AVTime:X[:SCALe]:MAXimum|:MINimum

**:DISPlay:AVTime:X[:SCALe]:MAXimum? (Query Only)**

Queries the upper limit of the horizontal scale setting range.

**Syntax** :DISPlay:AVTime:X[:SCALe]:MAXimum?

**Returns** <NRf> The upper limit of the horizontal scale setting range.

**Measurement Views** Amplitude versus Time

**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.

**Related Commands** :DISPlay:AVTime:X[:SCALe]:FULL

**:DISPlay:AVTime:X[:SCALe]:MINimum? (Query Only)**

Queries the lower limit of the horizontal scale setting range.

**Syntax** :DISPlay:AVTime:X[:SCALe]:MINimum?

**Returns** <NRf> The lower limit of the horizontal scale setting range.

**Measurement Views** Amplitude versus Time

**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.

**Related Commands** :DISPlay:AVTime:X[:SCALe]:FULL

**: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.

**Syntax** :DISPlay:AVTime:X[:SCALe]:OFFSet <value>  
:DISPlay:AVTime:X[:SCALe]:OFFSet?

**Arguments** <value>::=<NRf> specifies the minimum horizontal value.  
Use the :DISPlay:AVTime:X[:SCALe]:OFFSet:MAXimum|:MINimum? query to get the upper and lower limits of the setting range.

**Measurement Views** Amplitude versus Time

**Examples** :DISPlay:AVTime:X:SCALe:OFFSet 800ns  
sets the minimum horizontal value to 800 ns in the Amplitude versus Time graph.

**Related Commands** :DISPlay:AVTime:X[:SCALe]:AUTO:STATe,  
:DISPlay:AVTime:X[:SCALe]:FULL,  
:DISPlay:AVTime:X[:SCALe]:OFFSet:MAXimum|:MINimum?

**:DISPlay:AVTime:X[:SCALE]:OFFSet:MAXimum? (Query Only)**

Queries the upper limit of the horizontal offset setting range.

**Syntax** :DISPlay:AVTime:X[:SCALE]:OFFSet:MAXimum?

**Returns** <NRf> The upper limit of the horizontal offset setting range.

**Measurement Views** Amplitude versus Time

**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.

**Related Commands** :DISPlay:AVTime:X[:SCALE]:OFFSet

**:DISPlay:AVTime:X[:SCALE]:OFFSet:MINimum? (Query Only)**

Queries the lower limit of the horizontal offset setting range.

**Syntax** :DISPlay:AVTime:X[:SCALE]:OFFSet:MINimum?

**Returns** <NRf> The lower limit of the horizontal offset setting range.

**Measurement Views** Amplitude versus Time

**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.

**Related Commands** :DISPlay:AVTime:X[:SCALE]:OFFSet

## **: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.

**Syntax**     :DISPlay:AVTime:Y:RSCale

**Arguments**   None

**Measurement Views**   Amplitude versus Time

**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.

**Syntax**     :DISPlay:AVTime:Y[:SCALE]:FULL <value>  
               :DISPlay:AVTime:Y[:SCALE]:FULL?

**Arguments**   <value>::=<NRf> specifies the vertical range.  
 Range: 0.1 to 200 dB.

**Measurement Views**   Amplitude versus Time

**Examples**       :DISPlay:AVTime:Y:SCALE:FULL 100  
 sets the vertical range to 100 dB in the Amplitude versus Time graph.

**Related Commands**   :DISPlay:AVTime:Y[:SCALE]:OFFSet

**: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.

**Syntax**       :DISPlay:AVTime:Y[:SCALE]:OFFSet <value>  
               :DISPlay:AVTime:Y[:SCALE]:OFFSet?

**Arguments**   <value>::=<NRf> specifies the vertical offset.  
 Range: -170 to +50 dBm.

**Measurement Views**   Amplitude versus Time

**Examples**       :DISPlay:AVTime:Y:SCALE:OFFSet -12.5  
 sets the vertical offset to -12.5 dBm in the Amplitude versus Time graph.

**Related Commands**   :DISPlay:AVTime:Y[:SCALE]:FULL

## :DISPlay:CONSte Subgroup

The :DISPlay:CONSte commands control the constellation display.

Command Tree	Header	Parameter
	:DISPlay	
	:CONSte	
	:WINDow	
	:TRACe	
	:GRATicule	
	:GRID	
	:STATe	<boolean>

## :DISPlay:CONSte:WINDow:TRACe:GRATicule:GRID:STATe(?)

Determines whether to show or hide the graticule grid on the screen.

**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.

**Measurement Views**   Constellation

**Examples**    :DISPlay:CONSte:WINDow:TRACe:GRATicule:GRID:STATe ON  
 shows the graticule grid on the screen.



## :DISPlay:DDEMod Subgroup (Option 21 Only)

The :DISPlay:DDEMod commands control display in the General Purpose Digital Modulation measurements.

Command Tree	Header	Parameter
	:DISPlay	
	:DDEMod	
	:MEASview	
	:DELeTe	CONStE   EVM   MERRor   PERRor   SIGNa1qua1   STABle
	:NEW	CONStE   EVM   MERRor   PERRor   SIGNa1qua1   STABle
	:SELeCt	CONStE   EVM   MERRor   PERRor   SIGNa1qua1   STABle
	:RADix	BINary   HEXadecima1
	:X	
	[:SCALe]	<numeric_value>
	:AUTO	
	:SCALe	<boolean>
	:MAXimum?	
	:MINimum?	
	:OFFSet	<numeric_value>
	:MAXimum?	
	:MINimum?	
	:RESet	

## **:DISPlay:DDEMod:MEASview:DELeTe (No Query Form)**

Deletes the measurement view in the general purpose digital modulation measurements.

**Syntax**     :DISPlay:DDEMod:MEASview:DELeTe  
                   { CONStE | EVM | MERRor | PERRor | SIGNalqual | STABle }

**Arguments**   Table 2-33 lists the arguments. The arguments are the string type.

**Table 2-33: Modulation measurement views**

<b>Argument</b>	<b>View</b>
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.

**Measurement Views**   General purpose digital modulation

**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.

**Syntax**     :DISPlay:DDEMod:MEASview:NEW  
                  { CONSTe | EVM | MERRor | PERRor | SIGNa1qua1 | STAB1e }

**Arguments**   Refer to Table 2-33 on page 2-230.

**Measurement Views**   General purpose digital modulation

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.

**Syntax**     :DISPlay:DDEMod:MEASview:SElect  
                  { CONSTe | EVM | MERRor | PERRor | SIGNa1qua1 | STAB1e }

**Arguments**   Refer to Table 2-33 on page 2-230.

If you attempt to select a view that is not displayed on screen, the error (-200, “Execution error; Measurement not running”) will be returned.

**Measurement Views**   General purpose digital modulation

**Examples**     :DISPlay:DDEMod:MEASview:SElect CONSTe  
                  selects the constellation view.

**Related Commands**   :DISPlay:WINDow:OPTimized:MEASurement?

## **:DISPlay:DDEMod:RADix(?)**

Selects or queries the base of symbols.  
This command is effective in the symbol table.

**Syntax**     :DISPlay:DDEMod:RADix { BINary | HEXadecimal }  
              :DISPlay:DDEMod:RADix?

**Arguments**   BINary selects binary notation.  
              HEXadecimal selects hexadecimal notation.

**Measurement Views**   Symbol table

**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.

**Syntax** :DISPlay:DDEMod:X[:SCALe] <value>  
:DISPlay:DDEMod:X[:SCALe]?

**Arguments** <value>::=<NRf> specifies the horizontal scale in full-scale time.

Use the :DISPlay:DDEMod:X[:SCALe]:MAXimum|:MINimum? query to get the upper and lower limits of the setting range.

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

**Measurement Views** EVM versus Time, Magnitude error versus Time, Phase error versus Time

**Examples** :DISPlay:DDEMod:X:SCALe 1.5us  
sets the horizontal scale to 1.5  $\mu$ s.

**Related Commands** :DISPlay:DDEMod:X[:SCALe]:AUTO:STATe,  
:DISPlay:DDEMod:X[:SCALe]:OFFSet,  
:DISPlay:DDEMod:X[:SCALe]:MAXimum|:MINimum,  
[:SENSe]:DDEMod:TIME:UNITs

**: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.

**Syntax** :DISPlay:DDEMod:X[:SCALe]:AUTO

**Arguments** None

**Measurement Views** EVM versus Time, Magnitude error versus Time, Phase error versus Time

**Examples** :DISPlay:DDEMod:X:SCALe:AUTO  
sets the horizontal scale automatically to fit the waveform to the screen.

**Related Commands** :DISPlay:DDEMod:X[:SCALe]:AUTO:STATe

**:DISPlay:DDEMod:X[:SCALe]:AUTO:STATe(?)**

Determines whether to set the horizontal scale automatically or manually.

**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.

**Measurement Views** EVM versus Time, Magnitude error versus Time, Phase error versus Time

**Examples** :DISPlay:DDEMod:X:SCALe:AUTO:STATe ON  
specifies that the horizontal scale is set automatically.

**Related Commands** :DISPlay:DDEMod:X[:SCALe], :DISPlay:DDEMod:X[:SCALe]:OFFSet

**:DISPlay:DDEMod:X[:SCALE]:MAXimum? (Query Only)**

Queries the upper limit of the horizontal scale setting range.

**Syntax** :DISPlay:DDEMod:X[:SCALE]:MAXimum?

**Returns** <Nrf> The upper limit of the horizontal scale setting range.  
The unit can be changed by the [:SENSe]:DDEMod:TIME:UNITs command.

**Measurement Views** EVM versus Time, Magnitude error versus Time, Phase error versus Time

**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.

**Related Commands** :DISPlay:DDEMod:X[:SCALE], [:SENSe]:DDEMod:TIME:UNITs

**:DISPlay:DDEMod:X[:SCALE]:MINimum? (Query Only)**

Queries the lower limit of the horizontal scale setting range.

**Syntax** :DISPlay:DDEMod:X[:SCALE]:MINimum?

**Returns** <Nrf> The lower limit of the horizontal scale setting range.  
The unit can be changed by the [:SENSe]:DDEMod:TIME:UNITs command.

**Measurement Views** EVM versus Time, Magnitude error versus Time, Phase error versus Time

**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.

**Related Commands** :DISPlay:DDEMod:X[:SCALE], [:SENSe]:DDEMod:TIME:UNITs

**: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.

**Syntax** :DISPlay:DDEMod:X[:SCALe]:OFFSet <value>  
:DISPlay:DDEMod:X[:SCALe]:OFFSet?

**Arguments** <value>::=<NRf> specifies the minimum horizontal value.

Use the :DISPlay:DDEMod:X[:SCALe]:OFFSet:MAXimum|:MINimum? query to get the upper and lower limits of the setting range.

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

**Measurement Views** EVM versus Time, Magnitude error versus Time, Phase error versus Time

**Examples** :DISPlay:DDEMod:X:SCALe:OFFSet 20.075us  
sets the minimum horizontal value to 20.075  $\mu$ s.

**Related Commands** :DISPlay:DDEMod:X[:SCALe]:AUTO:STATe,  
:DISPlay:DDEMod:X[:SCALe],  
:DISPlay:DDEMod:X[:SCALe]:OFFSet:MAXimum|:MINimum?,  
[:SENSe]:DDEMod:TIME:UNITs



**:DISPlay:DDEMod:X[:SCALE]:OFFSet:MAXimum? (Query Only)**

Queries the upper limit of the horizontal offset setting range.

**Syntax** :DISPlay:DDEMod:X[:SCALE]:OFFSet:MAXimum?

**Returns** <NRf> The upper limit of the horizontal offset setting range.  
The unit can be changed by the [:SENSe]:DDEMod:TIME:UNITs command.

**Measurement Views** EVM versus Time, Magnitude error versus Time, Phase error versus Time

**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.

**Related Commands** :DISPlay:DDEMod:X[:SCALE]:OFFSet, [:SENSe]:DDEMod:TIME:UNITs

**:DISPlay:DDEMod:X[:SCALE]:OFFSet:MINimum? (Query Only)**

Queries the lower limit of the horizontal offset setting range.

**Syntax** :DISPlay:DDEMod:X[:SCALE]:OFFSet:MINimum?

**Returns** <NRf> The lower limit of the horizontal offset setting range.  
The unit can be changed by the [:SENSe]:DDEMod:TIME:UNITs command.

**Measurement Views** EVM versus Time, Magnitude error versus Time, Phase error versus Time

**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.

**Related Commands** :DISPlay:DDEMod:X[:SCALE]:OFFSet, [:SENSe]:DDEMod:TIME:UNITs

## **: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.

**Syntax**     :DISPlay:DDEMod:X[:SCALE]:RESet

**Arguments**   None

**Measurement Views**   EVM versus Time, Magnitude error versus Time, Phase error versus Time

**Examples**     :DISPlay:DDEMod:X:SCALE:RESet  
presets the horizontal scale to the default value.

## :DISPlay:DPSA Subgroup

The :DISPlay:DPSA commands control the DPX (Digital Phosphor) spectrum view.

Command Tree	Header	Parameter
	:DISPlay	
	:DPSA	
	:WINDow	
	:TRACe	
	:GRATicule	
	:GRID	
	:STATe	<boolean>
	:Y	
	[:SCALe]	
	:PDIVision	<numeric_value>

### :DISPlay:DPSA:WINDow:TRACe:GRATicule:GRID:STATe(?)

Determines whether to show or hide the graticule grid on the screen.

**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.

**Measurement Views** DPX spectrum

**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.

**Syntax**     :DISPlay:DPSA:Y[:SCALe]:PDIVision <value>  
              :DISPlay:DPSA:Y[:SCALe]:PDIVision?

**Arguments**   <value>::=<NRf> specifies the vertical scale (per division).  
                  Range: 2 to 20 dB/div.

**Measurement Views**   DPX spectrum

**Examples**       :SENSe:DPSA:Y:SCALe:PDIVision 0.5  
                  sets the vertical scale to 0.5 dB/div.

**Related Commands**   [:SENSe]:POWer:UNITs

## :DISPlay:EVM Subgroup (Option 21 Only)

The :DISPlay:EVM commands control the EVM (Error Vector Magnitude) versus Time view.

Command Tree	Header	Parameter
	:DISPlay	
	:EVM	
	:Y	
	[:SCALe]	<numeric_value>
	:AUTO	
	:OFFSet	<numeric_value>

## :DISPlay:EVM:Y[:SCALe](?)

Sets or queries the vertical range of the EVM versus Time graph.

**Syntax** :DISPlay:EVM:Y[:SCALe] <value>  
:DISPlay:EVM:Y[:SCALe]?

**Arguments** <value>::=<NRf> specifies the vertical range.  
Range: 1 to 100%.

**Measurement Views** EVM versus Time

**Examples** :DISPlay:EVM:Y:SCALe 50  
sets the vertical range to 50% in the EVM versus Time graph.

**Related Commands** :DISPlay:EVM:Y[:SCALe]:OFFSet

**: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.

**Syntax** :DISPlay:EVM:Y[:SCALe]:AUTO

**Arguments** None

**Measurement Views** EVM versus Time

**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.

**Syntax** :DISPlay:EVM:Y[:SCALe]:OFFSet <value>  
:DISPlay:EVM:Y[:SCALe]:OFFSet?

**Arguments** <value>::=<NRf> specifies the minimum vertical value.  
Range: -100 to 100%.

**Measurement Views** EVM versus Time

**Examples** :DISPlay:EVM:Y:SCALe:OFFSet -9.5  
sets the minimum vertical value to -9.5% in the EVM versus Time graph.

**Related Commands** :DISPlay:EVM:Y[:SCALe]

## :DISPlay:FVTime Subgroup

The :DISPlay:FVTime commands control the Frequency versus Time view.

Command Tree	Header	Parameter
	:DISPlay	
	:FVTime	
	:WINDow	
	:TRACe	
	:GRATicule	
	:GRID	
	:STATe	<boolean>
	:X	
	[:SCALe]	<numeric_value>
	:AUTO	
	:STATe	<boolean>
	:MAXimum?	
	:MINimum?	
	:OFFSet	<numeric_value>
	:MAXimum?	
	:MINimum?	
	:Y	
	[:SCALe]	<numeric_value>
	:AUTO	
	:OFFSet	<numeric_value>

## **:DISPlay:FVTime:WINDow:TRACe:GRATicule:GRID:STATe(?)**

Determines whether to show or hide the graticule grid on the screen.

**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.

**Measurement Views**   Frequency versus Time

**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.

**Syntax**     :DISPlay:FVTime:X[:SCALe] <value>  
              :DISPlay:FVTime:X[:SCALe]?

**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|MINimum? query to get the upper and lower limit values of the setting range.

**Measurement Views**   Frequency versus Time

**Examples**       :DISPlay:FVTime:X:SCALe 25.6us  
                  sets the horizontal scale to 25.6  $\mu$ s.

**Related Commands**   :DISPlay:FVTime:X[:SCALe]:AUTO:STATe,  
                      :DISPlay:FVTime:X[:SCALe]:OFFSet,  
                      :DISPlay:FVTime:X[:SCALe]:MAXimum|MINimum

**: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.

**Syntax** :DISPlay:FVTime:X[:SCALe]:AUTO

**Arguments** None

**Measurement Views** Frequency versus Time

**Examples** :DISPlay:FVTime:X:SCALe:AUTO  
sets the horizontal scale automatically to fit the waveform to the screen.

**Related Commands** :DISPlay:FVTime:X[:SCALe]:AUTO:STATe

**:DISPlay:FVTime:X[:SCALe]:AUTO:STATe(?)**

Determines whether to set the horizontal scale automatically or manually.

**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.

**Measurement Views** Frequency versus Time

**Examples** :DISPlay:FVTime:X:SCALe:AUTO:STATe ON  
specifies that the horizontal scale is set automatically.

**Related Commands** :DISPlay:FVTime:X[:SCALe], :DISPlay:FVTime:X[:SCALe]:OFFSet

**:DISPlay:FVTime:X[:SCALe]:MAXimum? (Query Only)**

Queries the upper limit of the horizontal scale setting range.

**Syntax** :DISPlay:FVTime:X[:SCALe]:MAXimum?

**Returns** <NRf> The upper limit of the horizontal scale setting range.

**Measurement Views** Frequency versus Time

**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.

**Related Commands** :DISPlay:FVTime:X[:SCALe]

**:DISPlay:FVTime:X[:SCALe]:MINimum? (Query Only)**

Queries the lower limit of the horizontal scale setting range.

**Syntax** :DISPlay:FVTime:X[:SCALe]:MINimum?

**Returns** <NRf> The lower limit of the horizontal scale setting range.

**Measurement Views** Frequency versus Time

**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.

**Related Commands** :DISPlay:FVTime:X[:SCALe]

**: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.

**Syntax** :DISPlay:FVTime:X[:SCALe]:OFFSet <value>  
:DISPlay:FVTime:X[:SCALe]:OFFSet?

**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|:MINimum? query to get the upper and lower limit values of the setting range.

**Measurement Views** Frequency versus Time

**Examples** :DISPlay:FVTime:X:SCALe:OFFSet 800ns  
sets the minimum horizontal value to 800 ns in the Frequency versus Time graph.

**Related Commands** :DISPlay:FVTime:X[:SCALe]:AUTO:STATe,  
:DISPlay:FVTime:X[:SCALe],  
:DISPlay:FVTime:X[:SCALe]:OFFSet:MAXimum|:MINimum?

**:DISPlay:FVTime:X[:SCALe]:OFFSet:MAXimum? (Query Only)**

Queries the upper limit of the horizontal offset setting range.

**Syntax** :DISPlay:FVTime:X[:SCALe]:OFFSet:MAXimum?

**Returns** <NRf> The upper limit of the horizontal offset setting range.

**Measurement Views** Frequency versus Time

**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.

**Related Commands** :DISPlay:FVTime:X[:SCALe]:OFFSet

**:DISPlay:FVTime:X[:SCALe]:OFFSet:MINimum? (Query Only)**

Queries the lower limit of the horizontal offset setting range.

**Syntax** :DISPlay:FVTime:X[:SCALe]:OFFSet:MINimum?

**Returns** <NRf> The lower limit of the horizontal offset setting range.

**Measurement Views** Frequency versus Time

**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.

**Related Commands** :DISPlay:FVTime:X[:SCALe]:OFFSet

**:DISPlay:FVTime:Y[:SCALe](?)**

Sets or queries the vertical range of the Frequency versus Time graph.

**Syntax**     :DISPlay:FVTime:Y[:SCALe] <value>  
              :DISPlay:FVTime:Y[:SCALe]?

**Arguments**   <value>::=<NRf> specifies the vertical range.  
                  Range: 10 Hz to 120 MHz.

**Measurement Views**   Frequency versus Time

**Examples**       :DISPlay:FVTime:Y:SCALe 30MHz  
                  sets the vertical range to 30 MHz in the Frequency versus Time graph.

**Related Commands**   :DISPlay:FVTime:Y[:SCALe]:OFFSet

**: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.

**Syntax**       :DISPlay:FVTime:Y[:SCALe]:AUTO

**Arguments**   None

**Measurement Views**   Frequency versus Time

**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.

**Syntax** :DISPlay:FVTime:Y[:SCALe]:OFFSet <value>

:DISPlay:FVTime:Y[:SCALe]:OFFSet?

**Arguments** <value>::=<NRf> specifies the vertical offset.  
Range: -60 MHz to +60 MHz.

**Measurement Views** Frequency versus Time

**Examples** :DISPlay:FVTime:Y:SCALe:OFFSet -14.5MHz  
sets the vertical offset to -14.5 MHz in the Frequency versus Time graph.

**Related Commands** :DISPlay:FVTime:Y[:SCALe]

## :DISPlay:GENeral Subgroup

The :DISPlay:GENeral commands control display in the general signal viewing.

Command Tree	Header	Parameter
	:DISPlay	
	:GENeral	
	:MEASview	
	:DElete	SPECtrum   DPSA   AVTime   FVTime   PHVTime   IQVTime   SGRam   TOVerview
	:NEW	SPECtrum   DPSA   AVTime   FVTime   PHVTime   IQVTime   SGRam   TOVerview
	:SElect	SPECtrum   DPSA   AVTime   FVTime   PHVTime   IQVTime   SGRam   TOVerview



**:DISPlay:GENeral:MEASview:DELeTe (No Query Form)**

Deletes a measurement view in the general signal viewing.

**Syntax** :DISPlay:GENeral:MEASview:DELeTe { SPECTrum | DPSA | AVTime  
| FVTime | PHVTime | IQVTime | SGRam | TOVerview }

**Arguments** Table 2-34 shows the arguments. The arguments are the string type.

**Table 2-34: 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.

**Measurement Views** General signal viewing

**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.

**Syntax**     :DISPlay:GENeral:MEASview:NEW { SPECTrum | DPSA | AVTime  
                  | FVTime | PHVTime | IQVTime | SGRam | TOVerview }

**Arguments**   Refer to Table 2-34 on page 2-253.

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.

**Measurement Views**   General signal viewing

**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.

**Syntax**     :DISPlay:GENeral:MEASview:SElect { SPECTrum | DPSA | AVTime  
                  | FVTime | PHVTime | IQVTime | SGRam | TOVerview }

**Arguments**   Refer to Table 2-34 on page 2-253.

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.

**Measurement Views**   General signal viewing

**Examples**           :DISPlay:GENeral:MEASview:SElect DPSA  
                  selects the DPX spectrum view.

**Related Commands**   :DISPlay:WINDow:OPTimized:MEASurement?

## :DISPlay:GPRF Subgroup

The :DISPlay:GPRF commands control display for the RF measurements.

Command Tree	Header	Parameter
	:DISPlay	
	:GPRF	
	:MEASview	
	:DELeTe	CCDF   ACPower   MCPower   OBW   PNOise   SPURious
	:NEW	CCDF   ACPower   MCPower   OBW   PNOise   SPURious
	:SELeCt	CCDF   ACPower   MCPower   OBW   PNOise   SPURious

### :DISPlay:GPRF:MEASview:DELeTe (No Query Form)

Deletes a selected measurement view in the RF measurements.

**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.

**Measurement Views** RF measurements

**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.

**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.

**Measurement Views**   RF measurements

**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.

**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.

**Measurement Views**   RF measurements

**Examples**     :DISPlay:GPRF:MEASview:SElect ACPower  
                  selects the Channel power and ACPR view.

**Related Commands**   :DISPlay:WINDow:OPTimized:MEASurement?

## :DISPlay:IQVTime Subgroup

The :DISPlay:IQVTime commands control the RF I&Q versus Time view.

Command Tree	Header	Parameter
	:DISPlay	
	:IQVTime	
	:WINDow	
	:TRACe	
	:GRATicule	
	:GRID	
	:STATe	<boolean>
	:X	
	[:SCALe]	<numeric_value>
	:AUTO	
	:STATe	<boolean>
	:MAXimum?	
	:MINimum?	
	:OFFSet	<numeric_value>
	:MAXimum?	
	:MINimum?	
	:Y	
	[:SCALe]	<numeric_value>
	:AUTO	
	:OFFSet	<numeric_value>
	:RESCale	

## **:DISPlay:IQVTime:WINDow:TRACe:GRATicule:GRID:STATe(?)**

Determines whether to show or hide the graticule grid on the screen.

**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.

**Measurement Views**   RF I&Q versus Time

**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.

**Syntax** :DISPlay:IQVTime:X[:SCALe] <value>  
:DISPlay:IQVTime:X[:SCALe]?

**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|:MINimum? query to get the upper and lower limit values of the setting range.

**Measurement Views** RF I&Q versus Time

**Examples** :DISPlay:IQVTime:X:SCALe 100us  
sets the horizontal scale to 100  $\mu$ s.

**Related Commands** :DISPlay:IQVTime:X[:SCALe]:AUTO:STATe,  
:DISPlay:IQVTime:X[:SCALe]:OFFSet,  
:DISPlay:IQVTime:X[:SCALe]:MAXimum|:MINimum

**: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.

**Syntax** :DISPlay:IQVTime:X[:SCALE]:AUTO

**Arguments** None

**Measurement Views** RF I&Q versus Time

**Examples** :DISPlay:IQVTime:X:SCALE:AUTO  
sets the horizontal scale automatically to fit the waveform to the screen.

**Related Commands** :DISPlay:IQVTime:X[:SCALE]:, :AUTO:STATe

**:DISPlay:IQVTime:X[:SCALE]:AUTO:STATe(?)**

Determines whether to set the horizontal scale automatically or manually.

**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.

**Measurement Views** RF I&Q versus Time

**Examples** :DISPlay:IQVTime:X:SCALE:AUTO:STATe ON  
specifies that the horizontal scale is set automatically.

**Related Commands** :DISPlay:IQVTime:X[:SCALE], :DISPlay:IQVTime:X[:SCALE]:OFFSet

**:DISPlay:IQVTime:X[:SCALe]:MAXimum? (Query Only)**

Queries the upper limit of the horizontal scale setting range.

**Syntax** :DISPlay:IQVTime:X[:SCALe]:MAXimum?

**Returns** <NRf> The upper limit of the horizontal scale setting range.

**Measurement Views** RF I&Q versus Time

**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.

**Related Commands** :DISPlay:IQVTime:X[:SCALe]

**:DISPlay:IQVTime:X[:SCALe]:MINimum? (Query Only)**

Queries the lower limit of the horizontal scale setting range.

**Syntax** :DISPlay:IQVTime:X[:SCALe]:MINimum?

**Returns** <NRf> The lower limit of the horizontal scale setting range.

**Measurement Views** RF I&Q versus Time

**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.

**Related Commands** :DISPlay:IQVTime:X[:SCALe]

**: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.

**Syntax** :DISPlay:IQVTime:X[:SCALe]:OFFSet <value>  
:DISPlay:IQVTime:X[:SCALe]:OFFSet?

**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|:MINimum? query to get the upper and lower limit values of the setting range.

**Measurement Views** RF I&Q versus Time

**Examples** :DISPlay:IQVTime:X:SCALe:OFFSet 800ns  
sets the minimum horizontal value to 800 ns in the RF I&Q versus Time graph.

**Related Commands** :DISPlay:IQVTime:X[:SCALe]:AUTO:STATe,  
:DISPlay:IQVTime:X[:SCALe],  
:DISPlay:IQVTime:X[:SCALe]:OFFSet:MAXimum|:MINimum?

**:DISPlay:IQVTime:X[:SCALe]:OFFSet:MAXimum? (Query Only)**

Queries the upper limit of the horizontal offset setting range.

**Syntax** :DISPlay:IQVTime:X[:SCALe]:OFFSet:MAXimum?

**Returns** <NRf> The upper limit of the horizontal offset setting range.

**Measurement Views** RF I&Q versus Time

**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.

**Related Commands** :DISPlay:IQVTime:X[:SCALe]:OFFSet

**:DISPlay:IQVTime:X[:SCALe]:OFFSet:MINimum? (Query Only)**

Queries the lower limit of the horizontal offset setting range.

**Syntax** :DISPlay:IQVTime:X[:SCALe]:OFFSet:MINimum?

**Returns** <NRf> The lower limit of the horizontal offset setting range.

**Measurement Views** RF I&Q versus Time

**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.

**Related Commands** :DISPlay:IQVTime:X[:SCALe]:OFFSet

**:DISPlay:IQVTime:Y[:SCALe](?)**

Sets or queries the vertical range of the RF I&Q versus Time graph.

**Syntax**     :DISPlay:IQVTime:Y[:SCALe] <value>  
              :DISPlay:IQVTime:Y[:SCALe]?

**Arguments**   <value>::=<NRf> specifies the vertical range.  
                  Range: 1 $\mu$  to 10 V.

**Measurement Views**   RF I&Q versus Time

**Examples**       :DISPlay:IQVTime:Y:SCALe 1.5  
                  sets the vertical range to 1.5 V in the RF I&Q versus Time graph.

**Related Commands**   :DISPlay:IQVTime:Y[:SCALe]:OFFSet

**: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.

**Syntax**       :DISPlay:IQVTime:Y[:SCALe]:AUTO

**Arguments**   None

**Measurement Views**   RF I&Q versus Time

**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.

**Syntax** :DISPlay:IQVTime:Y[:SCALe]:OFFSet <value>

:DISPlay:IQVTime:Y[:SCALe]:OFFSet

**Arguments** <value>::=<NRf> specifies the vertical offset.  
Range: -5 to +5 V.

**Measurement Views** RF I&Q versus Time

**Examples** :DISPlay:IQVTime:Y:SCALe:OFFSet -82.75mV  
sets the vertical offset to -82.75 mV in the RF I&Q versus Time graph.

**Related Commands** :DISPlay:IQVTime:Y[:SCALe]

**:DISPlay:IQVTime:Y[:SCALe]:RESCale (No Query Form)**

Rescales the vertical axis automatically to fit the waveform to the screen.

**Syntax** :DISPlay:IQVTime:Y[:SCALe]:RESCale

**Arguments** None

**Measurement Views** RF I&Q versus Time

**Examples** :DISPlay:IQVTime:Y:SCALe:RESCale  
rescales the vertical axis automatically to fit the waveform to the screen.

## :DISPlay:MCPower Subgroup

The :DISPlay:MCPower commands control the MCPR (Multiple Carrier Power Ratio) view.

Command Tree	Header	Parameter
	:DISPlay	
	:MCPower	
	:MARKer	
	:SHOW	
	:STATe	<boolean>
	:PLEVel	
	:SHOW	
	:STATe	<boolean>
	:RESet	
	:SCALE	
	:WINDow	
	:TRACe	
	:GRATICule	
	:GRID	
	:STATe	<boolean>
	:X	
	[:SCALE]	<numeric_value>
	:AUTO	
	:OFFSet	<numeric_value>
	:Y	
	[:SCALE]	<numeric_value>
	:AUTO	
	:OFFSet	<numeric_value>



## **:DISPlay:MCPower:MARKer:SHOW:STATe(?)**

Determines whether to show or hide the readout for the selected marker in the MCPR view.

**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.

**Measurement Views**   MCPR

**Examples**     :DISPlay:MCPower:MARKer:SHOW:STATe ON  
              shows the readout for the selected marker in the graph.

## **:DISPlay:MCPower:PLEVe1:SHOW:STATe(?)**

Determines whether to show or hide the power levels in the MCPR view.

**Syntax**     :DISPlay:MCPower:PLEVe1:SHOW:STATe { OFF | ON | 0 | 1 }  
              :DISPlay:MCPower:PLEVe1:SHOW:STATe?

**Arguments**   OFF or 0 hides the power levels in the graph.  
              ON or 1 shows the power levels in the graph.

**Measurement Views**   MCPR

**Examples**     :DISPlay:MCPower:PLEVe1: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  
Horizontal scale = Default span

**Syntax** :DISPlay:MCPower:RESet:SCALE

**Arguments** None

**Measurement Views** MCPR

**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.

**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.

**Measurement Views** MCPR

**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.

**Syntax**     :DISPlay:MCPower:X[:SCALE] <value>  
              :DISPlay:MCPower:X[:SCALE]?

**Arguments**   <value>::=<NRf> specifies the horizontal range.  
                  Range: 10 Hz to 6.2 GHz (RSA6106A) / 14 GHz (RSA6114A).

**Measurement Views**   MCPR

**Examples**       :DISPlay:MCPower:X:SCALE 10MHz  
                  sets the horizontal range to 10 MHz.

**Related Commands**   :DISPlay:MCPower:X[:SCALE]:OFFSet

**:DISPlay:MCPower:X[:SCALE]:AUTO (No Query Form)**

Rescales the horizontal axis automatically to fit the waveform to the screen in the MCPR view.

**Syntax**       :DISPlay:MCPower:X[:SCALE]:AUTO

**Arguments**   None

**Measurement Views**   MCPR

**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.

**Syntax**     :DISPlay:MCPower:X[:SCALe]:OFFSet <value>  
               :DISPlay:MCPower:X[:SCALe]:OFFSet?

**Arguments**   <value>::=<NRf> specifies the minimum horizontal value.  
 Range: [(center frequency) - (X scale) × 0.9] to  
           [(center frequency) + (X scale) × 0.9]

**Measurement Views**   MCPR

**Examples**       :DISPlay:MCPower:X:SCALe:OFFSet 1.45GHz  
 sets the minimum horizontal value to 1.45 GHz in the MCPR graph.

**Related Commands**   :DISPlay:MCPower:X[:SCALe]

## **:DISPlay:MCPower:Y[:SCALe](?)**

Sets or queries the vertical range of the MCPR graph.

**Syntax**       :DISPlay:MCPower:Y[:SCALe] <value>  
               :DISPlay:MCPower:Y[:SCALe]?

**Arguments**   <value>::=<NRf> specifies the vertical range.  
 Range: 0.1 to 200 dB.

**Measurement Views**   MCPR

**Examples**       :DISPlay:MCPower:Y:SCALe 100  
 sets the vertical range to 100 dB in the MCPR graph.

**Related Commands**   :DISPlay:MCPower:Y[:SCALe]:OFFSet

**:DISPlay:MCPower:Y[:SCALE]:AUTO (No Query Form)**

Rescales the vertical axis automatically to fit the waveform to the screen in the MCPR view.

**Syntax** :DISPlay:MCPower:Y[:SCALE]:AUTO

**Arguments** None

**Measurement Views** MCPR

**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.

**Syntax** :DISPlay:MCPower:Y[:SCALE]:OFFSet <value>  
:DISPlay:MCPower:Y[:SCALE]:OFFSet?

**Arguments** <value>::=<NRf> specifies the vertical offset.  
Range: -170 to +50 dBm.

**Measurement Views** MCPR

**Examples** :DISPlay:MCPower:Y:SCALE:OFFSet -12.5  
sets the vertical offset to -12.5 dBm in the MCPR graph.

**Related Commands** :DISPlay:MCPower:Y[:SCALE]

## :DISPlay:MERRor Subgroup (Option 21 Only)

The :DISPlay:MERRor commands set up the Magnitude error versus Time view.

Command Tree	Header	Parameter
	:DISPlay	
	:MERRor	
	:Y	
	[:SCALe]	<numeric_value>
	:AUTO	
	:OFFSet	<numeric_value>

### :DISPlay:MERRor:Y[:SCALe](?)

Sets or queries the vertical range of the Magnitude error versus Time graph.

**Syntax** :DISPlay:MERRor:Y[:SCALe] <value>  
 :DISPlay:MERRor:Y[:SCALe]?

**Arguments** <value>::=<NRf> specifies the vertical range.  
 Range: 1 to 100%.

**Measurement Views** Magnitude error versus Time

**Examples** :DISPlay:MERRor:Y:SCALe 50  
 sets the vertical range to 50% in the Magnitude error versus Time graph.

**Related Commands** :DISPlay:MERRor:Y[:SCALe]:OFFSet

**: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.

**Syntax** :DISPlay:MERRor:Y[:SCALe]:AUTO

**Arguments** None

**Measurement Views** Magnitude error versus Time

**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.

**Syntax** :DISPlay:MERRor:Y[:SCALe]:OFFSet <value>  
:DISPlay:MERRor:Y[:SCALe]:OFFSet

**Arguments** <value>::=<NRf> specifies the minimum vertical value.  
Range: -100 to +100%.

**Measurement Views** Magnitude error versus Time

**Examples** :DISPlay:MERRor:Y:SCALe:OFFSet -9.5  
sets the minimum vertical value to -9.5% in the Magnitude error versus Time graph.

**Related Commands** :DISPlay:MERRor:Y[:SCALe]

## :DISPlay:OBWidth Subgroup

The :DISPlay:OBWidth commands control the Occupied Bandwidth (OBW) view.

Command Tree	Header	Parameter
	:DISPlay	
	:OBWidth	
	:MARKer	
	:SHOW	
	:STATe	<boolean>
	:RESet	
	:SCALe	
	:SELected	
	:BANdwidth	OBWidth   XDBbandwidth
	:WINDow	
	:TRACe	
	:GRATicule	
	:GRID	
	:STATe	<boolean>
	:X	
	[:SCALe]	<numeric_value>
	:AUTO	
	:OFFSet	<numeric_value>
	:Y	
	[:SCALe]	<numeric_value>
	:AUTO	
	:OFFSet	<numeric_value>



**:DISPlay:OBWidth:MARKer:SHOW:STATe(?)**

Determines whether to show or hide the readout for the selected marker in the Occupied Bandwidth view.

**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.

**Measurement Views** Occupied Bandwidth

**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  
Horizontal scale = Default span

**Syntax** :DISPlay:OBWidth:RESet:SCALE

**Arguments** None

**Measurement Views** Occupied Bandwidth

**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.

**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.

**Measurement Views**   Occupied Bandwidth

**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.

**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.

**Measurement Views**   Occupied Bandwidth

**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.

**Syntax** :DISPlay:OBWidth:X[:SCALe] <value>  
:DISPlay:OBWidth:X[:SCALe]?

**Arguments** <value>::=<NRf> specifies the horizontal range.  
Range: 10 Hz to 6.2 GHz (RSA6106A) / 14 GHz (RSA6114A).

**Measurement Views** Occupied Bandwidth

**Examples** :DISPlay:OBWidth:X:SCALe 10MHz  
sets the horizontal range to 10 MHz.

**Related Commands** :DISPlay:OBWidth:X[:SCALe]:OFFSet

**: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.

**Syntax** :DISPlay:OBWidth:X[:SCALe]:AUTO

**Arguments** None

**Measurement Views** Occupied Bandwidth

**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.

**Syntax**     :DISPlay:OBWidth:X[:SCALe]:OFFSet <value>  
              :DISPlay:OBWidth:X[:SCALe]:OFFSet?

**Arguments**   <value>::=<NRf> specifies the minimum horizontal value.  
              Range: [(center frequency) - (X scale) × 0.9] to  
                      [(center frequency) + (X scale) × 0.9]

**Measurement Views**   Occupied Bandwidth

**Examples**       :DISPlay:OBWidth:X:SCALe:OFFSet 1.45GHz  
                  sets the minimum horizontal value to 1.45 GHz in the Occupied Bandwidth view.

**Related Commands**   :DISPlay:OBWidth:X[:SCALe]

**:DISPlay:OBWidth:Y[:SCALe](?)**

Sets or queries the vertical range of the Occupied Bandwidth view.

**Syntax**       :DISPlay:OBWidth:Y[:SCALe] <value>  
              :DISPlay:OBWidth:Y[:SCALe]?

**Arguments**   <value>::=<NRf> specifies the vertical range.  
              Range: 0.1 to 200 dB.

**Measurement Views**   Occupied Bandwidth

**Examples**       :DISPlay:OBWidth:Y:SCALe 100  
                  sets the vertical range to 100 dB in the Occupied Bandwidth view.

**Related Commands**   :DISPlay:OBWidth:Y[:SCALe]:OFFSet

**: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.

**Syntax** :DISPlay:OBWidth:Y[:SCALE]:AUTO

**Arguments** None

**Measurement Views** Occupied Bandwidth

**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.

**Syntax** :DISPlay:OBWidth:Y[:SCALE]:OFFSet <value>  
:DISPlay:OBWidth:Y[:SCALE]:OFFSet?

**Arguments** <value>::=<NRf> specifies the vertical offset.  
Range: -170 to +50 dBm.

**Measurement Views** Occupied Bandwidth

**Examples** :DISPlay:OBWidth:Y:SCALE:OFFSet -12.5  
sets the vertical offset to -12.5 dBm in the Occupied Bandwidth view.

**Related Commands** :DISPlay:OBWidth:Y[:SCALE]

## :DISPlay:PERRor Subgroup (Option 21 Only)

The :DISPlay:PERRor commands set up the Phase error versus Time view.

Command Tree	Header	Parameter
	:DISPlay	
	:PERRor	
	:Y	
	[:SCALe]	<numeric_value>
	:AUTO	
	:OFFSet	<numeric_value>

## :DISPlay:PERRor:Y[:SCALe](?)

Sets or queries the vertical range of the Phase error versus Time graph.

**Syntax** :DISPlay:PERRor:Y[:SCALe] <value>  
 :DISPlay:PERRor:Y[:SCALe]?

**Arguments** <value>::=<NRf> specifies the vertical range.  
 Range: 1 to 360°.

**Measurement Views** Phase error versus Time

**Examples** :DISPlay:PERRor:Y:SCALe 30  
 sets the vertical range to 30° in the Phase error versus Time graph.

**Related Commands** :DISPlay:PERRor:Y[:SCALe]:OFFSet

**: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.

**Syntax** :DISPlay:PERRor:Y[:SCALe]:AUTO

**Arguments** None

**Measurement Views** Phase error versus Time

**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.

**Syntax** :DISPlay:PERRor:Y[:SCALe]:OFFSet <value>  
:DISPlay:PERRor:Y[:SCALe]:OFFSet?

**Arguments** <value>::=<NRf> specifies the minimum vertical value.  
Range: -360 to +360°.

**Measurement Views** Phase error versus Time

**Examples** :DISPlay:PERRor:Y:SCALe:OFFSet -14.5  
sets the minimum vertical value to -14.5° in the Phase Error versus Time graph.

**Related Commands** :DISPlay:PERRor:Y[:SCALe]

## :DISPlay:PHVTime Subgroup

The :DISPlay:PHVTime commands control display in the Phase versus Time measurement.

Command Tree	Header	Parameter
	:DISPlay	
	:PHVTime	
	:WINDow	
	:TRACe	
	:GRATICule	
	:GRID	
	:STATe	<boolean>
	:X	
	[:SCALE]	<numeric_value>
	:AUTO	
	:STATe	<boolean>
	:MAXimum?	
	:MINimum?	
	:OFFSet	<numeric_value>
	:MAXimum?	
	:MINimum?	
	:Y	
	[:SCALE]	<numeric_value>
	:AUTO	
	:AXIS	MODulopi   CONTInuous
	:REFERence	<numeric_value>
	:OFFSet	<numeric_value>
	:RESCale	



**:DISPlay:PHVTime:WINDow:TRACe:GRATicule:GRID:STATe(?)**

Determines whether to show or hide the graticule grid on the screen.

**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.

**Measurement Views**   Phase versus Time

**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.

**Syntax**     :DISPlay:PHVTime:X[:SCALe] <value>  
              :DISPlay:PHVTime:X[:SCALe]?

**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|:MINimum? query to get the upper and lower limit values of the setting range.

**Measurement Views**   Phase versus Time

**Examples**       :DISPlay:PHVTime:X:SCALe 1.5ms  
                  sets the horizontal scale to 1.5 ms.

**Related Commands**   :DISPlay:PHVTime:X[:SCALe]:AUTO:STATe,  
                      :DISPlay:PHVTime:X[:SCALe]:OFFSet,  
                      :DISPlay:PHVTime:X[:SCALe]:MAXimum|:MINimum

**: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.

**Syntax** :DISPlay:PHVTime:X[:SCALe]:AUTO

**Arguments** None

**Measurement Views** Phase versus Time

**Examples** :DISPlay:PHVTime:X:SCALe:AUTO  
sets the horizontal scale automatically to fit the waveform to the screen.

**Related Commands** :DISPlay:PHVTime:X[:SCALe]:AUTO:STATe

**:DISPlay:PHVTime:X[:SCALe]:AUTO:STATe(?)**

Determines whether to set the horizontal scale automatically or manually.

**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.

**Measurement Views** Phase versus Time

**Examples** :DISPlay:PHVTime:X:SCALe:AUTO:STATe ON  
specifies that the horizontal scale is set automatically.

**Related Commands** :DISPlay:PHVTime:X[:SCALe], :DISPlay:PHVTime:X[:SCALe]:OFFSet

**:DISPlay:PHVTime:X[:SCALe]:MAXimum? (Query Only)**

Queries the upper limit of the horizontal scale setting range.

**Syntax** :DISPlay:PHVTime:X[:SCALe]:MAXimum?

**Returns** <NRf> The upper limit of the horizontal scale setting range.

**Measurement Views** Phase versus Time

**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.

**Related Commands** :DISPlay:PHVTime:X[:SCALe]

**:DISPlay:PHVTime:X[:SCALe]:MINimum? (Query Only)**

Queries the lower limit of the horizontal scale setting range.

**Syntax** :DISPlay:PHVTime:X[:SCALe]:MINimum?

**Returns** <NRf> The lower limit of the horizontal scale setting range.

**Measurement Views** Phase versus Time

**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.

**Related Commands** :DISPlay:PHVTime:X[:SCALe]

**: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.

**Syntax** :DISPlay:PHVTime:X[:SCALe]:OFFSet  
:DISPlay:PHVTime:X[:SCALe]:OFFSet?

**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|:MINimum? query to get the upper and lower limit values of the setting range.

**Measurement Views** Phase versus Time

**Examples** :DISPlay:PHVTime:X:SCALe:OFFSet 800ns  
sets the minimum horizontal value to 800 ns in the Phase versus Time graph.

**Related Commands** :DISPlay:PHVTime:X[:SCALe]:AUTO:STATe,  
:DISPlay:PHVTime:X[:SCALe],  
:DISPlay:PHVTime:X[:SCALe]:OFFSet:MAXimum|:MINimum?

**:DISPlay:PHVTime:X[:SCALe]:OFFSet:MAXimum? (Query Only)**

Queries the upper limit of the horizontal offset setting range.

**Syntax** :DISPlay:PHVTime:X[:SCALe]:OFFSet:MAXimum?

**Returns** <NRf> The upper limit of the horizontal offset setting range.

**Measurement Views** Phase versus Time

**Examples** :DISPlay:PHVTime:X:SCALe:OFFSet:MAXimum?  
might return  $-1.812\text{E}-3$ , indicating that the upper limit of the horizontal offset setting range is  $-1.812$  ms.

**Related Commands** :DISPlay:PHVTime:X[:SCALe]:OFFSet

**:DISPlay:PHVTime:X[:SCALe]:OFFSet:MINimum? (Query Only)**

Queries the lower limit of the horizontal offset setting range.

**Syntax** :DISPlay:PHVTime:X[:SCALe]:OFFSet:MINimum?

**Returns** <NRf> The lower limit of the horizontal offset setting range.

**Measurement Views** Phase versus Time

**Examples** :DISPlay:PHVTime:X:SCALe:OFFSet:MINimum?  
might return  $-16.28\text{E}-3$ , indicating that the lower limit of the horizontal offset setting range is  $-16.28$  ms.

**Related Commands** :DISPlay:PHVTime:X[:SCALe]:OFFSet

**:DISPlay:PHVTime:Y[:SCALe](?)**

Sets or queries the vertical range of the Phase versus Time graph.

**Syntax**     :DISPlay:PHVTime:Y[:SCALe] <value>  
               :DISPlay:PHVTime:Y[:SCALe]

**Arguments**   <value>::=<NRf> specifies the vertical range.  
 Range: 1 to 1T°.

**Measurement Views**   Phase versus Time

**Examples**       :DISPlay:PHVTime:Y:SCALe 180  
 sets the vertical range to 180° in the Phase versus Time graph.

**Related Commands**   :DISPlay:PHVTime:Y[:SCALe]:OFFSet

**: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.

**Syntax**       :DISPlay:PHVTime:Y[:SCALe]:AUTO

**Arguments**   None

**Measurement Views**   Phase versus Time

**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.

**Syntax**     :DISPlay:PHVTime:Y[:SCALe]:AXIS { MODulo $\pi$  | CONTinuous }  
              :DISPlay:PHVTime:Y[:SCALe]:AXIS?

**Arguments**   MODulo $\pi$  (modulo  $\pi$ ) shows the phase constrained within  $\pm 180^\circ$  along the vertical axis.  
              CONTinuous shows the phase as continuous quantity along the vertical axis.

**Measurement Views**   Phase versus Time

**Examples**     :DISPlay:PHVTime:Y:SCALe:AXIS MODulo $\pi$   
                  selects modulo  $\pi$  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.

**Syntax**     :DISPlay:PHVTime:Y[:SCALe]:AXIS:REFErence <value>  
              :DISPlay:PHVTime:Y[:SCALe]:AXIS:REFErence?

**Arguments**   <value>::=<NRf> specifies the phase reference time.

**Measurement Views**   Phase versus Time

**Examples**     :DISPlay:PHVTime:Y:SCALe:AXIS:REFErence 1.5us  
                  sets the phase reference time to 1.5  $\mu$ s.



**: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.

**Syntax** :DISPlay:PHVTime:Y[:SCALe]:OFFSet <value>

:DISPlay:PHVTime:Y[:SCALe]:OFFSet?

**Arguments** <value>::=<NRf> specifies the vertical offset.  
Range:  $-0.5T$  to  $+0.5T^\circ$ .

**Measurement Views** Phase versus Time

**Examples** :DISPlay:PHVTime:Y:SCALe:OFFSet -158.5  
sets the vertical offset to  $-158.5^\circ$  in the Phase versus Time graph.

**Related Commands** :DISPlay:PHVTime:Y[:SCALe]

**:DISPlay:PHVTime:Y[:SCALe]:RESCale (No Query Form)**

Rescales the vertical axis automatically to fit the Phase versus Time waveform to the screen.

**Syntax** :DISPlay:PHVTime:Y[:SCALe]:RESCale

**Arguments** None

**Measurement Views** Phase versus Time

**Examples** :DISPlay:PHVTime:Y:SCALe:RESCale  
rescales the vertical axis automatically to fit the Phase versus Time waveform to the screen.

## :DISPlay:PNOise Subgroup (Option 11 Only)

The :DISPlay:PNOise commands control the phase noise view.

Command Tree	Header	Parameter
	:DISPlay	
	:PNOise	
	:MARKer	
	:SHOW	
	:STATe	<boolean>
	:RESet	
	:SCALe	
	:WINDow	
	:TRACe	
	:GRATicule	
	:GRID	
	:STATe	<boolean>
	:X	
	[:SCALe]	
	:AUTO	
	:START	<numeric_value>
	:STOP	<numeric_value>
	:Y	
	[:SCALe]	<numeric_value>
	:AUTO	
	:OFFSet	<numeric_value>
	:PDIVision	<numeric_value>

**:DISPlay:PNOise:MARKer:SHOW:STATe(?)**

Determines whether to show or hide the readout for the selected marker in the phase noise view.

**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.

**Measurement Views** Phase noise

**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  
Horizontal stop = 1 GHz

**Syntax** :DISPlay:PNOise:RESet:SCALe

**Arguments** None

**Measurement Views** Phase noise

**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.

**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.

**Measurement Views**   Phase noise

**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.

**Syntax**     :DISPlay:PNOise:X[:SCALE]:AUTO

**Arguments**   None

**Measurement Views**   Phase noise

**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.

**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).

**Measurement Views**   Phase noise

**Examples**       :DISPlay:PNOise:X:SCALe:START 10Hz  
 sets the start frequency to 10 Hz in the phase noise graph.

**Related Commands**   :DISPlay:PNOise:X[:SCALe]:STOP

**:DISPlay:PNOise:X[:SCALe]:STOP(?)**

Sets or queries the stop frequency (right edge) of the phase noise graph.

**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).

**Measurement Views**   Phase noise

**Examples**       :DISPlay:PNOise:X:SCALe:STOP 2GHz  
 sets the stop frequency to 2 GHz in the phase noise graph.

**Related Commands**   :DISPlay:PNOise:X[:SCALe]:START

**:DISPlay:PNOise:Y[:SCALe](?)**

Sets or queries the vertical range of the phase noise graph.

**Syntax** :DISPlay:PNOise:Y[:SCALe] <value>  
:DISPlay:PNOise:Y[:SCALe]?

**Arguments** <value>::=<NRf> specifies the vertical range.  
Range: 0.1 to 200 dB.

**Measurement Views** Phase noise

**Examples** :DISPlay:PNOise:Y:SCALe 100  
sets the vertical range to 100 dB for the phase noise graph.

**Related Commands** :DISPlay:PNOise:Y[:SCALe]:OFFSet

**: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.

**Syntax** :DISPlay:PNOise:Y[:SCALe]:AUTO

**Arguments** None

**Measurement Views** Phase noise

**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.

**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.

**Measurement Views** Phase noise

**Examples** :DISPlay:PNOise:Y:SCALe:OFFSet -12.5  
sets the vertical offset to -12.5 dBc/Hz for the phase noise graph.

**Related Commands** :DISPlay:PNOise:Y[:SCALe]

**:DISPlay:PNOise:Y[:SCALe]:PDIVision(?)**

Sets or queries the vertical scale (per division) of the phase noise graph.

**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.

**Measurement Views** Phase noise

**Examples** :SENSe:DPSA:Y:SCALe:PDIVision 5  
sets the vertical scale to 5 dB/div.

## :DISPlay:PULSe Subgroup (Option 20 Only)

The :DISPlay:PULSe commands control display in the Pulsed RF measurements.

Command Tree	Header	Parameter
	:DISPlay	
	:PULSe	
	:MEASview	
	:DELeTe	RESuIt   TRACe   STATistics
	:NEW	RESuIt   TRACe   STATistics
	:SELeCt	RESuIt   TRACe   STATistics
	:RESuIt	
	:ATX	<boolean>
	:AVERage	<boolean>
	:DROp	<boolean>
	:DUTPct	<boolean>
	:DUTRatio	<boolean>
	:FALL	<boolean>
	:FRDeviation	<boolean>
	:MFRReqerror	<boolean>
	:MPHerror	<boolean>
	:PHDeviation	<boolean>
	:PPFRequency	<boolean>
	:PPOWer	<boolean>
	:PPPHase	<boolean>
	:RINTerval	<boolean>
	:RIPPlE	<boolean>
	:RISE	<boolean>
	:RMSFreqerror	<boolean>
	:RMSPherror	<boolean>
	:RRATe	<boolean>
	:TIME	<boolean>
	:WIDTh	<boolean>
	:SELeCt	
	:NUMBer	<numeric_value>
	:RESuIt	AVERage   PPOWer   ATX   WIDTh   RISE   FALL   RINTerval   RRATe   DUTPct   DUTRatio   RIPPlE   DROp   PPPHase   PPFRequency   RMSFreqerror   MFRReqerror   RMSPherror   MPHerror   FRDeviation   PHDeviation



```

:STATistics
  :MARKer
    :SHOW
      :STATe <boolean>
  :PLOT TRENd | FFT
  :WINDow
    :TRACe
      :GRATicule
      :GRID
      :STATe <boolean>
  :X
    :RSCale
    [:SCALE]
      :NUMBer <numeric_value>
      :OFFSet <numeric_value>
  :Y
    :RSCale
    [:SCALE]
      :FULL <numeric_value>
      :OFFSet <numeric_value>
      :STOP? <numeric_value>
:TRACe
  :MARKer
    :SHOW
      :STATe <boolean>
  :POINT
    :SHOW <boolean>
  :WINDow
    :TRACe
      :GRATicule
      :GRID
      :STATe <boolean>
  :X
    :RSCale
    [:SCALE] <numeric_value>
      :FULL SElecteD | MAXimum
      :OFFSet <numeric_value>
      :PDIVision <numeric_value>
  :Y
    :RSCale
    [:SCALE]
      :FULL <numeric_value>
      :OFFSet <numeric_value>
      :STOP? <numeric_value>

```

## **:DISPlay:PULSe:MEASview:DELeTe (No Query Form)**

Deletes the measurement view in the pulsed RF measurements.

**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.

**Measurement Views**   Pulsed RF measurements

**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.

**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.

**Measurement Views**   Pulsed RF measurements

**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.

**Syntax**     :DISPlay:PULSe:MEASview:SElect { RESult | TRACe | STATistics }

**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.

**Measurement Views**   Pulsed RF measurements

**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.

**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.

**Measurement Views**   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.

**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.

**Measurement Views**   Pulse table

**Examples**     :DISPlay:PULSe:RESuLt:AVERage ON  
              shows the average on power measurement result in the pulse table.

**:DISPlay:PULSe:RESuLt:DR0op(?)**

Determines whether or not to show the droop measurement result in the pulse table.

**Syntax**     :DISPlay:PULSe:RESuLt:DR0op { OFF | ON | 0 | 1 }  
              :DISPlay:PULSe:RESuLt:DR0op?

**Arguments**   OFF or 0 does not show the droop measurement result.  
              ON or 1 shows the droop measurement result in the pulse table.

**Measurement Views**   Pulse table

**Examples**     :DISPlay:PULSe:RESuLt:DR0op 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.

**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.

**Measurement Views** 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.

**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.

**Measurement Views** 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.

**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.

**Measurement Views**   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.

**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.

**Measurement Views**   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.

**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.

**Measurement Views** 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.

**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.

**Measurement Views** 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.

**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.

**Measurement Views**   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.

**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.

**Measurement Views**   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.

**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.

**Measurement Views**   Pulse table

**Examples**     :DISPlay:PULSe:RESult:PPower ON  
              shows the peak power measurement result in the pulse table.

## **:DISPlay:PULSe:RESult:PPHase(?)**

Determines whether or not to show the pulse-pulse carrier phase measurement result in the pulse table.

**Syntax**     :DISPlay:PULSe:RESult:PPHase { OFF | ON | 0 | 1 }  
              :DISPlay:PULSe:RESult:PPHase?

**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.

**Measurement Views**   Pulse table

**Examples**     :DISPlay:PULSe:RESult:PPHase 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.

**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.

**Measurement Views**   Pulse 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.

**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.

**Measurement Views**   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.

**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.

**Measurement Views** 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.

**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.

**Measurement Views** 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.

**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.

**Measurement Views**   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.

**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.

**Measurement Views**   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.

**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.

**Measurement Views** 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.

**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.

**Measurement Views** 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.

**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.

**Measurement Views**   Pulse statistics, Pulse table, Pulse trace

**Examples**       :DISPlay:WINDow:SElect:NUMBer -28  
                  measures the pulse #-28.

**Related Commands**   [:SENSe]:ANALysis:REFerence

**:DISPlay:PULSe:SElect:RESult(?)**

Selects or queries which result is shown in the pulse trace and statistics views.

**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 }

**Arguments** Table 2-35 lists the arguments.

**Table 2-35: 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

**Measurement Views** Pulse statistics, Pulse trace

**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.

**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.

**Measurement Views** Pulse statistics

**Examples** :DISPlay:PULSe:STATistics:MARKer:SHOW:STATE ON  
shows the marker readout in the statistics graph.

**Related Commands** :DISPlay:PULSe:STATistics:PLOT



**:DISPlay:PULSe:STATistics:PLOT(?)**

Selects or queries how to show the statistics graph.

**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.

**Measurement Views** Pulse statistics

**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.

**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.

**Measurement Views** Pulse statistics

**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.

**Syntax** :DISPlay:PULSe:STATistics:X:RSCale

**Arguments** None

**Measurement Views** Pulse statistics

**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.

**Syntax** :DISPlay:PULSe:STATistics:X[:SCALE]:NUMBER <value>  
:DISPlay:PULSe:STATistics:X[:SCALE]:NUMBER?

**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.

<b>:DISPlay:PULSe:STATistics:PLOT</b>	<b>Setting range</b>
TREND	1 to 1000
FFT	1 Hz to 120 MHz

**Measurement Views** Pulse statistics

**Examples** :DISPlay:PULSe:STATistics:X:SCALE:NUMBER 50  
sets the horizontal full scale to 50 pulses when the plot is trend.

**Related Commands** :DISPlay:PULSe:STATistics:PLOT,  
:DISPlay:PULSe:STATistics:X[:SCALE]:OFFSet

**:DISPlay:PULSe:STATistics:X[:SCALe]:OFFSet(?)**

Selects or queries the minimum horizontal value (the first pulse to show) in the statistics graph.

**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

**Measurement Views** Pulse statistics

**Examples** :DISPlay:PULSe:STATistics:X:SCALe:OFFSet 120  
sets the first pulse number to #120.

**Related Commands** :DISPlay:PULSe:STATistics:X[:SCALe]:NUMBer

**:DISPlay:PULSe:STATistics:Y:RSCale (No Query Form)**

Rescales the vertical axis to fit the waveform to the screen in the statistics graph.

**Syntax** :DISPlay:PULSe:STATistics:Y:RSCale

**Arguments** None

**Measurement Views** Pulse statistics

**Examples** :DISPlay:PULSe:STATistics:X: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.

**Syntax**     :DISPlay:PULSe:STATistics:Y[:SCALe]:FULL <value>  
               :DISPlay:PULSe:STATistics:Y[:SCALe]:FULL?

**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 table below.

<b>:DISPlay:PULSe:STATistics:PLOT</b>	<b>Measurement item</b>	<b>Setting range</b>
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

**Measurement Views**   Pulse statistics

**Examples**           :DISPlay:PULSe:STATistics:Y:SCALe:FULL 100  
 sets the vertical full scale to 100 dB.

**Related Commands**   :DISPlay:PULSe:STATistics:PLOT,  
 :DISPlay:PULSe:STATistics:X[:SCALe]:OFFSet

**:DISPlay:PULSe:STATistics:Y[:SCALe]:OFFSet(?)**

Sets or queries the vertical offset in the statistics graph.

**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 <sup>1</sup>
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
	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

<sup>1</sup> Indicates whether the offset is the value at the top or the bottom edge of the graph.

**Measurement Views** Pulse statistics

**Examples** :DISPlay:PULSe:STATistics:Y:SCALe:OFFSet 24.8  
sets the maximum vertical value to 24.8 dBm in the statistics graph.

**Related Commands** :DISPlay:PULSe:STATistics:PLOT,  
:DISPlay:PULSe:STATistics:X[:SCALe]:PDIVision

**:DISPlay:PULSe:STATistics:Y[:SCALe]:STOP? (Query Only)**

Queries the minimum vertical value (bottom edge) in the statistics graph.

**Syntax** :DISPlay:PULSe:STATistics:Y[:SCALe]:STOP?

**Returns** <y\_stop>::=<NRf> is the minimum vertical value (bottom edge).

**Measurement Views** Pulse statistics

**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.

**Related Commands** :DISPlay:PULSe:STATistics:X[:SCALe]:OFFSet

**:DISPlay:PULSe:TRACe:MARKer:SHOW:STATe(?)**

Determines whether to show or hide the marker readout in the pulse trace view.

**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.

**Measurement Views** Pulse trace

**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.

**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.

**Measurement Views** Pulse trace

**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.

**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.

**Measurement Views** Pulse trace

**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.

**Syntax**       :DISPlay:PULSe:TRACe:X[:SCALe]:RSCale

**Arguments**   None

**Measurement Views**   Pulse trace

**Examples**       :DISPlay:PULSe:TRACe:X:SCALe: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.

**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.

**Measurement Views**   Pulse trace

**Examples**       :DISPlay:PULSe:TRACe:X:SCALe 5.5E-6  
sets the horizontal scale to 5.5  $\mu$ s.



**:DISPlay:PULSe:TRACe:X[:SCALe]:FULL(?)**

Selects or queries the full-scale reference for the horizontal rescale.

**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.

**Measurement Views**   Pulse trace

**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.

**Syntax**     :DISPlay:PULSe:TRACe:X[:SCALe]:OFFSet <value>  
              :DISPlay:PULSe:TRACe:X[:SCALe]:OFFSet?

**Arguments**   <value>::=<NRf> specifies the minimum horizontal value.  
                  Range: [(analysis offset) - (X scale) × 0.9] to  
                              [(analysis offset) + (analysis length) - (X scale) × 0.1]

**Measurement Views**   Pulse trace

**Examples**     :DISPlay:PULSe:TRACe:X:SCALe:OFFSet 937.5E-9  
                  sets the minimum horizontal value to 937.5 ns.

**Related Commands**   :DISPlay:PULSe:TRACe:X[:SCALe]:PDIVision

**:DISPlay:PULSe:TRACe:X[:SCALe]:PDIVision(?)**

Sets or queries the horizontal full scale in the pulse trace view.

**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.

**Measurement Views**   Pulse trace

**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.

**Syntax** :DISPlay:PULSe:TRACe:Y[:SCALe]:RSCale

**Arguments** None

**Measurement Views** Pulse trace

**Examples** :DISPlay:PULSe:TRACe:Y:SCALe: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.

**Syntax** :DISPlay:PULSe:TRACe:Y[:SCALe]:FULL <value>  
:DISPlay:PULSe:TRACe:Y[:SCALe]:FULL?

**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.

**Measurement Views** Pulse trace

**Examples** :DISPlay:PULSe:TRACe:Y:SCALe:FULL 100  
sets the vertical full scale to 100 dB.

**Related Commands** [:SENSe]:POWer:UNITs, :DISPlay:PULSe:TRACe:Y[:SCALe]:OFFSet

**: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.

**Syntax** :DISPlay:PULSe:TRACe:Y[:SCALe]:OFFSet <value>

:DISPlay:PULSe:TRACe:Y[:SCALe]:OFFSet?

**Arguments** <value>::=<NRf> specifies the vertical offset.  
Range: -170 to +50 dBm.

**Measurement Views** Pulse trace

**Examples** :DISPlay:PULSe:TRACe:Y:SCALe:OFFSet 23.5  
sets the vertical offset to 23.5 dBm.

**Related Commands** :DISPlay:PULSe:TRACe:Y[:SCALe]:STOP?

**:DISPlay:PULSe:TRACe:Y[:SCALe]:STOP? (Query Only)**

Queries the minimum vertical value (bottom edge) in the pulse trace view.

**Syntax** :DISPlay:PULSe:TRACe:Y[:SCALe]:STOP?

**Returns** <y\_stop>::=<NRf> is the minimum vertical value (bottom edge).

**Measurement Views** Pulse trace

**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.

**Related Commands** :DISPlay:PULSe:TRACe:Y[:SCALe]:OFFSet

## :DISPlay:SGRam Subgroup

The :DISPlay:SGRam commands control the spectrogram view.

Command Tree	Header	Parameter
	:DISPlay:SGRam	
	:FREQuency	
	:AUTO	
	:OFFSet	<numeric_value>
	:SCALe	<numeric_value>
	:TIME	
	:AUTO	
	:OFFSet	<numeric_value>
	:OVERlap	<boolean>
	:SCALe	<numeric_value>

### :DISPlay:SGRam:FREQuency:AUTO (No Query Form)

Rescales the horizontal axis automatically to fit the waveform to the screen in the spectrogram view.

**Syntax** :DISPlay:SGRam:FREQuency:AUTO

**Arguments** None

**Measurement Views** Spectrogram

**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.

**Syntax**     :DISPlay:SGRam:FREQuency:OFFSet <value>  
              :DISPlay:SGRam:FREQuency:OFFSet?

**Arguments**   <value>::=<NRf> specifies the frequency offset.  
              Range: [(center frequency) - (X scale) × 0.9] to  
                      [(center frequency) + (X scale) × 0.9]

**Measurement Views**   Spectrogram

**Examples**       :DISPlay:SGRam:FREQuency:OFFSet 1.45GHz  
                  sets the frequency offset to 1.45 GHz in the spectrogram.

**Related Commands**   :DISPlay:SGRam:FREQuency:SCAlE

**:DISPlay:SGRam:FREQuency:SCAlE(?)**

Sets or queries the horizontal range of the spectrogram.

**Syntax**       :DISPlay:SGRam:FREQuency:SCAlE <value>  
              :DISPlay:SGRam:FREQuency:SCAlE?

**Arguments**   <value>::=<NRf> specifies the horizontal range.  
              Range: 10 Hz to 6.2 GHz (RSA6106A) / 14 GHz (RSA6114A).

**Measurement Views**   Spectrogram

**Examples**       :DISPlay:SGRam:FREQuency:SCAlE 10MHz  
                  sets the horizontal range to 10 MHz.

**Related Commands**   :DISPlay:SGRam:FREQuency:OFFSet

**:DISPlay:SGRam:TIME:AUTO (No Query Form)**

Rescales the vertical axis automatically to fit the graph to the screen in the spectrogram view.

**Syntax** :DISPlay:SGRam:TIME:AUTO

**Arguments** None

**Measurement Views** Spectrogram

**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.

**Syntax** :DISPlay:SGRam:TIME:OFFSet <value>  
:DISPlay:SGRam:TIME:OFFSet?

**Arguments** <value>::=<NRf> specifies the time offset.  
Range: Line #0 to 125000. Zero (0) represents the latest line.

**Measurement Views** Spectrogram

**Examples** :DISPlay:SGRam:TIME:OFFSet 15  
sets the time offset to Line #15.

**Related Commands** :DISPlay:SGRam:TIME:SCAlE

**:DISPlay:SGRam:TIME:OVERlap(?)**

Determines whether or not to allow overlap between adjacent FFT frames on the time axis in the spectrogram.

**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.

**Measurement Views** Spectrogram

**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.

**Syntax** :DISPlay:SGRam:TIME:SCALE <value>

:DISPlay:SGRam:TIME:SCALE?

**Arguments** <value>::=<NR1> specifies the vertical scale.

Range: -1023 to 0. Zero (0) displays every spectrum.

**Measurement Views** Spectrogram

**Examples** :DISPlay:SGRam:TIME:SCALE -5

displays one line every 5 spectra in the spectrogram.

**Related Commands** :DISPlay:SGRam:TIME:OFFSet



## :DISPlay:SPECTrum Subgroup

The :DISPlay:SPECTrum commands control display of the spectrum view.

Command Tree	Header	Parameter
	:DISPlay	
	:SPECTrum	
	:MARKer	
	:NOISe	
	:MODE	<boolean>
	:WINDow	
	:TRACe	
	:GRATicule	
	:GRID	
	:STATe	<boolean>
	:Y	
	[:SCAlE]	
	:PDIVision	<numeric_value>

**: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.

---

**Syntax**     :DISPlay:SPECTrum:MARKer:NOISe:MODE { OFF | ON | 0 | 1 }  
              :DISPlay:SPECTrum:MARKer:NOISe:MODE?

**Arguments**   OFF or 0 disables the marker noise mode.  
              ON or 1 enables the marker noise mode.

**Measurement Views**   Spectrum

**Examples**       :DISPlay:SPECTrum:MARKer:NOISe:MODE ON  
                  enables the marker noise mode.

**Related Commands**   [:SENSE]:POWER:UNITS

**:DISPlay:SPECTrum:WINDow:TRACe:GRATicule:GRID:STATe(?)**

Determines whether to show or hide the graticule grid on the screen.

**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.

**Measurement Views** Spectrum

**Examples** :DISPlay:SPECTrum:WINDow:TRACe:GRATicule:GRID:STATe ON  
 shows the graticule grid on the screen in the spectrum view.

**:DISPlay:SPECTrum:Y[:SCALe]:PDIVision(?)**

Sets or queries the vertical scale (per division) in the spectrum view.

**Syntax** :DISPlay:SPECTrum:Y[:SCALe]:PDIVision <value>  
 :DISPlay:SPECTrum:Y[:SCALe]:PDIVision?

**Arguments** <value>::=<NRf> specifies the vertical scale (per division).  
 Range: 0.01 to 20 dB/div.

**Measurement Views** Spectrum

**Examples** :SENSe:SPECTrum:Y:SCALe:PDIVision 0.5  
 sets the vertical scale to 0.5 dB/div.

**Related Commands** [:SENSe]:POWer:UNITs

## :DISPlay:SPURious Subgroup

The :DISPlay:SPURious commands control the Spurious view.

Command Tree	Header	Parameter
	:DISPlay	
	:SPURious	
	:MARKer	
	:SHOW	
	:STATe	<boolean>
	:RESet	
	:SCALE	
	:SElect	
	:NUMBer	<numeric_value>
	:SHOW	
	:LIMit	SHADed   LINE   OFF
	:WINDow	
	:TRACe	
	:GRATicule	
	:GRID	
	:STATe	<boolean>
	:X	
	[:SCALE]	
	:AUTO	
	:START	<numeric_value>
	:STOP	<numeric_value>
	:Y	
	[:SCALE]	<numeric_value>
	:AUTO	
	:OFFSet	<numeric_value>

**:DISPlay:SPURious:MARKer:SHOW:STATe(?)**

Determines whether to show or hide the readout for the selected marker in the Spurious view.

**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.

**Measurement Views** Spurious

**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  
Horizontal scale = Default span

**Syntax** :DISPlay:SPURious:RESet:SCALE

**Arguments** None

**Measurement Views** Spurious

**Examples** :DISPlay:SPURious:RESet:SCALE  
resets the horizontal and vertical scale to the default values.

**:DISPlay:SPURious:SElect:NUMBer(?)**

Selects or queries the spurious number in the Spurious view.

**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/:READ:SPURious:COUNT? query to get the number of  
                  spurious signals.

**Measurement Views**   Spurious

**Examples**       :DISPlay:SPURious:SElect:NUMBer 7  
                  selects the spurious #7.

**Related Commands**   :FETCh:SPURious:COUNT?, :READ:SPURious:COUNT?

**:DISPlay:SPURious:SHOW:LIMit(?)**

Selects or queries how to display the limits.

**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.

**Measurement Views**   Spurious

**Examples**       :DISPlay:SPURiousSHOW: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.

**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.

**Measurement Views** Spurious

**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.

**Syntax** :DISPlay:SPURious:X[:SCALE]:AUTO

**Arguments** None

**Measurement Views** Spurious

**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.

**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).

**Measurement Views** Spurious

**Examples** :DISPlay:SPURious:X:SCALe:START 1.61GHz  
sets the minimum horizontal value to 1.61 GHz in the spectrum graph.

**Related Commands** :DISPlay:SPURious:X[:SCALe]:STOP

**:DISPlay:SPURious:X[:SCALe]:STOP(?)**

Sets or queries the maximum horizontal value (right edge) of the spectrum graph in the Spurious view.

**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).

**Measurement Views** Spurious

**Examples** :DISPlay:SPURious:X:SCALe:STOP 2.16GHz  
sets the maximum horizontal value to 2.16 GHz in the spectrum graph.

**Related Commands** :DISPlay:SPURious:X[:SCALe]:START



**:DISPlay:SPURious:Y[:SCALe](?)**

Sets or queries the vertical range of the spectrum graph in the Spurious view.

**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.

**Measurement Views**   Spurious

**Examples**       :DISPlay:SPURious:Y:SCALe 100  
 sets the vertical range to 100 dB in the Spurious view.

**Related Commands**   :DISPlay:SPURious:Y[:SCALe]:OFFSet, [:SENSe]:POWer:UNITs

**:DISPlay:SPURious:Y[:SCALe]:AUTO (No Query Form)**

Rescales the vertical axis automatically to fit the waveform to the screen in the Spurious view.

**Syntax**       :DISPlay:SPURious:Y[:SCALe]:AUTO

**Arguments**   None

**Measurement Views**   Spurious

**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.

**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.

**Measurement Views**   Spurious

**Examples**       :DISPlay:SPURious:Y:SCALe:OFFSet -12.5  
sets the vertical offset to -12.5 dBm in the spectrum graph.

**Related Commands**   :DISPlay:SPURious:Y[:SCALe], [:SENSe]:POWer:UNITs

## :DISPlay:TOVerview Subgroup

The :DISPlay:TOVerview commands control display of the time overview.

Command Tree	Header	Parameter
	:DISPlay	
	:TOVerview	
	:WINDow	
	:TRACe	
	:GRATicule	
	:GRID	
	:STATe	<boolean>
	:X	
	[:SCALe]	<numeric_value>
	:AUTO	
	:OFFSet	<numeric_value>
	:Y	
	[:SCALe]	<numeric_value>
	:AUTO	
	:OFFSet	<numeric_value>
	:RESCale	

### :DISPlay:TOVerview:WINDow:TRACe:GRATicule:GRID:STATe(?)

Determines whether to show or hide the graticule grid on the screen.

**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.

**Measurement Views** Time overview

**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.

**Syntax**     :DISPlay:TOVerview:X[:SCALe] <value>  
              :DISPlay:TOVerview:X[:SCALe]?

**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.

**Measurement Views**   Time overview

**Examples**       :DISPlay:TOVerview:X:SCALe 12.5us  
                  sets the horizontal scale to 12.5  $\mu$ s.

**Related Commands**   :DISPlay:TOVerview:X[:SCALe]:OFFSet,  
                      [:SENSe]:ACQuisition:MEMory:CAPacity[:TIME]?

**: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.

**Syntax** :DISPlay:TOVerview:X[:SCALe]:AUTO

**Arguments** None

**Measurement Views** Time overview

**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.

**Syntax** :DISPlay:TOVerview:X[:SCALe]:OFFSet <value>  
:DISPlay:TOVerview:X[:SCALe]:OFFSet?

**Arguments** <value>::=<NRf> specifies the minimum horizontal value.  
Range: [(analysis offset) - (X scale) × 0.9] to  
[(analysis offset) + (analysis length) - (X scale) × 0.1]

**Measurement Views** Time overview

**Examples** :DISPlay:TOVerview:X:SCALe:OFFSet 800ns  
sets the minimum horizontal value to 800 ns in the time overview.

**Related Commands** :DISPlay:TOVerview:X[:SCALe]

**:DISPlay:TOVerview:Y[:SCALe](?)**

Sets or queries the vertical range of the time overview.

**Syntax**     :DISPlay:TOVerview:Y[:SCALe] <value>  
              :DISPlay:TOVerview:Y[:SCALe]?

**Arguments**   <value>::=<NRf> specifies the vertical range.  
                  Range: 0.1 to 200 dB.

**Measurement Views**   Time overview

**Examples**       :DISPlay:TOVerview:Y:SCALe 50  
                  sets the vertical range to 50 dBm in the time overview.

**Related Commands**   :DISPlay:TOVerview:Y[:SCALe]:OFFSet

**: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.

**Syntax**       :DISPlay:TOVerview:Y[:SCALe]:AUTO

**Arguments**   None

**Measurement Views**   Time overview

**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.

**Syntax** :DISPlay:TOVerview:Y[:SCALe]:OFFSet <value>

:DISPlay:TOVerview:Y[:SCALe]:OFFSet

**Arguments** <value>: :=<NRf> specifies the vertical offset.  
Range: -170 to +50 dBm.

**Measurement Views** Time overview

**Examples** :DISPlay:TOVerview:Y:SCALe:OFFSet -80  
sets the vertical offset to -80 dBm in the time overview.

**Related Commands** :DISPlay:TOVerview:Y[:SCALe]

**:DISPlay:TOVerview:Y[:SCALe]:RESCale (No Query Form)**

Sets the vertical scale automatically to fit the waveform to the screen in the time overview.

**Syntax** :DISPlay:TOVerview:Y[:SCALe]:RESCale

**Arguments** None

**Measurement Views** Time overview

**Examples** :DISPlay:TOVerview:Y:SCALe:RESCale  
sets the vertical scale automatically to fit the waveform to the screen.





# 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 on page 2-525. The :READ commands acquire a new input signal and fetch the measurement results from that data.

The FETCh commands are divided by measurement views into the subgroups listed in Table 2-36.

**Table 2-36: FETCh command subgroups**

Command header	Measurement view	Refer to:
:FETCh (basic)	All views, general fetch control	page 2-350
:FETCh:ACPower	Channel power and ACPR	page 2-355
:FETCh:AVTime	Amplitude versus Time	page 2-359
:FETCh:CCDF	CCDF	page 2-364
:FETCh:CONSte	Constellation (Option 21 only)	page 2-369
:FETCh:DDEMod	General purpose digital modulation measurements (Option 21 only)	page 2-371
:FETCh:DPsA	DPX (Digital Phosphor) spectrum	page 2-373
:FETCh:EVM	EVM versus Time (Option 21 only)	page 2-377
:FETCh:FVTime	Frequency versus Time	page 2-380
:FETCh:IQVTime	RF I&Q versus Time	page 2-384
:FETCh:MCPower	MCPR (Multiple Carrier Power Ratio)	page 2-389
:FETCh:MERRor	Magnitude error versus Time (Option 21 only)	page 2-394
:FETCh:OBW	OBW (Occupied Bandwidth)	page 2-397
:FETCh:PERRor	Phase error versus Time (Option 21 only)	page 2-406
:FETCh:PHVTime	Phase versus Time	page 2-409
:FETCh:PNOise	Phase noise (Option 11 only)	page 2-413
:FETCh:PULSe	Pulsed RF measurements (Option 20 only)	page 2-420
:FETCh:SGRam	Spectrogram	page 2-461
:FETCh:SPECTrum	Spectrum	page 2-462
:FETCh:SPURious	Spurious	page 2-464
:FETCh:SQUality	Signal quality (Option 21 only)	page 2-475
:FETCh:TOVerview	Time overview	page 2-487

## :FETCh Basic Command Subgroup

The :FETCh basic commands returns information on acquisition data.

Command Tree	Header	Parameter
	:FETCh	
	:RFIN?	
	:IQ?	<RecordID>, [<NumericDTO>, <NumericDTO>]
	:HEADer?	<RecordID>
	:SCALe?	
	:Record	
	:IDS?	

## :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).

**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.

---

**Measurement Views** All

**Examples** :FETCh:RFIN:IQ? 10,25,350  
might return #43848xxxx... (3848-byte data) for the record #10, ranging from 25<sup>th</sup> to 350<sup>th</sup> IQ pair.

**Related Commands** :FETCh:RFIN:IQ:HEADer?, :FETCh:RFIN:RECOrd:IDS?

**: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).

**Syntax** :FETCh:RFIN:IQ:HEADer? <rec\_ID>

**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.

**Measurement Views** All

**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 11:18:32 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  $\mu$ s  
Time stamp: 10/31/2007 11:18:32 AM  
Fast Frame ID: -1

**Related Commands** :FETCh:RFIN:RECOrd:IDS?

**: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.

**Syntax** :FETCh:RFIN:IQ:SCALE?

**Arguments** None

**Returns** <NR3> The RF linear data scaling factor.

**Measurement Views** All

**Examples** :FETCh:RFIN:IQ:SCALE?  
might return 19.553E-6 for the scaling factor.

**Related Commands** :FETCh:RFIN:RECOrd:IDS?

**:FETCh:RFIN:RECOrd:IDS? (Query Only)**

Returns the beginning and end ID numbers of acquisition data.  
The instrument needs to be in stopped mode. If not in stopped mode, it returns the execution error (-200).

**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.

**Measurement Views** All

**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:ACPower Subgroup**

The :FETCh:ACPower commands returns the results of the Channel power and ACPR (Adjacent Channel Power Ratio) measurement.

<b>Command Tree</b>	<b>Header</b>	<b>Parameter</b>
	:FETCh	
	:ACPower?	
	:CHANnel	
	:POWer?	
	:SPECTrum?	

**:FETCh:ACPower? (Query Only)**

Returns the Channel power and ACPR measurement results for all available channels.

**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.

**Measurement Views** Channel power and ACPR

**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.

**Related Commands** [:SENSe]:ACPower:CHANnel:PAIRs, [:SENSe]:POWer:UNITs



**:FETCh:ACPower:CHANnel:POWer? (Query Only)**

Returns the average power of the main channel (power reference) in the Channel power and ACPR measurement.

**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.

**Measurement Views** Channel power and ACPR

**Examples** :FETCh:ACPower:CHANnel:POWer?  
might return 4.227 indicating that the average power of the main channel is 4.227 dBm.

**Related Commands** [:SENSe]:POWer:UNITs

**:FETCh:ACPower:SPECTrum? (Query Only)**

Returns spectrum trace data of the Channel power and ACPR measurement.

**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.

**Measurement Views** Channel power and ACPR

**Examples** :FETCh:ACPower:SPECTrum?  
might return #43204xxxx... (3204-byte data) for the spectrum trace data of the Channel power and ACPR measurement.

**Related Commands** [:SENSE]:POWER:UNITs

## :FETCh:AVTime Subgroup

The :FETCh:AVTime commands returns the results of the Amplitude versus Time measurement.

Command Tree	Header	Parameter
	:FETCh	
	:AVTime	
	:AVERage?	
	:FIRSt :SECond :THIRd :FOURth	
	:MAXimum?	
	:MAXLocation?	
	:MINimum?	
	:MINLocation?	
	:RESult?	

---

**NOTE.** Except for the :FETCh:AVTime:FIRSt|:SECond|:THIRd|:FOURth command, select the trace using the :TRACe<x>:AVTime:SElect command.

---

## :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.

<b>Syntax</b>	:FETCh:AVTime:AVERage?
<b>Arguments</b>	None
<b>Returns</b>	<avg>::=<NRf> is the RMS amplitude in dBm. The unit can be changed by the [:SENSe]:POWer:UNITs command.
<b>Measurement Views</b>	Amplitude versus Time
<b>Examples</b>	:FETCh:AVTime:AVERage? might return -2.53 indicating the RMS amplitude is -2.53 dBm.
<b>Related Commands</b>	:TRACe<x>:AVTime:SElect, [:SENSe]:POWer:UNITs

**:FETCh:AVTime:FIRSt|:SECond|:THIRd|:FOURth? (Query Only)**

Returns the trace data in the Amplitude versus Time measurement.

**Syntax** :FETCh:AVTime:FIRSt|:SECond|:THIRd|:FOURth?

Where FIRSt, SECond, THIRd, and FOURth represent Trace 1 to 4, respectively. Each trace is defined as follows:

FIRSt: Trace 1  
SECond: Trace 2  
THIRd: Trace 3  
FOURth: Math trace

The traces can be specified by the :TRACe<x>:AVTime 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.

**Measurement Views** Amplitude versus Time

**Examples** :FETCh:AVTime:FIRSt?  
might return #3156xxxx... (156-byte data) for Trace 1.

**Related Commands** :TRACe<x>:AVTime command subgroup, [:SENSE]:POWER:UNITs

**: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.

**Syntax** :FETCh:AVTime:MAXimum?

**Arguments** None

**Returns** <max>::=<Nrf> is the maximum Amplitude in dBm.  
The unit can be changed by the [:SENSE]:POWer:UNITs command.

**Measurement Views** Amplitude versus Time

**Examples** :FETCh:AVTime:MAXimum?  
might return -2.84 indicating the maximum amplitude is -2.84 dBm.

**Related Commands** :FETCh:AVTime:MAXLocation?, :TRACe<x>:AVTime:SElect,  
[:SENSE]:POWer:UNITs

**: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.

**Syntax** :FETCh:AVTime:MAXLocation?

**Arguments** None

**Returns** <max\_time>::=<Nrf> is the time at the maximum in seconds.

**Measurement Views** Amplitude versus Time

**Examples** :FETCh:AVTime:MAXLocation?  
might return 25.03E-9 indicating the amplitude is maximum at 25.03 ns.

**Related Commands** :FETCh:AVTime:MAX?, :TRACe<x>:AVTime:SElect

**: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.

**Syntax** :FETCh:AVTime:MINimum?

**Arguments** None

**Returns** <min>::=<NRf> is the minimum amplitude in dBm.  
The unit can be changed by the [:SENSE]:POWER:UNITs command.

**Measurement Views** Amplitude versus Time

**Examples** :FETCh:AVTime:MINimum?  
might return -57.64 indicating the minimum amplitude is -57.64 dBm.

**Related Commands** :FETCh:AVTime:MINLocation?, :TRACe<x>:AVTime:SElect,  
[:SENSE]:POWER:UNITs

**: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.

**Syntax** :FETCh:AVTime:MINLocation?

**Arguments** None

**Returns** <min\_time>::=<NRf> is the time at the minimum in seconds.

**Measurement Views** Amplitude versus Time

**Examples** :FETCh:AVTime:MINLocation?  
might return 450.7E-9 indicating the amplitude is minimum at 450.7 ns.

**Related Commands** :FETCh:AVTime:MIN?, :TRACe<x>:AVTime:SElect

## :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.

**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.

**Measurement Views** Amplitude versus Time

**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  $\mu$ s, the minimum amplitude is -82.47 dBm at 22.11  $\mu$ s, and the RMS amplitude is -8.24 dBm

**Related Commands** :TRACe:AVTime:SElect, [:SENSE]:POWER:UNITs

## **:FETCh:CCDF Subgroup**

The :FETCh:CCDF commands returns the results of the CCDF (Complementary Cumulative Distribution Function) measurement.

<b>Command Tree</b>	<b>Header</b>	<b>Parameter</b>
	:FETCh	
	:CCDF?	
	:FIRSt   :SECond   :THIRd	
	:X?	
	:XY?	
	[:Y]?	



**:FETCh:CCDF? (Query Only)**

Returns the CCDF measurement results.

**Syntax** :FETCh:CCDF?

**Arguments** None

**Returns** <avg\_amp1>,<avg\_ccdf>,<crest\_factor>,<amp1\_10>,<amp1\_1>,<amp1\_p1>,<amp1\_p01>,<amp1\_p001>,<amp1\_p0001>

Where

<avg\_amp1> 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.

<amp1\_10> is the amplitude at CCDF of 10% in dB.

<amp1\_1> is the amplitude at CCDF of 1% in dB.

<amp1\_p1> is the amplitude at CCDF of 0.1% in dB.

<amp1\_p01> is the amplitude at CCDF of 0.01% in dB.

<amp1\_p001> is the amplitude at CCDF of 0.001% in dB.

<amp1\_p0001> is the amplitude at CCDF of 0.0001% in dB.

**Measurement Views** CCDF

**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.

**Related Commands** [:SENSe]:POWer:UNITs

**:FETCh:CCDF:FIRSt|:SECond|:THIRd:X? (Query Only)**

Returns the horizontal values of the specified trace in the CCDF measurement.

**Syntax** :FETCh:CCDF:FIRSt|:SECond|:THIRd:X?

Where FIRSt, SECond, and THIRd represent Trace 1 to 3, respectively.  
Each trace is defined as follows:

FIRSt: Trace 1  
SECond: Trace 2  
THIRd: Gaussian reference curve

**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.

**Measurement Views** CCDF

**Examples** :FETCh:CCDF:FIRSt:X?  
might return #41024xxxx... (1024-byte data) for the horizontal values of  
Trace 1.

**Related Commands** :FETCh:CCDF:FIRSt|:SECond|:THIRd[:Y]?

**:FETCh:CCDF:FIRSt|:SECond|:THIRd:XY? (Query Only)**

Returns the horizontal and vertical value pairs of the specified trace in the CCDF measurement.

**Syntax** :FETCh:CCDF:FIRSt|:SECond|:THIRd:XY?

Where FIRSt, SECond, and THIRd represent Trace 1 to 3, respectively. Each trace is defined as follows:

FIRSt: Trace 1  
SECond: Trace 2  
THIRd: Gaussian reference curve

**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.

**Measurement Views** CCDF

**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[:Y]? (Query Only)**

Returns the vertical values of the specified trace in the CCDF measurement.

**Syntax** :FETCh:CCDF:FIRSt|:SECond|:THIRd[:Y]?

Where FIRSt, SECond, and THIRd represent Trace 1 to 3, respectively.  
Each trace is defined as follows:

FIRSt: Trace 1  
SECond: Trace 2  
THIRd: Gaussian reference curve

**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.

**Measurement Views** CCDF

**Examples** :FETCh:CCDF:FIRSt:Y?  
might return #41024xxxx... (1024-byte data) for the vertical values of Trace 1.

**Related Commands** :FETCh:CCDF:FIRSt|:SECond|:THIRd:X?

## :FETCh:CONStE Subgroup (Option 21 Only)

The :FETCh:CONStE commands returns the results of the Constellation measurement.

Command Tree	Header	Parameter
	:FETCh	
	:CONStE	
	:RESuLts?	
	:TRACe?	

## :FETCh:CONStE:RESuLts? (Query Only)

Returns the constellation measurement results of EVM RMS, peak and location displayed on the bottom of the screen.

**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.

**Measurement Views**   Constellation

**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.

**Related Commands**   [:SENSE]:DDEMod:TIME:UNITs

**:FETCh:CONStE:TRACe? (Query Only)**

Returns the constellation trace data.

**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<sup>th</sup> data point. 4-byte little endian floating-point format specified in IEEE 488.2.

**Measurement Views** Constellation

**Examples** :FETCh:CONStE:TRACe?  
might return #43848xxxx... (3848-byte data) for the constellation trace data.

## :FETCh:DDEMod Subgroup (Option 21 Only)

The :FETCh:DDEMod commands returns the results of the General Purpose Digital Modulation measurements.

Command Tree	Header	Parameter
	:FETCh	
	:DDEMod	
	:STABle?	
	:SYNCh	
	:WORD	
	:LENGth?	
	:POSition?	

## :FETCh:DDEMod:STABle? (Query Only)

Returns the symbol table data.

**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.

**Measurement Views** Symbol table

**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.

**Syntax** :FETCh:DDEMod:SYNCh:WORD:LENGth?

**Arguments** None

**Returns** <NR1> indicates the length of the synch word in symbols.

**Measurement Views** Symbol table

**Examples** :FETCh:DDEMod:SYNCh:WORD:LENGth?  
might return 3, indicating the length of the synch word is three symbols.

**Related Commands** :FETCh:DDEMod:SYNCh:WORD:POSition?

**:FETCh:DDEMod:SYNCh:WORD:POSition? (Query Only)**

Returns the position of the synch word in the symbol table.

**Syntax** :FETCh:DDEMod:SYNCh:WORD:POSition?

**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.

**Measurement Views** Symbol table

**Examples** :FETCh:DDEMod:SYNCh:WORD:POSition?  
might return 10, indicating the synch word begins at 11<sup>th</sup> symbol in the table.

**Related Commands** :FETCh:DDEMod:SYNCh:WORD:LENGth?



## :FETCh:DPSA Subgroup

The :FETCh:DPSA commands returns the results of the DPX (Digital Phosphor) Spectrum measurement.

Command Tree	Header	Parameter
	:FETCh	
	:DPSA	
	:TRACe	
	:AVERAge?	
	:MATH?	
	:MAXimum?	
	:MINimum?	

### :FETCh:DPSA:TRACe:AVERAge? (Query Only)

Returns waveform data of the average trace in the DPX spectrum measurement.

**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.

**Measurement Views** DPX spectrum

**Examples** :FETCh:DPSA:TRACe:AVERAge?  
might return #42004xxxx... (2004-byte data) for the waveform data of the average trace.

**Related Commands** [:SENSe]:POWer:UNITs

**:FETCh:DPSA:TRACe:MATH? (Query Only)**

Returns waveform data of the math trace in the DPX spectrum measurement.

**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.

**Measurement Views** DPX spectrum

**Examples** :FETCh:DPSA:TRACe:MATH?  
might return #42004xxxx... (2004-byte data) for the waveform data of the math trace.

**Related Commands** [:SENSe]:POWer:UNITs

**:FETCh:DPSA:TRACe:MAXimum? (Query Only)**

Returns waveform data of the maximum trace in the DPX spectrum measurement.

**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.

**Measurement Views** DPX spectrum

**Examples** :FETCh:DPSA:TRACe:MAXimum?  
might return #42004xxxx... (2004-byte data) for the waveform data of the maximum trace.

**Related Commands** [:SENSe]:POWer:UNITs

**:FETCh:DPSA:TRACe:MINimum? (Query Only)**

Returns waveform data of the minimum trace in the DPX spectrum measurement.

**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.

**Measurement Views** DPX spectrum

**Examples** :FETCh:DPSA:TRACe:MINimum?  
might return #42004xxxx... (2004-byte data) for the waveform data of the minimum trace.

**Related Commands** [:SENSe]:POWer:UNITs

## :FETCh:EVM Subgroup (Option 21 Only)

The :FETCh:EVM commands returns the results of the EVM (Error Vector Magnitude) versus Time measurement.

Command Tree	Header	Parameter
	:FETCh	
	:EVM	
	:FERRor?	
	:PEAK?	
	:PINDeX?	
	:RMS?	
	:TRACe?	

## :FETCh:EVM:FERRor? (Query Only)

Returns the frequency error in the EVM versus Time measurement.

**Syntax** :FETCh:EVM:FERRor?

**Arguments** None

**Returns** <freq\_error>::=<NRf> is the frequency error in Hz.

**Measurement Views** EVM versus Time

**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.

**Syntax** :FETCh:EVM:PEAK?

**Arguments** None

**Returns** <peak>::=<NRf> is the peak EVM value in percent (%).

**Measurement Views** EVM versus Time

**Examples** :FETCh:EVM:PEAK?  
might return 1.32 indicating the peak EVM value is 1.32%.

**Related Commands** :FETCh:EVM:PINDEX?

## **:FETCh:EVM:PINDEX? (Query Only)**

Returns the time at the EVM peak.

**Syntax** :FETCh:EVM:PINDEX?

**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.

**Measurement Views** EVM versus Time

**Examples** :FETCh:EVM:PINDEX?  
might return 68.000, indicating that the EVM peak is at symbol #68.

**Related Commands** :FETCh:EVM:PEAK?, [:SENSE]:DDEMod:TIME:UNITS

**:FETCh:EVM:RMS? (Query Only)**

Returns the RMS (Root-Mean-Square) value in the EVM versus Time measurement.

**Syntax** :FETCh:EVM:RMS?

**Arguments** None

**Returns** <rms>::=<NRf> is the RMS EVM value in percent (%).

**Measurement Views** EVM versus Time

**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.

**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.

**Measurement Views** EVM versus Time

**Examples** :FETCh:EVM:TRACe?  
might return #42036xxxx... (2036-byte data) for the EVM versus Time trace.

## :FETCh:FVTime Subgroup

The :FETCh:EVM commands returns the results of the Frequency versus Time measurement.

Command Tree	Header	Parameter
	:FETCh	
	:FVTime?	
	:MAXimum?	
	:MAXLocation?	
	:MINimum?	
	:MINLocation?	
	:RESult?	

## :FETCh:FVTime? (Query Only)

Returns the Frequency versus Time trace data.

**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.

**Measurement Views**   Frequency versus Time

**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.

**Syntax** :FETCh:FVTime:MAXimum?

**Arguments** None

**Returns** <max>::=<NRf> is the maximum frequency drift in Hz.

**Measurement Views** Frequency versus Time

**Examples** :FETCh:FVTime:MAXimum?  
might return 2.625E+6 indicating the maximum frequency drift is 2.625 MHz.

**Related Commands** :FETCh:FVTime:MAXLocation?

**:FETCh:FVTime:MAXLocation? (Query Only)**

Returns the time at which the frequency drift is maximum.

**Syntax** :FETCh:FVTime:MAXLocation?

**Arguments** None

**Returns** <max\_time>::=<NRf> is the time in seconds at which the frequency drift is maximum.

**Measurement Views** Frequency versus Time

**Examples** :FETCh:FVTime:MAXLocation?  
might return 25.03E-9 indicating the frequency drift is maximum at 25.03 ns.

**Related Commands** :FETCh:FVTime:MAXimum?

**:FETCh:FVTime:MINimum? (Query Only)**

Returns the minimum value in the Frequency versus Time measurement.

**Syntax** :FETCh:FVTime:MINimum?

**Arguments** None

**Returns** <min>::=<NRf> is the minimum frequency drift in Hz.

**Measurement Views** Frequency versus Time

**Examples** :FETCh:FVTime:MINimum?  
might return -6.618E+6 indicating the minimum frequency drift is -6.618 MHz.

**Related Commands** :FETCh:FVTime:MINLocation?

**:FETCh:FVTime:MINLocation? (Query Only)**

Returns the time at which the frequency drift is minimum.

**Syntax** :FETCh:FVTime:MINLocation?

**Arguments** None

**Returns** <min\_time>::=<NRf> is the time in seconds at which the frequency drift is minimum.

**Measurement Views** Frequency versus Time

**Examples** :FETCh:FVTime:MAXLocation?  
might return 450.7E-9 indicating the frequency drift is minimum at 450.7 ns.

**Related Commands** :FETCh:FVTime:MINimum?

## :FETCh:FVTime:RESult? (Query Only)

Returns the Frequency versus Time measurement results.

**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.

**Measurement Views** Frequency versus Time

**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 Subgroup

The :FETCh:IQVTime commands returns the results of the RF I&Q versus Time measurement.

Command Tree	Header	Parameter
	:FETCh	
	:IQVTime	
	:I?	
	:MAXimum?	
	:MAXLocation?	
	:MINimum?	
	:MINLocation?	
	:Q?	
	:RESult?	

### :FETCh:IQVTime:I? (Query Only)

Returns the I versus Time trace data.

**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.

**Measurement Views**   RF I&Q versus Time

**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.

**Syntax** :FETCh:IQVTime:MAXimum?

**Arguments** None

**Returns** <max>::=<NRf> is the maximum I or Q level in volts.  
Use the [:SENSE]:IQVTime:TRACe:SElect:I or Q command to select the trace.

**Measurement Views** RF I&Q versus Time

**Examples** :FETCh:IQVTime:MAXimum?  
might return 1.214 indicating the maximum I or Q level is 1.214 V.

**Related Commands** :FETCh:IQVTime:MAXLocation?,  
[:SENSE]:IQVTime:TRACe:SElect:I, [:SENSE]:IQVTime:TRACe:SElect:Q

**:FETCh:IQVTime:MAXLocation? (Query Only)**

Returns the time at which the I or Q level is maximum.

**Syntax** :FETCh:IQVTime:MAXLocation?

**Arguments** None

**Returns** <max\_time>::=<NRf> is the time in seconds at which the I or Q level is maximum.

**Measurement Views** RF I&Q versus Time

**Examples** :FETCh:IQVTime:MAXLocation?  
might return 175.3E-9 indicating the I or Q level is maximum at 175.3 ns.

**Related Commands** :FETCh:IQVTime:MAXimum?

**:FETCh:IQVTime:MINimum? (Query Only)**

Returns the minimum value in the RF I&Q versus Time measurement.

**Syntax** :FETCh:IQVTime:MINimum?

**Arguments** None

**Returns** <min>::=<Nrf> is the minimum I or Q level in volts.  
Use the [:SENSe]:IQVTime:TRACe:SElect:I or Q command to select the trace.

**Measurement Views** RF I&Q versus Time

**Examples** :FETCh:IQVTime:MINimum?  
might return  $-370.5E-3$  indicating the minimum I or Q level is  $-370.5$  mV.

**Related Commands** :FETCh:IQVTime:MINLocation?,  
[:SENSe]:IQVTime:TRACe:SElect:I, [:SENSe]:IQVTime:TRACe:SElect:Q

**:FETCh:IQVTime:MINLocation? (Query Only)**

Returns the time at which the I or Q level is minimum.

**Syntax** :FETCh:IQVTime:MINLocation?

**Arguments** None

**Returns** <min\_time>::=<Nrf> is the time in seconds at which the I or Q level is minimum.

**Measurement Views** RF I&Q versus Time

**Examples** :FETCh:IQVTime:MAXLocation?  
might return  $450.7E-9$  indicating the I or Q level is minimum at  $450.7$  ns.

**Related Commands** :FETCh:IQVTime:MINimum?

## :FETCh:IQVTime:Q? (Query Only)

Returns the Q versus Time trace data.

**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.

**Measurement Views** RF I&Q versus Time

**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.

**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 [:SENSe]:IQVTime:TRACe:SElect:I or Q command to select the trace.

**Measurement Views** RF I&Q versus Time

**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.

**Related Commands** [:SENSe]:IQVTime:TRACe:SElect:I, [:SENSe]:IQVTime:TRACe:SElect:Q



## :FETCh:MCPower Subgroup

The :FETCh:MCPower commands returns the results of the MCPR (Multiple Carrier Power Ratio) measurement.

Command Tree	Header	Parameter
	:FETCh	
	:MCPower	
	:ADJacent	
	:CHANnels?	
	:CHANnel	
	:POWER?	
	:MAIN	
	:CHANnels?	
	:SPECTrum?	

**:FETCh:MCPower:ADJacent:CHANnels? (Query Only)**

Returns the power of adjacent channels in order of increasing frequency.

**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:ADD:ADJacent command.

**Measurement Views** MCPR

**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.

**Related Commands** [:SENSe]:MCPower:CHANnel:ADD:ADJacent

**:FETCh:MCPower:CHANnel:POWer? (Query Only)**

Returns the reference power in the MCPR measurement.

**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.

**Measurement Views** MCPR

**Examples** :FETCh:MCPower:CHANnel:POWer?  
might return 4.227 indicating that the reference power is 4.227 dBm.

**Related Commands** [:SENSe]:MCPower:RCHannels commands, [:SENSe]:POWer:UNITs

**:FETCh:MCPower:MAIN:CHANnels? (Query Only)**

Returns the power of main channels in order of increasing frequency.

**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.

**Measurement Views** MCPR

**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.

**Related Commands** [:SENSe]:MCPower:CHANnel:MAIN commands, [:SENSe]:POWer:UNITs

## :FETCh:MCPower:SPECTrum? (Query Only)

Returns spectrum trace data of the MCPR measurement.

**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.

**Measurement Views** MCPR

**Examples** :FETCh:MCPower:SPECTrum?  
might return #43204xxxx... (3204-byte data) for the spectrum trace data of the MCPR measurement.

**Related Commands** [:SENSE]:POWER:UNITs

## :FETCh:MERRor Subgroup (Option 21 Only)

The :FETCh:MERRor commands returns the results of the Magnitude error versus Time measurement.

Command Tree	Header	Parameter
	:FETCh	
	:MERRor	
	:FERRor?	
	:PEAK?	
	:PINDEX?	
	:RMS?	
	:TRACe?	

## :FETCh:MERRor:FERRor? (Query Only)

Returns the frequency error in the Magnitude error versus Time measurement.

**Syntax**     :FETCh:MERRor:FERRor?

**Arguments**   None

**Returns**     <freq\_error>::=<NRf> is the frequency error in Hz.

**Measurement Views**   Magnitude error versus Time

**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.

**Syntax** :FETCh:MERRor:PEAK?

**Arguments** None

**Returns** <peak>::=<NRf> is the peak magnitude error in percent (%).

**Measurement Views** Magnitude error versus Time

**Examples** :FETCh:MERRor:PEAK?  
might return 1.57 indicating the peak magnitude error is 1.57%.

**Related Commands** :FETCh:MERRor:PINDeX?

**:FETCh:MERRor:PINDeX? (Query Only)**

Returns the time at the magnitude error peak.

**Syntax** :FETCh:MERRor:PINDeX?

**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.

**Measurement Views** Magnitude error versus Time

**Examples** :FETCh:MERRor:PINDeX?  
might return 68.000, indicating that the magnitude error peak is at symbol #68.

**Related Commands** :FETCh:MERRor:PEAK?, [:SENSe]:DDEMod:TIME:UNITs

**:FETCh:MERRor:RMS? (Query Only)**

Returns the RMS (Root-Mean-Square) value in the Magnitude error versus Time measurement.

**Syntax** :FETCh:MERRor:RMS?

**Arguments** None

**Returns** <rms>::=<NRf> is the RMS magnitude error in percent (%).

**Measurement Views** Magnitude error versus Time

**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.

**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.

**Measurement Views** Magnitude error versus Time

**Examples** :FETCh:MERRor:TRACe?  
might return #42036xxxx... (2036-byte data) for the Magnitude error versus Time trace.



## :FETCh:OBWidth Subgroup

The :FETCh:OBWidth commands returns the results of the Occupied Bandwidth (OBW) measurement.

Command Tree	Header	Parameter
	:FETCh	
	:OBWidth	
	:FREQuency	
	:ERRor?	
	:OBWidth	
	:BANDwidth?	
	:LEFT	
	:FREQuency?	
	:LEVEl?	
	:POWer?	
	:RIGHT	
	:FREQuency?	
	:LEVEl?	
	:SPECTrum?	
	:XDBBbandwidth	
	:BANDwidth?	
	:LEFT	
	:FREQuency?	
	:LEVEl?	
	:POWer?	
	:RIGHT	
	:FREQuency?	
	:LEVEl?	

**:FETCh:OBWidth:FREQuency:ERRor? (Query Only)**

Returns the frequency error in the Occupied Bandwidth measurement.

**Syntax** :FETCh:OBWidth:FREQuency:ERRor?

**Arguments** None

**Returns** <freq\_error>::=<NRf> is the frequency error in Hz.

**Measurement Views** Occupied Bandwidth

**Examples** :FETCh:MERRor:FERRor?  
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.

**Syntax** :FETCh:OBWidth:OBWidth:BANDwidth?

**Arguments** None

**Returns** <OBW>::=<NRf> is the occupied bandwidth in Hz.

**Measurement Views** Occupied Bandwidth

**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.

**Syntax** :FETCh:OBWidth:OBWidth:LEFT:FREQuency?

**Arguments** None

**Returns** <OB\_left\_freq>::=<NRf> is the left frequency in Hz.

**Measurement Views** Occupied Bandwidth

**Examples** :FETCh:OBWidth:OBWidth:LEFT:FREQuency?  
might return 1.498E+9 indicating the left frequency is 1.498 GHz.

**Related Commands** :FETCh:OBWidth:OBWidth:RIGHT:FREQuency?

**:FETCh:OBWidth:OBWidth:LEFT:LEVel? (Query Only)**

Returns the level at the left frequency of the occupied bandwidth.

**Syntax** :FETCh:OBWidth:OBWidth:LEFT:LEVel?

**Arguments** None

**Returns** <OB\_left\_level>::=<NRf> is the level at the left frequency in dB.

**Measurement Views** Occupied Bandwidth

**Examples** :FETCh:OBWidth:OBWidth:LEFT:LEVel?  
might return -23.5 indicating the level at the left frequency is -23.5 dB.

**Related Commands** :FETCh:OBWidth:OBWidth:RIGHT:LEVel?

**:FETCh:OBWidth:OBWidth:POWer? (Query Only)**

Returns the reference power in the Occupied Bandwidth measurement.

**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.

**Measurement Views** Occupied Bandwidth

**Examples** :FETCh:OBWidth:OBWidth:POWer?  
might return -10.0 indicating the reference power is -10 dBm.

**Related Commands** [:SENSE]:POWer:UNITs

**:FETCh:OBWidth:OBWidth:RIGHT:FREQUENCY? (Query Only)**

Returns the right (higher) frequency of the occupied bandwidth.

**Syntax** :FETCh:OBWidth:OBWidth:RIGHT:FREQUENCY?

**Arguments** None

**Returns** <OB\_right\_freq>::=<NRf> is the right frequency in Hz.

**Measurement Views** Occupied Bandwidth

**Examples** :FETCh:OBWidth:OBWidth:RIGHT:FREQUENCY?  
might return 1.502E+9 indicating the right frequency is 1.502 GHz.

**Related Commands** :FETCh:OBWidth:OBWidth:LEFT:FREQUENCY?

**:FETCh:OBWidth:OBWidth:RIGHT:LEVEL? (Query Only)**

Returns the level at the right frequency of the occupied bandwidth.

**Syntax** :FETCh:OBWidth:OBWidth:RIGHT:LEVEL?

**Arguments** None

**Returns** <OB\_right\_level>::=<NRf> is the level at the right frequency in dB.

**Measurement Views** Occupied Bandwidth

**Examples** :FETCh:OBWidth:OBWidth:RIGHT:LEVEL?  
might return -23.5 indicating the level at the right frequency is -23.5 dB.

**Related Commands** :FETCh:OBWidth:OBWidth:LEFT:LEVEL?

**:FETCh:OBWidth:SPECTrum? (Query Only)**

Returns spectrum trace data of the Occupied Bandwidth measurement.

**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.

**Measurement Views** Occupied Bandwidth

**Examples** :FETCh:OBWidth:SPECTrum?  
might return #43204xxxx... (3204-byte data) for the spectrum trace data of the Occupied Bandwidth measurement.

**Related Commands** [:SENSE]:POWER:UNITs

**:FETCh:OBWidth:XDBBandwidth:BANDwidth? (Query Only)**

Returns the x dB bandwidth in the Occupied Bandwidth measurement.

**Syntax** :FETCh:OBWidth:XDBBandwidth:BANDwidth?

**Arguments** None

**Returns** <xdBW>::=<NRf> is the x dB bandwidth in Hz.

**Measurement Views** Occupied Bandwidth

**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.

**Syntax** :FETCh:OBWidth:XDBBandwidth:LEFT:FREQUENCY?

**Arguments** None

**Returns** <xdbBW\_left\_freq>::=<NRf> is the left frequency in Hz.

**Measurement Views** Occupied Bandwidth

**Examples** :FETCh:OBWidth:XDBBandwidth:LEFT:FREQUENCY?  
might return 1.498E+9 indicating the left frequency is 1.498 GHz.

**Related Commands** :FETCh:OBWidth:XDBBandwidth:RIGHT:FREQUENCY?

**:FETCh:OBWidth:XDBBandwidth:LEFT:LEVEL? (Query Only)**

Returns the level at the left frequency of the x dB bandwidth.

**Syntax** :FETCh:OBWidth:XDBBandwidth:LEFT:LEVEL?

**Arguments** None

**Returns** <xdbBW\_left\_level>::=<NRf> is the level at the left frequency in dB.

**Measurement Views** Occupied Bandwidth

**Examples** :FETCh:OBWidth:XDBBandwidth:LEFT:LEVEL?  
might return -23.5 indicating the level at the left frequency is -23.5 dB.

**Related Commands** :FETCh:OBWidth:XDBBandwidth:RIGHT:LEVEL?

**:FETCh:OBWidth:XDBBandwidth:POWer? (Query Only)**

Returns the reference power in the x dB bandwidth measurement.

**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.

**Measurement Views** Occupied Bandwidth

**Examples** :FETCh:OBWidth:XDBBandwidth:POWer?  
might return -10.0 indicating the reference power is -10 dBm.

**Related Commands** [:SENSE]:POWer:UNITs



**:FETCh:OBWidth:XDBBandwidth:RIGHT:FREQUENCY? (Query Only)**

Returns the right (higher) frequency of the x dB bandwidth.

**Syntax** :FETCh:OBWidth:XDBBandwidth:RIGHT:FREQUENCY?

**Arguments** None

**Returns** <xdbBW\_right\_freq>::=<NRf> is the right frequency in Hz.

**Measurement Views** Occupied Bandwidth

**Examples** :FETCh:OBWidth:XDBBandwidth:RIGHT:FREQUENCY?  
might return 1.502E+9 indicating the right frequency is 1.502 GHz.

**Related Commands** :FETCh:OBWidth:XDBBandwidth:LEFT:FREQUENCY?

**:FETCh:OBWidth:XDBBandwidth:RIGHT:LEVEL? (Query Only)**

Returns the level at the right frequency of the x dB bandwidth.

**Syntax** :FETCh:OBWidth:XDBBandwidth:RIGHT:LEVEL?

**Arguments** None

**Returns** <xdbBW\_right\_level>::=<NRf> is the level at the right frequency in dB.

**Measurement Views** Occupied Bandwidth

**Examples** :FETCh:OBWidth:XDBBandwidth:RIGHT:LEVEL?  
might return -23.5 indicating the level at the right frequency is -23.5 dB.

**Related Commands** :FETCh:OBWidth:XDBBandwidth:LEFT:LEVEL?

## :FETCh:PERRor Subgroup (Option 21 Only)

The :FETCh:PERRor commands returns the results of the Phase error versus Time measurement.

Command Tree	Header	Parameter
	:FETCh	
	:PERRor	
	:FERRor?	
	:PEAK?	
	:PINDEX?	
	:RMS?	
	:TRACe?	

## :FETCh:PERRor:FERRor? (Query Only)

Returns the frequency error in the Phase error versus Time measurement.

**Syntax**     :FETCh:PERRor:FERRor?

**Arguments**   None

**Returns**     <freq\_error>::=<NRf> is the frequency error in Hz.

**Measurement Views**   Phase error versus Time

**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.

**Syntax** :FETCh:PERRor:PEAK?

**Arguments** None

**Returns** <peak>::=<NRf> is the peak phase error in degrees.

**Measurement Views** Phase error versus Time

**Examples** :FETCh:PERRor:PEAK?  
might return 0.683 indicating the peak phase error is 0.683°.

**Related Commands** :FETCh:PERRor:PINDex?

**:FETCh:PERRor:PINDex? (Query Only)**

Returns the time at the phase error peak.

**Syntax** :FETCh:PERRor:PINDex?

**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.

**Measurement Views** Phase error versus Time

**Examples** :FETCh:PERRor:PINDex?  
might return 68.000, indicating that the phase error peak is at symbol #68.

**Related Commands** :FETCh:PERRor:PEAK?, [:SENSe]:DDEMod:TIME:UNITs

**:FETCh:PERRor:RMS? (Query Only)**

Returns the RMS (Root-Mean-Square) value in the Phase error versus Time measurement.

**Syntax** :FETCh:PERRor:RMS?

**Arguments** None

**Returns** <rms>::=<NRf> is the RMS phase error in degrees.

**Measurement Views** Phase error versus Time

**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.

**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.

**Measurement Views** Phase error versus Time

**Examples** :FETCh:PERRor:TRACe?  
might return #42036xxxx... (2036-byte data) for the Phase error versus Time trace.

## :FETCh:PHVTime Subgroup

The :FETCh:PHVTime commands returns the results of the Phase versus Time measurement.

Command Tree	Header	Parameter
	:FETCh	
	:PHVTime?	
	:MAXimum?	
	:MAXLocation?	
	:MINimum?	
	:MINLocation?	
	:RESult?	

## :FETCh:PHVTime? (Query Only)

Returns the Phase versus Time trace data.

**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.

**Measurement Views** Phase versus Time

**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.

**Syntax** :FETCh:PHVTime:MAXimum?

**Arguments** None

**Returns** <max>::=<NRf> is the maximum phase in degrees.

**Measurement Views** Phase versus Time

**Examples** :FETCh:PHVTime:MAXimum?  
might return 153.8 indicating the maximum phase is 153.8°.

**Related Commands** :FETCh:PHVTime:MAXLocation?

**:FETCh:PHVTime:MAXLocation? (Query Only)**

Returns the time at which the phase is maximum.

**Syntax** :FETCh:PHVTime:MAXLocation?

**Arguments** None

**Returns** <max\_time>::=<NRf> is the time in seconds at which the phase is maximum.

**Measurement Views** Phase versus Time

**Examples** :FETCh:PHVTime:MAXLocation?  
might return 175.3E-9 indicating the I or Q level is maximum at 175.3 ns.

**Related Commands** :FETCh:PHVTime:MAXimum?

## :FETCh:PHVTime:MINimum? (Query Only)

Returns the minimum value in the Phase versus Time measurement.

**Syntax** :FETCh:PHVTime:MINimum?

**Arguments** None

**Returns** <min>::=<NRf> is the minimum phase in degrees.

**Measurement Views** Phase versus Time

**Examples** :FETCh:PHVTime:MINimum?  
might return -176.3 indicating the minimum phase is -176.3°.

**Related Commands** :FETCh:PHVTime:MINLocation?

## :FETCh:PHVTime:MINLocation? (Query Only)

Returns the time at which the phase is minimum.

**Syntax** :FETCh:PHVTime:MINLocation?

**Arguments** None

**Returns** <min\_time>::=<NRf> is the time in seconds at which the phase is minimum.

**Measurement Views** Phase versus Time

**Examples** :FETCh:PHVTime:MAXLocation?  
might return 450.7E-9 indicating the phase is minimum at 450.7 ns.

**Related Commands** :FETCh:PHVTime:MINimum?

**:FETCh:PHVTime:RESult? (Query Only)**

Returns the Phase versus Time measurement results.

**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.

**Measurement Views** Phase versus Time

**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 Subgroup (Option 11 Only)

The :FETCh:PNOise commands returns results of the Phase Noise measurement.

Command Tree	Header	Parameter
	:FETCh	
	:PNOise	
	:ALL?	
	:CARRier	
	:FERRor?	
	:POWer?	
	:JITTer?	
	:RESidual	
	:FM?	
	:RMS	
	:PNOise?	
	:SPECTrum<x>	
	:X?	
	:XY?	
	[:Y]?	

**:FETCh:PNOise:ALL? (Query Only)**

Returns all results of the phase noise measurement.

**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.

**Measurement Views** Phase noise

**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

Residual FM: 14.58 Hz

**:FETCh:PNOise:CARRier:FERRor? (Query Only)**

Returns the carrier frequency error in the phase noise measurement.

**Syntax** :FETCh:PNOise:CARRier:FERRor?

**Arguments** None

**Returns** <NRf> Carrier frequency error in Hz.

**Measurement Views** Phase noise

**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.

**Syntax** :FETCh:PNOise:CARRier:POWer?

**Arguments** None

**Returns** <NRf> Carrier power in dBm.  
The unit can be changed by the [:SENSe]:POWer:UNITs command.

**Measurement Views** Phase noise

**Examples** :FETCh:PNOise:CARRier:POWer?  
might return -9.455, indicating that the carrier power is -9.455 dBm.

**Related Commands** [:SENSe]:POWer:UNITs

**:FETCh:PNOise:JITTer? (Query Only)**

Returns the jitter in the phase noise measurement.

**Syntax** :FETCh:PNOise:JITTer?

**Arguments** None

**Returns** <NRf> Jitter in seconds.

**Measurement Views** Phase noise

**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.

**Syntax** :FETCh:PNOise:RESidual:FM?

**Arguments** None

**Returns** <NRf> Residual FM in Hz.

**Measurement Views** Phase noise

**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.

**Syntax** :FETCh:PNOise:RMS:PNOise?

**Arguments** None

**Returns** <NRf> RMS phase noise in degrees.

**Measurement Views** Phase noise

**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.

**Syntax** :FETCh:PNOise:SPECTrum<x>:X?

Where <x> = 1 and 2, representing Trace 1 and Trace 2, respectively.

**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.

**Measurement Views** Phase noise

**Examples** :FETCh:PNOise:SPECTrum1:X?  
might return #516020xxxx... (16020-byte data) for the frequencies of Trace 1.

**Related Commands** :FETCh:PNOise:SPECTrum<x>[:Y]?

## **:FETCh:PNOise:SPECTrum<x>:XY? (Query Only)**

Returns the frequency and phase noise pairs of the specified trace.

**Syntax** :FETCh:PNOise:SPECTrum<x>:XY?

Where <x> = 1 and 2, representing Trace 1 and Trace 2, respectively.

**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.

**Measurement Views** Phase noise

**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.

**Syntax** :FETCh:PNOise:SPECTrum<x>[:Y]?

Where <x> = 1 and 2, representing Trace 1 and Trace 2, respectively.

**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.

**Measurement Views** Phase noise

**Examples** :FETCh:PNOise:SPECTrum1:Y?

might return #516020xxxx... (16020-byte data) for the phase noise values of Trace 1.

**Related Commands** :FETCh:PNOise:SPECTrum<x>[:Y]?

## :FETCh:PULSe Subgroup (Option 20 Only)

The :FETCh:PULSe commands returns the results of the Pulse Measurements.

Command Tree	Header	Parameter
	:FETCh	
	:PULSe	
	[:RESult]	
	:ATX?	
	:AVERage?	
	:DROop?	
	:DUTPct?	
	:DUTRatio?	
	:FALL?	
	:FRDeviation?	
	:MFReqerror?	
	:MPHerror?	
	:PHDeviation?	
	:PPFRequency?	
	:PPOWer?	
	:PPPase?	
	:RINTerval?	
	:RIPPlE?	
	:RISE?	
	:RMSFregerror?	
	:RMSPherror?	
	:RRATE?	
	:TIME?	
	:WIDTh?	
	:STATistics?	
	:ATX?	
	:AVERage?	
	:DROop?	
	:DUTPct?	
	:DUTRatio?	
	:FALL?	
	:FRDeviation?	
	:MFReqerror?	
	:MPHerror?	
	:PHDeviation?	
	:PPFRequency?	
	:PPOWer?	
	:PPPase?	
	:RINTerval?	
	:RIPPlE?	
	:RISE?	



```

:RMSFreqerror?
:RMSPherror?
:RRATe?
:WIDTh?
:TRACe
:X?
:XY?
[:Y]?

```

## :FETCh:PULSe[:RESUlt]:ATX? (Query Only)

Returns the average transmitted power in the pulse table.

**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.

**Measurement Views** Pulse table

**Examples** :FETCh:PULSe:RESUlt:ATX?  
might return 0,-18.57,-18.73,-18.20,-18.53 for Pulse 0 to 3.

**Related Commands** [:SENSe]:POWer:UNITs

**:FETCh:PULSe[:RESult]:AVERAge? (Query Only)**

Returns the average on power in the pulse table.

**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.

**Measurement Views** Pulse table

**Examples** :FETCh:PULSe:RESult:AVERAge?  
might return 0,-2.354,-2.368,-2.343,-2.358 for Pulse 0 to 3.

**Related Commands** [:SENSe]:POWer:UNITs

**:FETCh:PULSe[:RESUlt]:DROop? (Query Only)**

Returns the droop in the pulse table.

**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 (%).

**Measurement Views** Pulse table

**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.

**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 (%).

**Measurement Views** Pulse table

**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.

**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).

**Measurement Views** Pulse table

**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.

**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.

**Measurement Views** Pulse table

**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.

**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.

**Measurement Views** Pulse table

**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]:MFRReqerror? (Query Only)**

Returns the maximum frequency error in the pulse table.

**Syntax** :FETCh:PULSe[:RESult]:MFRReqerror?

**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.

**Measurement Views** Pulse table

**Examples** :FETCh:PULSe:RESult:MFRReqerror?  
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.

**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.

**Measurement Views** Pulse table

**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.

**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.

**Measurement Views** Pulse table

**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.

**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.

**Measurement Views** Pulse table

**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.

**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.

**Measurement Views** Pulse table

**Examples** :FETCh:PULSe:RESult:PPOWer?  
might return 0,-2.26,-2.27,-2.23,-2.25 for Pulse 0 to 3.

**Related Commands** [:SENSe]:POWer:UNITs

**:FETCh:PULSe[:RESult]:PPHase? (Query Only)**

Returns the pulse-pulse carrier phase in the pulse table.

**Syntax** :FETCh:PULSe[:RESult]:PPHase?

**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.

**Measurement Views** Pulse table

**Examples** :FETCh:PULSe:RESult:PPHase?  
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.

**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.

**Measurement Views** Pulse table

**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.

**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 (%).

**Measurement Views** Pulse table

**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.

**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.

**Measurement Views** Pulse table

**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.

**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.

**Measurement Views** Pulse table

**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.

**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.

**Measurement Views** Pulse table

**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.

**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.

**Measurement Views** Pulse table

**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.

**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.

**Measurement Views** Pulse table

**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.

**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.

**Measurement Views** Pulse table

**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.

---

**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.

**Measurement Views** Pulse statistics

**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.

**Related Commands** :DISPlay:PULSe:SElect:RESult, :DISPlay:PULSe:STATistics:PLOT

**: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.

**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.

**Measurement Views** Pulse statistics

**Examples** :FETCh:PULSe:STATistics:ATX?  
might return -18.51,-18.74,-18.12 for the average transmitted power in the pulse statistics.

**Related Commands** :DISPlay:PULSe:STATistics:PLOT, [:SENSe]:POWer:UNITs

## **: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.

**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 by the [:SENSe]:POWer:UNITs command.

**Measurement Views** Pulse statistics

**Examples** :FETCh:PULSe:STATistics:AVERage?  
might return -2.35, -2.36, -2.34 for the average on power in the pulse statistics.

**Related Commands** :DISPlay:PULSe:STATistics:PLOT, [:SENSe]:POWer:UNITs

**:FETCh:PULSe:STATistics:DR0op? (Query Only)**

Returns the droop in the pulse statistics.

This command is valid when :DISPlay:PULSe:STATistics:PLOT is set to TREND.

**Syntax** :FETCh:PULSe:STATistics:DR0op?

**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.

**Measurement Views** Pulse statistics

**Examples** :FETCh:PULSe:STATistics:DR0op?  
might return 22.67E-3, -613.5E-3, 633.8E-3 for the droop in the pulse statistics.

**Related Commands** :DISPlay:PULSe:STATistics:PLOT

## :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.

**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 (%).

**Measurement Views** Pulse statistics

**Examples** :FETCh:PULSe:STATistics:DUTPct?  
might return 2.437,2.310,2.657 for the duty factor (%) in the pulse statistics.

**Related Commands** :DISPlay:PULSe:STATistics:PLOT

**: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.

**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.

**Measurement Views** Pulse statistics

**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.

**Related Commands** :DISPlay:PULSe:STATistics:PLOT

## :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.

**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.

**Measurement Views** Pulse statistics

**Examples** :FETCh:PULSe:STATistics:FALL?  
might return 70.27E-9,69.62E-9,71.27E-9 for the fall time in the pulse statistics.

**Related Commands** :DISPlay:PULSe:STATistics:PLOT

**: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.

**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.

**Measurement Views** Pulse statistics

**Examples** :FETCh:PULSe:STATistics:FRDeviation?  
might return 754.1E+3,660.5E+3,835.2E+3 for the frequency deviation in the pulse statistics.

**Related Commands** :DISPlay:PULSe:STATistics:PLOT



## :FETCh:PULSe:STATistics:MFRReqerror? (Query Only)

Returns the maximum frequency error in the pulse statistics.

This command is valid when :DISPlay:PULSe:STATistics:PLOT is set to TREND.

**Syntax** :FETCh:PULSe:STATistics:MFRReqerror?

**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.

**Measurement Views** Pulse statistics

**Examples** :FETCh:PULSe:STATistics:MFRReqerror?  
might return 645.0E+3,555.6E+3,738.8E+3 for the maximum frequency error  
in the pulse statistics.

**Related Commands** :DISPlay:PULSe:STATistics:PLOT

**: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.

**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.

**Measurement Views** Pulse statistics

**Examples** :FETCh:PULSe:STATistics:MPHerror?  
might return -11.47,-17.18,-7.61 for the maximum phase error in the pulse statistics.

**Related Commands** :DISPlay:PULSe:STATistics:PLOT

## :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.

**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.

**Measurement Views** Pulse statistics

**Examples** :FETCh:PULSe:STATistics:PHDeviation?  
might return 11.678,7.694,17.374 for the phase deviation in the pulse statistics.

**Related Commands** :DISPlay:PULSe:STATistics:PLOT

**: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.

**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.

**Measurement Views** Pulse statistics

**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.

**Related Commands** :DISPlay:PULSe:STATistics:PLOT

**: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.

**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.

**Measurement Views** Pulse statistics

**Examples** :FETCh:PULSe:STATistics:PPOWer?  
might return -2.273,-2.313,-2.235 for the peak power in the pulse statistics.

**Related Commands** :DISPlay:PULSe:STATistics:PLOT, [:SENSe]:POWer:UNITs

**: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.

**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.

**Measurement Views** Pulse statistics

**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.

**Related Commands** :DISPlay:PULSe:STATistics:PLOT

## :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.

**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.

**Measurement Views** Pulse statistics

**Examples** :FETCh:PULSe:STATistics:RINTerval?  
might return 240.5E-6,217.9E-6,281.2E-6 for the repetition interval in the pulse statistics.

**Related Commands** :DISPlay:PULSe:STATistics:PLOT

**: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.

**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.

**Measurement Views** Pulse statistics

**Examples** :FETCh:PULSe:STATistics:RIPPlE?  
might return 160.4E-3,83.78E-3,287.7E-3 for the ripple in the pulse statistics.

**Related Commands** :DISPlay:PULSe:STATistics:PLOT



## :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.

**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.

**Measurement Views** Pulse statistics

**Examples** :FETCh:PULSe:STATistics:RISE?  
might return 105.4E-9,91.65E-9,116.2E-9 for the rise time in the pulse statistics.

**Related Commands** :DISPlay:PULSe:STATistics:PLOT

**: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.

**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.

**Measurement Views** Pulse statistics

**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.

**Related Commands** :DISPlay:PULSe:STATistics:PLOT

**: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.

**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.

**Measurement Views** Pulse statistics

**Examples** :FETCh:PULSe:STATistics:RMSPherror?  
might return 1.032,604.5E-3,1.606 for the RMS phase error in the pulse statistics.

**Related Commands** :DISPlay:PULSe:STATistics:PLOT

**: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.

**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.

**Measurement Views** Pulse statistics

**Examples** :FETCh:PULSe:STATistics:RRATe?  
might return 62.50E+3,62.49E+3,62.52E+3 for the repetition rate in the pulse statistics.

**Related Commands** :DISPlay:PULSe:STATistics:PLOT

## :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.

**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.

**Measurement Views** Pulse statistics

**Examples** :FETCh:PULSe:STATistics:WIDTh?  
might return 4.636E-6,4.630E-6,4.643E-6 for the pulse width in the pulse statistics.

**Related Commands** :DISPlay:PULSe:STATistics:PLOT

**: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.

**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.

**Measurement Views** Pulse trace

**Examples** :FETCh:PULSe:TRACe:X?  
might return #43204xxxx... (3204-byte data) for the time values of the trace.

**Related Commands** :DISPlay:PULSe:SElect:NUMBer, :DISPlay:PULSe:SElect:RESult,  
:FETCh:PULSe:TRACe[:Y]?

**: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.

**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.

**Measurement Views** Pulse trace

**Examples** :FETCh:PULSe:TRACe:XY?  
might return #43204xxxx... (3204-byte data) for the horizontal (time) and vertical value pairs of the pulse trace.

**Related Commands** :DISPlay:PULSe:SElect:NUMBer, :DISPlay:PULSe:SElect:RESult, [:SENSe]:POWer:UNITs

**: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.

**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.

**Measurement Views** Pulse trace

**Examples** :FETCh:PULSe:TRACe:Y?  
might return #43204xxxx... (3204-byte data) for the vertical values of the pulse trace.

**Related Commands** :DISPlay:PULSe:SElect:NUMBer, :DISPlay:PULSe:SElect:RESult,  
:FETCh:PULSe:TRACe:X?, [:SENSE]:POWer:UNITs



## :FETCh:SGRam Subgroup

The :FETCh:SGRam commands returns the results of the spectrogram measurement.

Command Tree	Header	Parameter
	:FETCh	
	:SGRam?	

## :FETCh:SGRam? (Query Only)

Returns trace data of a line in the spectrogram.  
The line is selected using the :TRACe:SGRam:SElect:LINE command.

**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.

**Measurement Views** Spectrogram

**Examples** :FETCh:SGRam?  
might return #43204xxxx... (3204-byte data) for the line in the spectrogram.

**Related Commands** :TRACe:SGRam:SElect:LINE, [:SENSE]:POWER:UNITs

## **:FETCh:SPECTrum Subgroup**

The :FETCh:SPECTrum commands returns results of the Spectrum measurement.

<b>Command Tree</b>	<b>Header</b>	<b>Parameter</b>
	:FETCh	
	:SPECTrum	
	:TRACe<x>?	

Where <x> = 1 to 5. Each trace is defined as follows:

- TRACe1: Trace 1
- TRACe2: Trace 2
- TRACe3: Trace 3
- TRACe4: Math trace
- TRACe5: Spectrogram

The traces can be specified by the :TRACe<x>:SPECTrum command subgroup.

**:FETCh:SPECTrum:TRACe<x>? (Query Only)**

Returns the trace data in the Spectrum measurement.

**Syntax** :FETCh:SPECTrum:TRACe<x>?

Where <x> = 1 to 5.

---

**NOTE.** *TRACe5 (spectrogram) is valid when the spectrum and spectrogram measurements are running.*

---

**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.

**Measurement Views** Spectrum

**Examples** :FETCh:SPECTrum:TRACe1?  
might return #43204xxxx... (3204-byte data) for Trace 1 in the Spectrum measurement.

**Related Commands** :TRACe<x>:SPECTrum command subgroup, [:SENSe]:POWer:UNITs

## :FETCh:SPURious Subgroup

The :FETCh:SPURious commands returns results of the Spurious measurement.

Command Tree	Header	Parameter
	:FETCh	
	:SPURious	
	:CARRier	
	:POWer?	
	:COUNt?	
	:PASS?	
	:SPECTrum	
	:X?	
	:XY?	
	[:Y]?	
	:SPUR<x>	
	:AMPLitude	
	:ABSolute?	
	:RELative?	
	:FREQuency	
	:ABSolute?	
	:RELative?	
	:LIMit	
	:ABSolute?	
	:RELative?	
	:VIOLation?	
	:RANGe?	

Where <x> represents a spurious signal number from 1 to the total number of spurious signals.

## **:FETCh:SPURious:CARRier:POWer? (Query Only)**

Returns the carrier power in the Spurious measurement.

**Syntax** :FETCh:SPURious:CARRier:POWer?

**Arguments** None

**Returns** <NRf> Carrier power in dBm.  
The unit can be changed by the [:SENSe]:POWer:UNITs command.

**Measurement Views** Spurious

**Examples** :FETCh:SPURious:CARRier:POWer?  
might return 4.227, indicating that the carrier power is 4.227 dBm.

**Related Commands** [:SENSe]:POWer:UNITs

## **:FETCh:SPURious:COUNT? (Query Only)**

Returns the number of spurious signals in the Spurious measurement.

**Syntax** :FETCh:SPURious:COUNT?

**Arguments** None

**Returns** <NRf> The number of spurious signals.

**Measurement Views** Spurious

**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.

**Syntax** :FETCh:SPURious:PASS?

**Arguments** None

**Returns** 0 (fail) or 1 (pass).

**Measurement Views** Spurious

**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.

**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.

**Measurement Views** Spurious

**Examples** :FETCh:SPURious:SPECTrum:X?  
might return #516020xxxx... (16020-byte data) for the frequencies of the spectrum trace in the Spurious measurement.

**Related Commands** :FETCh:SPURious:SPECTrum[:Y]?, [:SENSe]:POWer:UNITs

**:FETCh:SPURious:SPECTrum:XY? (Query Only)**

Returns the frequency and amplitude pairs of the spectrum trace in the Spurious measurement.

**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.

**Measurement Views** Spurious

**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.

**Related Commands** [:SENSE]:POWER:UNITs

**:FETCh:SPURious:SPECTrum[:Y]? (Query Only)**

Returns the amplitudes of the spectrum trace in the Spurious measurement.

**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.

**Measurement Views** Spurious

**Examples** :FETCh:SPURious:SPECTrum:Y?  
might return #516020xxxx... (16020-byte data) for the amplitudes of the spectrum trace in the Spurious measurement.

**Related Commands** :FETCh:SPURious:SPECTrum:X?, [:SENSe]:POWer:UNITs



**:FETCh:SPURious:SPUR<x>:AMPLitude:ABSolute? (Query Only)**

Returns the absolute amplitude of the specified spurious signal in the Spurious measurement.

**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.

**Measurement Views** Spurious

**Examples** :FETCh:SPURious:SPUR1:AMPLitude:ABSolute?  
might return -19.782, indicating that the absolute amplitude of Spurious #1 is -19.782 dBm.

**Related Commands** [:SENSE]:POWER:UNITs

**:FETCh:SPURious:SPUR<x>:AMPLitude:RELative? (Query Only)**

Returns the relative amplitude of the specified spurious signal in the Spurious measurement.

**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.

**Measurement Views** Spurious

**Examples** :FETCh:SPURious:SPUR1:AMPLitude:RELative?  
might return -9.782, indicating that the relative amplitude of Spurious #1 is -9.782 dB.

**Related Commands** [:SENSE]:SPURious:REFerence

**:FETCh:SPURious:SPUR<x>:FREQuency:ABSolute? (Query Only)**

Returns the absolute frequency of the specified spurious signal in the Spurious measurement.

**Syntax** :FETCh:SPURious:SPUR<x>:FREQuency:ABSolute?

**Arguments** None

**Returns** <NRf> Absolute frequency of the spurious signal in Hz.

**Measurement Views** Spurious

**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.

**Syntax** :FETCh:SPURious:SPUR<x>:FREQuency:RELative?

**Arguments** None

**Returns** <NRf> Relative frequency of the spurious signal to the carrier in Hz.

**Measurement Views** Spurious

**Examples** :FETCh:SPURious:SPUR1:FREQuency:RELative?  
might return 3.634E+6, indicating that the relative frequency of Spurious #1 is 3.634 MHz.

**Related Commands** [:SENSe]:SPURious:REFEreNce

**:FETCh:SPURious:SPUR<x>:LIMit:ABSolute? (Query Only)**

Returns the absolute amplitude of the limit for the specified spurious signal in the Spurious measurement.

**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.

**Measurement Views** Spurious

**Examples** :FETCh:SPURious:SPUR1:LIMit:ABSolute?  
might return -50.0, indicating that the absolute amplitude of the limit for Spurious #1 is -50 dBm.

**Related Commands** [:SENSe]:POWer:UNITs

**:FETCh:SPURious:SPUR<x>:LIMit:RELative? (Query Only)**

Returns the relative amplitude of the limit for the specified spurious signal in the Spurious measurement.

**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.

**Measurement Views** Spurious

**Examples** :FETCh:SPURious:SPUR1:LIMit:RELative?  
might return -10.0, indicating that the relative amplitude of the limit for Spurious #1 is -10 dB.

**Related Commands** [:SENSE]:SPURious:REFerence

**:FETCh:SPURious:SPUR<x>:LIMit:VIOLation? (Query Only)**

Returns whether the specified spurious signal exceeds the limit or not.

**Syntax** :FETCh:SPURious:SPUR<x>:LIMit:VIOLation?

**Arguments** None

**Returns** 0 (under the limit) or 1 (over the limit).

**Measurement Views** Spurious

**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.

**Syntax** :FETCh:SPURious:SPUR<x>:RANGe?

**Arguments** None

**Returns** <string> "A" to "T" representing Range A to T, respectively.

**Measurement Views** Spurious

**Examples** :FETCh:SPURious:SPUR1:RANGe?  
might return "E", indicating that Spurious #1 is in Range E.

## :FETCh:SQQuality Subgroup (Option 21 Only)

The :FETCh:SQQuality commands return the results of the signal quality measurement.

Command Tree	Header	Parameter
	:FETCh	
	:SQQuality	
	:FREQuency	
	:ERRor?	
	:GAIN	
	:IMBalance?	
	:ORIGin	
	:OFFSet?	
	:PEAK	
	:EVM?	
	:DB?	
	:LOCation?	
	:MERRor?	
	:DB?	
	:LOCation?	
	:PERRor?	
	:LOCation?	
	:QUADrature	
	:ERRor?	
	:RHO	
	:RMS	
	:EVM?	
	:DB?	
	:MER	
	:DB?	
	:MERRor?	
	:DB?	
	:PERRor?	

**:FETCh:SQUality:FREQuency:ERRor? (Query Only)**

Returns the frequency error in the signal quality measurement.

**Syntax** :FETCh:SQUality:FREQuency:ERRor?

**Arguments** None

**Returns** <NRf> Frequency error in Hz.

**Measurement Views** Signal quality

**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.

**Syntax** :FETCh:SQUality:GAIN:IMBalance?

**Arguments** None

**Returns** <NRf> Gain imbalance in dB.

**Measurement Views** Signal quality

**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.

**Syntax** :FETCh:SQUality:ORIGin:OFFSet?

**Arguments** None

**Returns** <NRf> Origin offset in dB.

**Measurement Views** Signal quality

**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.

**Syntax** :FETCh:SQUality:PEAK:EVM?

**Arguments** None

**Returns** <NRf> Peak EVM in percent (%).

**Measurement Views** Signal quality

**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.

**Syntax** :FETCh:SQUality:PEAK:EVM:DB?

**Arguments** None

**Returns** <NRf> Peak EVM in dB.

**Measurement Views** Signal quality

**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.

**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.

**Measurement Views** Signal quality

**Examples** :FETCh:SQUality:PEAK:EVM:LOCation?  
might return 68.000, indicating that the EVM is peak at symbol #68.000.

**Related Commands** [:SENSe]:DDEMod:TIME:UNITs

**:FETCh:SQUality:PEAK:MERRor? (Query Only)**

Returns the peak magnitude error (%) in the signal quality measurement.

**Syntax** :FETCh:SQUality:PEAK:MERRor?

**Arguments** None

**Returns** <NRf> Peak magnitude error in percent (%).

**Measurement Views** Signal quality

**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.

**Syntax** :FETCh:SQUality:PEAK:MERRor:DB?

**Arguments** None

**Returns** <NRf> Peak magnitude error in dB.

**Measurement Views** Signal quality

**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.

**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.

**Measurement Views** Signal quality

**Examples** :FETCh:SQUality:PEAK:MERRor:LOCation?  
might return 68.000, indicating that the magnitude error is peak at symbol #68.

**Related Commands** [:SENSe]:DDEMod:TIME:UNITs

**:FETCh:SQUality:PEAK:PERRor? (Query Only)**

Returns the peak phase error in the signal quality measurement.

**Syntax** :FETCh:SQUality:PEAK:PERRor?

**Arguments** None

**Returns** <NRf> Peak phase error in degrees.

**Measurement Views** Signal quality

**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.

**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.

**Measurement Views** Signal quality

**Examples** :FETCh:SQUality:PEAK:PERRor:LOCation?  
might return 68.000, indicating that the phase error is peak at symbol #68.

**Related Commands** [:SENSe]:DDEMod:TIME:UNITs

**:FETCh:SQUality:QUADrature:ERRor? (Query Only)**

Returns the quadrature error in the signal quality measurement.

**Syntax** :FETCh:SQUality:QUADrature:ERRor?

**Arguments** None

**Returns** <NRf> Quadrature error in degrees.

**Measurement Views** Signal quality

**Examples** :FETCh:SQUality:QUADrature:ERRor?  
might return  $-14.264E-3$ , indicating that the quadrature error is  $-0.014264^\circ$ .

**:FETCh:SQUality:RHO? (Query Only)**

Returns the  $\rho$  (waveform quality) value in the signal quality measurement.

**Syntax** :FETCh:SQUality:RHO?

**Arguments** None

**Returns** <NRf>  $\rho$  value.

**Measurement Views** Signal quality

**Examples** :FETCh:SQUality:RHO?  
might return  $998.703E-3$ , indicating that  $\rho$  is 0.998703.

**:FETCh:SQUality:RMS:EVM? (Query Only)**

Returns the RMS EVM (%) in the signal quality measurement.

**Syntax** :FETCh:SQUality:RMS:EVM?

**Arguments** None

**Returns** <NRf> RMS EVM in percent (%).

**Measurement Views** Signal quality

**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.

**Syntax** :FETCh:SQUality:RMS:EVM:DB?

**Arguments** None

**Returns** <NRf> RMS EVM in dB.

**Measurement Views** Signal quality

**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.

**Syntax** :FETCh:SQUality:RMS:MER:DB?

**Arguments** None

**Returns** <NRf> RMS MER in dB.

**Measurement Views** Signal quality

**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.

**Syntax** :FETCh:SQUality:RMS:MERRor?

**Arguments** None

**Returns** <NRf> RMS magnitude error in percent (%).

**Measurement Views** Signal quality

**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.

**Syntax** :FETCh:SQUality:RMS:MERRor:DB?

**Arguments** None

**Returns** <NRf> RMS MERRor in dB.

**Measurement Views** Signal quality

**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.

**Syntax** :FETCh:SQUality:RMS:PERRor?

**Arguments** None

**Returns** <NRf> RMS phase error in degrees.

**Measurement Views** Signal quality

**Examples** :FETCh:SQUality:RMS:PERRor?  
might return 893.472E-3, indicating that the RMS phase error is 0.893472°.

## :FETCh:TOVerview Subgroup

The :FETCh:TOVerview commands returns the results of the time overview measurement.

Command Tree	Header	Parameter
	:FETCh	
	:TOVerview?	

## :FETCh:TOVerview? (Query Only)

Returns the trace data in the time overview.

**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.

**Measurement Views** Time overview

**Examples** :FETCh:TOVerview?  
might return #43204xxxx... (3204-byte data) for the trace in the time overview.

**Related Commands** [:SENSe]:POWer:UNITs



# INITiate Commands

The INITiate commands control data acquisition.

## Command Tree

Header	Parameter
:INITiate	
:CONTinuous	<boolean>
[:IMMediate]	

## :INITiate:CONTinuous(?)

Determines whether to place the analyzer in the single or the continuous acquisition mode.

**Syntax**     :INITiate:CONTinuous { OFF | ON | 0 | 1 }  
              :INITiate:CONTinuous?

**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

**Measurement Views**   All

**Examples**         :INITiate:CONTinuous ON  
                      places the analyzer in the continuous acquisition mode.

**Related Commands**   :INITiate[:IMMediate]

**: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.*

---

<b>Syntax</b>	:INITiate[:IMMEDIATE]
<b>Arguments</b>	None
<b>Measurement Views</b>	All
<b>Examples</b>	:INITiate:IMMEDIATE starts input signal acquisition.
<b>Related Commands</b>	*OPC, *TRG, *WAI, :INITiate:CONTinuous

# INPut Commands

The INPut commands control the characteristics of the signal input.

## Command Tree

Header	Parameter
:INPut	
[:RF]	
:ATTenuation	<numeric_value>
:AUTO	<boolean>
:MONitor	
:STATe	<boolean>
:GAIN	
:STATe	<boolean>
:MLEVel :RLEVel	<numeric_value>

**:INPut[:RF]:ATTenuation(?)**

Sets or queries the input attenuation. Programming a specified attenuation sets :INPut[:RF]:ATTenuation:AUTO OFF.

**Syntax** :INPut[:RF]:ATTenuation <value>  
:INPut[:RF]:ATTenuation?

**Arguments** <value>::=<NR1> specifies the input attenuation.  
Range: 0 to 75 dB in 5 dB steps.

**Measurement Views** All

**Examples** :INPut:RF:ATTenuation 20  
sets the input attenuation to 20 dB.

**Related Commands** :INPut[:RF]:ATTenuation:AUTO

**:INPut[:RF]:ATTenuation:AUTO(?)**

Determines whether to set the input attenuation automatically or manually.

**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.

**Measurement Views** All

**Examples** :INPut:RF:ATTenuation:AUTO ON  
specifies that the input attenuation is set automatically.

**Related Commands** :INPut[:RF]:ATTenuation



**:INPut[:RF]:ATTenuation:MONitor:STATe(?)**

Determines whether to enable or disable to monitor attenuator use.

**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.

**Measurement Views** All

**Examples** :INPut:RF:ATTenuation:MONitor:STATe ON  
enables to monitor attenuator use.

**:INPut[:RF]:GAIN:STATe(?)****Option 01 Only**

Determines whether to enable or disable the internal pre-amp.

**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.

**Measurement Views**   All

**Examples**     :INPut:RF:GAIN:STATe ON  
              enables the internal pre-amp.

**:INPut:MLEVel|:RLEVel(?)**

Sets or queries the reference level.

**Syntax**     :INPut:MLEVel|:RLEVel <value>  
              :INPut:MLEVel|:RLEVel?

**Arguments**   <value>::=<NRf> specifies the reference level.  
              Range: -170 to +50 dBm.

**Measurement Views**   All

**Examples**     :INPut:RLEVel -10  
              sets the reference level to -10 dBm.

# MMEMory Commands

The :MMEMory commands allow you to manipulate files on the mass memory devices. For handling files, refer to the *RSA6100A Series Quick Reference User Manual*. These commands are divided by measurement views into the subgroups listed in Table 2-37.

**Table 2-37: MMEMory command subgroups**

Command header	Measurement view	Refer to:
:MMEMory (basic)	All views, general file control	page 2-496
:MMEMory:AVTime	Amplitude versus Time	page 2-504
:MMEMory:CCDF	CCDF	page 2-506
:MMEMory:DPSA	DPX (Digital Phosphor) spectrum	page 2-508
:MMEMory:FVTime	Frequency versus Time	page 2-510
:MMEMory:IQVTime	RF I&Q versus Time	page 2-512
:MMEMory:PHVTime	Phase versus Time	page 2-515
:MMEMory:SPECTrum	Spectrum	page 2-517
:MMEMory:SPURious	Spurious	page 2-519

**NOTE.** For the data file formats, refer to the RSA6100A Series Online Help.

## 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.

## :MMEMory Basic Command Subgroup

The :MMEMory basic commands control loading and storing files.

Command Tree	Header	Parameter
	:MMEMory	
	:CALibration	
	:LOAD	
	:CORRection	
	:EXTErnal	
	:EDIT<x>	<file_name>
	:STORE	
	:CORRection	
	:EXTErnal	
	:EDIT<x>	<file_name>
	:LOAD	
	:IQ	<file_name>
	:STATE	<file_name>
	:TRACe	<file_name>
	:STORE	
	:IQ	<file_name>
	:CSV	<file_name>
	:MAT	<file_name>
	:MState	<file_name>
	:RESuIts	<file_name>
	:STATE	<file_name>
	:TRACe	<file_name>

## **:MMEMory:CALibration:LOAD:CORRection:EXTErnal:EDIT<x> (No Query Form)**

Loads the external loss table from a specified file.

**Syntax** :MMEMory:CALibration:LOAD:CORRection:EXTErnal:EDIT<x>  
<file\_name>

Where <x> = 1 to 3 represent External Loss Table 1 to 3, respectively.

**Arguments** <file\_name>::=<string> specifies the file to load the external loss table from. The file extension is .csv. You can omit the extension.

**Measurement Views** All

**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.

**Syntax** :MMEMory:CALibration:STORe:CORRection:EXTErnal:EDIT<x>  
<file\_name>

Where <x> = 1 to 3 represent External Loss Table 1 to 3, respectively.

**Arguments** <file\_name>::=<string> specifies the file to store the external loss table to. The file extension is .csv. You can omit the extension.

**Measurement Views** All

**Examples** :MMEMory:CALibration:STORe:CORRection:EXTErnal:EDIT1 "Table1"  
stores the External Loss Table 1 to the *Table1.csv* file.

**:MMEMory:LOAD:IQ (No Query Form)**

Loads time-domain IQ waveform into the acquisition memory from a file.

**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.

**Measurement Views** All

**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.

**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.

**Measurement Views** All

**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.

**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 Table 2-38. 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 Table 2-38. 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 2-38: 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
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

**Table 2-38: Trace file extension and trace selection command (Cont.)**

Display group	Measurement view	File extension	Trace selection
Pulsed RF (Option 20 only)	Pulse statistics	NA	NA
	Pulse table	NA	NA
	Pulse trace	NA	NA

**Measurement Views** Refer to Table 2-38.

**Examples** :MMEMemory:LOAD:TRACe "TRACE1"  
loads the trace data from the *TRACE1.Specan* file when the spectrum view is selected.

**Related Commands** :TRACe<x>:AVTime:SElect, :TRACe<x>:CCDF:SElect,  
:TRACe<x>:DPSA:SElect, :TRACe<x>:PNOise:SElect,  
:TRACe<x>:SPEctrum:SElect

## :MMEMemory:STORe:IQ (No Query Form)

Saves time-domain IQ waveform in the acquisition memory to a specified file.

**Syntax** :MMEMemory: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.

**Measurement Views** All

**Examples** :MMEMemory: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.

**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.

**Measurement Views** All

**Examples** :MMEMory:STORe:IQ:CSV "IQ2"  
saves IQ data to the *IQ2.csv* 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.

**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.

**Measurement Views** All

**Examples** :MMEMory:STORe:IQ:MAT "IQ3"  
saves IQ data to the *IQ3.mat* file.

**:MMEMory:STORe:MSate (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.

**Syntax** :MMEMory:STORe:MSate <file\_name>

**Arguments** <file\_name>::=<string> specifies the file to store the measurement parameters. The file extension is .txt. You can omit the extension.

**Measurement Views** All

**Examples** :MMEMory:STORe:MSate "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.

**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.

**Measurement Views** All

**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.

**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.

**Measurement Views** All

**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.

**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 Table 2-38 on page 2-499. 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.

**Measurement Views** Refer to Table 2-38 on page 2-499.

**Examples** :MMEMory:STORe:TRACe "TRACE1"  
stores the trace data in the *TRACE1.Specan* file when the spectrum view is selected.

## :MMEMory:AVTime Subgroup

The :MMEMory:AVTime commands control the mass storage in the Amplitude versus Time measurement.

Command Tree	Header	Parameter
	:MMEMory	
	:AVTime	
	:LOAD	
	:TRACe<x>	<file_name>
	:STORE	
	:TRACe<x>	<file_name>

Where <x> = 1 to 4.

TRACe1: Trace 1

TRACe2: Trace 2

TRACe3: Trace 3

TRACe4: Math trace (for store only)

**:MMEMory:AVTime:LOAD:TRACe<x> (No Query Form)**

Loads the Amplitude versus Time trace data from the specified file.

**Syntax** :MMEMory:AVTime:LOAD:TRACe<x> <file\_name>

Where <x> = 1 to 3; Trace 4 (math trace) is invalid.

**Arguments** <file\_name>::=<string> specifies the file from which to load trace data. The file extension is .AmplVsTime. You can omit the extension.

**Measurement Views** Amplitude versus Time

**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.

**Syntax** :MMEMory:AVTime:STORE:TRACe<x> <file\_name>

Where <x> = 1 to 4; All traces are valid.

**Arguments** <file\_name>::=<string> specifies the file in which to store trace data. The file extension is .AmplVsTime. You can omit the extension.

**Measurement Views** Amplitude versus Time

**Examples** :MMEMory:AVTime:STORE:TRACe1 "Sample1"  
stores Trace 1 data in the *Sample1.AmplVsTime* file.

## :MMEMemory:CCDF Subgroup

The :MMEMemory:CCDF commands control the mass storage in the CCDF measurement.

Command Tree	Header	Parameter
	:MMEMemory	
	:CCDF	
	:LOAD	
	:TRACe<x>	<file_name>
	:STORE	
	:TRACe<x>	<file_name>

Where <x> = 1 or 2.

TRACe1: Trace 1

TRACe2: Trace 2

TRACe3 (Gaussian reference curve) is invalid.

**:MMEMory:CCDF:LOAD:TRACe<x> (No Query Form)**

Loads the CCDF trace data from the specified file.

**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.

**Measurement Views** CCDF

**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.

**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.

**Measurement Views** CCDF

**Examples** :MMEMory:CCDF:STORe:TRACe1 "Sample1"  
stores Trace 1 data in the *Sample1.CCDF* file.

## :MMEMory:DPSA Subgroup

The :MMEMory:DPSA commands control the mass storage in the DPX (Digital Phosphor) spectrum measurement.

Command Tree	Header	Parameter
	:MMEMory	
	:DPSA	
	:LOAD	
	:TRACe<x>	<file_name>
	:STORE	
	:TRACe<x>	<file_name>

Where <x> = 1 to 5. Each trace is defined as follows:

- TRACe1: Maximum trace
- TRACe2: Minimum trace
- TRACe3: Average trace
- TRACe4: Math trace (for store only)
- TRACe5: Bitmap trace



**:MMEMory:DPSA:LOAD:TRACe<x> (No Query Form)**

Loads the DPX spectrum trace data from the specified file.

**Syntax** :MMEMory:DPSA:LOAD:TRACe<x> <file\_name>

Where <x> = 1, 2, 3, or 5; TRACe4 (math trace) is invalid.

**Arguments** <file\_name>::=<string> specifies the file from which to load trace data. The file extension is .dpt. You can omit the extension.

**Measurement Views** DPX spectrum

**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.

**Syntax** :MMEMory:DPSA:STORe:TRACe<x> <file\_name>

Where <x> = 1 to 5; All traces are valid.

**Arguments** <file\_name>::=<string> specifies the file in which to store trace data. The file extension is .dpt. You can omit the extension.

**Measurement Views** DPX spectrum

**Examples** :MMEMory:DPSA:STORe:TRACe1 "Sample1"  
stores Trace 1 data in the *Sample1.dpt* file.

## :MMEMory:FVTime Subgroup

The :MMEMory:FVTime commands control the mass storage in the Frequency versus Time measurement.

Command Tree	Header	Parameter
	:MMEMory	
	:FVTime	
	:LOAD	
	:TRACe	<file_name>
	:STORe	
	:TRACe	<file_name>

**:MMEMory:FVTime:LOAD:TRACe (No Query Form)**

Loads the Frequency versus Time trace data from the specified file.

**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.

**Measurement Views** Frequency versus Time

**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.

**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.

**Measurement Views** Frequency versus Time

**Examples** :MMEMory:FVTime:STORe:TRACe "Sample1"  
stores the Frequency versus Time trace data in the *Sample1.FreqVsTime* file.

## :MMEMory:IQVTime Subgroup

The :MMEMory:IQVTime commands control the mass storage in the RF I&Q versus Time measurement.

Command Tree	Header	Parameter
	:MMEMory	
	:IQVTime	
	:LOAD	
	:TRACe	
	:I	<file_name>
	:Q	<file_name>
	:STORe	
	:TRACe	
	:I	<file_name>
	:Q	<file_name>

**:MMEMory:IQVTime:LOAD:TRACe:I (No Query Form)**

Loads I trace data from the specified file.

**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.

**Measurement Views** RF I&Q versus Time

**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.

**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.

**Measurement Views** RF I&Q versus Time

**Examples** :MMEMory:IQVTime:LOAD:TRACe:Q "Sample2"  
loads the Q trace data from the *Sample2.RFIQVsTime* file.

## **:MMEMemory:IQVTime:STORe:TRACe:I (No Query Form)**

Stores I trace data in the specified file.

**Syntax** :MMEMemory:IQVTime:STORe:TRACe <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.

**Measurement Views** RF I&Q versus Time

**Examples** :MMEMemory:IQVTime:STORe:TRACe "Sample1"  
stores the I trace data in the *Sample1.RFIQVsTime* file.

## **:MMEMemory:IQVTime:STORe:TRACe:Q (No Query Form)**

Stores Q trace data in the specified file.

**Syntax** :MMEMemory:IQVTime:STORe:TRACe <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.

**Measurement Views** RF I&Q versus Time

**Examples** :MMEMemory:IQVTime:STORe:TRACe "Sample2"  
stores the Q trace data in the *Sample2.RFIQVsTime* file.

## :MMEMory:PHVTime Subgroup

The :MMEMory:PHVTime commands control the mass storage in the Phase versus Time measurement.

Command Tree	Header	Parameter
	:MMEMory	
	:PHVTime	
	:LOAD	
	:TRACe	<file_name>
	:STORe	
	:TRACe	<file_name>

## **:MMEMory:PHVTime:LOAD:TRACe (No Query Form)**

Loads the Phase versus Time trace data from the specified file.

**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.

**Measurement Views** Phase versus Time

**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.

**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.

**Measurement Views** Phase versus Time

**Examples** :MMEMory:PHVTime:STORe:TRACe "Sample1"  
stores the Phase versus Time trace data in the *Sample1.PhaseVsTime* file.



## :MMEMory:SPECTrum Subgroup

The :MMEMory:SPECTrum commands control the mass storage in the Spectrum measurement.

Command Tree	Header	Parameter
	:MMEMory	
	:SPECTrum	
	:LOAD	
	:TRACe<x>	<file_name>
	:STORe	
	:TRACe<x>	<file_name>

Where <x> = 1 to 5. Each trace is defined as follows:

TRACe1: Trace 1  
 TRACe2: Trace 2  
 TRACe3: Trace 3  
 TRACe4: Math trace (for store only)  
 TRACe5: Spectrogram (for store only)

**:MMEMory:SPECTrum:LOAD:TRACe<x> (No Query Form)**

Loads the spectrum trace data from the specified file.

**Syntax** :MMEMory:SPECTrum:LOAD:TRACe<x> <file\_name>

Where <x> = 1 to 3; Trace 4 (math trace) and Trace 5 (spectrogram) are invalid.

**Arguments** <file\_name>::=<string> specifies the file from which to load trace data. The file extension is .Specan. You can omit the extension.

**Measurement Views** Spectrum

**Examples** :MMEMory:SPECTrum:LOAD:TRACe1 "Sample1"  
loads Trace 1 data from the *Sample1.Specan* file.

**:MMEMory:SPECTrum:STORe:TRACe<x> (No Query Form)**

Stores the spectrum trace data in the specified file.

**Syntax** :MMEMory:SPECTrum:STORe:TRACe<x> <file\_name>

Where <x> = 1 to 5; All traces are valid.

---

**NOTE.** *TRACe5 (spectrogram) is valid when the spectrum and spectrogram measurements are running.*

---

**Arguments** <file\_name>::=<string> specifies the file in which to store trace data. The file extension is .Specan. You can omit the extension.

**Measurement Views** Spectrum

**Examples** :MMEMory:SPECTrum:STORe:TRACe1 "Sample1"  
stores Trace 1 data in the *Sample1.Specan* file.

## :MMEMory:SPURious Subgroup

The :MMEMory:SPURious commands control the mass storage in the Spurious measurement.

Command Tree	Header	Parameter
	:MMEMory	
	:SPURious	
	:LOAD	
	:TABLE	<file_name>
	:STORE	
	:TABLE	<file_name>

**:MMEMory:SPURious:LOAD:TABLE (No Query Form)**

Loads the spurious table containing the limits for enabled ranges from the specified file.

**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.

**Measurement Views** Spurious

**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.

**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.

**Measurement Views** Spurious

**Examples** :MMEMory:SPURious:STORe:TRACe1 "Table1"  
stores the spurious table in the *Table1.csv* file.

# OUTPut Commands

The :OUTPut commands control the characteristics of the signal output.

## Command Tree

Header	Parameter
:OUTPut	
:IF	<boolean>
:BAWdwidth	
:BWIDTH	FLATtop   GAUSSian
[:STATe]	<boolean>
:IQ	<boolean>
:NOISe	
[:STATe]	<boolean>

## **:OUTPut:IF:BANDwidth|:BWIDth(?)**

*Option 05 Only*

Selects or queries the IF output filter when :OUTPut:IF[:STATe] is set to ON.

**Syntax** :OUTPut:IF:BANDwidth|:BWIDth { FLATtop | GAUSSian }  
:OUTPut:IF:BANDwidth|:BWIDth?

**Arguments** FLATtop selects the flattop filter.  
GAUSSian selects the Gaussian filter.

**Measurement Views** All

**Examples** :OUTPut:IF:BANDwidth GAUSSian  
selects the Gaussian filter for the IF output.

**Related Commands** :OUTPut:IF[:STATe]

## **:OUTPut:IF[:STATe](?)**

*Option 05 Only*

Determines whether to turn on or off the 500 MHz IF Out on the rear panel.

**Syntax** :OUTPut:IF[:STATe] { OFF | ON | 0 | 1 }  
:OUTPut:IF[:STATe]?

**Arguments** OFF or 0 turns off IF Out.  
ON or 1 turns on IF Out.

**Measurement Views** All

**Examples** :OUTPut:IF:STATe ON  
turns on IF Out.

**Related Commands** :OUTPut:IF:BANDwidth|:BWIDth

**:OUTPut:IQ[:STATe](?)***Option 05 Only*

Determines whether to enable or disable the digital IQ output data stream from the rear panel connectors.

**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.

**Measurement Modes** All

**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.

**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.

**Measurement Views** All

**Examples** :OUTPut:NOISe:STATe ON  
turns on +28 V DC Out.





# 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.

The READ commands are divided by measurement views into the subgroups listed in Table 2-39.

**Table 2-39: READ command subgroups**

Command header	Measurement view	Refer to:
:READ:ACPower	Channel power and ACPR	page 2-526
:READ:AVTime	Amplitude versus Time	page 2-530
:READ:CCDF	CCDF	page 2-535
:READ:CONSte	Constellation (Option 21 only)	page 2-540
:READ:DDEMod	General purpose digital modulation measurements (Option 21 only)	page 2-542
:READ:DPXA	DPX (Digital Phosphor) spectrum	page 2-543
:READ:EVM	EVM versus Time (Option 21 only)	page 2-547
:READ:FVTime	Frequency versus Time	page 2-550
:READ:IQVTime	RF I&Q versus Time	page 2-554
:READ:MCPower	MCPR (Multiple Carrier Power Ratio)	page 2-559
:READ:MERRor	Magnitude error versus Time (Option 21 only)	page 2-564
:READ:OBW	OBW (Occupied Bandwidth)	page 2-567
:READ:PERRor	Phase error versus Time (Option 21 only)	page 2-576
:READ:PHVTime	Phase versus Time	page 2-579
:READ:PNOise	Phase noise (Option 11 only)	page 2-583
:READ:PULSe	Pulsed RF measurements (Option 20 only)	page 2-590
:READ:SGRam	Spectrogram	page 2-631
:READ:SPECtrum	Spectrum	page 2-632
:READ:SPURious	Spurious	page 2-634
:READ:SQUality	Signal quality (Option 21 only)	page 2-645
:READ:TOVerview	Time overview	page 2-657

## **:READ:ACPower Subgroup**

The :READ:ACPower commands returns the results of the Channel power and ACPR (Adjacent Channel Power Ratio) measurement.

<b>Command Tree</b>	<b>Header</b>	<b>Parameter</b>
	:READ	
	:ACPower?	
	:CHANnel	
	:POWer?	
	:SPECTrum?	

**:READ:ACPower? (Query Only)**

Returns the Channel power and ACPR measurement results for all available channels.

**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.

**Measurement Views** Channel power and ACPR

**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.

**Related Commands** [:SENSe]:ACPower:CHANnel:PAIRs, [:SENSe]:POWer:UNITs

**:READ:ACPower:CHANnel:POWer? (Query Only)**

Returns the average power of the main channel (power reference) in the Channel power and ACPR measurement.

**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.

**Measurement Views**    Channel power and ACPR

**Examples**      :READ:ACPower:CHANnel:POWer?  
might return 4.227 indicating that the average power of the main channel is 4.227 dBm.

**Related Commands**    [:SENSe]:POWer:UNITs

## **:READ:ACPower:SPECTrum? (Query Only)**

Returns spectrum trace data of the Channel power and ACPR measurement.

**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.

**Measurement Views** Channel power and ACPR

**Examples** :READ:ACPower:SPECTrum?  
might return #43204xxxx... (3204-byte data) for the spectrum trace data of the Channel power and ACPR measurement.

**Related Commands** [:SENSe]:POWer:UNITs

## :READ:AVTime Subgroup

The :READ:AVTime commands returns the results of the Amplitude versus Time measurement.

Command Tree	Header	Parameter
	:READ	
	:AVTime	
	:AVERage	
	:FIRSt :SECond :THIRd :FOURth	
	:MAXimum	
	:MAXLocation	
	:MINimum	
	:MINLocation	
	:RESult	

---

**NOTE.** Except for the :READ:AVTime:FIRSt|:SECond|:THIRd|:FOURth command, select the trace using the :TRACe<x>:AVTime:SElect command.

---

## :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.

<b>Syntax</b>	:READ:AVTime:AVERage?
<b>Arguments</b>	None
<b>Returns</b>	<avg>::=<NRf> is the RMS amplitude in dBm. The unit can be changed by the [:SENSe]:POWer:UNITs command.
<b>Measurement Views</b>	Amplitude versus Time
<b>Examples</b>	:READ:AVTime:AVERage? might return -2.53 indicating the RMS amplitude is -2.53 dBm.
<b>Related Commands</b>	:TRACe<x>:AVTime:SElect, [:SENSe]:POWer:UNITs

**:READ:AVTime:FIRSt|:SECond|:THIRd|:FOURth? (Query Only)**

Returns the trace data in the Amplitude versus Time measurement.

**Syntax** :READ:AVTime:FIRSt|:SECond|:THIRd|:FOURth?

Where FIRSt, SECond, THIRd, and FOURth represent Trace 1 to 4, respectively. Each trace is defined as follows:

FIRSt: Trace 1  
 SECond: Trace 2  
 THIRd: Trace 3  
 FOURth: Math trace

The traces can be specified by the :TRACe<x>:AVTime 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.

**Measurement Views** Amplitude versus Time

**Examples** :READ:AVTime:FIRSt?  
 might return #3156xxxx... (156-byte data) for Trace 1.

**Related Commands** :TRACe<x>:AVTime command subgroup, [:SENSe]:POWer:UNITs

## **: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.

**Syntax** :READ:AVTime:MAXimum?

**Arguments** None

**Returns** <max>::=<NRf> is the maximum Amplitude in dBm.  
The unit can be changed by the [:SENSE]:POWER:UNITs command.

**Measurement Views** Amplitude versus Time

**Examples** :READ:AVTime:MAXimum?  
might return -2.84 indicating the maximum amplitude is -2.84 dBm.

**Related Commands** :READ:AVTime:MAXLocation?, :TRACe<x>:AVTime:SElect,  
[:SENSE]:POWER:UNITs

## **: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:SElec command.

**Syntax** :READ:AVTime:MAXLocation?

**Arguments** None

**Returns** <max\_time>::=<NRf> is the time at the maximum in seconds.

**Measurement Views** Amplitude versus Time

**Examples** :READ:AVTime:MAXLocation?  
might return 25.03E-9 indicating the amplitude is maximum at 25.03 ns.

**Related Commands** :READ:AVTime:MAXimum?, :TRACe<x>:AVTime:SElect



**: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.

**Syntax** :READ:AVTime:MINimum?

**Arguments** None

**Returns** <min>::=<Nrf> is the minimum amplitude in dBm.  
The unit can be changed by the [:SENSE]:POWER:UNITs command.

**Measurement Views** Amplitude versus Time

**Examples** :READ:AVTime:MINimum?  
might return -57.64 indicating the minimum amplitude is -57.64 dBm.

**Related Commands** :READ:AVTime:MINLocation?, :TRACe<x>:AVTime:SElect,  
[:SENSE]:POWER:UNITs

**: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.

**Syntax** :READ:AVTime:MINLocation?

**Arguments** None

**Returns** <min\_time>::=<Nrf> is the time at the minimum in seconds.

**Measurement Views** Amplitude versus Time

**Examples** :READ:AVTime:MINLocation?  
might return 450.7E-9 indicating the amplitude is minimum at 450.7 ns.

**Related Commands** :READ:AVTime:MINimum?, :TRACe:AVTime:SElect

## **: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.

**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.

**Measurement Views** Amplitude versus Time

**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  $\mu$ s,  
 the minimum amplitude is -82.47 dBm at 22.11  $\mu$ s, and  
 the RMS amplitude is -8.24 dBm

**Related Commands** :TRACe<x>:AVTime:SElect, [:SENSE]:POWER:UNITs

## :READ:CCDF Subgroup

The :READ:CCDF commands returns the results of the CCDF (Complementary Cumulative Distribution Function) measurement.

Command Tree	Header	Parameter
	:READ	
	:CCDF?	
	:FIRSt   :SECond   :THIRd	
	:X?	
	:XY?	
	[:Y]?	

## :READ:CCDF? (Query Only)

Returns the CCDF measurement results.

**Syntax** :READ:CCDF?

**Arguments** None

**Returns** <avg\_amp1>,<avg\_ccdf>,<crest\_factor>,<ampl\_10>,<ampl\_1>,<ampl\_p1>,<ampl\_p01>,<ampl\_p001>,,<ampl\_p0001>

Where

<avg\_amp1> 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.

**Measurement Views** CCDF

**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.

**Related Commands** [:SENSE]:POWER:UNITs

**:READ:CCDF:FIRSt|:SECond|:THIRd:X? (Query Only)**

Returns the horizontal values of the specified trace in the CCDF measurement.

---

**NOTE.** This query is invalid when [:SENSe]:CCDF:TIME:TYPE is set to CONTInuous or TOTal.

---

**Syntax** :READ:CCDF:FIRSt|:SECond|:THIRd:X?

Where FIRSt, SECond, and THIRd represent Trace 1 to 3, respectively. Each trace is defined as follows:

FIRSt:	Trace 1
SECond:	Trace 2
THIRd:	Gaussian reference curve

**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.

**Measurement Views** CCDF

**Examples** :READ:CCDF:FIRSt:X?  
might return #41024xxxx... (1024-byte data) for the horizontal values of Trace 1.

**Related Commands** :READ:CCDF:FIRSt|:SECond|:THIRd[:Y]?, [:SENSe]:CCDF:TIME:TYPE

## **:READ:CCDF:FIRSt|:SECond|:THIRd:XY? (Query Only)**

Returns the horizontal and vertical value pairs of the specified trace in the CCDF measurement.

---

**NOTE.** This query is invalid when [:SENSe]:CCDF:TIME:TYPE is set to CONTInuous or TOTal.

---

**Syntax**     :READ:CCDF:FIRSt | :SECond | :THIRd:XY?

Where FIRSt, SECond, and THIRd represent Trace 1 to 3, respectively. Each trace is defined as follows:

FIRSt:     Trace 1  
 SECond:    Trace 2  
 THIRd:     Gaussian reference curve

**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.

**Measurement Views**   CCDF

**Examples**     :READ:CCDF:FIRSt:XY?  
 might return #41024xxxx... (1024-byte data) for the horizontal and vertical value pairs of Trace 1.

**Related Commands**   [:SENSe]:CCDF:TIME:TYPE

**:READ:CCDF:FIRSt|:SECond|:THIRd[:Y]? (Query Only)**

Returns the vertical values of the specified trace in the CCDF measurement.

---

**NOTE.** This query is invalid when [:SENSe]:CCDF:TIME:TYPE is set to CONTInuous or TOTal.

---

**Syntax** :READ:CCDF:FIRSt|:SECond|:THIRd[:Y]?

Where FIRSt, SECond, and THIRd represent Trace 1 to 3, respectively. Each trace is defined as follows:

FIRSt:	Trace 1
SECond:	Trace 2
THIRd:	Gaussian reference curve

**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.

**Measurement Views** CCDF

**Examples** :READ:CCDF:FIRSt:Y?  
might return #41024xxxx... (1024-byte data) for the vertical values of Trace 1.

**Related Commands** :READ:CCDF:FIRSt|:SECond|:THIRd:X?, [:SENSe]:CCDF:TIME:TYPE

## :READ:CONSte Subgroup (Option 21 Only)

The :READ:CONSte commands returns the results of the Constellation measurement.

Command Tree	Header	Parameter
	:READ	
	:CONSte	
	:RESu1ts?	
	:TRACe?	

## :READ:CONSte:RESu1ts? (Query Only)

Returns the constellation measurement results of EVM RMS, peak and location displayed on the bottom of the screen.

**Syntax**       :READ:CONSte:RESu1ts?

**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.

**Measurement Views**   Constellation

**Examples**       :READ:CONSte:RESu1ts?  
 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.

**Related Commands**   [:SENSe]:DDEMod:TIME:UNITs



## **:READ:CONStE:TRACe? (Query Only)**

Returns the constellation trace data.

**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  $n^{\text{th}}$  data point. 4-byte little endian floating-point format specified in IEEE 488.2.

**Measurement Views** Constellation

**Examples** :READ:CONStE:TRACe?  
might return #43848xxxx... (3848-byte data) for the constellation trace data.

## :READ:DDEMod Subgroup (Option 21 Only)

The :READ:DDEMod commands returns the results of the General Purpose Digital Modulation measurements.

Command Tree	Header	Parameter
	:READ	
	:DDEMod	
	:STABLe?	

## :READ:DDEMod:STABLe? (Query Only)

Returns the symbol table data.

**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.

**Measurement Views**   Symbol table

**Examples**     :READ:DDEMod:STABLe?  
might return #3512xxxx... (512-byte data) for the symbol table.

## :READ:DPSA Subgroup

The :READ:DPSA commands returns the results of the DPX (Digital Phosphor) Spectrum measurement.

Command Tree	Header	Parameter
	:READ	
	:DPSA	
	:TRACe	
	:AVERAge?	
	:MATH?	
	:MAXimum?	
	:MINimum?	

### :READ:DPSA:TRACe:AVERAge? (Query Only)

Returns waveform data of the average trace in the DPX spectrum measurement.

**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.

**Measurement Views** DPX spectrum

**Examples** :READ:DPSA:TRACe:AVERAge?  
might return #42004xxxx... (2004-byte data) for the waveform data of the average trace.

**Related Commands** [:SENSe]:POWer:UNITs

## **:READ:DPSA:TRACe:MATH? (Query Only)**

Returns waveform data of the math trace in the DPX spectrum measurement.

**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.

**Measurement Views** DPX spectrum

**Examples** :READ:DPSA:TRACe:MATH?  
might return #42004xxxx... (2004-byte data) for the waveform data of the math trace.

**Related Commands** [:SENSe]:POWer:UNITs

## **:READ:DPSA:TRACe:MAXimum? (Query Only)**

Returns waveform data of the maximum trace in the DPX spectrum measurement.

**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.

**Measurement Views**   DPX spectrum

**Examples**         :READ:DPSA:TRACe:MAXimum?  
might return #42004xxxx... (2004-byte data) for the waveform data of the maximum trace.

**Related Commands**   [:SENSe]:POWer:UNITs

## **:READ:DPSA:TRACe:MINimum? (Query Only)**

Returns waveform data of the minimum trace in the DPX spectrum measurement.

**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.

**Measurement Views**    DPX spectrum

**Examples**      :READ:DPSA:TRACe:MINimum?  
 might return #42004xxxx... (2004-byte data) for the waveform data of the minimum trace.

**Related Commands**    [:SENSe]:POWer:UNITs

## :READ:EVM Subgroup (Option 21)

The :READ:EVM commands returns the results of the EVM (Error Vector Magnitude) versus Time measurement.

Command Tree	Header	Parameter
	:READ	
	:EVM	
	:FERRor?	
	:PEAK?	
	:PINDEX?	
	:RMS?	
	:TRACe?	

## :READ:EVM:FERRor? (Query Only)

Returns the frequency error in the EVM versus Time measurement.

**Syntax** :READ:EVM:FERRor?

**Arguments** None

**Returns** <freq\_error>::=<NRf> is the frequency error in Hz.

**Measurement Views** EVM versus Time

**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.

**Syntax** :READ:EVM:PEAK?

**Arguments** None

**Returns** <peak>::=<NRf> is the peak EVM value in percent (%).

**Measurement Views** EVM versus Time

**Examples** :READ:EVM:PEAK?  
might return 1.32 indicating the peak EVM value is 1.32%.

**Related Commands** :READ:EVM:PINDEX?

## **:READ:EVM:PINDEX? (Query Only)**

Returns the time at the EVM peak.

**Syntax** :READ:EVM:PINDEX?

**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.

**Measurement Views** EVM versus Time

**Examples** :READ:EVM:PINDEX?  
might return 68.000, indicating that the EVM peak is at symbol #68.

**Related Commands** :READ:EVM:PEAK?, [:SENSE]:DDEMod:TIME:UNITs



**:READ:EVM:RMS? (Query Only)**

Returns the RMS (Root-Mean-Square) value in the EVM versus Time measurement.

**Syntax** :READ:EVM:RMS?

**Arguments** None

**Returns** <rms>::=<NRf> is the RMS EVM value in percent (%).

**Measurement Views** EVM versus Time

**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.

**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.

**Measurement Views** EVM versus Time

**Examples** :READ:EVM:TRACe?  
might return #42036xxxx... (2036-byte data) for the EVM versus Time trace.

## :READ:FVTime Subgroup

The :READ:EVM commands returns the results of the Frequency versus Time measurement.

Command Tree	Header	Parameter
	:READ	
	:FVTime?	
	:MAXimum?	
	:MAXLocation?	
	:MINimum?	
	:MINLocation?	
	:RESult?	

## :READ:FVTime? (Query Only)

Returns the Frequency versus Time trace data.

**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.

**Measurement Views**   Frequency versus Time

**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.

**Syntax** :READ:FVTime:MAXimum?

**Arguments** None

**Returns** <max>::=<NRf> is the maximum frequency drift in Hz.

**Measurement Views** Frequency versus Time

**Examples** :READ:FVTime:MAXimum?  
might return 2.625E+6 indicating the maximum frequency drift is 2.625 MHz.

**Related Commands** :READ:FVTime:MAXLocation?

## :READ:FVTime:MAXLocation? (Query Only)

Returns the time at which the frequency drift is maximum.

**Syntax** :READ:FVTime:MAXLocation?

**Arguments** None

**Returns** <max\_time>::=<NRf> is the time in seconds at which the frequency drift is maximum.

**Measurement Views** Frequency versus Time

**Examples** :READ:FVTime:MAXLocation?  
might return 25.03E-9 indicating the frequency drift is maximum at 25.03 ns.

**Related Commands** :READ:FVTime:MAXimum?

## **:READ:FVTime:MINimum? (Query Only)**

Returns the minimum value in the Frequency versus Time measurement.

**Syntax** :READ:FVTime:MINimum?

**Arguments** None

**Returns** <min>::=<NRf> is the minimum frequency drift in Hz.

**Measurement Views** Frequency versus Time

**Examples** :READ:FVTime:MINimum?  
might return -6.618E+6 indicating the minimum frequency drift is -6.618 MHz.

**Related Commands** :READ:FVTime:MINLocation?

## **:READ:FVTime:MINLocation? (Query Only)**

Returns the time at which the frequency drift is minimum.

**Syntax** :READ:FVTime:MINLocation?

**Arguments** None

**Returns** <min\_time>::=<NRf> is the time in seconds at which the frequency drift is minimum.

**Measurement Views** Frequency versus Time

**Examples** :READ:FVTime:MAXLocation?  
might return 450.7E-9 indicating the frequency drift is minimum at 450.7 ns.

**Related Commands** :READ:FVTime:MINimum?

## :READ:FVTime:RESult? (Query Only)

Returns the Frequency versus Time measurement results.

**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.

**Measurement Views** Frequency versus Time

**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 Subgroup

The :READ:IQVTime commands returns the results of the RF I&Q versus Time measurement.

Command Tree	Header	Parameter
	:READ	
	:IQVTime?	
	:I?	
	:MAXimum?	
	:MAXLocation?	
	:MINimum?	
	:MINLocation?	
	:Q?	
	:RESult?	

### :READ:IQVTime:I? (Query Only)

Returns the I versus Time trace data.

**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.

**Measurement Views**   RF I&Q versus Time

**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.

**Syntax** :READ:IQVTime:MAXimum?

**Arguments** None

**Returns** <max>::=<NRf> is the maximum I or Q level in volts.

**Measurement Views** RF I&Q versus Time

**Examples** :READ:IQVTime:MAXimum?  
might return 1.214 indicating the maximum I or Q level is 1.214 V.

**Related Commands** :READ:IQVTime:MAXLocation?

## :READ:IQVTime:MAXLocation? (Query Only)

Returns the time at which the I or Q level is maximum.

**Syntax** :READ:IQVTime:MAXLocation?

**Arguments** None

**Returns** <max\_time>::=<NRf> is the time in seconds at which the I or Q level is maximum.

**Measurement Views** RF I&Q versus Time

**Examples** :READ:IQVTime:MAXLocation?  
might return 175.3E-9 indicating the I or Q level is maximum at 175.3 ns.

**Related Commands** :READ:IQVTime:MAXimum?

## **:READ:IQVTime:MINimum? (Query Only)**

Returns the minimum value in the RF I&Q versus Time measurement.

**Syntax** :READ:IQVTime:MINimum?

**Arguments** None

**Returns** <min>::=<NRf> is the minimum I or Q level in volts.

**Measurement Views** RF I&Q versus Time

**Examples** :READ:IQVTime:MINimum?  
might return  $-370.5E-3$  indicating the minimum I or Q level is  $-370.5$  mV.

**Related Commands** :READ:IQVTime:MINLocation?

## **:READ:IQVTime:MINLocation? (Query Only)**

Returns the time at which the I or Q level is minimum.

**Syntax** :READ:IQVTime:MINLocation?

**Arguments** None

**Returns** <min\_time>::=<NRf> is the time in seconds at which the I or Q level is minimum.

**Measurement Views** RF I&Q versus Time

**Examples** :READ:IQVTime:MAXLocation?  
might return  $450.7E-9$  indicating the I or Q level is minimum at  $450.7$  ns.

**Related Commands** :READ:IQVTime:MINimum?



## **:READ:IQVTime:Q? (Query Only)**

Returns the Q versus Time trace data.

**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.

**Measurement Views** IQ versus Time

**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.

**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.

Use the [:SENSe]:IQVTime:TRACe:SElect:I or Q command to select the trace.

**Measurement Views** RF I&Q versus Time

**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.

**Related Commands** [:SENSe]:IQVTime:TRACe:SElect:I, [:SENSe]:IQVTime:TRACe:SElect:Q

## :READ:MCPower Subgroup

The :READ:MCPower commands returns the results of the MCPR (Multiple Carrier Power Ratio) measurement.

Command Tree	Header	Parameter
	:READ	
	:MCPower?	
	:ADJacent	
	:CHANnels	
	:CHANnel	
	:POWER?	
	:MAIN	
	:CHANnels	
	:SPECTrum?	

## **:READ:MCPower:ADJacent:CHANnels? (Query Only)**

Returns the power of adjacent channels in order of increasing frequency.

**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:ADD:ADJacent command.

**Measurement Views** MCPR

**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.

**Related Commands** [:SENSe]:MCPower:CHANnel:ADD:ADJacent

**:READ:MCPower:CHANnel:POWer? (Query Only)**

Returns the reference power in the MCPR measurement.

**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.

**Measurement Views** MCPR

**Examples** :READ:MCPower:CHANnel:POWer?  
might return 4.227 indicating that the reference power is 4.227 dBm.

**Related Commands** [:SENSe]:MCPower:RCHannels commands, [:SENSe]:POWer:UNITs

## **:READ:MCPower:MAIN:CHANnels? (Query Only)**

Returns the power of main channels in order of increasing frequency.

**Syntax**     :READ: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.

**Measurement Views**   MCPR

**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.

**Related Commands**   [:SENSe]:MCPower:CHANnel:MAIN commands, [:SENSe]:POWer:UNITs

## :READ:MCPower:SPECTrum? (Query Only)

Returns spectrum trace data of the MCPR measurement.

**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.

**Measurement Views** MCPR

**Examples** :READ:MCPower:SPECTrum?  
might return #43204xxxx... (3204-byte data) for the spectrum trace data of the MCPR measurement.

**Related Commands** [:SENSE]:POWER:UNITs

## :READ:MERRor Subgroup (Option 21)

The :READ:MERRor commands returns the results of the Magnitude error versus Time measurement.

Command Tree	Header	Parameter
	:READ	
	:MERRor	
	:FERRor?	
	:PEAK?	
	:PINDEX?	
	:RMS?	
	:TRACe?	

## :READ:MERRor:FERRor? (Query Only)

Returns the frequency error in the Magnitude error versus Time measurement.

**Syntax**     :READ:MERRor:FERRor?

**Arguments**   None

**Returns**     <freq\_error>::=<Nrf> is the frequency error in Hz.

**Measurement Views**   Magnitude error versus Time

**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.

**Syntax** :READ:MERRor:PEAK?

**Arguments** None

**Returns** <peak>::=<NRf> is the peak magnitude error in percent (%).

**Measurement Views** Magnitude error versus Time

**Examples** :READ:MERRor:PEAK?  
might return 1.57 indicating the peak magnitude error is 1.57%.

**Related Commands** :READ:MERRor:PINDEX?

**:READ:MERRor:PINDEX? (Query Only)**

Returns the time at the magnitude error peak.

**Syntax** :READ:MERRor:PINDEX?

**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.

**Measurement Views** Magnitude error versus Time

**Examples** :READ:MERRor:PINDEX?  
might return 68.000, indicating that the magnitude error peak is at symbol #68.

**Related Commands** :READ:MERRor:PEAK?, [:SENSe]:DDEMod:TIME:UNITs

## **:READ:MERRor:RMS? (Query Only)**

Returns the RMS (Root-Mean-Square) value in the Magnitude error versus Time measurement.

**Syntax** :READ:MERRor:RMS?

**Arguments** None

**Returns** <rms>::=<NRf> is the RMS magnitude error in percent (%).

**Measurement Views** Magnitude error versus Time

**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.

**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.

**Measurement Views** Magnitude error versus Time

**Examples** :READ:MERRor:TRACe?  
might return #42036xxxx... (2036-byte data) for the Magnitude error versus Time trace.

## :READ:OBWidth Subgroup

The :READ:OBWidth commands returns the results of the Occupied Bandwidth (OBW) measurement.

Command Tree	Header	Parameter
	:READ	
	:OBWidth	
	:FREQuency	
	:ERRor?	
	:OBWidth	
	:BANDwidth?	
	:LEFT	
	:FREQuency?	
	:LEVe1?	
	:POWer?	
	:RIGHt	
	:FREQuency?	
	:LEVe1?	
	:SPECTrum?	
	:XDBBandwidth	
	:BANDwidth?	
	:LEFT	
	:FREQuency?	
	:LEVe1?	
	:POWer?	
	:RIGHt	
	:FREQuency?	
	:LEVe1?	

## **:READ:OBWidth:FREQuency:ERRor? (Query Only)**

Returns the frequency error in the Occupied Bandwidth measurement.

**Syntax** :READ:OBWidth:FREQuency:ERRor?

**Arguments** None

**Returns** <freq\_error>::=<NRf> is the frequency error in Hz.

**Measurement Views** Occupied Bandwidth

**Examples** :READ:MERRor:FERRor?  
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.

**Syntax** :READ:OBWidth:OBWidth:BANDwidth?

**Arguments** None

**Returns** <OBW>::=<NRf> is the occupied bandwidth in Hz.

**Measurement Views** Occupied Bandwidth

**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.

**Syntax** :READ:OBWidth:OBWidth:LEFT:FREQUENCY?

**Arguments** None

**Returns** <OB\_left\_freq>::=<Nrf> is the left frequency in Hz.

**Measurement Views** Occupied Bandwidth

**Examples** :READ:OBWidth:OBWidth:LEFT:FREQUENCY?  
might return 1.498E+9 indicating the left frequency is 1.498 GHz.

**Related Commands** :READ:OBWidth:OBWidth:RIGHT:FREQUENCY?

**:READ:OBWidth:OBWidth:LEFT:LEVEL? (Query Only)**

Returns the level at the left frequency of the occupied bandwidth.

**Syntax** :READ:OBWidth:OBWidth:LEFT:LEVEL?

**Arguments** None

**Returns** <OB\_left\_level>::=<Nrf> is the level at the left frequency in dB.

**Measurement Views** Occupied Bandwidth

**Examples** :READ:OBWidth:OBWidth:LEFT:LEVEL?  
might return -23.5 indicating the level at the left frequency is -23.5 dB.

**Related Commands** :READ:OBWidth:OBWidth:RIGHT:LEVEL?

## **:READ:OBWidth:OBWidth:POWer? (Query Only)**

Returns the reference power in the Occupied Bandwidth measurement.

**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.

**Measurement Views**    Occupied Bandwidth

**Examples**      :READ:OBWidth:OBWidth:POWer?  
might return -10.0 indicating the reference power is -10 dBm.

**Related Commands**    [:SENSE]:POWer:UNITs

## **:READ:OBWidth:OBWidth:RIGHT:FREQUENCY? (Query Only)**

Returns the right (higher) frequency of the occupied bandwidth.

**Syntax** :READ:OBWidth:OBWidth:RIGHT:FREQUENCY?

**Arguments** None

**Returns** <OB\_right\_freq>::=<NRf> is the right frequency in Hz.

**Measurement Views** Occupied Bandwidth

**Examples** :READ:OBWidth:OBWidth:RIGHT:FREQUENCY?  
might return 1.502E+9 indicating the right frequency is 1.502 GHz.

**Related Commands** :READ:OBWidth:OBWidth:LEFT:FREQUENCY?

## **:READ:OBWidth:OBWidth:RIGHT:LEVEL? (Query Only)**

Returns the level at the right frequency of the occupied bandwidth.

**Syntax** :READ:OBWidth:OBWidth:RIGHT:LEVEL?

**Arguments** None

**Returns** <OB\_right\_level>::=<NRf> is the level at the right frequency in dB.

**Measurement Views** Occupied Bandwidth

**Examples** :READ:OBWidth:OBWidth:RIGHT:LEVEL?  
might return -23.5 indicating the level at the right frequency is -23.5 dB.

**Related Commands** :READ:OBWidth:OBWidth:LEFT:LEVEL?

## **:READ:OBWidth:SPECTrum? (Query Only)**

Returns spectrum trace data of the Occupied Bandwidth measurement.

**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.

**Measurement Views** Occupied Bandwidth

**Examples** :READ:OBWidth:SPECTrum?  
might return #43204xxxx... (3204-byte data) for the spectrum trace data of the Occupied Bandwidth measurement.

**Related Commands** [:SENSE]:POWER:UNITs

## **:READ:OBWidth:XDBBandwidth:BANDwidth? (Query Only)**

Returns the x dB bandwidth in the Occupied Bandwidth measurement.

**Syntax** :READ:OBWidth:XDBBandwidth:BANDwidth?

**Arguments** None

**Returns** <xdBW>::=<NRf> is the x dB bandwidth in Hz.

**Measurement Views** Occupied Bandwidth

**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.

**Syntax** :READ:OBWidth:XDBBandwidth:LEFT:FREQUENCY?

**Arguments** None

**Returns** <xdbbw\_left\_freq>::=<NRf> is the left frequency in Hz.

**Measurement Views** Occupied Bandwidth

**Examples** :READ:OBWidth:XDBBandwidth:LEFT:FREQUENCY?  
might return 1.498E+9 indicating the left frequency is 1.498 GHz.

**Related Commands** :READ:OBWidth:XDBBandwidth:RIGHT:FREQUENCY?

**:READ:OBWidth:XDBBandwidth:LEFT:LEVEL? (Query Only)**

Returns the level at the left frequency of the x dB bandwidth.

**Syntax** :READ:OBWidth:XDBBandwidth:LEFT:LEVEL?

**Arguments** None

**Returns** <xdbbw\_left\_level>::=<NRf> is the level at the left frequency in dB.

**Measurement Views** Occupied Bandwidth

**Examples** :READ:OBWidth:XDBBandwidth:LEFT:LEVEL?  
might return -23.5 indicating the level at the left frequency is -23.5 dB.

**Related Commands** :READ:OBWidth:XDBBandwidth:RIGHT:LEVEL?

## **:READ:OBWidth:XDBBandwidth:POWer? (Query Only)**

Returns the reference power in the x dB bandwidth measurement.

**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.

**Measurement Views**   Occupied Bandwidth

**Examples**     :READ:OBWidth:XDBBandwidth:POWer?  
might return -10.0 indicating the reference power is -10 dBm.

**Related Commands**   [:SENSE]:POWer:UNITs

**:READ:OBWidth:XDBBandwidth:RIGHT:FREQUENCY? (Query Only)**

Returns the right (higher) frequency of the x dB bandwidth.

**Syntax** :READ:OBWidth:XDBBandwidth:RIGHT:FREQUENCY?

**Arguments** None

**Returns** <xdbbw\_right\_freq>::=<NRf> is the right frequency in Hz.

**Measurement Views** Occupied Bandwidth

**Examples** :READ:OBWidth:XDBBandwidth:RIGHT:FREQUENCY?  
might return 1.502E+9 indicating the right frequency is 1.502 GHz.

**Related Commands** :READ:OBWidth:XDBBandwidth:LEFT:FREQUENCY?

**:READ:OBWidth:XDBBandwidth:RIGHT:LEVEL? (Query Only)**

Returns the level at the right frequency of the x dB bandwidth.

**Syntax** :READ:OBWidth:XDBBandwidth:RIGHT:LEVEL?

**Arguments** None

**Returns** <xdbbw\_right\_level>::=<NRf> is the level at the right frequency in dB.

**Measurement Views** Occupied Bandwidth

**Examples** :READ:OBWidth:XDBBandwidth:RIGHT:LEVEL?  
might return -23.5 indicating the level at the right frequency is -23.5 dB.

**Related Commands** :READ:OBWidth:XDBBandwidth:LEFT:LEVEL?

## :READ:PERRor Subgroup (Option 21)

The :READ:PERRor commands returns the results of the Phase error versus Time measurement.

Command Tree	Header	Parameter
	:READ	
	:PERRor	
	:FERRor?	
	:PEAK?	
	:PINDex?	
	:RMS?	
	:TRACe?	

## :READ:PERRor:FERRor? (Query Only)

Returns the frequency error in the Phase error versus Time measurement.

**Syntax**     :READ:PERRor:FERRor?

**Arguments**   None

**Returns**     <freq\_error>::=<Nrf> is the frequency error in Hz.

**Measurement Views**   Phase error versus Time

**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.

**Syntax** :READ:PERRor:PEAK?

**Arguments** None

**Returns** <peak>::=<NRf> is the peak phase error in degrees.

**Measurement Views** Phase error versus Time

**Examples** :READ:PERRor:PEAK?  
might return 0.683 indicating the peak phase error is 0.683°.

**Related Commands** :READ:PERRor:PINDex?

## **:READ:PERRor:PINDex? (Query Only)**

Returns the time at the phase error peak.

**Syntax** :READ:PERRor:PINDex?

**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.

**Measurement Views** Phase error versus Time

**Examples** :READ:PERRor:PINDex?  
might return 68.000, indicating that the phase error peak is at symbol #68.

**Related Commands** :READ:PERRor:PEAK?, [:SENSe]:DDEMod:TIME:UNITs

## **:READ:PERRor:RMS? (Query Only)**

Returns the RMS (Root-Mean-Square) value in the Phase error versus Time measurement.

**Syntax** :READ:PERRor:RMS?

**Arguments** None

**Returns** <rms>::=<NRf> is the RMS phase error in degrees.

**Measurement Views** Phase error versus Time

**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.

**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.

**Measurement Views** Phase error versus Time

**Examples** :READ:PERRor:TRACe?  
might return #42036xxxx... (2036-byte data) for the Phase error versus Time trace.

## :READ:PHVTime Subgroup

The :READ:PHVTime commands returns the results of the Phase versus Time measurement.

Command Tree	Header	Parameter
	:READ	
	:PHVTime?	
	:MAXimum?	
	:MAXLocation?	
	:MINimum?	
	:MINLocation?	
	:RESult?	

## :READ:PHVTime? (Query Only)

Returns the Phase versus Time trace data.

**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.

**Measurement Views** Phase versus Time

**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.

**Syntax** :READ:PHVTime:MAXimum?

**Arguments** None

**Returns** <max>::=<NRf> is the maximum phase in degrees.

**Measurement Views** Phase versus Time

**Examples** :READ:PHVTime:MAXimum?  
might return 153.8 indicating the maximum phase is 153.8°.

**Related Commands** :READ:PHVTime:MAXLocation?

## **:READ:PHVTime:MAXLocation? (Query Only)**

Returns the time at which the phase is maximum.

**Syntax** :READ:PHVTime:MAXLocation?

**Arguments** None

**Returns** <max\_time>::=<NRf> is the time in seconds at which the phase is maximum.

**Measurement Views** Phase versus Time

**Examples** :READ:PHVTime:MAXLocation?  
might return 175.3E-9 indicating the I or Q level is maximum at 175.3 ns.

**Related Commands** :READ:PHVTime:MAXimum?



## :READ:PHVTime:MINimum? (Query Only)

Returns the minimum value in the Phase versus Time measurement.

**Syntax** :READ:PHVTime:MINimum?

**Arguments** None

**Returns** <min>::=<NRf> is the minimum phase in degrees.

**Measurement Views** Phase versus Time

**Examples** :READ:PHVTime:MINimum?  
might return -176.3 indicating the minimum phase is -176.3°.

**Related Commands** :READ:PHVTime:MINLocation?

## :READ:PHVTime:MINLocation? (Query Only)

Returns the time at which the phase is minimum.

**Syntax** :READ:PHVTime:MINLocation?

**Arguments** None

**Returns** <min\_time>::=<NRf> is the time in seconds at which the phase is minimum.

**Measurement Views** Phase versus Time

**Examples** :READ:PHVTime:MAXLocation?  
might return 450.7E-9 indicating the phase is minimum at 450.7 ns.

**Related Commands** :READ:PHVTime:MINimum?

**:READ:PHVTime:RESult? (Query Only)**

Returns the Phase versus Time measurement results.

**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.

**Measurement Views** Phase versus Time

**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 Subgroup (Option 11 Only)

The :READ:PNOise commands returns results of the Phase Noise measurement.

Command Tree	Header	Parameter
	:READ	
	:PNOise	
	:ALL	
	:CARRier	
	:FREQuency?	
	:POWer?	
	:JITTer?	
	:RESidual	
	:FM?	
	:RMS	
	:PNOise?	
	:SPECTrum<x>	
	:X?	
	:XY?	
	[:Y]?	

## **:READ:PNOise:ALL? (Query Only)**

Returns all results of the phase noise measurement.

**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.

**Measurement Views** Phase noise

**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  
     Residual FM: 14.58 Hz

**:READ:PNOise:CARRier:FERRor? (Query Only)**

Returns the carrier frequency error in the phase noise measurement.

**Syntax** :READ:PNOise:CARRier:FERRor?

**Arguments** None

**Returns** <NRf> Carrier frequency error in Hz.

**Measurement Views** Phase noise

**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.

**Syntax** :READ:PNOise:CARRier:POWer?

**Arguments** None

**Returns** <NRf> Carrier power in dBm.  
The unit can be changed by the [:SENSe]:POWer:UNITs command.

**Measurement Views** Phase noise

**Examples** :READ:PNOise:CARRier:POWer?  
might return -9.455, indicating that the carrier power is -9.455 dBm.

**Related Commands** [:SENSe]:POWer:UNITs

### **:READ:PNOise:JITTer? (Query Only)**

Returns the jitter in the phase noise measurement.

**Syntax** :READ:PNOise:JITTer?

**Arguments** None

**Returns** <NRf> Jitter in seconds.

**Measurement Views** Phase noise

**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.

**Syntax** :READ:PNOise:RESidual:FM?

**Arguments** None

**Returns** <NRf> Residual FM in Hz.

**Measurement Views** Phase noise

**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.

**Syntax** :READ:PNOise:RMS:PNOise?

**Arguments** None

**Returns** <NRf> RMS phase noise in degrees.

**Measurement Views** Phase noise

**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.

**Syntax** :READ:PNOise:SPECTrum<x>:X?

Where <x> = 1 and 2, representing Trace 1 and Trace 2, respectively.

**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.

**Measurement Views** Phase noise

**Examples** :READ:PNOise:SPECTrum1:X?  
might return #516020xxxx... (16020-byte data) for the frequencies of Trace 1.

**Related Commands** :READ:PNOise:SPECTrum<x>[:Y]?

## **:READ:PNOise:SPECTrum<x>:XY? (Query Only)**

Returns the frequency and phase noise pairs of the specified trace.

**Syntax** :READ:PNOise:SPECTrum<x>:XY?

Where <x> = 1 and 2, representing Trace 1 and Trace 2, respectively.

**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.

**Measurement Views** Phase noise

**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.

**Syntax** :READ:PNOise:SPECTrum<x>[:Y]?

Where <x> = 1 and 2, representing Trace 1 and Trace 2, respectively.

**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.

**Measurement Views** Phase noise

**Examples** :READ:PNOise:SPECTrum1:Y?

might return #516020xxxx... (16020-byte data) for the phase noise values of Trace 1.

**Related Commands** :READ:PNOise:SPECTrum<x>[:Y]?

## **:READ:PULSe Subgroup (Option 20 Only)**

The :READ:PULSe commands returns the results of the Pulse Measurements.

<b>Command Tree</b>	<b>Header</b>	<b>Parameter</b>
	:READ	
	:PULSe	
	[:RESuLt]	
	:ATX?	
	:AVERAge?	
	:DROOp?	
	:DUTPct?	
	:DUTRatio?	
	:FALL?	
	:FRDeviation?	
	:MFReqerror?	
	:MPHerror?	
	:PHDeviation?	
	:PPFRequency?	
	:PPOWer?	
	:PPPHas?	
	:RINTerval?	
	:RIPPlE?	
	:RISE?	
	:RMSFreqerror?	
	:RMSPherror?	
	:RRATe?	
	:TIME?	
	:WIDTh?	
	:STATistics?	
	:ATX?	
	:AVERAge?	
	:DROOp?	
	:DUTPct?	
	:DUTRatio?	
	:FALL?	
	:FRDeviation?	
	:MFReqerror?	
	:MPHerror?	
	:PHDeviation?	
	:PPFRequency?	
	:PPOWer?	
	:PPPHas?	
	:RINTerval?	
	:RIPPlE?	
	:RISE?	

```

:RMSFreqerror?
:RMSPherror?
:RRATE?
:WIDTh?
:TRACe
:X?
:XY?
[:Y]?

```

## **:READ:PULSe[:RESult]:ATX? (Query Only)**

Returns the average transmitted power in the pulse table.

**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.

**Measurement Views** Pulse table

**Examples** :READ:PULSe:RESult:ATX?  
might return 0,-18.57,-18.73,-18.20,-18.53 for Pulse 0 to 3.

**Related Commands** [:SENSe]:POWer:UNITs

## **:READ:PULSE[:RESult]:AVERage? (Query Only)**

Returns the average on power in the pulse table.

**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.

**Measurement Views**   Pulse table

**Examples**     :READ:PULSE:RESult:AVERage?  
might return 0,-2.354,-2.368,-2.343,-2.358 for Pulse 0 to 3.

**Related Commands**   [:SENSe]:POWer:UNITs

**:READ:PULSe[:RESult]:DROop? (Query Only)**

Returns the average on power in the pulse table.

**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 (%).

**Measurement Views** Pulse table

**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.

**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 (%).

**Measurement Views** Pulse table

**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.

**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).

**Measurement Views** Pulse table

**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.

**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.

**Measurement Views** Pulse table

**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.

**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.

**Measurement Views** Pulse table

**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]:MFRReqerror? (Query Only)**

Returns the maximum frequency error in the pulse table.

**Syntax** :READ:PULSe[:RESult]:MFRReqerror?

**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.

**Measurement Views** Pulse table

**Examples** :READ:PULSe:RESult:MFRReqerror?  
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.

**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.

**Measurement Views** Pulse table

**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.

**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.

**Measurement Views**   Pulse table

**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.

**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.

**Measurement Views** Pulse table

**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.

**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.

**Measurement Views**   Pulse table

**Examples**         :READ:PULSe:RESult:PPOWer?  
                   might return 0, -2.26, -2.27, -2.23, -2.25 for Pulse 0 to 3.

**Related Commands**   [:SENSe]:POWer:UNITs

**:READ:PULSE[:RESult]:PPHase? (Query Only)**

Returns the pulse-pulse carrier phase in the pulse table.

**Syntax** :READ:PULSE[:RESult]:PPHase?

**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.

**Measurement Views** Pulse table

**Examples** :READ:PULSE:RESult:PPHase?  
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.

**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.

**Measurement Views** Pulse table

**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.

**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 (%).

**Measurement Views** Pulse table

**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.

**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.

**Measurement Views** Pulse table

**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.

**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.

**Measurement Views** Pulse table

**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.

**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.

**Measurement Views**   Pulse table

**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.

**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.

**Measurement Views** Pulse table

**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.

**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.

**Measurement Views** Pulse table

**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.

**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.

**Measurement Views** Pulse table

**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.

---

**Syntax** :READ: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.

**Measurement Views** Pulse statistics

**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.

**Related Commands** :DISPlay:PULSe:SElect:RESult, :DISPlay:PULSe:STATistics:PLOT

## **: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.

**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.

**Measurement Views** Pulse statistics

**Examples** :READ:PULSe:STATistics:ATX?  
might return -18.51,-18.74,-18.12 for the average transmitted power in the pulse statistics.

**Related Commands** :DISPlay:PULSe:STATistics:PLOT, [:SENSe]:POWer:UNITs

## **: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.

**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.

**Measurement Views** Pulse statistics

**Examples** :READ:PULSe:STATistics:AVERage?  
might return -2.35, -2.36, -2.34 for the average on power in the pulse statistics.

**Related Commands** :DISPlay:PULSe:STATistics:PLOT, [:SENSe]:POWer:UNITs

## **:READ:PULSe:STATistics:DR0op? (Query Only)**

Returns the droop in the pulse statistics.

This command is valid when :DISPlay:PULSe:STATistics:PLOT is set to TREND.

**Syntax** :READ:PULSe:STATistics:DR0op?

**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.

**Measurement Views** Pulse statistics

**Examples** :READ:PULSe:STATistics:DR0op?  
might return 22.67E-3, -613.5E-3, 633.8E-3 for the droop in the pulse statistics.

**Related Commands** :DISPlay:PULSe:STATistics:PLOT

## **: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.

**Syntax** :READ: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 (%).

**Measurement Views** Pulse statistics

**Examples** :READ:PULSe:STATistics:DUTPct?  
might return 2.437,2.310,2.657 for the duty factor (%) in the pulse statistics.

**Related Commands** :DISPlay:PULSe:STATistics:PLOT

**: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.

**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.

**Measurement Views** Pulse statistics

**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.

**Related Commands** :DISPlay:PULSe:STATistics:PLOT



## **: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.

**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.

**Measurement Views** Pulse statistics

**Examples** :READ:PULSe:STATistics:FALL?  
might return 70.27E-9,69.62E-9,71.27E-9 for the fall time in the pulse statistics.

**Related Commands** :DISPlay:PULSe:STATistics:PLOT

## **: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.

**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.

**Measurement Views** Pulse statistics

**Examples** :READ:PULSe:STATistics:FRDeviation?  
might return 754.1E+3,660.5E+3,835.2E+3 for the frequency deviation in the pulse statistics.

**Related Commands** :DISPlay:PULSe:STATistics:PLOT

## **: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.

**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.

**Measurement Views** Pulse statistics

**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.

**Related Commands** :DISPlay:PULSe:STATistics:PLOT

## **: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.

**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.

**Measurement Views** Pulse statistics

**Examples** :READ:PULSe:STATistics:MPHerror?  
might return -11.47,-17.18,-7.61 for the maximum phase error in the pulse statistics.

**Related Commands** :DISPlay:PULSe:STATistics:PLOT

## **: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.

**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.

**Measurement Views** Pulse statistics

**Examples** :READ:PULSe:STATistics:PHDeviation?  
might return 11.678,7.694,17.374 for the phase deviation in the pulse statistics.

**Related Commands** :DISPlay:PULSe:STATistics:PLOT

**: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.

**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.

**Measurement Views** Pulse statistics

**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.

**Related Commands** :DISPlay:PULSe:STATistics:PLOT

## **: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.

**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.

**Measurement Views** Pulse statistics

**Examples** :READ:PULSe:STATistics:PPOWer?  
might return -2.273,-2.313,-2.235 for the peak power in the pulse statistics.

**Related Commands** :DISPlay:PULSe:STATistics:PLOT, [:SENSe]:POWer:UNITs

## **: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.

**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.

**Measurement Views** Pulse statistics

**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.

**Related Commands** :DISPlay:PULSe:STATistics:PLOT



## **: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.

**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.

**Measurement Views** Pulse statistics

**Examples** :READ:PULSe:STATistics:RINTerval?  
might return 240.5E-6,217.9E-6,281.2E-6 for the repetition interval in the pulse statistics.

**Related Commands** :DISPlay:PULSe:STATistics:PLOT

## **: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.

**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.

**Measurement Views** Pulse statistics

**Examples** :READ:PULSe:STATistics:RIPPlE?  
might return 160.4E-3,83.78E-3,287.7E-3 for the ripple in the pulse statistics.

**Related Commands** :DISPlay:PULSe:STATistics:PLOT

## **: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.

**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.

**Measurement Views** Pulse statistics

**Examples** :READ:PULSe:STATistics:RISE?  
might return 105.4E-9,91.65E-9,116.2E-9 for the rise time in the pulse statistics.

**Related Commands** :DISPlay:PULSe:STATistics:PLOT

## **: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.

**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.

**Measurement Views** Pulse statistics

**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.

**Related Commands** :DISPlay:PULSe:STATistics:PLOT

## **: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.

**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.

**Measurement Views** Pulse statistics

**Examples** :READ:PULSe:STATistics:RMSPherror?  
might return 1.032,604.5E-3,1.606 for the RMS phase error in the pulse statistics.

**Related Commands** :DISPlay:PULSe:STATistics:PLOT

**: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.

**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.

**Measurement Views** Pulse statistics

**Examples** :READ:PULSe:STATistics:RRATe?  
might return 62.50E+3,62.49E+3,62.52E+3 for the repetition rate in the pulse statistics.

**Related Commands** :DISPlay:PULSe:STATistics:PLOT

## **: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.

**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.

**Measurement Views** Pulse statistics

**Examples** :READ:PULSe:STATistics:WIDTh?  
might return 4.636E-6,4.630E-6,4.643E-6 for the pulse width in the pulse statistics.

**Related Commands** :DISPlay:PULSe:STATistics:PLOT

## **: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.

**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.

**Measurement Views** Pulse trace

**Examples** :READ:PULSe:TRACe:X?  
might return #43204xxxx... (3204-byte data) for the time values of the trace.

**Related Commands** :DISPlay:PULSe:SElect:NUMBer, :DISPlay:PULSe:SElect:RESult,  
:READ:PULSe:TRACe[:Y]?



## **: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.

**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.

**Measurement Views** Pulse trace

**Examples** :READ:PULSe:TRACe:XY?  
might return #43204xxxx... (3204-byte data) for the horizontal (time) and vertical value pairs of the pulse trace.

**Related Commands** :DISPlay:PULSe:SElect:NUMBer, :DISPlay:PULSe:SElect:RESult, [:SENSe]:POWer:UNITs

## **: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.

**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.

**Measurement Views** Pulse trace

**Examples** :READ:PULSe:TRACe:Y?  
might return #43204xxxx... (3204-byte data) for the vertical values of the pulse trace.

**Related Commands** :DISPlay:PULSe:SElect:NUMBer, :DISPlay:PULSe:SElect:RESult,  
:READ:PULSe:TRACe:X?, [:SENSE]:POWer:UNITs

## :READ:SGRam Subgroup

The :READ:SGRam commands returns the results of the spectrogram measurement.

Command Tree	Header	Parameter
	:READ	
	:SGRam?	

## :READ:SGRam? (Query Only)

Returns the spectrogram trace data.  
The line is selected using the :TRACe:SGRam:SElect:LINE command.

**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.

**Measurement Views** Spectrogram

**Examples** :READ:SGRam:TRACe1?  
might return #43204xxxx... (3204-byte data) for the spectrogram trace.

**Related Commands** :TRACe:SGRam:SElect:LINE, [:SENSE]:POWER:UNITs

## **:READ:SPECTrum Subgroup**

The :READ:SPECTrum commands returns results of the Spectrum measurement.

<b>Command Tree</b>	<b>Header</b>	<b>Parameter</b>
	:READ	
	:SPECTrum	
	:TRACe<x>?	

Where <x> = 1 to 5. Each trace is defined as follows:

- TRACe1: Trace 1
- TRACe2: Trace 2
- TRACe3: Trace 3
- TRACe4: Math trace
- TRACe5: Spectrogram

The traces can be specified by the :TRACe<x>:SPECTrum command subgroup.

**:READ:SPECTrum:TRACe<x>? (Query Only)**

Returns the trace data in the Spectrum measurement.

**Syntax** :READ:SPECTrum:TRACe<x>?

Where <x> = 1 to 5.

---

**NOTE.** TRACe5 (spectrogram) is valid when the spectrum and spectrogram measurements are running.

---

**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.

**Measurement Views** Spectrum

**Examples** :READ:SPECTrum:TRACe1?  
might return #43204xxxx... (3204-byte data) for Trace 1 in the Spectrum measurement.

**Related Commands** :TRACe<x>:SPECTrum command subgroup, [:SENSe]:POWer:UNITs

## :READ:SPURious Subgroup

The :READ:SPURious commands returns results of the Spurious measurement.

Command Tree	Header	Parameter
	:READ	
	:SPURious	
	:CARRier	
	:POWer?	
	:COUnT?	
	:PASS?	
	:SPECTrum	
	:X?	
	:XY?	
	[:Y]?	
	:SPUR<x>	
	:AMPLitude	
	:ABSolute?	
	:RELative?	
	:FREQuency	
	:ABSolute?	
	:RELative?	
	:LIMit	
	:ABSolute?	
	:RELative?	
	:VIOLation?	
	:RANGe?	

Where <x> represents a spurious signal number from 1 to the total number of spurious signals.

## **:READ:SPURious:CARRier:POWer? (Query Only)**

Returns the carrier power in the Spurious measurement.

**Syntax** :READ:SPURious:CARRier:POWer?

**Arguments** None

**Returns** <NRf> Carrier power in dBm.  
The unit can be changed by the [:SENSe]:POWer:UNITs command.

**Measurement Views** Spurious

**Examples** :READ:SPURious:CARRier:POWer?  
might return 4.227, indicating that the carrier power is 4.227 dBm.

**Related Commands** [:SENSe]:POWer:UNITs

## **:READ:SPURious:COUNT? (Query Only)**

Returns the number of spurious signals in the Spurious measurement.

**Syntax** :READ:SPURious:COUNT?

**Arguments** None

**Returns** <NRf> The spurious count.

**Measurement Views** Spurious

**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.

**Syntax** :READ:SPURious:PASS?

**Arguments** None

**Returns** 0 (fail) or 1 (pass).

**Measurement Views** Spurious

**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.

**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.

**Measurement Views** Spurious

**Examples** :READ:SPURious:SPECTrum:X?  
might return #516020xxxx... (16020-byte data) for the frequencies of the spectrum trace in the Spurious measurement.

**Related Commands** :READ:SPURious:SPECTrum[:Y]?, [:SENSe]:POWer:UNITs



## **:READ:SPURious:SPECTrum:XY? (Query Only)**

Returns the frequency and amplitude pairs of the spectrum trace in the Spurious measurement.

**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.

**Measurement Views** Spurious

**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.

**Related Commands** [:SENSE]:POWER:UNITs

## **:READ:SPURious:SPECTrum[:Y]? (Query Only)**

Returns the amplitudes of the spectrum trace in the Spurious measurement.

**Syntax**     :READ: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.

**Measurement Views**   Spurious

**Examples**     :READ:SPURious:SPECTrum:Y?  
 might return #516020xxxx... (16020-byte data) for the amplitudes of the spectrum trace in the Spurious measurement.

**Related Commands**   :READ:SPURious:SPECTrum:X?, [:SENSe]:POWer:UNITs

**:READ:SPURious:SPUR<x>:AMPLitude:ABSolute? (Query Only)**

Returns the absolute amplitude of the specified spurious signal in the Spurious measurement.

**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.

**Measurement Views** Spurious

**Examples** :READ:SPURious:SPUR1:AMPLitude:ABSolute?  
might return -19.782, indicating that the absolute amplitude of Spurious #1 is -19.782 dBm.

**Related Commands** [:SENSE]:POWER:UNITs

**:READ:SPURious:SPUR<x>:AMPLitude:RELative? (Query Only)**

Returns the relative amplitude of the specified spurious signal in the Spurious measurement.

**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.

**Measurement Views** Spurious

**Examples** :READ:SPURious:SPUR1:AMPLitude:RELative?  
might return -9.782, indicating that the relative amplitude of Spurious #1 is -9.782 dB.

**Related Commands** [:SENSe]:SPURious:REFeRence

**:READ:SPURious:SPUR<x>:FREQUENCY:ABSolute? (Query Only)**

Returns the absolute frequency of the specified spurious signal in the Spurious measurement.

**Syntax** :READ:SPURious:SPUR<x>:FREQUENCY:ABSolute?

**Arguments** None

**Returns** <NRf> Absolute frequency of the spurious signal in Hz.

**Measurement Views** Spurious

**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.

**Syntax** :READ:SPURious:SPUR<x>:FREQUENCY:RELative?

**Arguments** None

**Returns** <NRf> Relative frequency of the spurious signal to the carrier in Hz.

**Measurement Views** Spurious

**Examples** :READ:SPURious:SPUR1:FREQUENCY:RELative?  
might return 3.634E+6, indicating that the relative frequency of Spurious #1 is 3.634 MHz.

**Related Commands** [:SENSe]:SPURious:REFEreNce

**:READ:SPURious:SPUR<x>:LIMit:ABSolute? (Query Only)**

Returns the absolute amplitude of the limit for the specified spurious signal in the Spurious measurement.

**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.

**Measurement Views** Spurious

**Examples** :READ:SPURious:SPUR1:LIMit:ABSolute?  
might return -50.0, indicating that the absolute amplitude of the limit for Spurious #1 is -50 dBm.

**Related Commands** [:SENSe]:POWer:UNITs

**:READ:SPURious:SPUR<x>:LIMit:RELative? (Query Only)**

Returns the relative amplitude of the limit for the specified spurious signal in the Spurious measurement.

**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.

**Measurement Views** Spurious

**Examples** :READ:SPURious:SPUR1:LIMit:RELative?  
might return -10.0, indicating that the relative amplitude of the limit for Spurious #1 is -10 dB.

**Related Commands** [:SENSE]:SPURious:REFerence

## **:READ:SPURious:SPUR<x>:LIMit:VIOLation? (Query Only)**

Returns whether the specified spurious signal exceeds the limit or not.

**Syntax** :READ:SPURious:SPUR<x>:LIMit:VIOLation?

**Arguments** None

**Returns** 0 (under the limit) or 1 (over the limit).

**Measurement Views** Spurious

**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.

**Syntax** :READ:SPURious:SPUR<x>:RANGe?

**Arguments** None

**Returns** <string> "A" to "T" representing Range A to T, respectively.

**Measurement Views** Spurious

**Examples** :READ:SPURious:SPUR1:RANGe?  
might return "E", indicating that Spurious #1 is in Range E.



## :READ:SQuality Subgroup (Option 21)

The :READ:SQuality commands return the results of the signal quality measurement.

Command Tree	Header	Parameter
	:READ	
	:SQuality	
	:FREQuency	
	:ERRor?	
	:GAIN	
	:IMBalance?	
	:ORIGin	
	:OFFSet?	
	:PEAK	
	:EVM?	
	:DB?	
	:LOCation?	
	:MERRor?	
	:DB?	
	:LOCation?	
	:PERRor?	
	:LOCation?	
	:QUADrature	
	:ERRor?	
	:RHO	
	:RMS	
	:EVM?	
	:DB?	
	:MER	
	:DB?	
	:MERRor?	
	:DB?	
	:PERRor?	

### **:READ:SQUality:FREQuency:ERRor? (Query Only)**

Returns the frequency error in the signal quality measurement.

**Syntax** :READ:SQUality:FREQuency:ERRor?

**Arguments** None

**Returns** <NRf> Frequency error in Hz.

**Measurement Views** Signal quality

**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.

**Syntax** :READ:SQUality:GAIN:IMBalance?

**Arguments** None

**Returns** <NRf> Gain imbalance in dB.

**Measurement Views** Signal quality

**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.

**Syntax** :READ:SQUality:ORIGin:OFFSet?

**Arguments** None

**Returns** <NRf> Origin offset in dB.

**Measurement Views** Signal quality

**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.

**Syntax** :READ:SQUality:PEAK:EVM?

**Arguments** None

**Returns** <NRf> Peak EVM in percent (%).

**Measurement Views** Signal quality

**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.

**Syntax** :READ:SQUality:PEAK:EVM:DB?

**Arguments** None

**Returns** <NRf> Peak EVM in dB.

**Measurement Views** Signal quality

**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.

**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.

**Measurement Views** Signal quality

**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.

**Syntax** :READ:SQUality:PEAK:MERRor?

**Arguments** None

**Returns** <NRf> Peak magnitude error in percent (%).

**Measurement Views** Signal quality

**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.

**Syntax** :READ:SQUality:PEAK:MERRor:DB?

**Arguments** None

**Returns** <NRf> Peak magnitude error in dB.

**Measurement Views** Signal quality

**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.

**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.

**Measurement Views**   Signal quality

**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.

**Syntax** :READ:SQUality:PEAK:PERRor?

**Arguments** None

**Returns** <NRf> Peak phase error in degrees.

**Measurement Views** Signal quality

**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.

**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.

**Measurement Views** Signal quality

**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.

**Syntax** :READ:SQUality:QUADrature:ERRor?

**Arguments** None

**Returns** <NRf> Quadrature error in degrees.

**Measurement Views** Signal quality

**Examples** :READ:SQUality:QUADrature:ERRor?  
might return  $-14.264E-3$ , indicating that the quadrature error is  $-0.014264^\circ$ .

## **:READ:SQUality:RHO? (Query Only)**

Returns the  $\rho$  (waveform quality) value in the signal quality measurement.

**Syntax** :READ:SQUality:RHO?

**Arguments** None

**Returns** <NRf>  $\rho$  value.

**Measurement Views** Signal quality

**Examples** :READ:SQUality:RHO?  
might return  $998.703E-3$ , indicating that  $\rho$  is 0.998703.



## **:READ:SQUality:RMS:EVM? (Query Only)**

Returns the RMS EVM (%) in the signal quality measurement.

**Syntax** :READ:SQUality:RMS:EVM?

**Arguments** None

**Returns** <NRf> RMS EVM in percent (%).

**Measurement Views** Signal quality

**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.

**Syntax** :READ:SQUality:RMS:EVM:DB?

**Arguments** None

**Returns** <NRf> RMS EVM in dB.

**Measurement Views** Signal quality

**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.

**Syntax**      :READ:SQUality:RMS:MER:DB?

**Arguments**    None

**Returns**      <NRf>      RMS MER in dB.

**Measurement Views**    Signal quality

**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.

**Syntax** :READ:SQUality:RMS:MERRor?

**Arguments** None

**Returns** <NRf> RMS magnitude error in percent (%).

**Measurement Views** Signal quality

**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.

**Syntax** :READ:SQUality:RMS:MERRor:DB?

**Arguments** None

**Returns** <NRf> RMS magnitude error in dB.

**Measurement Views** Signal quality

**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.

**Syntax** :READ:SQUality:RMS:PERRor?

**Arguments** None

**Returns** <NRf> RMS phase error in degrees.

**Measurement Views** Signal quality

**Examples** :READ:SQUality:RMS:PERRor?  
 might return 893.472E-3, indicating that the RMS phase error is 0.893472°.

## :READ:TOVerview Subgroup

The :READ:TOVerview commands returns the results of the time overview measurement.

Command Tree	Header	Parameter
	:READ	
	:TOVerview?	

## :READ:TOVerview? (Query Only)

Returns the trace data in the time overview.

**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.

**Measurement Views** Time overview

**Examples** :READ:TOVerview?  
might return #43204xxxx... (3204-byte data) for the trace in the time overview.

**Related Commands** [:SENSe]:POWer:UNITs



# SENSe Commands

The SENSe commands set the details for each of the measurement sessions. They are divided by measurement views into the subgroups listed in Table 2-40.

**Table 2-40: SENSe command subgroups**

Command header	Measurement view	Refer to:
[:SENSe] (basic)	All views, general analysis parameter control	page 2-660
[:SENSe]:ACPower	Channel power and ACPR	page 2-680
[:SENSe]:AVTime	Amplitude versus Time	page 2-692
[:SENSe]:CCDF	CCDF	page 2-697
[:SENSe]:DDEMod	General purpose digital modulation measurements (Option 21 only)	page 2-700
[:SENSe]:DPSA	DPX (Digital Phosphor) spectrum	page 2-713
[:SENSe]:FVTime	Frequency versus Time	page 2-725
[:SENSe]:IQVTime	IQ level versus Time	page 2-731
[:SENSe]:MCPower	MCPR (Multiple Carrier Power Ratio)	page 2-737
[:SENSe]:OBWidth	Occupied Bandwidth (OBW)	page 2-751
[:SENSe]:PHVTime	Phase versus Time	page 2-759
[:SENSe]:PNOise	Phase noise (Option 11 only)	page 2-765
[:SENSe]:PULSe	Pulsed RF measurements (Option 20 only)	page 2-772
[:SENSe]:SGRam	Spectrogram	page 2-788
[:SENSe]:SPECTrum	Spectrum	page 2-803
[:SENSe]:SPURious	Spurious	page 2-818
[:SENSe]:TOVerview	Time overview	page 2-846

## [[:SENSe]] Basic Command Subgroup

The [[:SENSe]] basic commands globally set up the analysis parameters.

Command Tree	Header	Parameter
	[[:SENSe]]	
	:ACQuisition	
	:BANDwidth :BWIDTH	<numeric_value>
	:FFRame	
	:ACTual?	
	:LIMit	<numeric_value>
	:STATe	<boolean>
	:MEMory	
	:AVAilable	
	:SAMPles?	
	:CAPacity	
	[:TIME]?	
	:USED	
	[:PERCent]?	
	:MODE	AUTO   SAMPles   LENGth
	:SAMPles	<numeric_value>
	:SEConds	<numeric_value>
	:ANALysis	
	:ADVanced	
	:DITHer	<boolean>
	:HWARE	
	:STATus?	
	:LENGth	<numeric_value>
	:ACTual?	
	:AUTO	<boolean>
	:REFerence	ACQSTART   TRIGGER
	:START	<numeric_value>
	:AUTO	<boolean>
	:MEASurement	
	:FREQuency	<numeric_value>
	:POWer	
	:UNITs	DBM   DBV   VOLTs   WATTs   DBUW   DBW   DBUV   DBMV   DBUA   DBUV_M   DBUA_M   AMPS
	:REANalyze	
	:ROSCillator	
	SOURce	INTernal   EXTernal



:SPECTrum	
:LENGth	<numeric_value>
:ACTual?	
:AUTO	<boolean>
:STARt	<numeric_value>
:TIME	
:MODE	INdependent   COMMOn
:USETtings	

**[[: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.

**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).

**Measurement Views** All

**Examples** :SENSe:ACQuisition:BANDwidth 30MHz  
sets the acquisition bandwidth to 30 MHz.

**Related Commands** [[:SENSe]:ACQuisition:MODE]

**[[:SENSe]:ACQuisition:FFRame:ACTual? (Query Only)**

Queries the actual number of Fast Frames.

**Syntax** [[:SENSe]:ACQuisition:FFRame:ACTual?

**Arguments** None

**Returns** <NR1> The actual number of Fast Frames.

**Measurement Views** All

**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.

**Syntax**    [:SENSE]:ACquisition:FFrame:Limit <value>  
              [:SENSE]:ACquisition:FFrame:Limit?

**Arguments**    <value>::=<NR1> specifies the fast frame limit.  
                  Range: 1 to 65535 frames.

**Measurement Views**    All

**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.

**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.

**Measurement Views**    All

**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.

**Syntax** [[:SENSe]:ACQuisition:MEMory:AVAIlable:SAMPles?

**Arguments** None

**Returns** <NRf> The amount of acquisition memory available in samples.

**Measurement Views** All

**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).

**Syntax** [[:SENSe]:ACQuisition:MEMory:CAPacity[:TIME]?

**Arguments** None

**Returns** <NRf> The acquisition memory capacity in seconds.

**Measurement Views** All

**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.

**Syntax**     [:SENSe]:ACQuisition:MEMory:USED[:PERCent]?

**Arguments**   None

**Returns**     <Nrf>     The percentage of the capacity used.

**Measurement Views**   All

**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).

**Syntax**     [:SENSe]:ACQuisition:MODE { AUTO | SAMPlEs | LENGth }  
[:SENSe]:ACQuisition:MODE?

**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.

**Measurement Views**   All

**Examples**     :SENSe:ACQuisition:MODE AUTO  
sets the all sampling parameters automatically.

**Related Commands**   [:SENSe]:ACQuisition:BANDwidth|:BWIDth,  
[:SENSe]:ACQuisition:SAMPlEs, [:SENSe]:ACQuisition:SECOnds

**[[: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.

**Syntax** [[:SENSe]:ACQuisition:SAMPles <value>  
[[:SENSe]:ACQuisition:SAMPles?

**Arguments** <value>::=<NR1> specifies the acquisition samples.  
Range: 2 to 1 G samples.

**Measurement Views** All

**Examples** :SENSe:ACQuisition:SAMPles 1114  
sets the acquisition samples to 1114.

**Related Commands** [[:SENSe]:ACQuisition:MODE]

**[[:SENSe]:ACQuisition:SEConds(?)]**

Sets or queries the acquisition length (time over which the acquisition occurs) when [[:SENSe]:ACQuisition:MODE] is set to LENGth.

**Syntax** [[:SENSe]:ACQuisition:SEConds <value>  
[[:SENSe]:ACQuisition:SEConds?

**Arguments** <value>::=<NRf> specifies the acquisition length.

**Measurement Views** All

**Examples** :SENSe:ACQuisition:SAMPles 12.5ms  
sets the acquisition length to 12.5 ms.

**Related Commands** [[:SENSe]:ACQuisition:MODE]

## **[: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.

**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.

**Measurement Views**   All

**Examples**     :SENSe:ANALysis:ADVanced:DITHer ON  
enables dithering.

**[:SENSE]:ANALysis:ADVanced:DITHer:HWARe:STATus? (Query Only)**

Queries the dithering hardware status.

**Syntax**     [:SENSE]:ANALysis:ADVanced:DITHer:HWARe:STATus?

**Arguments**     None

**Returns**     One of the following status information.

**Table 2-41: Dithering status**

Status	Description
DUNaligned	Dithering is disabled and unaligned.
ON	Dithering is enabled
OFF	Dithering is disabled.

**Measurement Views**     All

**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.

**Syntax**     [:SENSe]:ANALysis:LENGth <value>  
              [:SENSe]:ANALysis:LENGth?

**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].

**Measurement Views**     All

**Examples**        :SENSe:ANALysis:LENGth 25.625us  
                  sets the analysis length to 25.625  $\mu$ s.

**Related Commands**     [:SENSe]:ANALysis:AUTO

**[[:SENSe]:ANALysis:LENGth:ACTual? (Query Only)**

Queries the actual analysis length.

**Syntax** [[:SENSe]:ANALysis:LENGth:ACTual?

**Arguments** None

**Returns** <NRf> Actual analysis length in seconds.

**Measurement Views** All

**Examples** :SENSe:ANALysis:LENGth:ACTual?  
might return 25.625E-6, indicating that the actual analysis length is 25.625  $\mu$ s.

**[[:SENSe]:ANALysis:LENGth:AUTO(?)]**

Determines whether to set the analysis length automatically or manually.

**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.

**Measurement Views** All

**Examples** :SENSe:ANALysis:LENGth:AUTO ON  
sets the analysis length automatically.

**Related Commands** [[:SENSe]:ANALysis:LENGth

**[:SENSe]:ANALysis:REFerence(?)**

Selects or queries the analysis time reference.

**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.

**Measurement Views**    All

**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.

**Syntax**        [:SENSe]:ANALysis:START <value>  
                   [:SENSe]:ANALysis:START?

**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].

**Measurement Views**    All

**Examples**        :SENSe:ANALysis:START 23.5us  
                   sets the analysis offset to 23.5  $\mu$ s.

**Related Commands**    [:SENSe]:ANALysis:LENGth, [:SENSe]:ANALysis:START:AUTO

**[[:SENSe]:ANALysis:START:AUTO(?)]**

Determines whether to set the analysis offset automatically or manually.

**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.

**Measurement Views** All

**Examples** :SENSe:ANALysis:START:AUTO ON  
sets the analysis offset automatically.

**Related Commands** [[:SENSe]:ANALysis:START

**[[:SENSe]:MEASurement:FREQuency(?)]**

Sets or queries the measurement frequency.

**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).

**Measurement Views** All

**Examples** :SENSe:MEASurement:FREQuency 7.5GHz  
sets the measurement frequency to 7.5 GHz.

## [:SENSe]:POWer:UNITs(?)

Selects or queries the fundamental unit of power.

**Syntax** [:SENSe]:POWer:UNITs { DBM | DBV | VOLTs | WATTs | DBUW | DBW  
| DBUV | DBMV | DBUA | DBUV\_M | DBUA\_M | AMPS }

[:SENSe]:POWer:UNITs?

**Arguments** Table 2-42 lists the arguments.

**Table 2-42: Power units**

Argument	Power unit
DBM	dBm
DBV	dBV
VOLTs	Volts
WATTs	Watts
DBUW	dB $\mu$ W
DBW	dBW
DBUV	dB $\mu$ V
DBMV	dBmV
DBUA	dB $\mu$ A
DBUV_M	dB $\mu$ V/m
DBUA_M	dB $\mu$ A/m
AMPS	Amps

**NOTE.** Select dB $\mu$ V/m or dB $\mu$ A/m unit when using an antenna table.

**Measurement Views** All

**Examples** :SENSe:POWer:UNITs DBM  
specifies the fundamental unit of power as dBm.

**[ :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.*

---

<b>Syntax</b>	[ :SENSe ]:REANalyze
<b>Arguments</b>	None
<b>Measurement Views</b>	All
<b>Examples</b>	:SENSe:REANalyze have all measurements reanalyze the current acquisition record.
<b>Related Commands</b>	*OPC, *WAI

## **[:SENSe]:ROSCillator:SOURce(?)**

Selects or queries the frequency reference oscillator source.

**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.

**Measurement Views**    All

**Examples**    :SENSe:ROSCillator:SOURce INTernal  
uses the 10 MHz internal oscillator for the frequency reference.

**[:SENSe]:SPEctrum:LENGth(?)**

Sets or queries the spectrum length when [:SENSe]:SPEctrum:TIME:MODE is INDEpendent. Programming a specified length sets [:SENSe]:SPEctrum:AUTO OFF.

**Syntax**     [:SENSe]:SPEctrum:LENGth <value>  
                 [:SENSe]:SPEctrum:LENGth?

**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.

**Measurement Views**     All

**Examples**         :SENSe:SPEctrum:LENGth 38.5us  
                 sets the SPEctrum length to 38.5  $\mu$ s.

**Related Commands**     [:SENSe]:SPEctrum:AUTO, [:SENSe]:SPEctrum:START,  
                 [:SENSe]:SPEctrum:TIME:MODE



**[:SENSe]:SPECTrum:LENGth:ACTual? (Query Only)**

Queries the actual spectrum length.

**Syntax**    [:SENSe]:SPECTrum:LENGth:ACTual?

**Arguments**    None

**Returns**    <NRf>    Actual spectrum length in seconds.

**Measurement Views**    All

**Examples**    :SENSe:SPECTrum:LENGth:ACTual?  
might return 6.337E-6, indicating that the actual spectrum length is 6.337  $\mu$ s.

**[:SENSe]:SPECTrum:LENGth:AUTO(?)**

Determines whether to set the spectrum length automatically or manually when [:SENSe]:SPECTrum:TIME:MODE is INdependent.

**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.

**Measurement Views**    All

**Examples**    :SENSe:SPECTrum:LENGth:AUTO ON  
sets the spectrum length automatically.

**Related Commands**    [:SENSe]:SPECTrum:LENGth, [:SENSe]:SPECTrum:TIME:MODE

**[[:SENSe]:SPECTrum:START(?)]**

Sets or queries the spectrum offset time when [[:SENSe]:SPECTrum:TIME:MODE] is INDEpendent.

**Syntax** [[:SENSe]:SPECTrum:START <value>

[[:SENSe]:SPECTrum:START?

**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.

**Measurement Views** All

**Examples** :SENSe:SPECTrum:START 23.5us  
sets the analysis offset to 23.5  $\mu$ s.

**Related Commands** [[:SENSe]:SPECTrum:LENGth, [[:SENSe]:SPECTrum:TIME:MODE]

**[[:SENSe]:SPECTrum:TIME:MODE(?)]**

Determines whether to set the spectrum time parameters automatically or manually.

**Syntax** [[:SENSe]:SPECTrum:TIME:MODE { INDEpendent | COMMon }

[[:SENSe]:SPECTrum:TIME:MODE?

**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.

**Measurement Views** All

**Examples** :SENSe:SPECTrum:TIME:MODE COMMon  
sets the spectrum time parameters automatically.

**Related Commands** [[:SENSe]:ANALysis commands  
[[:SENSe]:SPECTrum:STARt, [[:SENSe]:SPECTrum:LENGth

**[[: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.

**Syntax** [[:SENSe]:USETtings

**Arguments** None

**Measurement Views** All

**Examples** :SENSe:USETtings  
updates settings.

## [[:SENSe]:ACPower Subgroup

The [[:SENSe]:ACPower commands set up the conditions related to the Channel power and ACPR (Adjacent Channel Power Ratio) measurement.

Command Tree	Header	Parameter
	[SENSe]	
	:ACPower	
	:AVERage	OFF   TIME   FREQUENCY
	:COUNT	<numeric_value>
	:BANDwidth :BWIDTH	
	[:RESolution]	<numeric_value>
	:ACTual?	
	:AUTO	<boolean>
	:VIDeo	<numeric_value>
	:STATE	<boolean>
	:CHANnel	
	:BANDwidth	
	:BWIDTH	<numeric_value>
	:FILTer	RRCosine   NONE
	:PAIRs	<numeric_value>
	:SPACing	<numeric_value>
	:CHIPrate	<numeric_value>
	:CLEar	
	:RESuIts	
	:FREQUency	<numeric_value>
	:STEP	<numeric_value>
	:AUTO	<boolean>
	:NFLoor	
	:STATE	<boolean>
	:OPTimize	
	:SPAN	RTBandwidth   DRANge
	:RRCRolloff	<numeric_value>

**[:SENSe]:ACPower:AVERage(?)**

Selects or queries how to average waveform in the Channel power and ACPR measurement.

**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.

**Measurement Views** Channel power and ACPR

**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.

**Syntax** [:SENSe]:ACPower:AVERage:COUNT <number>

[:SENSe]:ACPower:AVERage:COUNT?

**Arguments** <number>::=<NR1> specifies the average count. Range: 2 to 10000.

**Measurement Views** Channel power and ACPR

**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.

**Syntax** [:SENSe]:ACPower:BANDwidth]:BWIDth[:RESolution] <value>

[:SENSe]:ACPower:BANDwidth]:BWIDth[:RESolution]?

**Arguments** <value>::=<NRf> specifies the RBW.  
Range: 100 Hz to 5 MHz.

**Measurement Views** Channel power and ACPR

**Examples** :SENSe:ACPower:BANDwidth:RESolution 200kHz  
sets the RBW to 200 kHz.

**Related Commands** [:SENSe]:ACPower:BANDwidth[:RESolution]:AUTO

**[[:SENSe]:ACPower:BANDwidth]:BWIDth[:RESolution]:ACTual? (Query Only)**

Queries the actual resolution bandwidth (RBW) in the Channel power and ACPR measurement.

**Syntax** [:SENSe]:ACPower:BANDwidth]:BWIDth[:RESolution]:ACTual?

**Arguments** None

**Returns** <NRf> The actual RBW in Hz.

**Measurement Views** Channel power and ACPR

**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.

**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.

**Measurement Views** Channel power and ACPR

**Examples** :SENSe:ACPower:BANDwidth:AUTO ON  
sets the RBW automatically.

**Related Commands** [[:SENSe]:ACPower:BANDwidth|:BWIDth[:RESolution]

**[[:SENSe]:ACPower:BANDwidth]:BWIDth:VIDeo(?)**

Sets or queries the video bandwidth (VBW). Programming a specified VBW sets [:SENSe]:ACPower:BANDwidth]:BWIDth:VIDeo:STATe OFF.

**Syntax** [:SENSe]:ACPower:BANDwidth]:BWIDth:VIDeo <value>  
[:SENSe]:ACPower:BANDwidth]:BWIDth:VIDeo?

**Arguments** <value>: :=<NRf> specifies the VBW.  
Range: Current RBW/10<sup>4</sup> (1 Hz minimum) to Current RBW.

**Measurement Views** Channel power and ACPR

**Examples** :SENSe:ACPower:BANDwidth:VIDeo 200kHz  
sets the VBW to 200 kHz.

**Related Commands** [:SENSe]:ACPower:BANDwidth]:BWIDth:VIDeo:STATe

**[[:SENSe]:ACPower:BANDwidth]:BWIDth:VIDeo:STATe(?)**

Determines whether to enable or disable the video bandwidth (VBW) in the Channel power and ACPR measurement.

**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.

**Measurement Views** Channel power and ACPR

**Examples** :SENSe:ACPower:BANDwidth:VIDeo:STATe ON  
enables the VBW.

**Related Commands** [:SENSe]:ACPower:BANDwidth]:BWIDth:VIDeo



**[: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.

**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.

**Measurement Views** Channel power and ACPR

**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.

**Syntax** [:SENSe]:ACPower:CHANnel:FILTer { RRCosine | NONE }

[:SENSe]:ACPower:CHANnel:FILTer?

**Arguments** RRCosine uses the Root-Raised Cosine filter.  
NONE uses no filter.

**Measurement Views** Channel power and ACPR

**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.

**Syntax** `[[:SENSe]:ACPower:CHANnel:PAIRs <number>`

`[[:SENSe]:ACPower:CHANnel:PAIRs?`

**Arguments** `<number>::=<NR1>` specifies the number of adjacent pairs.  
Range: 0 to 50.

**Measurement Views** Channel power and ACPR

**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.

**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.

**Measurement Views** Channel power and ACPR

**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).

**Syntax** [:SENSe]:ACPower:CHIPrate <value>

[:SENSe]:ACPower:CHIPrate?

**Arguments** <value>::=<NRf> specifies the chip rate.  
Range: 100 Hz to 105 MHz.

**Measurement Views** Channel power and ACPR

**Examples** :SENSe:ACPower:CHIPrate 5kHz  
sets the chip rate to 5 kHz.

**Related Commands** [:SENSe]:ACPower:CHANnel:FILTer

## **[:SENSe]:ACPower:CLEar:RESults (No Query Form)**

Restarts the average trace.

**Syntax** [:SENSe]:ACPower:CLEar:RESults

**Arguments** None

**Measurement Views** Channel power and ACPR

**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.

**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).

**Measurement Views**    Channel power and ACPR

**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.

**Syntax** [:SENSe]:ACPower:FREQUENCY:STEP <value>  
[:SENSe]:ACPower:FREQUENCY:STEP?

**Arguments** <value>::=<NRf> specifies the frequency step size.  
Range: 0 to 6.2 GHz (RSA6106A) / 14 GHz (RSA6114A).

**Measurement Views** Channel power and ACPR

**Examples** :SENSe:ACPower:FREQUENCY:STEP 1kHz  
sets the frequency step size to 1 kHz.

**Related Commands** [:SENSe]:ACPower:FREQUENCY:STEP:AUTO

**[:SENSe]:ACPower:FREQUENCY:STEP:AUTO(?)**

Determines whether to set the frequency step size automatically or manually in the Channel power and ACPR measurement.

**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.

**Measurement Views** Channel power and ACPR

**Examples** :SENSe:ACPower:FREQUENCY:STEP:AUTO ON  
specifies that the frequency step size is set automatically.

**Related Commands** [:SENSe]:ACPower:FREQUENCY:STEP

**[[:SENSE]:ACPower:NFLoor:STATe(?]**

Determines whether to enable or disable the correction for noise floor.

**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.

**Measurement Views**    Channel power and ACPR

**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.

**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.

**Measurement Views**    Channel power and ACPR

**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.

**Syntax** [[:SENSe]:ACPower:RRCRolloff <value>

[[:SENSe]:ACPower:RRCRolloff?

**Arguments** <value>: :=<NRf> specifies the filter parameter.  
Range: 0.0001 to 1 in 0.0001 steps.

**Measurement Views** Channel power and ACPR

**Examples** :SENSe:ACPower:RRCRolloff 0.3  
sets the filter parameter to 0.3.

**Related Commands** [[:SENSe]:ACPower:CHANnel:FILTer

## **[[:SENSE]:AVTime Subgroup**

The [[:SENSE]:AVTime commands set up the conditions related to the Amplitude versus Time measurement.

<b>Command Tree</b>	<b>Header</b>	<b>Parameter</b>
	[[:SENSE]	
	:AVTime	
	:Bandwith :BWIDth	<numeric_value>
	:ACTual?	
	:CLEar	
	:RESults	
	:MAXTracepoints	ONEK   TENK   HUNDredk   NEVerdecimate
	:METHod	SPAN   TDBW
	:SPAN	<numeric_value>



**[[:SENSE]:AVTime:BANDwith|: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.

**Syntax** [[:SENSE]:AVTime:BANDwith|:BWIDth <value>  
[[:SENSE]:AVTime:BANDwith|:BWIDth?

**Arguments** <value>::=<NRf> specifies the filter bandwidth.  
Range: 1 Hz to 20 MHz (Standard) / 60 MHz (Option 110).

**Measurement Views** Amplitude versus Time

**Examples** :SENSE:AVTime:BANDwith 10MHz  
sets the filter bandwidth to 10 MHz.

**Related Commands** [[:SENSE]:AVTime:SPAN]

**[[:SENSE]:AVTime:BANDwidth|:BWIDth:ACTual? (Query Only)]**

Queries the actual time-domain bandwidth in the Amplitude versus Time measurement.

**Syntax** [[:SENSE]:AVTime:BANDwidth|:BWIDth:ACTual?

**Arguments** None

**Returns** <NRf> The actual time-domain bandwidth in Hz.

**Measurement Views** Amplitude versus Time

**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).

**Syntax** [[:SENSe]:AVTime:CLEAr:RESuLts

**Arguments** None

**Measurement Views** Amplitude versus Time

**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.

**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.

**Measurement Views** Amplitude versus Time

**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.

**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.

**Measurement Views**    Amplitude versus Time

**Examples**    :SENSe:AVTime:METhod SPAN  
specifies that the measurement bandwidth is set by the frequency span.

**Related Commands**    [:SENSe]:AVTime:BANDwidth|:BWIDth, [:SENSe]:AVTime: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.

**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)

**Measurement Views**   Amplitude versus Time

**Examples**        :SENSe:AVTime:SPAN 5MHz  
                    sets the frequency span to 5 MHz.

**Related Commands**   [:SENSe]:AVTime:BANDwidth[:BWIDth]

## [:SENSE]:CCDF Subgroup

The [:SENSE]:CCDF commands set up the conditions related to the CCDF (Complementary Cumulative Distribution Function) measurement.

Command Tree	Header	Parameter
	[SENSE]	
	:CCDF	
	:BANDwidth :BWIDth	
	:CLEAr	
	:TIME	
	:TOTal	
	:LENGth	<numeric_value>
	:TYPE	SINGLE   TOTal   CONTinuous

## [:SENSE]:CCDF:BANDwidth|:BWIDth(?)

Sets or queries the CCDF measurement bandwidth (frequency span).

**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).

**Measurement Views** CCDF

**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.

**Syntax** [[:SENSe]:CCDF:CLEAr

**Arguments** None

**Measurement Views** CCDF

**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.

**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.

**Measurement Views** CCDF

**Examples** :SENSe:CCDF:TIME:TOTAl:LENGth 10  
sets the CCDF measurement time to 10 s.

**Related Commands** [[:SENSe]:CCDF:TIME:TYPE

## **[[:SENSe]:CCDF:TIME:TYPE(?)]**

Determines how to repeat the CCDF measurement.

**Syntax**    [[:SENSe]:CCDF:TIME:TYPE { SINGLe | TOTa1 | CONTInuous }  
[[:SENSe]:CCDF:TIME:TYPE?

**Arguments**    SINGLe specifies that the analyzer sets the analysis length to 1 ms and then acquire data once to calculate CCDF.

TOTa1 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: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.

**Measurement Views**    CCDF

**Examples**    :SENSe:CCDF:TIME:TYPE ON  
sets the CCDF measurement time automatically.

**Related Commands**    INITiate commands, [[:SENSe]:CCDF:CLEar], [[:SENSe]:CCDF:TIME:LENGth]

## [[:SENSE]:DDEMod Subgroup (Option 21 Only)

The [[:SENSE]:DDEMod commands set up conditions related to the General Purpose Digital Modulation measurements.

Command Tree	Header	Parameter
	[[:SENSE]	
	:DDEMod	
	:ANALysis	
	:LENGth	<numeric_value>
	:ACTual?	
	:AUTO	<boolean>
	:BURSt	
	:DETEct	ON   OFF
	:THREshold	<numeric_value>
	:CARRier	
	:OFFSet	<numeric_value>
	:FILTer	
	:ALPHa	<numeric_value>
	:MEASurement	OFF   RRCosine   RCOSine   GAUSSian   RECTangular   IS95TXEQ_MEA   IS95TX_MEA   IS95REF
	:REFerence	OFF   RRCosine   RCOSine   GAUSSian   RECTangular   IS95TXEQ_MEA   IS95TX_MEA   IS95REF
	:MAGNitude	
	:NORMalize	RSYMBOL   MSYMBOL
	:MODulation	
	:TYPE	QPSK   PSK8   D8PSK   DQPSK   PIOVER4DQPSK   BPSK   QAM16   QAM32   QAM64   QAM128   QAM256   MSK
	:SRATE	<numeric_value>
	:SWAP	
	:IQ	<boolean>
	:SYMBOL	
	:POINTs	ONE   TWO   FOUR   EIGHT
	:SYNCh	
	:WORD	<boolean>
	:SYMBOL	<binary_block>
	:TIME	
	:UNITs	SECONDS   SYMBOLS



## **[[:SENSe]:DDEMod:ANALysis:LENGth(?)**

Sets or queries the analysis length. Programming a specified length sets [[:SENSe]:DDEMod:ANALysis:AUTO OFF.

**Syntax** [[:SENSe]:DDEMod:ANALysis:LENGth <value>

[[:SENSe]:DDEMod:ANALysis:LENGth?

**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].

**Measurement Views** General purpose digital modulation

**Examples** :SENSe:DDEMod:ANALysis:LENGth 25.625us  
sets the analysis length to 25.625  $\mu$ s.

**Related Commands** [[:SENSe]:DDEMod:ANALysis:AUTO

**[[:SENSe]:DDEMod:ANALysis:LENGth:ACTual? (Query Only)**

Queries the actual analysis length.

**Syntax** [[:SENSe]:DDEMod:ANALysis:LENGth:ACTual?

**Arguments** None

**Returns** <NRf> Actual analysis length in seconds.

**Measurement Views** General purpose digital modulation

**Examples** :SENSe:DDEMod:ANALysis:LENGth:ACTual?  
might return 25.625E-6, indicating that the actual analysis length is 25.625  $\mu$ s.

**[[:SENSe]:DDEMod:ANALysis:LENGth:AUTO(?]**

Determines whether to set the analysis length automatically or manually.

**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.

**Measurement Views** General purpose digital modulation

**Examples** :SENSe:DDEMod:ANALysis:LENGth:AUTO ON  
sets the analysis length automatically.

**Related Commands** [[:SENSe]:DDEMod:ANALysis:LENGth

## [:SENSe]:DDEMod:BURSt:DETECT(?)

Determines how to detect bursts.

**Syntax**    [:SENSe]:DDEMod:BURSt:DETECT { ON | OFF }  
[:SENSe]:DDEMod:BURSt:DETECT?

**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.

---

**Measurement Views**    General purpose digital modulation

**Examples**    :SENSe:DDEMod:BURSt:DETECT OFF  
analyzes the whole analysis length.

**Related Commands**    [:SENSe]:DDEMod:BURSt:THRESHOLD

**[[:SENSe]:DDEMod:BURSt:THReshold(?)]**

Sets or queries the threshold level above which the input signal is determined to be a burst.

**Syntax** [[:SENSe]:DDEMod:BURSt:THReshold <value>

[[:SENSe]:DDEMod:BURSt:THReshold?

**Arguments** <value>::=<NRf> specifies the threshold level for detecting bursts.  
Range: -100 to -10 dBc.

**Measurement Views** General purpose digital modulation

**Examples** :SENSe:DDEMod:BURSt:THReshold -25  
sets the threshold level to -25 dBc.

**Related Commands** [[:SENSe]:DDEMod:BURSt:DETECT

**[[:SENSe]:DDEMod:CARRier:OFFSet(?)]**

Sets or queries the carrier frequency offset in the digital modulation analysis.

**Syntax** [[:SENSe]:DDEMod:CARRier:OFFSet <value>

[[:SENSe]:DDEMod:CARRier:OFFSet?

**Arguments** <value>::=<NRf> specifies the carrier frequency offset.  
Range: -50 kHz to +50 kHz.

**Measurement Views** General purpose digital modulation

**Examples** :SENSe:DDEMod:CARRier:OFFSet 2kHz  
sets the carrier frequency offset to 2 kHz.

**[:SENSe]:DDEMod:FILTer:ALPHa(?)**

Sets or queries the filter factor ( $\alpha/BT$ ) in the digital modulation analysis.

**Syntax**     [:SENSe]:DDEMod:FILTer:ALPHa <value>  
              [:SENSe]:DDEMod:FILTer:ALPHa?

**Arguments**   <value>: :=<NRf> specifies the filter factor.  
                  Range: 0.001 to 1.

**Measurement Views**   General purpose digital modulation

**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.

**Syntax** `[[:SENSe]:DDEMod:FILTer:MEASurement`  
`{ OFF | RRCosine | RCOSine | GAUSSian | RECTangular`  
`| IS95TXEQ_MEA | IS95TX_MEA | IS95REF }`  
`[[:SENSe]:DDEMod:FILTer:MEASurement?`

**Arguments** Table 2-43 lists the arguments.

**Table 2-43: Digital modulation measurement filter**

Argument	Measurement filter
OFF	No filter
RRCosine	Root Raised Cosine
RCOSine	Raised Cosine
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.

**Measurement Views** General purpose digital modulation

**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.

**Syntax**    [:SENSe]:DDEMod:FILTer:REFeRence  
                   { OFF | RRCosine | RCOSine | GAUSSian | RECTangular  
                   | IS95TXEQ\_MEA | IS95TX\_MEA | IS95REF }  
 [:SENSe]:DDEMod:FILTer:REFeRence?

**Arguments**    Refer to Table 2-30 on page 2-87.

**Measurement Views**    General purpose digital modulation

**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.

**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.

**Measurement Views**    General purpose digital modulation

**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.

**Syntax** `[[:SENSE]:DDEMod:MODulation:TYPE { QPSK | PSK8 | D8PSK | DQPSK  
| PIOVER4DQPSK | BPSK | QAM16 | QAM32 | QAM64 | QAM128 | QAM256  
| MSK }  
[:SENSE]:DDEMod:MODulation:TYPE?`

**Arguments** Table 2-44 lists the arguments and corresponding modulation type.

**Table 2-44: Modulation type**

Argument	Modulation type
QPSK	QPSK
PSK8	8PSK
D8PSK	D8PSK
DQPSK	DQPSK
PIOVER4DQPSK	$\pi/4$ QPSK
BPSK	BPSK
QAM16	16QAM
QAM32	32QAM
QAM64	64QAM
QAM128	128QAM
QAM256	256QAM
MSK	MSK

**Measurement Views** General purpose digital modulation

**Examples** `:SENSE:DDEMod:MODulation:TYPE QPSK`  
selects QPSK modulation system.



## **[[:SENSe]:DDEMod:SRATe(?)]**

Sets or queries the symbol rate in the digital modulation analysis.

**Syntax**    [:SENSe]:DDEMod:SRATe <value>  
              [:SENSe]:DDEMod:SRATe?

**Arguments**    <value>::=<NRf> specifies the symbol rate.  
                  Range: 100 Hz to 122.9 MHz.

**Measurement Views**    General purpose digital modulation

**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.

**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.

**Measurement Views**    General purpose digital modulation

**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<x>:CONStE:MODE command first to change Points/Symbol.*

---

**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 .

**Measurement Views**   Constellation, EVM versus Time, Magnitude error versus Time, Phase error versus Time

**Examples**       :SENSE:DDEMod:SYMBOL:POINTS FOUR  
sets the number of points per symbol to four.

**Related Commands**   :TRACe<x>:CONStE:MODE

**[:SENSe]:DDEMod:SYNCh:WORD(?)**

Determines whether to enable or disable the synchronization word.

**Syntax**    [:SENSe]:DDEMod:SYNCh:WORD { OFF | ON | 0 | 1 }  
              [:SENSe]:DDEMod:SYNCh:WORD?

**Arguments**    OFF or 0 disables the synchronization word.  
                  ON or 1 enables the synchronization word.

**Measurement Views**    General purpose digital modulation

**Examples**        :SENSe:DDEMod:SYNCh:WORD ON  
                  enables the synchronization word.

**Related Commands**    [:SENSe]:DDEMod:SYNCh:WORD:SYMBOL

**[[: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.

**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<sup>th</sup> symbol value of the sync word. 32-bit integer. n: Max 256.

**Measurement Views** General purpose digital modulation

**Examples** :SENSe:DDEMod:SYNCh:WORD:SYMBol #216xxxx (4 symbols)  
sets a sync word composed of four symbols.

**Related Commands** [:SENSe]:DDEMod:MODulation:TYPE, [:SENSe]:DDEMod:SYNCh:WORD

**[[:SENSe]:DDEMod:TIME:UNITs(?)]**

Selects or queries the fundamental unit of time.

**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.

**Measurement Views** General purpose digital modulation

**Examples** :SENSe:DDEMod:TIME:UNITs SEConds  
specifies the fundamental unit of time as seconds.

## [:SENSe]:DPSA Subgroup

The [:SENSe]:DPSA commands set up the conditions related to the DPX (Digital Phosphor) Spectrum measurement.

Command Tree	Header	Parameter
	[:SENSe]	
	:DPSA	
	:AUDio	
	:DEMod	
	:GAIN	<numeric_value>
	:RXBWidth	<numeric_value>
	:RXFRrequency?	
	:STATe	<boolean>
	:TUNE	MR   MARK1   MARK2   MARK3   MARK4   SMARker   FREQcontrol
	:TYPE	AM   FM
	:BANDwidth :BWIDth	
	[:RESolution]	<numeric_value>
	:AUTO	<boolean>
	:CLEar	
	:RESults	
	:COLor	RED   GREen   BLUe   CYAN   BCYan   YELLow   MAGenta   GRAY   TEMPerature   SPECTral
	:MAXimum	<numeric_value>
	:MINimum	<numeric_value>
	:FREQuency	
	:CENTer	<numeric_value>
	:SPAN	<numeric_value>
	:START	<numeric_value>
	:STEP	<numeric_value>
	:AUTO	<boolean>
	:STOP	<numeric_value>

**[[:SENSe]:DPSA:AUDio:DEMod:GAIN(?)]**

Sets or queries the audio gain.

---

**NOTE.** *The sound level is also affected by the Windows volume control.*

---

**Syntax** [[:SENSe]:DPSA:AUDio:DEMod:GAIN <value>

[[:SENSe]:DPSA:AUDio:DEMod:GAIN?

**Arguments** <value>::=<NR1> specifies the audio gain.  
Range: 0 to 15 (integer).

**Measurement Views** DPX spectrum

**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.

**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.

**Measurement Views** DPX spectrum

**Examples** :SENSe:DPSA:AUDio:DEMod:RXBWidth 30kHz  
sets the receiver bandwidth to 30 kHz.

**[:SENSe]:DPSA:AUDio:DEMod:RXFRrequency? (Query Only)**

Returns the receiver frequency in the audio demodulation.

The frequency depends on the setting of the [:SENSe]:DPSA:AUDio:DEMod:TUNE command.

**Syntax** [:SENSe]:DPSA:AUDio:DEMod:RXFRrequency?

**Returns** <frequency>::=<NRf> is the receiver frequency in the audio demodulation.

**Measurement Views** DPX spectrum

**Examples** :SENSe:DPSA:AUDio:DEMod:RXFRrequency?  
might return 80.3E+6, indicating that the receiver frequency is 80.3 MHz.

**Related Commands** [:SENSe]:DPSA:AUDio:DEMod:TUNE

**[:SENSe]:DPSA:AUDio:DEMod:STATe(?)**

Determines whether to enable or disable the audio demodulation.

**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.

**Measurement Views** DPX spectrum

**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.

**Syntax** `[[:SENSe]:DPSA:AUDio:DEMod:TUNE { MR | MARK1 | MARK2 | MARK3 | MARK4 | SMARker | FREQcontrol }]`  
`[[:SENSe]:DPSA:AUDio:DEMod:TUNE?`

**Arguments** Table 2-45 lists the arguments.

**Table 2-45: 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

**Measurement Views** DPX spectrum

**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.

**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).

**Measurement Views**    DPX spectrum

**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.

**Syntax**        [[:SENSe]:DPSA:BANDwidth[:BWIDth[:RESolution] <value>  
                   [[:SENSe]:DPSA:BANDwidth[:BWIDth[:RESolution]?

**Arguments**    <value>::=<NRf> is the resolution bandwidth.  
                   Range: 1 Hz to 5 MHz.

**Measurement Views**    DPX spectrum

**Examples**        :SENSe:DPSA:BANDwidth:RESolution 200kHz  
                   sets the resolution bandwidth to 200 kHz.

**Related Commands**    [[:SENSe]:DPSA:BANDwidth[:BWIDth[:RESolution]:AUTO

**[[:SENSE]:DPSA:BANDwidth]:BWIDth[:RESolution]:AUTO(?)**

Determines whether to set the resolution bandwidth (RBW) automatically or manually in the DPX spectrum measurement.

**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.

**Measurement Views** DPX spectrum

**Examples** `:SENSE:DPSA:BANDwidth:AUTO ON`  
sets the resolution bandwidth automatically.

**Related Commands** `[[:SENSE]:DPSA:BANDwidth]:BWIDth[:RESolution]`

**[[:SENSE]:DPSA:CLEar:RESults (No Query Form)**

Restarts multi-trace functions (Average and Max/Min Hold).

**Syntax** `[[:SENSE]:DPSA:CLEar:RESults`

**Arguments** None

**Measurement Views** DPX spectrum

**Examples** `:SENSE:DPSA:CLEar:RESults`  
restarts multi-trace functions.

**[:SENSe]:DPSA:COLor(?)**

Selects or queries the color palette of three-dimensional graphs.

**Syntax**    [:SENSe]:DPSA:COLor { RED | GREen | BLUe | CYAN | BCYan  
| YELLow | MAGenta | GRAY | TEMPerature | SPECtral }

[:SENSe]:DPSA:COLor?

**Arguments**    Table 2-46 lists the arguments.

**Table 2-46: 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

**Measurement Views**    DPX spectrum

**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.

**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:MIN command.

**Measurement Views**   DPX spectrum

**Examples**       :SENSe:DPSA:COLor:MAXimum 90  
                  sets the maximum value of the color axis to 90%.

**Related Commands**   [:SENSe]:DPSA:COLor:MINimum

**[[:SENSe]:DPSA:COLor:MINimum(?)]**

Sets or queries the minimum value of the color axis in the DPX spectrum measurement.

**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:MAX command.

**Measurement Views**   DPX spectrum

**Examples**       :SENSe:DPSA:COLor:MINimum 10  
                  sets the minimum value of the color axis to 10%.

**Related Commands**   [:SENSe]:DPSA:COLor:MAXimum

**[[: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:

$$(\text{start frequency}) = (\text{center frequency}) - (\text{span})/2$$

$$(\text{stop frequency}) = (\text{center frequency}) + (\text{span})/2$$


---

**Syntax** [[:SENSE]:DPSA:FREQUENCY:CENTER <value>

[[:SENSE]:DPSA:FREQUENCY:CENTER?

**Arguments** <value>::=<Nrf> specifies the center frequency.  
Range: 0 Hz to 6.2 GHz (RSA6106A) / 14 GHz (RSA6114A).

**Measurement Views** DPX spectrum

**Examples** :SENSE:DPSA:FREQUENCY:CENTER 7.5GHz  
sets the center frequency to 7.5 GHz.

**Related Commands** [[:SENSE]:DPSA:FREQUENCY:START, [[:SENSE]:DPSA:FREQUENCY:STOP

**[[:SENSE]:DPSA:FREQUENCY:SPAN(?)]**

Sets or queries the frequency span in the DPX spectrum measurement.

**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)

**Measurement Views** DPX spectrum

**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.

**Syntax** [[:SENSe]:DPSA:FREQUENCY:START <value>

[[:SENSe]:DPSA:FREQUENCY:START?

**Arguments** <value>::=<NRf> is the measurement start frequency.  
Range: (center frequency)  $\pm$  (span)/2.

**Measurement Views** DPX spectrum

**Examples** :SENSe:DPSA:FREQUENCY:START 6.95GHz  
sets the start frequency to 6.95 GHz.

**Related Commands** [[:SENSe]:DPSA:FREQUENCY:CENTer], [[:SENSe]:DPSA:FREQUENCY:STOP]

**[: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.

**Syntax**     [:SENSe]:DPSA:FREQuency:STEP <value>  
                  [:SENSe]:DPSA:FREQuency:STEP?

**Arguments**   <value>::=<NRf> specifies the frequency step size.  
 Range: 0 Hz to 6.2 GHz (RSA6106A) / 14 GHz (RSA6114A).

**Measurement Views**   DPX spectrum

**Examples**         :SENSe:DPSA:FREQuency:STEP 1.5kHz  
 sets the step size to 1.5 kHz.

**Related Commands**   [:SENSe]:DPSA:FREQuency:STEP:AUTO

**[:SENSe]:DPSA:FREQuency:STEP:AUTO(?)**

Determines whether to set the frequency step size automatically or manually.

**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.

**Measurement Views**   DPX spectrum

**Examples**         :SENSe:DPSA:BANDwidth:AUTO ON  
 sets the frequency step size automatically.

**Related Commands**   [:SENSe]:DPSA:FREQuency:STEP

**[[: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.

**Syntax** [[:SENSe]:DPSA:FREQUENCY:STOP <value>  
[[:SENSe]:DPSA:FREQUENCY:STOP?

**Arguments** <value>::=<NRf> is the measurement stop frequency.  
Range: (center frequency)  $\pm$  (span)/2.

**Measurement Views** DPX spectrum

**Examples** :SENSe:DPSA:FREQUENCY:STOP 7.05GHz  
sets the stop frequency to 7.05 GHz.

**Related Commands** [[:SENSe]:DPSA:FREQUENCY:CENTer], [[:SENSe]:DPSA:FREQUENCY:START



## [:SENSE]:FVTime Subgroup

The [:SENSE]:FVTime commands set up the conditions related to the Frequency versus Time measurement.

Command Tree	Header	Parameter
	[:SENSE]	
	:FVTime	
	:CLEar	
	:RESults	
	:FREQuency	
	:CENTer	<numeric_value>
	:SPAN	<numeric_value>
	:START	<numeric_value>
	:STEP	<numeric_value>
	:AUTO	<boolean>
	:STOP	<numeric_value>
	:MAXTracepoints	ONEK   TENK   HUNDredk   NEVerdecimate

## [:SENSE]:FVTime:CLEar:RESults (No Query Form)

Restarts multi-trace functions (Average and Max/Min Hold).

**Syntax** [:SENSE]:FVTime:CLEar:RESults

**Arguments** None

**Measurement Views** Frequency versus Time

**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:

$$(start\ frequency) = (center\ frequency) - (span)/2$$

$$(stop\ frequency) = (center\ frequency) + (span)/2$$

---

**Syntax** [[:SENSe]:FVTime:FREQuency:CENTer <value>

[[:SENSe]:FVTime:FREQuency:CENTer?

**Arguments** <value>::=<NRf> specifies the center frequency.  
Range: 0 Hz to 6.2 GHz (RSA6106A) / 14 GHz (RSA6114A).

**Measurement Views** Frequency versus Time

**Examples** :SENSe:FVTime:FREQuency:CENTer 7.5GHz  
sets the center frequency to 7.5 GHz.

**Related Commands** [[:SENSe]:FVTime:FREQuency:START, [[:SENSe]:FVTime:FREQuency:STOP

**[::SENSe]:FVTime:FREQuency:SPAN(?)**

Sets or queries the frequency span in the Frequency versus Time measurement.

**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)

**Measurement Views** Frequency versus Time

**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.

**Syntax** [::SENSe]:FVTime:FREQuency:START <value>  
[::SENSe]:FVTime:FREQuency:START?

**Arguments** <value>::=<NRf> is the measurement start frequency.  
Range: (center frequency) ± (span)/2.

**Measurement Views** Frequency versus Time

**Examples** :SENSe:FVTime:FREQuency:START 6.95GHz  
sets the start frequency to 6.95 GHz.

**Related Commands** [::SENSe]:FVTime:FREQuency:CENTer, [::SENSe]:FVTime:FREQuency:STOP

**[[: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.

**Syntax**     [:SENSe]:FVTime:FREQuency:STEP <value>  
              [:SENSe]:FVTime:FREQuency:STEP?

**Arguments**   <value>::=<NRf> specifies the frequency step size.  
                  Range: 0 Hz to 6.2 GHz (RSA6106A) / 14 GHz (RSA6114A).

**Measurement Views**   Frequency versus Time

**Examples**       :SENSe:FVTime:FREQuency:STEP 1.5kHz  
                  sets the step size to 1.5 kHz.

**Related Commands**   [:SENSe]:FVTime:FREQuency:STEP:AUTO

**[[:SENSe]:FVTime:FREQuency:STEP:AUTO(?)]**

Determines whether to set the frequency step size automatically or manually.

**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.

**Measurement Views**   Frequency versus Time

**Examples**       :SENSe:FVTime:BANDwidth:AUTO ON  
                  sets the frequency step size automatically.

**Related Commands**   [:SENSe]:FVTime:FREQuency:STEP

## **[: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.

**Syntax**     [:SENSe]:FVTime:FREQuency:STOP <value>

[:SENSe]:FVTime:FREQuency:STOP?

**Arguments**     <value>::=<Nrf> is the measurement stop frequency.  
Range: (center frequency) ± (span)/2.

**Measurement Views**     Frequency versus Time

**Examples**     :SENSe:FVTime:FREQuency:STOP 7.05GHz  
sets the stop frequency to 7.05 GHz.

**Related Commands**     [:SENSe]:FVTime:FREQuency:CENTer, [:SENSe]:FVTime:FREQuency:START

**[:SENSe]:FVTime:MAXTracepoints(?)**

Selects or queries the maximum trace points in the Frequency versus Time measurement.

**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.

**Measurement Views**    Frequency versus Time

**Examples**        :SENSe:FVTime:MAXTracepoints TENK  
                  sets the maximum trace points to 10 k.

## [:SENSE]:IQVTime Subgroup

The [:SENSE]:IQVTime commands set up the conditions related to the RF I&Q versus Time measurement.

Command Tree	Header	Parameter
	[:SENSE]	
	:IQVTime	
	:CLEar	
	:RESults	
	:FREQuency	
	:CENTer	<numeric_value>
	:SPAN	<numeric_value>
	:START	<numeric_value>
	:STEP	<numeric_value>
	:AUTO	<boolean>
	:STOP	<numeric_value>
	MAXTracepoints	ONEK   TENK   HUNDredk   NEVerdecimate

## [:SENSE]:IQVTime:CLEar:RESults (No Query Form)

Restarts multi-trace functions (Average and Max/Min Hold).

**Syntax** [:SENSE]:IQVTime:CLEar:RESults

**Arguments** None

**Measurement Views** RF I&Q versus Time

**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:

$$(start\ frequency) = (center\ frequency) - (span)/2$$

$$(stop\ frequency) = (center\ frequency) + (span)/2$$

---

**Syntax** [[:SENSe]:IQVTime:FREQuency:CENTer <value>

[[:SENSe]:IQVTime:FREQuency:CENTer?

**Arguments** <value>::=<NRf> specifies the center frequency.  
Range: 0 Hz to 6.2 GHz (RSA6106A) / 14 GHz (RSA6114A).

**Measurement Views** RF I&Q versus Time

**Examples** :SENSe:IQVTime:FREQuency:CENTer 7.5GHz  
sets the center frequency to 7.5 GHz.

**Related Commands** [[:SENSe]:IQVTime:FREQuency:START, [[:SENSe]:IQVTime:FREQuency:STOP



**[:SENSe]:IQVTime:FREQUENCY:SPAN(?)**

Sets or queries the frequency span in the RF I&Q versus Time measurement.

**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)

**Measurement Views**   RF I&Q versus Time

**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.

**Syntax**     [:SENSe]:IQVTime:FREQUENCY:START <value>  
                  [:SENSe]:IQVTime:FREQUENCY:START?

**Arguments**   <value>::=<Nrf> is the measurement start frequency.  
 Range: (center frequency) ± (span)/2.

**Measurement Views**   RF I&Q versus Time

**Examples**       :SENSe:IQVTime:FREQUENCY:START 6.95GHz  
 sets the start frequency to 6.95 GHz.

**Related Commands**   [:SENSe]:IQVTime:FREQUENCY:CENTer,  
 [:SENSe]:IQVTime:FREQUENCY:STOP

**[: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.

**Syntax**     [:SENSe]:IQVTime:FREQuency:STEP <value>  
              [:SENSe]:IQVTime:FREQuency:STEP?

**Arguments**   <value>::=<NRf> specifies the frequency step size.  
                  Range: 0 Hz to 6.2 GHz (RSA6106A) / 14 GHz (RSA6114A).

**Measurement Views**   RF I&Q versus Time

**Examples**         :SENSe:IQVTime:FREQuency:STEP 1.5kHz  
                      sets the step size to 1.5 kHz.

**Related Commands**   [:SENSe]:IQVTime:FREQuency:STEP:AUTO

**[:SENSe]:IQVTime:FREQuency:STEP:AUTO(?)**

Determines whether to set the frequency step size automatically or manually.

**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.

**Measurement Views**   RF I&Q versus Time

**Examples**         :SENSe:IQVTime:BANDwidth:AUTO ON  
                      sets the frequency step size automatically.

**Related Commands**   [:SENSe]:IQVTime:FREQuency:STEP

## **[[: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.

**Syntax** [[:SENSe]:IQVTime:FREQUENCY:STOP <value>

[[:SENSe]:IQVTime:FREQUENCY:STOP?

**Arguments** <value>::=<NRf> is the measurement stop frequency.  
Range: (center frequency)  $\pm$  (span)/2.

**Measurement Views** RF I&Q versus Time

**Examples** :SENSe:IQVTime:FREQUENCY:STOP 7.05GHz  
sets the stop frequency to 7.05 GHz.

**Related Commands** [[:SENSe]:IQVTime:FREQUENCY:CENTer,  
[[:SENSe]:IQVTime:FREQUENCY:START

**[:SENSe]:IQVTime:MAXTracepoints(?)**

Selects or queries the maximum trace points in the RF I&Q versus Time measurement.

**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.

**Measurement Views**    RF I&Q versus Time

**Examples**        :SENSe:IQVTime:MAXTracepoints TENK  
                  sets the maximum trace points to 10 k.

## [:SENSE]:MCPower Subgroup

The [:SENSE]:MCPower commands set up the conditions related to the MCPR (Multiple Carrier Power Ratio) measurement.

Command Tree	Header	Parameter
	[:SENSE]	
	:MCPower	
	:AVERage	OFF   TIME   FREQUENCY
	:COUNT	<numeric_value>
	:BANDwidth :BWIDth	
	[:RESolution]	<numeric_value>
	:ACTual?	
	:AUTO	<boolean>
	:VIDeo	<numeric_value>
	:STATe	<boolean>
	:CHANnel	
	:ADJacent	
	:ADD	<numeric_value>,<numeric_value>
	:DELete	<string>
	:FILTer	RRCosine   NONE
	:MAIN	
	:BANDwidth	
	:BWIDth	<numeric_value>
	:COUNT	<numeric_value>
	:INACTive	<string>,<boolean>
	:SPACing	<numeric_value>
	:CHIPrate	<numeric_value>
	:CLEar	
	:RESulTs	
	:FREQuency	<numeric_value>
	:STEP	<numeric_value>
	:AUTO	<boolean>
	:NFLoor	
	:STATe	<boolean>
	:OPTimize	
	:SPAN	RTBandwidth   DRANge
	:RCHannels?	
	:MAIN<x>	
	:TOTal	
	:RRCRolloff	<numeric_value>

**[[:SENSE]:MCPower:AVERage(?)]**

Selects or queries the average method in the MCPR measurement.

**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.

**Measurement Views**    MCPR

**Examples**        :SENSE:MCPower:AVERage TIME  
                      performs averaging for time samples.

**[[:SENSE]:MCPower:AVERage:COUNT(?)]**

Sets or queries the average count in the MCPR measurement.

**Syntax**        [[:SENSE]:MCPower:AVERage:COUNT <value>  
                  [[:SENSE]:MCPower:AVERage:COUNT?

**Arguments**    <value>::=<NR1> specifies the average count.  
                  Range: 2 to 10000.

**Measurement Views**    MCPR

**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.

**Syntax** [:SENSe]:MCPower:BANDwidth|:BWIDth[:RESolution] <value>

[:SENSe]:MCPower:BANDwidth|:BWIDth[:RESolution]?

**Arguments** <value>::=<NRf> specifies the RBW.  
Range: 100 Hz to 5 MHz.

**Measurement Views** MCPR

**Examples** :SENSe:MCPower:BANDwidth:RESolution 200kHz  
sets the RBW to 200 kHz.

**Related Commands** [:SENSe]:MCPower:BANDwidth|:BWIDth[:RESolution]:AUTO

**[:SENSe]:MCPower:BANDwidth|:BWIDth[:RESolution]:ACTual? (Query Only)**

Queries the actual resolution bandwidth (RBW) in the MCPR measurement.

**Syntax** [:SENSe]:MCPower:BANDwidth|:BWIDth[:RESolution]:ACTual?

**Arguments** None

**Returns** <NRf> The actual RBW in Hz.

**Measurement Views** MCPR

**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.

**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.

**Measurement Views** MCPR

**Examples** `:SENSe:MCPower:BANDwidth:AUTO ON`  
sets the resolution bandwidth automatically.

**Related Commands** `[[:SENSe]:MCPower:BANDwidth]:BWIDth[:RESolution]`



**[[:SENSe]:MCPower:BANDwidth]:BWIDth:VIDeo(?)**

Sets or queries the video bandwidth (VBW). Programming a specified VBW sets [[:SENSe]:MCPower:BANDwidth]:BWIDth:VIDeo:STATe OFF.

**Syntax** [[:SENSe]:MCPower:BANDwidth]:BWIDth:VIDeo <value>

[[:SENSe]:MCPower:BANDwidth]:BWIDth:VIDeo?

**Arguments** <value>: :=<NRf> specifies the VBW.  
Range: Current RBW/10<sup>4</sup> (1 Hz minimum) to Current RBW.

**Measurement Views** MCPR

**Examples** :SENSe:MCPower:BANDwidth:VIDeo 200kHz  
sets the VBW to 200 kHz.

**Related Commands** [[:SENSe]:MCPower:BANDwidth]:BWIDth:VIDeo:STATe

**[[:SENSe]:MCPower:BANDwidth]:BWIDth:VIDeo:STATe(?)**

Determines whether to enable or disable the video bandwidth (VBW) in the MCPR measurement.

**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.

**Measurement Views** MCPR

**Examples** :SENSe:MCPower:BANDwidth:VIDeo:STATe ON  
enables the VBW.

**Related Commands** [[:SENSe]:MCPower:BANDwidth]:BWIDth:VIDeo

**[:SENSe]:MCPower:CHANnel:ADJacent:ADD (No Query Form)**

Adds a pair of upper and lower adjacent channels in the MCPR measurement.

**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).

**Measurement Views** MCPR

**Examples** :SENSe:MCPower:CHANnel:ADJacent:ADD 200kHz,80kHz  
adds a pair of upper and lower adjacent channels with the offset of  $\pm 200$  kHz and the bandwidth of 80 kHz.

**[:SENSe]:MCPower:CHANnel:ADJacent:DELeTe (No Query Form)**

Deletes a selected adjacent channel in the MCPR measurement.

**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.

**Measurement Views** MCPR

**Examples** :SENSe:MCPower:CHANnel:DELeTe "A2"  
deletes A2 (the adjacent channel 2).

**[:SENSe]:MCPower:CHANnel:FILTer(?)**

Selects or queries the adjacent channel filter in the MCPR measurement.

**Syntax**    [:SENSe]:MCPower:CHANnel:FILTer { RRCosine | NONE }  
 [:SENSe]:MCPower:CHANnel:FILTer?

**Arguments**    RRCosine selects the Root-Raised-Cosine filter.  
 NONE uses no filter.

**Measurement Views**    MCPR

**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.

**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.

**Measurement Views**    MCPR

**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.

**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.

**Measurement Views** MCPR

**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.

**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).

**Measurement Views** MCPR

**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.

**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.

**Measurement Views** MCPR

**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).

**Syntax** [[:SENSE]:MCPower:CHIPrate <value>

[[:SENSE]:MCPower:CHIPrate?

**Arguments** <value>::=<NRf> specifies the chip rate.  
Range: 100 Hz to 105 MHz.

**Measurement Views** MCPR

**Examples** :SENSE:MCPower:CHIPrate 1kHz  
sets the chip rate to 1 kHz.

**Related Commands** [[:SENSE]:MCPower:CHANnel:FILTer]

**[:SENSe]:MCPower:CLEar:RESuIts (No Query Form)**

Restarts the average trace.

**Syntax** [:SENSe]:MCPower:CLEar:RESuIts

**Arguments** None

**Measurement Views** MCPR

**Examples** :SENSe:MCPower:CLEar:RESuIts  
restarts the average trace.

**[:SENSe]:MCPower:FREQuency(?)**

Sets or queries the center frequency in the MCPR measurement.

**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).

**Measurement Views** MCPR

**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.

**Syntax** [:SENSe]:MCPower:FREQuency:STEP <value>  
[:SENSe]:MCPower:FREQuency:STEP?

**Arguments** <value>: :=<NRf> specifies the frequency step size.  
Range: 0 to 6.2 GHz (RSA6106A) / 14 GHz (RSA6114A).

**Measurement Views** MCPR

**Examples** :SENSe:MCPower:FREQuency:STEP 50kHz  
sets the frequency step size to 50 kHz.

**Related Commands** [:SENSe]:MCPower:FREQuency:STEP:AUTO

**[:SENSe]:MCPower:FREQuency:STEP:AUTO(?)**

Determines whether to set the frequency step size automatically or manually in the MCPR measurement.

**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.

**Measurement Views** MCPR

**Examples** :SENSe:MCPower:FREQuency:STEP:AUTO ON  
specifies that the frequency step size is set automatically.

**Related Commands** [:SENSe]:MCPower:FREQuency:STEP

**[[:SENSE]:MCPower:NFLoor:STATE(?)]**

Determines whether to enable or disable correction for noise floor.

**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.

**Measurement Views** MCPR

**Examples** :SENSE:MCPower:NFLoor:STATE ON  
enables correction for noise floor.

**[[:SENSE]:MCPower:OPTimize(?)]**

Selects or queries the optimization method in the MCPR measurement.

**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.

**Measurement Views** MCPR

**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.

**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>).

**Measurement Views**   MCPR

**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.

**Syntax**     [:SENSe]:MCPower:RChannels:MAIN<x>

Where <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.

**Arguments**   None

**Measurement Views**   MCPR

**Examples**     :SENSe:MCPower:RChannels:MAIN3  
selects Main 3 for the power reference channel.

**Related Commands**   [:SENSe]:MCPower:CHANnel:MAIN commands

**[[:SENSE]:MCPower:RChannels:TOTal (No Query Form)**

Sets the power reference to the total power of all the active channels in the MCPR measurement.

**Syntax** [[:SENSE]:MCPower:RChannels:TOTal

**Arguments** None

**Measurement Views** MCPR

**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.

**Syntax** [[:SENSE]:MCPower:RRCRolloff <value>  
[[:SENSE]:MCPower:RRCRolloff?

**Arguments** <value>: :=<NRf> specifies the filter parameter.  
Range: 0.001 to 1, 0.0001 step.

**Measurement Views** MCPR

**Examples** :SENSE:MCPower:RRCRolloff 0.3  
sets the filter parameter to 0.3.

**Related Commands** [[:SENSE]:MCPower:CHANnel:FILTer

## [:SENSe]:OBWidth Subgroup

The [:SENSe]:OBWidth commands set up the conditions related to the Occupied Bandwidth (OBW) measurement.

Command Tree	Header	Parameter
	[SENSe]	
	:OBWidth	
	:AVERage	<boolean>
	:COUNt	<numeric_value>
	:BANDwidth :BWIDth	
	:MEASurement	<numeric_value>
	[:RESolution]	<numeric_value>
	:ACTual?	
	:AUTO	<boolean>
	:VIDeo	<numeric_value>
	:STATe	<boolean>
	:CLEar	
	:RESults	
	:FREQuency	
	:CENTer	<numeric_value>
	:STEP	<numeric_value>
	:AUTO	<boolean>
	:PERCent	<numeric_value>
	:XDBLevEl	<numeric_value>

**[[:SENSe]:OBWidth:AVERage(?]**

Selects or queries whether to enable or disable averaging in the Occupied Bandwidth measurement.

**Syntax**    [:SENSe]:OBWidth:AVERage { OFF | ON | 0 | 1 }  
[:SENSe]:OBWidth:AVERage?

**Arguments**    OFF or 0 disables averaging.  
ON or 1 enables averaging.

**Measurement Views**    Occupied Bandwidth

**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.

**Syntax**    [:SENSe]:OBWidth:AVERage:COUNT <number>  
[:SENSe]:OBWidth:AVERage:COUNT?

**Arguments**    <number>::=<NR1> specifies the average count. Range: 2 to 10000.

**Measurement Views**    Occupied Bandwidth

**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.

**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.

**Measurement Views** Occupied Bandwidth

**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.

**Syntax** [[:SENSe]:OBWidth:BANDwidth]:BWIDth[:RESolution] <value>

[[:SENSe]:OBWidth:BANDwidth]:BWIDth[:RESolution]?

**Arguments** <value>::=<NRf> specifies the RBW.  
Range: 100 Hz to 5 MHz.

**Measurement Views** Occupied Bandwidth

**Examples** :SENSe:OBWidth:BANDwidth:RESolution 200kHz  
sets the RBW to 200 kHz.

**Related Commands** [[:SENSe]:OBWidth:BANDwidth[:RESolution]:AUTO

**[[:SENSe]:OBWidth:BANDwidth]:BWIDth[:RESolution]:ACTual? (Query Only)**

Queries the actual resolution bandwidth (RBW) in the Occupied Bandwidth measurement.

**Syntax** `[[:SENSe]:OBWidth:BANDwidth]:BWIDth[:RESolution]:ACTual?`

**Arguments** None

**Returns** <NRf> The actual RBW in Hz.

**Measurement Views** Occupied Bandwidth

**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.

**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.

**Measurement Views** Occupied Bandwidth

**Examples** `:SENSe:OBWidth:BANDwidth:AUTO ON`  
sets the RBW automatically.

**Related Commands** `[[:SENSe]:OBWidth:BANDwidth]:BWIDth[:RESolution]`

**[:SENSe]:OBWidth:BANDwidth|:BWIDth:VIDeo(?)**

Sets or queries the video bandwidth (VBW). Programming a specified VBW sets [:SENSe]:OBWidth:BANDwidth|:BWIDth:VIDeo:STATe OFF.

**Syntax** [:SENSe]:OBWidth:BANDwidth|:BWIDth:VIDeo <value>

[:SENSe]:OBWidth:BANDwidth|:BWIDth:VIDeo?

**Arguments** <value>::=<NRf> specifies the VBW.  
Range: Current RBW/10<sup>4</sup> (1 Hz minimum) to Current RBW.

**Measurement Views** Occupied Bandwidth

**Examples** :SENSe:OBWidth:BANDwidth:VIDeo 200kHz  
sets the VBW to 200 kHz.

**Related Commands** [:SENSe]:OBWidth:BANDwidth|:BWIDth:VIDeo:STATe

**[:SENSe]:OBWidth:BANDwidth|:BWIDth:VIDeo:STATe(?)**

Determines whether to enable or disable the video bandwidth (VBW) in the Occupied Bandwidth measurement.

**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.

**Measurement Views** Occupied Bandwidth

**Examples** :SENSe:OBWidth:BANDwidth:VIDeo:STATe ON  
enables the VBW.

**Related Commands** [:SENSe]:OBWidth:BANDwidth|:BWIDth:VIDeo

**[:SENSe]:OBWidth:CLEar:RESuLts (No Query Form)**

Restarts the average trace.

This command is valid when [:SENSe]:OBWidth:AVERage is set to ON.

**Syntax** [:SENSe]:OBWidth:CLEar:RESuLts

**Arguments** None

**Measurement Views** Occupied Bandwidth

**Examples** :SENSe:OBWidth:CLEar:RESuLts  
restarts the average trace.

**Related Commands** [:SENSe]:OBWidth:AVERage

**[:SENSe]:OBWidth:FREQuency:CENTer(?)**

Sets or queries the center frequency in the Occupied Bandwidth measurement.

**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).

**Measurement Views** Occupied Bandwidth

**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.

**Syntax** [:SENSe]:OBWidth:FREQuency:STEP <value>  
[:SENSe]:OBWidth:FREQuency:STEP?

**Arguments** <value>::=<NRf> specifies the frequency step size.  
Range: 0 to 6.2 GHz (RSA6106A) / 14 GHz (RSA6114A).

**Measurement Views** Occupied Bandwidth

**Examples** :SENSe:OBWidth:FREQuency:STEP 1kHz  
sets the frequency step size to 1 kHz.

**Related Commands** [:SENSe]:OBWidth:FREQuency:STEP:AUTO

**[:SENSe]:OBWidth:FREQuency:STEP:AUTO(?)**

Determines whether to set the frequency step size automatically or manually in the Occupied Bandwidth measurement.

**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.

**Measurement Views** Occupied Bandwidth

**Examples** :SENSe:OBWidth:FREQuency:STEP:AUTO ON  
specifies that the frequency step size is set automatically.

**Related Commands** [:SENSe]:OBWidth:FREQuency:STEP

**[:SENSe]:OBWidth:PERCent(?)**

Sets or queries the occupied bandwidth percent power (power ratio of the occupied bandwidth to the measurement bandwidth).

**Syntax**     [:SENSe]:OBWidth:PERCent <value>

              [:SENSe]:OBWidth:PERCent?

**Arguments**   <value>::=<NRf> specifies the occupied bandwidth percent power.  
Range: 50 to 99.9%.

**Measurement Views**   Occupied Bandwidth

**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.

**Syntax**       [:SENSe]:OBWidth:XDBLevel <value>

              [:SENSe]:OBWidth:XDBLevel?

**Arguments**   <value>::=<NRf> specifies the x dB level.  
Range: -80 to -1 dB.

**Measurement Views**   Occupied Bandwidth

**Examples**       :SENSe:OBWidth:XDBLevel -10  
                  sets the x dB level to -10 dB.

## [[:SENSE]:PHVTime Subgroup

The [[:SENSE]:PHVTime commands set up the conditions related to the Phase versus Time measurement.

Command Tree	Header	Parameter
	[[:SENSE]	
	:PHVTime	
	:CLEar	
	:RESults	
	:FREQuency	
	:CENTer	<numeric_value>
	:SPAN	<numeric_value>
	:START	<numeric_value>
	:STEP	<numeric_value>
	:AUTO	<boolean>
	:STOP	<numeric_value>
	:MAXTracepoints	ONEK   TENK   HUNDredk   NEVerdecimate

## [[:SENSE]:PHVTime:CLEar:RESults (No Query Form)

Restarts multi-trace functions (Average and Max/Min Hold).

**Syntax** [[:SENSE]:PHVTime:CLEar:RESults

**Arguments** None

**Measurement Views** Phase versus Time

**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:

$$(start\ frequency) = (center\ frequency) - (span)/2$$

$$(stop\ frequency) = (center\ frequency) + (span)/2$$

---

**Syntax** [[:SENSe]:PHVTime:FREQuency:CENTer <value>

[[:SENSe]:PHVTime:FREQuency:CENTer?

**Arguments** <value>::=<NRf> specifies the center frequency.  
Range: 0 Hz to 6.2 GHz (RSA6106A) / 14 GHz (RSA6114A).

**Measurement Views** Phase versus Time

**Examples** :SENSe:PHVTime:FREQuency:CENTer 7.5GHz  
sets the center frequency to 7.5 GHz.

**Related Commands** [[:SENSe]:PHVTime:FREQuency:START,  
[:SENSe]:PHVTime:FREQuency:STOP

**[:SENSe]:PHVTime:FREQuency:SPAN(?)**

Sets or queries the frequency span in the Phase versus Time measurement.

**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)

**Measurement Views**    Phase versus Time

**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.

**Syntax**        [:SENSe]:PHVTime:FREQuency:START <value>  
                   [:SENSe]:PHVTime:FREQuency:START?

**Arguments**    <value>::=<NRf> is the measurement start frequency.  
                   Range: (center frequency)  $\pm$  (span)/2.

**Measurement Views**    Phase versus Time

**Examples**        :SENSe:PHVTime:FREQuency:START 6.95GHz  
                   sets the start frequency to 6.95 GHz.

**Related Commands**    [:SENSe]:PHVTime:FREQuency:CENTer,  
                               [:SENSe]:PHVTime:FREQuency:STOP

**[: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.

**Syntax** [:SENSe]:PHVTime:FREQUENCY:STEP <value>  
[:SENSe]:PHVTime:FREQUENCY:STEP?

**Arguments** <value>::=<NRf> specifies the frequency step size.  
Range: 0 Hz to 6.2 GHz (RSA6106A) / 14 GHz (RSA6114A).

**Measurement Views** Phase versus Time

**Examples** :SENSe:PHVTime:FREQUENCY:STEP 1.5kHz  
sets the step size to 1.5 kHz.

**Related Commands** [:SENSe]:PHVTime:FREQUENCY:STEP:AUTO

**[:SENSe]:PHVTime:FREQUENCY:STEP:AUTO(?)**

Determines whether to set the frequency step size automatically or manually.

**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.

**Measurement Views** Phase versus Time

**Examples** :SENSe:PHVTime:BANDwidth:AUTO ON  
sets the frequency step size automatically.

**Related Commands** [:SENSe]:PHVTime:FREQUENCY:STEP

## **[[: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.

**Syntax** [[:SENSe]:PHVTime:FREQUENCY:STOP <value>

[[:SENSe]:PHVTime:FREQUENCY:STOP?

**Arguments** <value>::=<NRf> is the measurement stop frequency.  
Range: (center frequency)  $\pm$  (span)/2.

**Measurement Views** Phase versus Time

**Examples** :SENSe:PHVTime:FREQUENCY:STOP 7.05GHz  
sets the stop frequency to 7.05 GHz.

**Related Commands** [[:SENSe]:PHVTime:FREQUENCY:CENTer,  
[[:SENSe]:PHVTime:FREQUENCY:START

**[:SENSe]:PHVTime:MAXTracepoints(?)**

Selects or queries the maximum trace points in the Phase versus Time measurement.

**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.

**Measurement Views**    Phase versus Time

**Examples**        :SENSe:PHVTime:MAXTracepoints TENK  
                  sets the maximum trace points to 10 k.



## [:SENSe]:PNOise Subgroup (Option 11 Only)

The [:SENSe]:PNOise commands set up the conditions related to the Phase Noise measurement.

Command Tree	Header	Parameter
	[:SENSe]	
	:PNOise	
	:AVERage	
	:COUNT	<numeric_value>
	:ENABLE	<boolean>
	:CARRier	
	:FREQuency	
	:TRACk	<boolean>
	:THReshold	<numeric_value>
	:CLEar	
	:RESults	
	:FREQuency	
	:INTEgration	
	:OFFSet	
	:START	<numeric_value>
	:STOP	<numeric_value>
	:PLOT	
	:OFFSet	
	:START	<numeric_value>
	:STOP	<numeric_value>
	:OPTimization	SPEEd   DRANge

**[: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.

**Syntax**     [:SENSe]:PNOise:AVERage:COUNT <number>  
              [:SENSe]:PNOise:AVERage:COUNT?

**Arguments**   <number>::=<NR1> specifies the average count.  
                  Range: 2 to 10000.

**Measurement Views**   Phase noise

**Examples**       :SENSe:PNOise:AVERage:COUNT 64  
                  sets the average count to 64.

**Related Commands**   [:SENSe]:PNOise:AVERage:ENABle

**[:SENSe]:PNOise:AVERage:ENABle(?)**

Determines whether to enable or disable averaging trace in the phase noise measurement.

**Syntax**       [:SENSe]:PNOise:AVERage:ENABle { OFF | ON | 0 | 1 }  
              [:SENSe]:PNOise:AVERage:ENABle?

**Arguments**    OFF disables averaging trace.  
                  ON enables averaging trace.

**Measurement Views**   Phase noise

**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.

**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.

**Measurement Views**   Phase noise

**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.

**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.

**Measurement Views**   Phase noise

**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.

**Syntax**     [:SENSe]:PNOise:CLEar:RESults

**Arguments**   None

**Measurement Views**   Phase noise

**Examples**       :SENSe:PNOise:CLEar:RESults  
                  restarts the average process.

**Related Commands**   [:SENSe]:PNOise:AVERage subgroup

**[:SENSE]:PNOise:FREQUENCY:INTEGRation:OFFSet:START(?)**

Sets or queries the start offset frequency for integration in the phase noise measurement.

**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.

**Measurement Views**   Phase noise

**Examples**       :SENSE:PNOise:FREQUENCY:INTEGRation:OFFSet:START 100kHz  
 sets the start offset frequency for integration to 100 kHz.

**Related Commands**   [:SENSE]:PNOise:FREQUENCY:INTEGRation:OFFSet:STOP

**[:SENSE]:PNOise:FREQUENCY:INTEGRation:OFFSet:STOP(?)**

Sets or queries the stop offset frequency for integration in the phase noise measurement.

**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.

**Measurement Views**   Phase noise

**Examples**       :SENSE:PNOise:FREQUENCY:INTEGRation:OFFSet:STOP 100MHz  
 sets the stop offset frequency for integration to 100 MHz.

**Related Commands**   [:SENSE]:PNOise:FREQUENCY:INTEGRation:OFFSet:START

**[:SENSe]:PNOise:FREQuency:PLOT:OFFSet:START(?)**

Sets or queries the start offset frequency for plotting the phase noise trace.

**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.

**Measurement Views**   Phase noise

**Examples**       :SENSe:PNOise:FREQuency:PLOT:OFFSet:START 100kHz  
                  sets the start offset frequency for plot to 100 kHz.

**Related Commands**   [:SENSe]:PNOise:FREQuency:PLOT:OFFSet:STOP

**[:SENSe]:PNOise:FREQuency:PLOT:OFFSet:STOP(?)**

Sets or queries the stop offset frequency for plotting the phase noise trace.

**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.

**Measurement Views**   Phase noise

**Examples**       :SENSe:PNOise:FREQuency:PLOT:OFFSet:STOP 100MHz  
                  sets the stop offset frequency for plot to 100 MHz.

**Related Commands**   [:SENSe]:PNOise:FREQuency:PLOT:OFFSet:START

**[:SENSe]:PNOise:OPTimization(?)**

Selects or queries the method of optimizing the gain and input bandwidth in the phase noise measurement.

**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.

**Measurement Views**    Phase noise

**Examples**    :SENSe:PNOise:OPTimization DRANge  
optimizes the gain and input bandwidth to maximize the dynamic range.

## [[:SENSE]:PULSE Subgroup (Option 20 Only)

The [[:SENSE]:PULSE commands set up the conditions related to the pulsed RF measurements.

Command Tree	Header	Parameter
	[[:SENSE]	
	:PULSE	
	:ANALyze	
	:LEVel	VOLTage   POWer
	:FIFTy	VOLTage   POWer
	:HUNDred	AVERage   INDEpendent
	:MEASurement	
	:TIME	
	:AUTO	<boolean>
	:STARt	<numeric_value>
	:STOP	<numeric_value>
	:PMLOCation	<numeric_value>
	:POINT	
	:LOCation	AVERage   INDEpendent
	:RFALl	WIDE   NARRow
	:RIPple	<numeric_value>
	:CARRier	
	:OFFSet	<numeric_value>
	:SEARCh	AUTO   MANuaL
	:DETECT	
	:MEASurement	<boolean>
	:NUMBer	<numeric_value>
	:POWer	
	[:THReshold]	<numeric_value>
	:TIME	
	[:THReshold]	<numeric_value>
	:FILTer	
	:BANdwidth :BWIDTH	<numeric_value>
	:MEASurement	GAUSSian   NONE   MAXRtbw
	:FREFerence	
	:AUTO	<boolean>
	:CHIRpbw	<numeric_value>
	:OFFSet	<numeric_value>
	:MODulation	
	:TYPE	CWConst   CWCHange   LCHirp
	:SIGNal	
	:TYPE	CWConst   CWCHange   LCHirp



**[[:SENSe]:PULSe:ANALyze:LEVEl(?]**

Selects or queries how to determine the 50% level for the pulsed RF measurements.

**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.

**Measurement Views** Pulse statistics, Pulse table, Pulse trace

**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.

**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.

**Measurement Views** Pulse statistics, Pulse table, Pulse trace

**Examples** :SENSe:PULSe:ANALyze:LEVEl:FIFTy POWer  
uses -3 dB to determine the 50% level.

**Related Commands** [[:SENSe]:PULSe:ANALyze: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.

**Syntax** [[:SENSe]:PULSe:ANALyze:LEVEl:HUNDred { AVERAge | INDEpendent }  
[[:SENSe]:PULSe:ANALyze:LEVEl:HUNDred?

**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.

**Measurement Views** Pulse statistics, Pulse table, Pulse trace

**Examples** :SENSe:PULSe:ANALyze:LEVEl:HUNDred AVERAge  
uses pulse average amplitude to set the 100% level.

**Related Commands** [[:SENSe]:PULSe:ANALyze:POINt:LOCation

## **[[: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.

**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.

**Measurement Views** Pulse statistics, Pulse table, Pulse trace

**Examples** `:SENSe:PULSe:ANALyze:MEASurement:TIME:AUTO ON`  
sets the measurement time for frequency and phase results automatically.

**Related Commands** `[[:SENSe]:PULSe:ANALyze:MEASurement:TIME:STARt`,  
`[[:SENSe]:PULSe:ANALyze:MEASurement:TIME:STOP`

**[ :SENSe ]:PULSe:ANALyze:MEASurement:TIME:START(?)**

Sets or queries the measurement start time for frequency and phase results in the pulsed RF measurements.

**Syntax** [ :SENSe ]:PULSe:ANALyze:MEASurement:TIME:START <value>

[ :SENSe ]:PULSe:ANALyze:MEASurement:TIME:START?

**Arguments** <value>::=<NRf> specifies the measurement start time from the 50% level of the pulse rising edge. Range: -100 to 100 ms.

**Measurement Views** Pulse statistics, Pulse table, Pulse trace

**Examples** :SENSe:PULSe:ANALyze:MEASurement:TIME:AUTO 2.8us  
sets the start time to 2.8  $\mu$ s from the 50% level of the pulse rising edge.

**Related Commands** [ :SENSe ]:PULSe:ANALyze:MEASurement:TIME:STOP

**[ :SENSe ]:PULSe:ANALyze:MEASurement:TIME:STOP(?)**

Sets or queries the measurement stop time for frequency and phase results in the pulsed RF measurements.

**Syntax** [ :SENSe ]:PULSe:ANALyze:MEASurement:TIME:STOP <value>

[ :SENSe ]:PULSe:ANALyze:MEASurement:TIME:STOP?

**Arguments** <value>::=<NRf> specifies the measurement stop time from the 50% level of the pulse falling edge. Range: -100 to 100 ms.

**Measurement Views** Pulse statistics, Pulse table, Pulse trace

**Examples** :SENSe:PULSe:ANALyze:MEASurement:TIME:STOP 1.2us  
sets the stop time to 1.2  $\mu$ s from the 50% level of the pulse falling edge.

**Related Commands** [ :SENSe ]:PULSe:ANALyze:MEASurement:TIME:START

**[::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.

**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.

**Measurement Views** Pulse statistics, Pulse table, Pulse trace

**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.

**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.

**Measurement Views** Pulse statistics, Pulse table, Pulse trace

**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.

**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.

**Measurement Views**    Pulse statistics, Pulse table, Pulse trace

**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).

**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.

**Measurement Views**    Pulse statistics, Pulse table, Pulse trace

**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.

**Syntax**    [:SENSe]:PULSe:CARRier:OFFSet <value>  
               [:SENSe]:PULSe:CARRier:OFFSet?

**Arguments**    <value>::=<NRf> specifies the carrier frequency offset.  
 Range: -50 kHz to +50 kHz.

**Measurement Views**    Pulse statistics, Pulse table, Pulse trace

**Examples**        :SENSe:PULSe:CARRier:OFFSet 2.5kHz  
 sets the offset frequency to 2.5 kHz to the carrier.

**Related Commands**    [:SENSe]:PULSe:CARRier:SEARch

**[:SENSe]:PULSe:CARRier:SEARch(?)**

Selects or queries how to detect the carrier in the pulsed RF measurements.

**Syntax**        [:SENSe]:PULSe:CARRier:SEARch { AUTO | MANua1 }  
               [:SENSe]:PULSe:CARRier:SEARch?

**Arguments**    AUTO specifies that the carrier is detected automatically.  
 MANua1 specifies that the carrier frequency offset is set manually, using the  
 [:SENSe]:PULSe:CARRier:OFFSet command.

**Measurement Views**    Pulse statistics, Pulse table, Pulse trace

**Examples**        :SENSe:PULSe:CARRier:SEARch AUTO  
 specifies that the carrier is detected automatically.

**Related Commands**    [:SENSe]:PULSe:CARRier:OFFSet

**[[:SENSe]:PULSe:DETECT:MEASurement(?)]**

Determines whether or not to set the maximum number of pulses to measure within the analysis time.

**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.

**Measurement Views**    Pulse statistics, Pulse table, Pulse trace

**Examples**    :SENSe:PULSe:DETECT:MEASurement ON  
specifies that the maximum number of pulses is set manually.

**Related Commands**    [:SENSe]:PULSe:DETECT:NUMBER



## **[: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.

**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.

**Measurement Views**     Pulse statistics, Pulse table, Pulse trace

**Examples**        :SENSe:PULSe:DETECT:MEASurement 850  
                      sets the maximum number of pulses to 850.

**Related Commands**     [:SENSe]:PULSe:DETECT:MEASurement

**[[:SENSe]:PULSe:DETECT:POWer[:THReshold](?)**

Sets or queries the power threshold to detect pulses.

**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.

**Measurement Views**   Pulse statistics, Pulse table, Pulse trace

**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.

**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.

**Measurement Views**   Pulse statistics, Pulse table, Pulse trace

**Examples**       :SENSe:PULSe:DETECT:TIME:THReshold 1.5ms  
                  sets the time threshold to 1.5 ms.

**[[:SENSe]:PULSe:FILTer:BAWdwidth]:BWIDth(?)**

Sets or queries the filter or acquisition bandwidth when [[:SENSe]:PULSe:FILTer:MEASurement] is set to GAUSSian or NONE (No filter), respectively.

**Syntax** [[:SENSe]:PULSe:FILTer:BAWdwidth]:BWIDth <value>

[[:SENSe]:PULSe:FILTer:BAWdwidth]:BWIDth?

**Arguments** <value>: :=<Nrf> specifies the filter/acquisition bandwidth depending the [[:SENSe]:PULSe:FILTer:MEASurement] setting. The table below shows the setting range.

[[: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).

You can enter any value, but it is rounded up to the next valid number.

**Measurement Views** Pulse statistics, Pulse table, Pulse trace

**Examples** :SENSe:PULSe:FILTer:MEASurement 10MHz  
sets the filter bandwidth to 10 MHz.

**Related Commands** [[:SENSe]:PULSe:FILTer:MEASurement]

**[[:SENSe]:PULSe:FILTer:MEASurement(?)**

Selects or queries the measurement filter in the pulsed RF measurements.

**Syntax** `[[:SENSe]:PULSe:FILTer:MEASurement { GAUSSian | NONE | MAXRtbw }  
[:SENSe]:PULSe:FILTer:MEASurement?`

**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).

**Measurement Views** Pulse statistics, Pulse table, Pulse trace

**Examples** `:SENSe:PULSe:FILTer:MEASurement GAUSSian`  
uses the Gaussian filter in the pulsed RF measurements.

**Related Commands** `[[:SENSe]:PULSe:FILTer:BANDwidth]:BWIDth`

## **[[:SENSe]:PULSe:FREFerence:AUTO(?)]**

Determines whether to estimate the pulse frequency reference automatically or manually in the pulsed RF measurements.

**Syntax**    [[:SENSe]:PULSe:FREFerence:AUTO { OFF | ON | 0 | 1 }  
[[:SENSe]:PULSe:FREFerence:AUTO?

**Measurement Views**    Pulse statistics, Pulse table, Pulse trace

**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.

**Related Commands**    [[:SENSe]:PULSe:FREFerence:CHIRpbw,  
[[:SENSe]:PULSe:FREFerence:OFFSet

**[[: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.

**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).

**Measurement Views** Pulse statistics, Pulse table, Pulse trace

**Examples** :SENSe:PULSe:FREference:CHIRpbw 1.5MHz  
sets the chirp bandwidth to 1.5 MHz.

**Related Commands** [:SENSe]:PULSe:FREference:AUTO, [:SENSe]:PULSe:MODulation:TYPE

**[[:SENSe]:PULSe:FREference:OFFSet(?)]**

Sets or queries the frequency reference offset. This command is valid when [:SENSe]:PULSe:FREference:AUTO is set to OFF.

**Syntax** [:SENSe]:PULSe:FREference:OFFSet <value>  
[:SENSe]:PULSe:FREference:OFFSet?

**Arguments** <value>::=<NRf> specifies the frequency reference offset.  
Range: -50 kHz to +50 kHz.

**Measurement Views** Pulse statistics, Pulse table, Pulse trace

**Examples** :SENSe:PULSe:FREference:OFFSet 2.5kHz  
sets the frequency offset to 2.5 kHz to the reference.

**Related Commands** [:SENSe]:PULSe:FREference:AUTO

**[[: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.

**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.

**Measurement Views** Pulse statistics, Pulse table, Pulse trace

**Examples** :SENSe:PULSe:MODulation:TYPE CWCHange  
selects the CW with changing phase as the modulation type.

**Related Commands** [[:SENSe]:PULSe:SIGNal:TYPE

**[[:SENSe]:PULSe:SIGNal:TYPE(?)]**

Selects or queries the signal type in the pulsed RF measurements.

**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.

**Measurement Views** Pulse statistics, Pulse table, Pulse trace

**Examples** :SENSe:PULSe:SIGNal:TYPE CWCHange  
selects the CW with changing phase as the signal type.

## [[:SENSE]:SGRam Subgroup

The [[:SENSE]:SGRam commands set up conditions related to the spectrogram.

Command Tree	Header	Parameter
	[[:SENSE]	
	:SGRam	
	:BANDwidth :BWIDTH	
	:OPTimization	AUTO   MAXDynrange   MINNoise   MINTime
	[:RESolution]	<numeric_value>
	:ACTual?	
	:AUTO	<boolean>
	:MODE	<boolean>
	:VIDeo	<numeric_value>
	:STATe	<boolean>
	:COLor	RED   GREen   BLUe   CYAN   BCYan   YELLow   MAGenta   GRAY   TEMPerature   SPECTral
	:MAXimum	
	:MINimum	
	:FFT	
	:WINDow	KAISer   MIL6db   CISPr   BH4B   UNIFORM   FLATtop   HANNing
	:FILTer	
	[:SHAPE]	KAISer   MIL6db   CISPr   BH4B   UNIFORM   FLATtop   HANNing
	:FREQuency	
	:CENTer	<numeric_value>
	:SPAN	<numeric_value>
	:BANDwidth	
	[:RESolution]	
	:RATio	<numeric_value>
	:MAXimum	
	:START	<numeric_value>
	:STEP	<numeric_value>
	:AUTO	<boolean>
	:STOP	<numeric_value>



**[[:SENSe]:SGRam:BANDwidth|:BWIDth:OPTimization(?)**

Selects or queries the method of optimizing the gain and input bandwidth in the spectrogram.

**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.

**Measurement Views**    Spectrogram

**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.

**Syntax** [:SENSe]:SGRam:BANDwidth]:BWIDth[:RESolution] <value>  
[:SENSe]:SGRam:BANDwidth]:BWIDth[:RESolution]?

**Arguments** <value>::=<NRf> specifies the RBW.  
Range: 1 Hz to 10 MHz.

**Measurement Views** Spectrogram

**Examples** :SENSe:SGRam:BANDwidth:RESolution 200kHz  
sets the RBW to 200 kHz.

**Related Commands** [:SENSe]:SGRam:BANDwidth]:BWIDth[:RESolution]:AUTO

**[[:SENSe]:SGRam:BANDwidth]:BWIDth[:RESolution]:ACTual? (Query Only)**

Queries the actual resolution bandwidth (RBW) in the spectrogram.

**Syntax** [:SENSe]:SGRam:BANDwidth]:BWIDth[:RESolution]:ACTual?

**Arguments** None

**Returns** <NRf> The actual RBW in Hz.

**Measurement Views** Spectrogram

**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.

**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]:SGRam:BANDwidth[:BWIDth[:RESolution] command.

ON or 1 specifies that the resolution bandwidth is set automatically.

**Measurement Views** Spectrogram

**Examples** :SENSe:SGRam:BANDwidth:AUTO ON  
sets the resolution bandwidth automatically.

**Related Commands** [[:SENSe]:DPSA:BANDwidth[:RESolution]

**[[:SENSE]:SGRam:BANDwidth]:BWIDth[:RESolution]:MODE(?)**

Determines whether to enable or disable the RBW processing in the spectrogram.

**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.

**Measurement Views** Spectrogram

**Examples** `:SENSE:SGRam:BANDwidth:MODE ON`  
enables the RBW processing.

**Related Commands** `[[:SENSE]:SGRam:BANDwidth]:BWIDth[:RESolution]`,  
`[[:SENSE]:SGRam:FFT:WINDow]`

**[[:SENSe]:SGRam:BANDwidth]:BWIDth:VIDeo(?)**

Sets or queries the video bandwidth (VBW). Programming a specified VBW sets [[:SENSe]:SGRam:BANDwidth]:BWIDth:VIDeo:STATe OFF.

**Syntax** [[:SENSe]:SGRam:BANDwidth]:BWIDth:VIDeo <value>

[[:SENSe]:SGRam:BANDwidth]:BWIDth:VIDeo?

**Arguments** <value>::=<NRf> specifies the VBW.  
Range: Current RBW/10<sup>4</sup> (1 Hz minimum) to Current RBW.

**Measurement Views** Spectrogram

**Examples** :SENSe:SGRam:BANDwidth:VIDeo 200kHz  
sets the VBW to 200 kHz.

**Related Commands** [[:SENSe]:SGRam:BANDwidth]:BWIDth:VIDeo:STATe

**[[:SENSe]:SGRam:BANDwidth]:BWIDth:VIDeo:STATe(?)**

Determines whether to enable or disable the video bandwidth (VBW) in the spectrogram measurement.

**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.

**Measurement Views** Spectrogram

**Examples** :SENSe:SGRam:BANDwidth:VIDeo:STATe ON  
enables the VBW.

**Related Commands** [[:SENSe]:SGRam:BANDwidth]:BWIDth:VIDeo

**[:SENSe]:SGRam:COLor(?)**

Selects or queries the color palette of three-dimensional graphs.

**Syntax** `[:SENSe]:SGRam:COLor { RED | GREen | BLUe | CYAN | BCYan  
| YELLow | MAGenta | GRAY | TEMPerature | SPECtral }`

`[:SENSe]:SGRam:COLor?`

**Arguments** Table 2-47 lists the arguments.

**Table 2-47: Color palette for spectrogram**

Argument	Palette
RED	Red
GREen	Green
BLUe	Blue
CYAN	Cyan
BCYan	Binary cyan
YELLow	Yellow
MAGenta	Magenta
GRAY	Gray
TEMPerature	Temperature
SPECtral	Spectral

**Measurement Views** Spectrogram

**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.

**Syntax**     [:SENSe]:SGRam:COLor:MAXimum <value>  
                  [:SENSe]:SGRam:COLor:MAXimum?

**Arguments**   <value>::=<NRf> specifies the maximum value of the color axis.  
 Range: -100 to +100 dBm.

**Measurement Views**   Spectrogram

**Examples**       :SENSe:SGRam:COLor:MAXimum 10  
 sets the maximum value of the color axis to 10 dBm.

**Related Commands**   [:SENSe]:SGRam:COLor:MINimum

**[:SENSe]:SGRam:COLor:MINimum(?)**

Sets or queries the minimum value of the color axis in the spectrogram.

**Syntax**       [:SENSe]:SGRam:COLor:MINimum <value>  
                  [:SENSe]:SGRam:COLor:MINimum?

**Arguments**   <value>::=<NRf> specifies the minimum value of the color axis.  
 Range: -100 to +100 dBm.

**Measurement Views**   Spectrogram

**Examples**       :SENSe:SGRam:COLor:MINimum 10  
 sets the minimum value of the color axis to 10 dBm.

**Related Commands**   [:SENSe]:SGRam:COLor:MAXimum

**[:SENSe]:SGRam:FFT:WINDow(?)**

Selects or queries the FFT window in the spectrogram.  
This command is equivalent to [:SENSe]:SGRam:FILTer[:SHAPE].

**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.

**Measurement Views** Spectrogram

**Examples** :SENSe:SGRam:FFT:WINDow HANNing  
selects the Hanning window.

**Related Commands** [:SENSe]:SGRam:FILTer[:SHAPE]



## **[[:SENSe]:SGRam:FILTer[:SHAPE](?)**

Selects or queries the filter shape in the spectrogram.  
This command is equivalent to [[:SENSe]:SGRam:FFT:WINDow.

**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.

**Measurement Views** Spectrogram

**Examples** :SENSe:SGRam:FILTer:SHAPE HANNing  
selects the Hanning window.

**Related Commands** [[:SENSe]:SGRam:FFT: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:

$$(start\ frequency) = (center\ frequency) - (span)/2$$

$$(stop\ frequency) = (center\ frequency) + (span)/2$$

---

**Syntax** [[:SENSe]:SGRam:FREQuency:CENTer <value>

[[:SENSe]:SGRam:FREQuency:CENTer?

**Arguments** <value>::=<NRf> specifies the center frequency.  
Range: 0 Hz to 6.2 GHz (RSA6106A) / 14 GHz (RSA6114A).

**Measurement Views** Spectrogram

**Examples** :SENSe:SGRam:FREQuency:CENTer 7.5GHz  
sets the center frequency to 7.5 GHz.

**Related Commands** [[:SENSe]:SGRam:FREQuency:STARt, [[:SENSe]:SGRam:FREQuency:STOP

**[:SENSe]:SGRam:FREQuency:SPAN(?)**

Sets or queries the frequency span in the spectrogram.

**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).

**Measurement Views**    Spectrogram

**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.

**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.

**Measurement Views**    Spectrogram

**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.

**Related Commands**    [:SENSe]:SGRam:BANDwidth[:BWIDth[:RESolution]:AUTO]

**[[:SENSe]:SGRam:FREQuency:SPAN:MAXimum (No Query Form)**

Sets the frequency range to the maximum real-time span in the spectrogram.

**Syntax** [[:SENSe]:SGRam:FREQuency:SPAN:MAXimum <value>

**Arguments** None

**Measurement Views** Spectrogram

**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.

**Syntax** [[:SENSe]:SGRam:FREQuency:START <value>

[[:SENSe]:SGRam:FREQuency:START?

**Arguments** <value>::=<Nrf> is the measurement start frequency.  
Range: (center frequency)  $\pm$  (span)/2.

**Measurement Views** Spectrogram

**Examples** :SENSe:SGRam:FREQuency:START 6.95GHz  
sets the start frequency to 6.95 GHz.

**Related Commands** [[:SENSe]:SGRam:FREQuency:CENTer, [[:SENSe]:SGRam:FREQuency:STOP

**[:SENSe]:SGRam:FREQuency:STEP(?)**

Sets or queries the frequency step size. Programming a specified step size sets [:SENSe]:SGRam:FREQuency:STEP:AUTO OFF.

**Syntax**     [:SENSe]:SGRam:FREQuency:STEP <value>  
                  [:SENSe]:SGRam:FREQuency:STEP?

**Arguments**     <value>::=<NRf> specifies the frequency step size.  
                  Range: 0 to 6.2 GHz (RSA6106A) / 14 GHz (RSA6114A).

**Measurement Views**     Spectrogram

**Examples**     :SENSe:SGRam:FREQuency:STEP 1kHz  
                  sets the frequency step size to 1 kHz.

**Related Commands**     [:SENSe]:SGRam:FREQuency:STEP:AUTO

**[:SENSe]:SGRam:FREQuency:STEP:AUTO(?)**

Determines whether to set the frequency step size automatically or manually in the spectrogram.

**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.

**Measurement Views**     Spectrogram

**Examples**     :SENSe:SGRam:FREQuency:STEP:AUTO ON  
                  specifies that the frequency step size is set automatically.

**Related Commands**     [:SENSe]:SGRam:FREQuency:STEP

**[[: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.

**Syntax** [[:SENSe]:SGRam:FREQuency:STOP <value>

[[:SENSe]:SGRam:FREQuency:STOP?

**Arguments** <value>::=<NRf> is the measurement start frequency.  
Range: (center frequency)  $\pm$  (span)/2.

**Measurement Views** Spectrogram

**Examples** :SENSe:SGRam:FREQuency:STOP 7.05GHz  
sets the stop frequency to 7.05 GHz.

**Related Commands** [[:SENSe]:SGRam:FREQuency:CENTer,  
[[:SENSe]:SGRam:FREQuency:START

## [:SENSE]:SPECTrum Subgroup

The [:SENSE]:SPECTrum commands set up the conditions related to the Spectrum measurement.

Command Tree	Header	Parameter
	[:SENSE]	
	:SPECTrum	
	:BANDwidth :BWIDth	
	:OPTimization	AUTO   MAXDynrange   MINNoise   MINTime
	[:RESolution]	<numeric_value>
	:ACTual?	
	:AUTO	<boolean>
	:MODE	<boolean>
	:VIDeo	<numeric_value>
	:STATe	<boolean>
	:CLEar	
	:RESults	
	:FFT	
	:WINDow	KAISer   MIL6db   CISPr   BH4B   UNIFORM   FLATtop   HANNing
	:FILTer	
	[:SHAPE]	KAISer   MIL6db   CISPr   BH4B   UNIFORM   FLATtop   HANNing
	:FREQuency	
	:CENTer	<numeric_value>
	:SPAN	<numeric_value>
	:BANDwidth	
	[:RESolution]	
	:RATio	<numeric_value>
	:START	<numeric_value>
	:STEP	<numeric_value>
	:AUTO	<boolean>
	:STOP	<numeric_value>
	:MAX	
	:SPAN	
	:POINTs	
	:COUNT	P801   P2401   P4001   P8001   P10401

**[[:SENSe]:SPECTrum:BANDwidth]:BWIDth:OPTimization(?)**

Selects or queries the method of optimizing the gain and input bandwidth in the spectrum measurement.

**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.

**Measurement Views** Spectrum

**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.

**Syntax** [[:SENSe]:SPECTrum:BANDwidth]:BWIDth[:RESolution] <value>

[[:SENSe]:SPECTrum:BANDwidth]:BWIDth[:RESolution]?

**Arguments** <value>::=<NRf> specifies the RBW.  
Range: 1 Hz to 10 MHz.

**Measurement Views** Spectrum

**Examples** :SENSe:SPECTrum:BANDwidth:RESolution 200kHz  
sets the RBW to 200 kHz.

**Related Commands** [[:SENSe]:SPECTrum:BANDwidth]:BWIDth[:RESolution]:AUTO

**[[:SENSe]:SPECTrum:BANDwidth]:BWIDth[:RESolution]:ACTual? (Query Only)**

Queries the actual resolution bandwidth (RBW) in the spectrum measurement.

**Syntax** [[:SENSe]:SPECTrum:BANDwidth]:BWIDth[:RESolution]:ACTual?

**Arguments** None

**Returns** <NRf> The actual RBW in Hz.

**Measurement Views** Spectrum

**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.

**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.

**Measurement Views** Spectrum

**Examples** `:SENSe:SPECTrum:BANDwidth:AUTO ON`  
sets the resolution bandwidth automatically.

**Related Commands** `[[:SENSe]:SPECTrum:BANDwidth]:BWIDth[:RESolution]`

**[[:SENSE]:SPECTrum:BANDwidth]:BWIDth[:RESolution]:MODE(?)**

Determines whether to enable or disable the RBW process.

**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.

**Measurement Views**    Spectrum

**Examples**        :SENSE:SPECTrum:BANDwidth:MODE ON  
                      enables the RBW process.

**Related Commands**    [[:SENSE]:SPECTrum:BANDwidth]:BWIDth[:RESolution],  
                          [[:SENSE]:SPECTrum:FFT:WINDow]

**[[:SENSe]:SPECTrum:BANDwidth]:BWIDth:VIDeo(?)**

Sets or queries the video bandwidth (VBW). Programming a specified VBW sets [:SENSe]:SPECTrum:BANDwidth]:BWIDth:VIDeo:STATe OFF.

**Syntax** [:SENSe]:SPECTrum:BANDwidth]:BWIDth:VIDeo <value>  
[:SENSe]:SPECTrum:BANDwidth]:BWIDth:VIDeo?

**Arguments** <value>: :=<NRf> specifies the VBW.  
Range: Current RBW/10<sup>4</sup> (1 Hz minimum) to Current RBW.

**Measurement Views** Spectrum

**Examples** :SENSe:SPECTrum:BANDwidth:VIDeo 200kHz  
sets the VBW to 200 kHz.

**Related Commands** [:SENSe]:SPECTrum:BANDwidth]:BWIDth:VIDeo:STATe

**[[:SENSe]:SPECTrum:BANDwidth]:BWIDth:VIDeo:STATe(?)**

Determines whether to enable or disable the video bandwidth (VBW) in the spectrum measurement.

**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.

**Measurement Views** Spectrum

**Examples** :SENSe:SPECTrum:BANDwidth:VIDeo:STATe ON  
enables the VBW.

**Related Commands** [:SENSe]:SPECTrum:BANDwidth]:BWIDth:VIDeo

**[:SENSe]:SPECtrum:CLEar:RESults (No Query Form)**

Restarts multi-trace functions (Average and Max/Min Hold).

**Syntax**    [:SENSe]:SPECtrum:CLEar:RESults

**Arguments**    None

**Measurement Views**    Spectrum

**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].

**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.  
                 FLATtop selects the flat-top window.  
                 HANNing selects the Hanning window.  
                 BH4B selects the Blackman-Harris 4B type window.  
                 UNIFORM selects the uniform window.

**Measurement Views**     Spectrum

**Examples**     :SENSe:SPEctrum:FFT:WINDow HANNing  
                 selects the Hanning window.

**Related Commands**     [:SENSe]:SPEctrum:FILTer[:SHAPE]

## **[[:SENSe]:SPECTrum:FILTer[:SHAPE](?)**

Selects or queries the filter shape in the spectrum measurement.  
This command is equivalent to [[:SENSe]:SPECTrum:FFT:WINDow.

**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.  
FLATtop selects the flat-top window.  
HANNing selects the Hanning window.  
BH4B selects the Blackman-Harris 4B type window.  
UNIFORM selects the uniform window.

**Measurement Views** Spectrum

**Examples** :SENSe:SPECTrum:FILTer:SHAPE HANNing  
selects the Hanning window.

**Related Commands** [[:SENSe]:SPECTrum:FFT: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:

$$(start\ frequency) = (center\ frequency) - (span)/2$$

$$(stop\ frequency) = (center\ frequency) + (span)/2$$

---

**Syntax** [[:SENSe]:SPECTrum:FREQuency:CENTer <value>

[[:SENSe]:SPECTrum:FREQuency:CENTer?

**Arguments** <value>::=<NRf> specifies the center frequency.  
Range: 0 Hz to 6.2 GHz (RSA6106A) / 14 GHz (RSA6114A).

**Measurement Views** Spectrum

**Examples** :SENSe:SPECTrum:FREQuency:CENTer 7.5GHz  
sets the center frequency to 7.5 GHz.

**Related Commands** [[:SENSe]:SPECTrum:FREQuency:START,  
[:SENSe]:SPECTrum:FREQuency:STOP



**[[:SENSe]:SPECTrum:FREQUency:SPAN(?)]**

Sets or queries the frequency span in the spectrum measurement.

**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).

**Measurement Views** Spectrum

**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.

**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.

**Measurement Views** Spectrum

**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.

**Related Commands** [[:SENSe]:SPECTrum:BANDwidth[:BWIDth[:RESolution]:AUTO

**[[: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.

**Syntax** [[:SENSe]:SPECTrum:FREQuency:STARt <value>

[[:SENSe]:SPECTrum:FREQuency:STARt?

**Arguments** <value>::=<NRf> is the measurement start frequency.  
Range: (center frequency)  $\pm$  (span)/2.

**Measurement Views** Spectrum

**Examples** :SENSe:SPECTrum:FREQuency:STARt 6.95GHz  
sets the start frequency to 6.95 GHz.

**Related Commands** [[:SENSe]:SPECTrum:FREQuency:CENTer,  
[[:SENSe]:SPECTrum:FREQuency:STOP

**[: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.

**Syntax** [:SENSe]:SPEctrum:FREQuency:STEP <value>  
[:SENSe]:SPEctrum:FREQuency:STEP?

**Arguments** <value>::=<NRf> specifies the frequency step size.  
Range: 0 Hz to 6.2 GHz (RSA6106A) / 14 GHz (RSA6114A).

**Measurement Views** Spectrum

**Examples** :SENSe:SPEctrum:FREQuency:STEP 1.5kHz  
sets the step size to 1.5 kHz.

**Related Commands** [:SENSe]:SPEctrum:FREQuency:STEP:AUTO

**[:SENSe]:SPEctrum:FREQuency:STEP:AUTO(?)**

Determines whether to set the frequency step size automatically or manually.

**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.

**Measurement Views** Spectrum

**Examples** :SENSe:SPEctrum:BANDwidth:AUTO ON  
sets the frequency step size automatically.

**Related Commands** [:SENSe]:SPEctrum:FREQuency:STEP

**[[: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.

**Syntax** [[:SENSe]:SPECTrum:FREQuency:STOP <value>

[[:SENSe]:SPECTrum:FREQuency:STOP?

**Arguments** <value>::=<NRf> is the measurement start frequency.  
Range: (center frequency)  $\pm$  (span)/2.

**Measurement Views** Spectrum

**Examples** :SENSe:SPECTrum:FREQuency:STOP 7.05GHz  
sets the stop frequency to 7.05 GHz.

**Related Commands** [[:SENSe]:SPECTrum:FREQuency:CENTer,  
[[:SENSe]:SPECTrum:FREQuency:START

## **[:SENSe]:SPEcTrum:MAX:SPAN (No Query Form)**

Sets the frequency span to the maximum real-time span.

**Syntax**     [:SENSe]:SPEcTrum:MAX:SPAN

**Arguments**     None

**Measurement Views**     Spectrum

**Examples**     :SENSe:SPEcTrum::MAX:SPAN  
sets the frequency span to the maximum real-time span.

## **[:SENSe]:SPEcTrum:POINts:COUNT(?)**

Sets or queries the number of sample points on the signal spectrum.

**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.

**Measurement Views**     Spectrum

**Examples**     :SENSe:SPEcTrum:POINts:COUNT P801  
sets the number of sample points to 801.

## [[:SENSe]:SPURious Subgroup

The [[:SENSe]:SPURious commands set up the conditions related to the Spurious measurement.

Command Tree	Header	Parameter
	[[:SENSe]	
	:SPURious	
	:CARRier	
	:BANDwidth	<numeric_value>
	:INTegration	<numeric_value>
	[:RESolution]	<numeric_value>
	:AUTO	<boolean>
	:DETection	AVERAge   PEAK
	:FREQuency	<numeric_value>
	:THReshold	<numeric_value>
	:CLEar	
	:RESuIts	
	[:FREQuency]	
	:OVERlap?	
	:LIST	ALL   OVERlimit
	:MODE	MULTi   SINGLE
	:OPTimization	AUTO   MAXDynrange   MINNoise
	:POINts	
	:COUNT	P801   P2401   P4001   P8001   P10401
	:RANGe<x>	
	:BANDwidth	
	:VIDeo	<numeric_value>
	:STATe	<boolean>
	:DETection	AVERAge   PEAK   QUASipeak   CAverage   CPEak
	:EXCursion	<numeric_value>
	:FILTer	
	[:SHAPE]	RBW   MIL6db   CISPr
	:BANDwidth	<numeric_value>
	:AUTO	<boolean>
	:FREQuency	
	:START	<numeric_value>
	:STOP	<numeric_value>

---

```
:LIMit
  :ABSolute
    :START <numeric_value>
    :STOP <numeric_value>
  :MASK ABS | REL | AND | OR | OFF
  :RELative
    :START <numeric_value>
    :STOP <numeric_value>
  :STATE <boolean>
  :THReshold <numeric_value>
:REFerence CARRier | MANUal | NONE
  :MANUal
    :POWER <numeric_value>
```

**[[:SENSE]:SPURious:CARRier:BANDwidth(?)]**

Sets or queries the channel bandwidth for the carrier as power reference.  
This command is valid when [[:SENSE]:SPURious:REFeRence] is set to CARRier.

**Syntax** [[:SENSE]:SPURious:CARRier:BANDwidth <value>  
[[:SENSE]:SPURious:CARRier:BANDwidth?

**Arguments** <value>::=<NRf> specifies the channel bandwidth.  
Range: 100 Hz to 40 MHz (Standard) / 110 MHz (Option 110).

**Measurement Views** Spurious

**Examples** :SENSE:SPURious:CARRier:BANDwidth 20MHz  
sets the channel bandwidth to 20 MHz.

**Related Commands** [[:SENSE]:SPURious:REFeRence]

**[[:SENSE]:SPURious:CARRier:BANDwidth:INTEgration(?)]**

Sets or queries the integration bandwidth to calculate the carrier power.  
This command is valid when [[:SENSE]:SPURious:REFeRence] is set to CARRier.

**Syntax** [[:SENSE]:SPURious:CARRier:BANDwidth:INTEgration  
<value>  
[[:SENSE]:SPURious:CARRier:BANDwidth:INTEgration?

**Arguments** <value>::=<NRf> specifies the integration bandwidth.  
Range: 100 Hz to 40 MHz (Standard) / 110 MHz (Option 110).

**Measurement Views** Spurious

**Examples** :SENSE:SPURious:CARRier:BANDwidth:INTEgration 2MHz  
sets the integration bandwidth to 2 MHz.

**Related Commands** [[:SENSE]:SPURious:REFeRence]



**[[:SENSe]:SPURious:CARRier:BANDwidth[: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.

**Syntax** [[:SENSe]:SPURious:CARRier:BANDwidth[:RESolution] <value>  
[[:SENSe]:SPURious:CARRier:BANDwidth[:RESolution]?

**Arguments** <value>::=<Nrf> specifies the RBW.  
Range: 1 Hz to 10 MHz.

**Measurement Views** Spurious

**Examples** :SENSe:SPURious:CARRier:BANDwidth:RESolution 200kHz  
sets the RBW to 200 kHz.

**Related Commands** [[:SENSe]:SPURious:CARRier:BANDwidth[:RESolution]:AUTO,  
[[:SENSe]:SPURious:REFeRence

**[[:SENSe]:SPURious:CARRier:BANDwidth[: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.

**Syntax**     [:SENSe]:SPURious:CARRier:BANDwidth[:RESolution]:AUTO  
                  { OFF | ON | 0 | 1 }

[:SENSe]:SPURious:CARRier:BANDwidth[:RESolution]:AUTO?

**Arguments**   OFF or 0 specifies that the resolution bandwidth is set manually using the [:SENSe]:SPURious:CARRier:BANDwidth[:RESolution] command.

ON or 1 specifies that the resolution bandwidth is set automatically.

**Measurement Views**   Spurious

**Examples**       :SENSe:SPURious:CARRier:BANDwidth:RESolution:AUTO ON  
sets the resolution bandwidth automatically.

**Related Commands**   [:SENSe]:SPURious:CARRier:BANDwidth[:RESolution],  
[:SENSe]:SPURious:REFerence

**[:SENSe]:SPURious:CARRier:DETection(?)**

Selects or queries the carrier detection method.

This command is valid when [:SENSe]:SPURious:REFeRence is set to CARRier.

**Syntax** [:SENSe]:SPURious:CARRier:DETection { AVERAge | PEAK }

[:SENSe]:SPURious:CARRier:DETection?

**Arguments** AVERAge selects the average detection.

PEAK selects the peak detection.

**Measurement Views** Spurious

**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.

**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).

**Measurement Views** Spurious

**Examples** :SENSe:SPURious:CARRier:FREQuency 7.5GHz  
sets the carrier frequency to 7.5 GHz.

**Related Commands** [:SENSe]:SPURious:REFeRence

**[[: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.

**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.

**Measurement Views** Spurious

**Examples** :SENSe:SPURious:CARRier:THReshold -25  
sets the carrier threshold level to -25 dBm.

**Related Commands** [[:SENSe]:POWEr:UNITs], [[:SENSe]:SPURious:REFEreNce], :UNIT:POWEr

**[[:SENSe]:SPURious:CLEAr:RESults (No Query Form)]**

Restarts multi-trace functions (Average and Max Hold).

**Syntax** [[:SENSe]:SPURious:CLEAr:RESults

**Arguments** None

**Measurement Views** Spurious

**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.*

---

**Syntax** [:SENSe]:SPURious[:FREQuency]:OVERlap?

**Arguments** None

**Returns** 0 (no overlap) or 1 (overlap).

**Measurement Views** Spurious

**Examples** :SENSe:SPURious:FREQuency:OVERlap?  
might return 1, indicating that some frequency ranges overlap.

**Related Commands** [:SENSe]:SPURious:REFEreNce

**[:SENSe]:SPURious:LIST(?)**

Selects or queries how to list the spurious signals in the Spurious measurement.

**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.

**Measurement Views**   Spurious

**Examples**       :SENSe:SPURious:LIST ALL  
                  lists all of the detected spurious signals.

**Related Commands**   [:SENSe]:SPURious:RANGe<x>:LIMit command group

**[:SENSe]:SPURious:MODE(?)**

Selects or queries the frequency range mode in the Spurious measurement.

**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.

**Measurement Views**    Spurious

**Examples**    :SENSe:SPURious:MODE MULTi  
 displays all of the enabled ranges.

**Related Commands**    :DISPlay:SPURious:SElect:NUMBer, [:SENSe]:SPURious:RANGe<x>:STATe

**[:SENSe]:SPURious:OPTimization(?)**

Selects or queries the method of optimizing the gain and input bandwidth in the Spurious measurement.

**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.

**Measurement Views**    Spurious

**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.

**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.

**Measurement Views**    Spurious

**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.

**Syntax** [:SENSe]:SPURious:RANGe<x>:BANDwidth:VIDeo <value>  
[:SENSe]:SPURious:RANGe<x>:BANDwidth:VIDeo?

**Arguments** <value>::=<NRf> specifies the VBW.  
Range: Current RBW/10<sup>4</sup> (1 Hz minimum) to Current RBW.

**Measurement Views** Spurious

**Examples** :SENSe:SPURious:RANGe1:BANDwidth:VIDeo 200kHz  
sets the VBW to 200 kHz for Range A.

**Related Commands** [:SENSe]:SPURious:BANDwidth:VIDeo:STATe

**[:SENSe]:SPURious:RANGe<x>:BANDwidth:VIDeo:STATe(?)**

Determines whether to enable or disable the video bandwidth (VBW) in the specified frequency range.

**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.

**Measurement Views** Spurious

**Examples** :SENSe:SPURious:RANGe1:BANDwidth:VIDeo:STATe ON  
enables the VBW for Range A.

**Related Commands** [:SENSe]:SPURious:RANGe<x>:BANDwidth:VIDeo

**[[:SENSE]:SPURious:RANGe<x>:DETection(?)]**

Selects or queries the spurious detection method in the specified frequency range.

**Syntax**    [:SENSE]:SPURious:RANGe<x>:DETection  
              { AVERage | PEAK | QUASipeak | CAVerage | CPEak }  
[:SENSE]:SPURious:RANGe<x>:DETection?

Where <x> = 1 to 20, representing Range A to T, respectively.

**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.

**Measurement Views**    Spurious

**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.

**Syntax** [[:SENSe]:SPURious:RANGe<x>:EXCursion <value>

[[:SENSe]:SPURious:RANGe<x>:EXCursion?

Where <x> = 1 to 20, representing Range A to T, respectively.

**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.

**Measurement Views** Spurious

**Examples** :SENSe:SPURious:RANGe1:EXCursion 8  
sets the excursion level to 8 dB.

**Related Commands** [[:SENSe]:SPURious:RANGe<x>:THReshold

**[[:SENSE]:SPURious:RANGe<x>:FILTer[:SHAPE](?)**

Selects or queries the filter shape to search the specified frequency range for spurious signals.

**Syntax** [[:SENSE]:SPURious:RANGe<x>:FILTer[:SHAPE]  
{ RBW | MIL6db | CISPr }

[[:SENSE]:SPURious:RANGe<x>:FILTer[:SHAPE]?

Where <x> = 1 to 20, representing Range A to T, respectively.

**Arguments** RBW selects the RBW filter.

MIL6db selects the -6 dB RBW (MIL) filter.

CISPr selects the CISPR filter.

**Measurement Views** Spurious

**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.

**Syntax** [[:SENSe]:SPURious:RANGe<x>:FILTer[:SHAPE]:BANDwidth <value>

[[:SENSe]:SPURious:RANGe<x>:FILTer[:SHAPE]:BANDwidth?

Where <x> = 1 to 20, representing Range A to T, respectively.

**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.

**Measurement Views** Spurious

**Examples** :SENSe:SPURious:RANGe1:FILTer:SHAPE:BANDwidth 200kHz  
sets the filter bandwidth to 200 kHz for Range A.

**Related Commands** [[:SENSe]:SPURious:RANGe<x>:DETection,  
[[:SENSe]:SPURious:RANGe<x>:FILTer[:SHAPE]:BANDwidth:AUTO

**[[: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.

**Syntax** `[[:SENSE]:SPURious:RANGE<x>:FILTER[:SHAPE]:BANDwidth:AUTO  
{ OFF | ON | 0 | 1 }`

`[[:SENSE]:SPURious:RANGE<x>:FILTER[:SHAPE]:BANDwidth:AUTO?`

Where <x> = 1 to 20, representing Range A to T, respectively.

**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.

**Measurement Views** Spurious

**Examples** `:SENSE:SPURious:RANGE1:FILTER:SHAPE:BANDwidth:AUTO ON`  
sets the filter bandwidth automatically for Range A.

**Related Commands** `[[:SENSE]:SPURious:RANGE<x>:FILTER[:SHAPE]:BANDwidth`

**[[:SENSe]:SPURious:RANGe<x>:FREQuency:START(?]**

Sets or queries the start frequency of the specified range in the Spurious measurement.

---

**NOTE.** *The frequency ranges must not be overlapped. Use the [[:SENSe]:SPURious[:FREQuency]:OVERlap? query to check whether there is any overlap.*

---

**Syntax** [[:SENSe]:SPURious:RANGe<x>:FREQuency:START <value>

[[:SENSe]:SPURious:RANGe<x>:FREQuency:START?

Where <x> = 1 to 20, representing Range A to T, respectively.

**Arguments** <value>: :=<Nrf> specifies the start frequency.  
Range: 0 Hz to 6.2 GHz (RSA6106A) / 14 GHz (RSA6114A).

**Measurement Views** Spurious

**Examples** :SENSe:SPURious:RANGe1:FREQuency:START 1.615GHz  
sets the start frequency of Range A to 1.615 GHz.

**Related Commands** [[:SENSe]:SPURious[:FREQuency]:OVERlap?,  
[[:SENSe]:SPURious:RANGe<x>:FREQuency:STOP

**[[:SENSe]:SPURious:RANGe<x>:FREQuency:STOP(?]**

Sets or queries the stop frequency of the specified range in the Spurious measurement.

---

**NOTE.** *The frequency ranges must not be overlapped. Use the [[:SENSe]:SPURious[:FREQuency]:OVERlap? query to check whether there is any overlap.*

---

**Syntax** [[:SENSe]:SPURious:RANGe<x>:FREQuency:STOP <value>

[[:SENSe]:SPURious:RANGe<x>:FREQuency:STOP?

Where <x> = 1 to 20, representing Range A to T, respectively.

**Arguments** <value>::=<NRf> specifies the stop frequency.  
Range: 0 Hz to 6.2 GHz (RSA6106A) / 14 GHz (RSA6114A).

**Measurement Views** Spurious

**Examples** :SENSe:SPURious:RANGe1:FREQuency:STOP 1.715GHz  
sets the stop frequency of Range A to 1.715 GHz.

**Related Commands** [[:SENSe]:SPURious[:FREQuency]:OVERlap?,  
[[:SENSe]:SPURious:RANGe<x>:FREQuency:START



**[[: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.

**Syntax** [[:SENSe]:SPURious:RANGe<x>:LIMit:ABSolute:START <value>  
[[:SENSe]:SPURious:RANGe<x>:LIMit:ABSolute:START?

Where <x> = 1 to 20, representing Range A to T, respectively.

**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.

**Measurement Views** Spurious

**Examples** :SENSe:SPURious:RANGe1:LIMit:ABSolute:START -30  
sets the absolute start amplitude of the limits for Range A to -30 dBm.

**Related Commands** [[:SENSe]:POWER:UNITs,  
[[:SENSe]:SPURious:RANGe<x>:LIMit:ABSolute:STOP,  
[[:SENSe]:SPURious:RANGe<x>:LIMit:MASK, :UNIT:POWER

**[[: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.

**Syntax** [[:SENSe]:SPURious:RANGe<x>:LIMit:ABSolute:STOP <value>  
[[:SENSe]:SPURious:RANGe<x>:LIMit:ABSolute:STOP?

Where <x> = 1 to 20, representing Range A to T, respectively.

**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.

**Measurement Views** Spurious

**Examples** :SENSe:SPURious:RANGe1:LIMit:ABSolute:STOP -10  
sets the absolute stop amplitude of the limits for Range A to -10 dBm.

**Related Commands** [[:SENSe]:POWer:UNITs],  
[[:SENSe]:SPURious:RANGe<x>:LIMit:ABSolute:STOP],  
[[:SENSe]:SPURious:RANGe<x>:LIMit:MASK], [:UNIT:POWer]

**[[:SENSe]:SPURious:RANGe<x>:LIMit:MASK(?]**

Selects or queries the limit mask function mode for the specified range in the Spurious measurement.

**Syntax** [[:SENSe]:SPURious:RANGe<x>:LIMit:MASK  
{ ABS | REL | AND | OR | OFF }

[[:SENSe]:SPURious:RANGe<x>:LIMit:MASK?

Where <x> = 1 to 20, representing Range A to T, respectively.

**Arguments** Table 2-48 lists the arguments.

**Table 2-48: 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 :STOP commands.

To set the relative amplitude limits, use the [[:SENSe]:SPURious:RANGe<x>:LIMit:RELative:STARt and :STOP commands.

**Measurement Views** Spurious

**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.

**Related Commands** [[:SENSe]:SPURious:RANGe<x>:LIMit:ABSolute:STARt and :STOP,  
[:SENSe]:SPURious:RANGe<x>:LIMit:RELative:STARt and :STOP

**[[: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.

**Syntax** [[:SENSe]:SPURious:RANGe<x>:LIMit:RELative:START <value>  
[[:SENSe]:SPURious:RANGe<x>:LIMit:RELative:START?

Where <x> = 1 to 20, representing Range A to T, respectively.

**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.

**Measurement Views** Spurious

**Examples** :SENSe:SPURious:RANGe1:LIMit:RELative:START -30  
sets the relative start amplitude of the limits for Range A to -30 dB.

**Related Commands** [[:SENSe]:SPURious:RANGe<x>:LIMit:RELative:STOP,  
[[:SENSe]:SPURious:RANGe<x>:LIMit:MASK,  
[[:SENSe]:SPURious:REFerence

**[[: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.

**Syntax** [[:SENSe]:SPURious:RANGe<x>:LIMit:RELative:STOP <value>  
[[:SENSe]:SPURious:RANGe<x>:LIMit:RELative:STOP?

Where <x> = 1 to 20, representing Range A to T, respectively.

**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.

**Measurement Views** Spurious

**Examples** :SENSe:SPURious:RANGe1:LIMit:RELative:STOP -10  
sets the relative stop amplitude of the limits for Range A to -10 dB.

**Related Commands** [[:SENSe]:SPURious:RANGe<x>:LIMit:RELative:START,  
[[:SENSe]:SPURious:RANGe<x>:LIMit:MASK,  
[[:SENSe]:SPURious:REference

**[[:SENSe]:SPURious:RANGe<x>:STATe(?]**

Determines whether to enable or disable the frequency range in the Spurious measurement.

**Syntax** [[:SENSe]:SPURious:RANGe<x>:STATe { OFF | ON | 0 | 1 }

[[:SENSe]:SPURious:RANGe<x>:STATe?

Where <x> = 1 to 20, representing Range A to T, respectively.

**Arguments** OFF or 0 disables the frequency range.

ON or 1 enables the frequency range.

**Measurement Views** Spurious

**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.

**Syntax** [[:SENSe]:SPURious:RANGe<x>:THReshold <value>

[[:SENSe]:SPURious:RANGe<x>:THReshold?

Where <x> = 1 to 20, representing Range A to T, respectively.

**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.

**Measurement Views** Spurious

**Examples** :SENSe:SPURious:RANGe1:THReshold -25  
sets the threshold level to -25 dBm in Range A.

**Related Commands** [[:SENSe]:POWer:UNITs, :UNIT:POWer

**[[:SENSe]:SPURious:REFeRence(?)]**

Selects or queries the power reference in the Spurious measurement.

**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.

**Measurement Views**    Spurious

**Examples**        :SENSe:SPURious:REFeRence CARRier  
                  uses the carrier as the power reference.

**Related Commands**    [[:SENSe]:SPURious:REFeRence:MANual:POWer



## **[: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.

**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.

**Measurement Views**     Spurious

**Examples**        :SENSe:SPURious:REFerence:MANual:POWer -25  
                  sets the reference power level to -25 dBm.

**Related Commands**     [:SENSe]:POWer:UNITs, [:SENSe]:SPURious:REFerence, :UNIT:POWer

## **[[:SENSe]:TOVerview Subgroup**

The [[:SENSe]:TOVerview commands set up conditions for the Time Overview.

<b>Command Tree</b>	<b>Header</b>	<b>Parameter</b>
	[[:SENSe]	
	:TOVerview	
	:FREQuency	
	:CENTer	<numeric_value>
	:MAXTracepoints	ONEK   TĒNK   HUNDredk   NEVerdecimate

**[:SENSe]:TOVerview:FREQuency:CENTer(?)**

Sets or queries the center frequency in the time overview.

**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).

**Measurement Views**    Time overview

**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.

**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.

**Measurement Views**    Time overview

**Examples**        :SENSe:TOVerview:MAXTracepoints TENK  
                   sets the maximum trace points to 10 k.



# STATus Commands

The STATus commands control the SCPI-defined status reporting structures. In addition to those in IEEE 488.2, the analyzer has questionable and operation registers defined in SCPI. These registers conform to the IEEE 488.2 specification and each is comprised of a condition register, an event register, an enable register, and negative and positive transition filters.

---

**NOTE.** For details on the registers, refer to Status and Events beginning on page 3-1.

---

## Command Tree

Header	Parameter
:STATus	
:ACPower	
:EVENTs?	
:AVTime	
:EVENTs?	
:CCDF	
:EVENTs?	
:CONSte	
:EVENTs?	
:DPSA	
:EVENTs?	
:EVM	
:EVENTs?	
:FVTime	
:EVENTs?	
:IQVTime	
:EVENTs?	
:MCPower	
:EVENTs?	
:MERRor	
:EVENTs?	
:OBWidth	
:EVENTs?	

```

:OPERation
  :CONDition?
  :ENABle          <bit_value>
  [:EVENT]?
  :NTRansition    <bit_value>
  :PTRansition    <bit_value>
:PERRor
  :EVENTs?
:PHVTime
  :EVENTs?
:PNOise
  :EVENTs?
:PRESet
:PULSe
  :RESult
  :EVENTs?
  :STATistics
  :EVENTs?
  :TRACe
  :EVENTs?
:QUEStionable
  :CONDition?
  :ENABle          <bit_value>
  [:EVENT]?
  :NTRansition    <bit_value>
  :PTRansition    <bit_value>
  :CALibration
  :CONDition?
  :ENABle          <bit_value>
  [:EVENT]?
  :NTRansition    <bit_value>
  :PTRansition    <bit_value>
  :FREQuency
  :CONDition?
  :ENABle          <bit_value>
  [:EVENT]?
  :NTRansition    <bit_value>
  :PTRansition    <bit_value>
:SGRam
  :EVENTs?
:SPECTrum
  :EVENTs?
:SPURious
  :EVENTs?
:SQUality
  :EVENTs?

```

## :STATus:ACPower:EVENTs? (Query Only)

Returns the current events for the Channel power and ACPR measurement.

**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".

**Measurement Views** Channel power and ACPR

**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.

**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".

**Measurement Views** Amplitude versus Time

**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.

**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".

**Measurement Views** CCDF

**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.

**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".

**Measurement Views** Constellation

**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.

**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".

**Measurement Views** DPX spectrum

**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.

**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".

**Measurement Views** EVM versus Time

**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.

**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".

**Measurement Views** Frequency versus Time

**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.

**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".

**Measurement Views** RF I&Q versus Time

**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.

**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".

**Measurement Views** MCPR

**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.

**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".

**Measurement Views** Magnitude error versus Time

**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.

**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".

**Measurement Views** Occupied Bandwidth

**Examples** STATus:OBW: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).

**Syntax** :STATus:OPERation:CONDition?

**Arguments** None

**Returns** <NR1> is a decimal number showing the contents of the OCR.

**Measurement Views** All

**Examples** :STATus:OPERation:CONDition?  
might return 16, showing that the bits in the OCR have the binary value 000000000010000, 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.

**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.)

**Measurement Views** All

**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.

**Syntax** :STATus:OPERation[:EVENT]?

**Arguments** None

**Returns** <NR1> is a decimal number showing the contents of the OEVR.

**Measurement Views** All

**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).

**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.)

**Measurement Views** All

**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).

**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.)

**Measurement Views**   All

**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.

**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".

**Measurement Views** Phase error versus Time

**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.

**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".

**Measurement Views** Phase versus Time

**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)

### Option 11 Only

Returns the current events for the phase noise measurement.

**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".

**Measurement Views** Spurious

**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.

**Syntax** :STATus:PRESet

**Arguments** None

**Measurement Views** All

**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.

**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".

**Measurement Views**   Pulse table

**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.

**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".

**Measurement Views** Pulse statistics

**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.

**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".

**Measurement Views** Pulse trace

**Examples** STATus:PULSe:TRACe:EVENTs?  
 might return 2026,"Acq Sampling Params: manual control", indicating that  
 the sampling parameters are controlled manually.

## :STATus:QUEStionable:CONDition? (Query Only)

Returns the contents of the Questionable Condition Register (QCR).

**Syntax** :STATus:QUEStionable:CONDition?

**Arguments** None

**Returns** <NR1> is a decimal number showing the contents of the QCR.

**Measurement Views** All

**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.

**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.)

**Measurement Views** All

**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.

**Syntax** :STATus:QUESTionable[:EVENT]?

**Arguments** None

**Returns** <NR1> is a decimal number showing the contents of the QEVr.

**Measurement Views** All

**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:NTRansition(?)**

Sets or queries the negative transition filter value of the Questionable Transition Register (QTR).

**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.)

**Measurement Views** All

**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).

**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.)

**Measurement Views** All

**Examples** :STATus:QUEStionable:PTRansition 0  
sets the positive transition filter value to 0.

:STATus:QUEStionable:PTRansition?  
might return 0.

**:STATus:QUEStionable:CALibration:CONDition? (Query Only)**

Returns the contents of the questionable calibration condition register.

**Syntax** :STATus:QUEStionable:CALibration:CONDition?

**Arguments** None

**Returns** <NR1> is a decimal number showing the contents of the questionable calibration condition register.

**Measurement Views** All

**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.

**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.)

**Measurement Views**   All

**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.

**Syntax** :STATus:QUESTionable:CALibration[:EVENT]?

**Arguments** None

**Returns** <NR1> is a decimal number showing the contents of the questionable calibration event register.

**Measurement Views** All

**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.

**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.)

**Measurement Views** All

**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.

**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.)

**Measurement Views**   All

**Examples**     :STATUS:QUESTIONABLE:CALIBRATION:PTRANSITION 0  
 sets the positive transition filter value to 0.  
  
 :STATUS:QUESTIONABLE:CALIBRATION:PTRANSITION?  
 might return 0.

## **:STATUS:QUESTIONABLE:FREQUENCY:CONDITION? (Query Only)**

Returns the contents of the questionable frequency condition register.

**Syntax**     :STATUS:QUESTIONABLE:FREQUENCY:CONDITION?

**Arguments**   None

**Returns**     <NR1> is a decimal number showing the contents of the questionable frequency condition register.

**Measurement Views**   All

**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.

**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.)

**Measurement Views**   All

**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.

**Syntax** :STATus:QUESTionable:FREQuency[:EVENT]?

**Arguments** None

**Returns** <NR1> is a decimal number showing the contents of the questionable frequency event register.

**Measurement Views** All

**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.

**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.)

**Measurement Views**   All

**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.

**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.)

**Measurement Views**   All

**Examples**     :STATUS:QUESTIONABLE:FREQUENCY:PTRANSITION 0  
 sets the positive transition filter value to 0.  
  
 :STATUS:QUESTIONABLE:FREQUENCY:PTRANSITION?  
 might return 0.

## **:STATus:SGRam:EVENTs? (Query Only)**

Returns the current events for the spectrogram measurement.

**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".

**Measurement Views**   Spectrogram

**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.

**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".

**Measurement Views** Spectrum

**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.

**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".

**Measurement Views** Spurious

**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.

**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".

**Measurement Views** Signal quality

**Examples** STATus:SQUALity:EVENTs?  
might return 2026,"Acq Sampling Params: manual control", indicating that the sampling parameters are controlled manually.



# SYSTEM Commands

The SYSTem commands control system-related conditions.

## Command Tree

Header	Parameter
:SYSTem	
:COMMunicate	
:GPIB	
[:SELF]	
:ADDRess	<numeric_value>
:DATE	<year>,<month>,<day>
:ERRor	
:ALL?	
:CODE	
:ALL?	
[:NEXT]?	
:COUNT?	
[:NEXT]?	
:OPTions?	
:PRESet	
:TIME	<hour>,<minute>,<second>
:VERSion?	

## **:SYSTem:COMMunicate:GPIB[:SELF]:ADDRess(?)**

Sets or queries the GPIB address of the instrument.

**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.

**Measurement Views**   All

**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.

**Syntax**       :SYSTem:DATE <year>,<month>,<day>  
               :SYSTem:DATE?

**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.

**Measurement Views**   All

**Examples**       :SYSTem:DATE 2002,3,19  
                   sets the internal calendar to March 19, 2002.

**Related Commands**   :SYSTem:TIME

**: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 page 3-15.

**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".

**Measurement Views** All

**Examples** :SYSTem:ERRor:ALL?  
might return  
-130, "Suffix error; Unrecognized suffix, INPut:MLEVel -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 page 3-15.

**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.

**Measurement Views** All

**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 page 3-15.

**Syntax** :SYSTem:ERRor:CODE[:NEXT]?

**Arguments** None

**Returns** <code>::=<NR1> is the error/event code, ranging from -32768 to 32767.

**Measurement Views** All

**Examples** :SYSTem:ERRor:CODE:NEXT?  
might return -101 of the error code.

**Related Commands** :SYSTem:ERRor[:NEXT]?

**: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.

**Syntax** :SYSTem:ERRor:COUNT?

**Arguments** None

**Returns** <enum>::=<NR1> is the number of errors/events.  
If the queue is empty, the response is 0.

**Measurement Views** All

**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 page 3-15.

**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.

**Measurement Views**   All

**Examples**     :SYSTem:ERRor:NEXT?  
 might return  
 -130, "Suffix error; Unrecognized suffix, INPut:MLEVel -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?.

**Syntax** :SYSTem:OPTions?

**Arguments** None

**Returns** <option>::=<string> contains the comma-separated option numbers.

**Measurement Modes** All

**Examples** :SYSTem:OPTions?  
might return "01,02,20", indicating that Option 01, 02, and 20 are currently installed in the analyzer.

**Related Commands** \*OPT?, :INSTrument[:SElect]

## :SYSTem:PRESet (No Query Form)

Restores the analyzer to the defaults.

This command is equivalent to the PRESET key on the front panel.

**Syntax** :SYSTem:PRESet

**Arguments** None

**Measurement Modes** All

**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.

**Syntax**     :SYSTem:TIME <hour>,<minute>,<second>  
               :SYSTem:TIME?

**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.

**Measurement Modes**   All

**Examples**       :SYSTem:TIME 10,15,30  
                   sets the time to 10:15:30.

**Related Commands**   :SYSTem:DATE

## **:SYSTem:VERSion? (Query Only)**

Returns the SCPI version number for which the analyzer complies.

**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.

**Measurement Modes**   All

**Examples**       :SYSTem:VERSion?  
                   might return 1999.0 for the SCPI version.

# TRACe Commands

The TRACe commands select trace type and control trace arithmetic. These commands are divided by measurement views into the subgroups listed in Table 2-49.

**Table 2-49: TRACe command subgroups**

Command header	Measurement view	Refer to:
:TRACe<x>:AVTime	Amplitude versus Time	page 2-896
:TRACe<x>:CCDF	CCDF	page 2-906
:TRACe:CONSte	Constellation (Option 21 only)	page 2-911
:TRACe<x>:DPsA	DPX (Digital Phosphor) spectrum	page 2-912
:TRACe:FVTime	Frequency versus Time	page 2-922
:TRACe:IQVTime	RF I&Q versus Time	page 2-926
:TRACe:OBWidth	Occupied Bandwidth (OBW)	page 2-933
:TRACe:PHVTime	Phase versus Time	page 2-934
:TRACe<x>:PNOise	Phase noise (Option 11 only)	page 2-938
:TRACe:SGRam	Spectrogram	page 2-942
:TRACe<x>:SPEctrum	Spectrum	page 2-946
:TRACe:SPURious	Spurious	page 2-958

## :TRACe<x>:AVTime Subgroup

The :TRACe:AVTime commands control traces in the Amplitude versus Time measurement.

Command Tree	Header	Parameter
	:TRACe<x>	
	:AVTime	<boolean>
	:AVERage	
	:COUNt	<numeric_value>
	:RESet	
	:COUNt	<numeric_value>
	:ENABle	<boolean>
	:RESet	
	:FREeze	<boolean>
	:FUNction	NORMal   AVERage   MAXHold   MINHold
	:LEFToperand	TRACE1   TRACE2   TRACE3
	:RIGHToperand	TRACE1   TRACE2   TRACE3
	:SElect	

Where <x> = 1 to 4. Each trace is defined as follows:

TRACe1: Trace 1  
 TRACe2: Trace 2  
 TRACe3: Trace 3  
 TRACe4: Math trace

---

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

---

**:TRACe<x>:AVTime(?)**

Determines whether or not to show the specified trace in the Amplitude versus Time view.

**Syntax** :TRACe<x>:AVTime { OFF | ON | 0 | 1 }

:TRACe<x>:AVTime?

Where <x> = 1 to 4; All traces are valid.

**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.

**Measurement Views** Amplitude versus Time

**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.

**Syntax** :TRACe<x>:AVTime:AVERAge:COUNT <number>

:TRACe<x>:AVTime:AVERAge:COUNT?

Where <x> = 1 to 3; Trace 4 (math trace) is invalid.

**Arguments** <number>::=<NR1> specifies the number of traces to combine for averaging.  
Range: 1 to 10000.

**Measurement Views** Amplitude versus Time

**Examples** :TRACe1:AVTime:AVERAge:COUNT 64  
sets the average count to 64 for Trace 1.

**Related Commands** :TRACe<x>:AVTime:FUNCTION

**: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.

**Syntax** :TRACe<x>:AVTime:AVERage:RESet

Where <x> = 1 to 3; Trace 4 (math trace) is invalid.

**Arguments** None

**Measurement Views** Amplitude versus Time

**Examples** :TRACe1:AVTime:AVERage:RESet  
restarts acquisition and display of waveforms for Trace 1.

**Related Commands** :TRACe<x>:AVTime:FUNctIon

**: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.

**Syntax** :TRACe<x>:AVTime:COUNT <number>

:TRACe<x>:AVTime:COUNT?

Where <x> = 1 to 3; Trace 4 (math trace) is invalid.

**Arguments** <number>: :=<NR1> specifies the count for Max/Min Hold.  
Range: 1 to 10000.

**Measurement Views** Amplitude versus Time

**Examples** :TRACe1:AVTime:COUNT 32  
sets the count to 32 for Trace 1.

**Related Commands** :INITiate:CONTInuous, :TRACe<x>:AVTime:FUNCTION

**: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.

**Syntax** :TRACe<x>:AVTime:COUNT:ENABLE { OFF | ON | 0 | 1 }  
:TRACe<x>:AVTime:COUNT:ENABLE?

Where <x> = 1 to 3; Trace 4 (math trace) is invalid.

**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.

**Measurement Views** Amplitude versus Time

**Examples** :TRACe1:AVTime:COUNT:ENABLE ON  
enables the Max/Min Hold count for Trace 1.

**Related Commands** :TRACe<x>:AVTime:FUNCTION



**: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.

**Syntax** :TRACe<x>:AVTime:COUNT:RESet

Where <x> = 1 to 3; Trace 4 (math trace) is invalid.

**Arguments** None

**Measurement Views** Amplitude versus Time

**Examples** :TRACe1:AVTime:COUNT:RESet  
clears the Max/Min Hold data and counter, and restarts the process for Trace 1.

**Related Commands** :TRACe<x>:AVTime:FUNCTION

**:TRACe<x>:AVTime:FREeze(?)**

Determines whether or not to freeze the display of the specified trace in the Amplitude versus Time view.

**Syntax** :TRACe<x>:AVTime:FREeze { OFF | ON | 0 | 1 }

:TRACe<x>:AVTime:FREeze?

Where <x> = 1 to 4; All traces are valid.

**Arguments** OFF or 0 updates the display of the specified trace normally.  
ON or 1 stops updating the display of the specified trace.

**Measurement Views** Amplitude versus Time

**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.

**Syntax**     :TRACe<x>:AVTime:FUNction  
                  { NORMal | AVERage | MAXHold | MINHold }  
  
:TRACe<x>:AVTime:FUNction?

Where <x> = 1 to 3; Trace 4 (math trace) is invalid.

**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.

**Measurement Views**   Amplitude versus Time

**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.

**Syntax** :TRACe<x>:AVTime:LEFToperand { TRACE1 | TRACE2 | TRACE3 }

:TRACe<x>:AVTime:LEFToperand?

Where <x> = 4; Only Trace 4 (math trace) is valid.

**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.

**Measurement Views** Amplitude versus Time

**Examples** :TRACe4:AVTime:LEFToperand TRACE2  
selects Trace 2 as the left operand for the math trace.

**Related Commands** :TRACe<x>:AVTime:RIGHToperand

**:TRACe<x>:AVTime:RIGHtooperand(?)**

Selects or queries the right operand for the math trace (Trace 4) in the Amplitude versus Time view.

**Syntax** :TRACe<x>:AVTime:RIGHtooperand { TRACE1 | TRACE2 | TRACE3 }

:TRACe<x>:AVTime:RIGHtooperand?

Where <x> = 4; Only Trace 4 (math trace) is valid.

**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.

**Measurement Views** Amplitude versus Time

**Examples** :TRACe4:AVTime:RIGHtooperand TRACE1  
selects Trace 1 as the right operand for the math trace.

**Related Commands** :TRACe<x>:AVTime:LEFTooperand

**: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.

**Syntax** :TRACe<x>:AVTime:SElect  
:TRACe<x>:AVTime:SElect?

Where <x> = 1 to 4; All traces are valid.

**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.

**Measurement Views** Amplitude versus Time

**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 Subgroup

The :TRACe:CCDF commands control traces in the CCDF measurement.

Command Tree	Header	Parameter
	:TRACe<x>	
	:CCDF	
	:FREeze	<boolean>
	:SElect	
	:SHOW	<boolean>
	:X	<numeric_value>
	:Y?	

Where <x> = 1 to 3. Each trace is defined as follows:

TRACe1: Trace 1

TRACe2: Trace 2

TRACe3: Gaussian reference curve

---

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

---

## :TRACe<x>:CCDF:FREeze(?)

Determines whether or not to freeze the display of the specified trace (Trace 1 or 2) in the CCDF view.

**Syntax** :TRACe<x>:CCDF:FREeze { OFF | ON | 0 | 1 }  
:TRACe<x>:CCDF:FREeze?

Where <x> = 1 or 2; Trace 3 (Gaussian curve) is invalid.

**Arguments** OFF or 0 updates the display of the specified trace normally.  
ON or 1 stops updating the display of the specified trace.

**Measurement Views** CCDF

**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.

**Syntax** :TRACe<x>:CCDF:SElect  
:TRACe<x>:CCDF:SElect?

Where <x> = 1 to 3; All traces are valid.

**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.

**Measurement Views** CCDF

**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.

**Syntax** :TRACe<x>:CCDF:SHOW { OFF | ON | 0 | 1 }

:TRACe<x>:CCDF:SHOW?

Where <x> = 1 to 3; All traces are valid.

**Arguments** OFF or 0 hides the specified trace.

ON or 1 shows the specified trace.

**Measurement Views** CCDF

**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.

---

**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.

---

**Syntax** :TRACe<x>:CCDF:X <value>

:TRACe<x>:CCDF:X?

Where <x> = 1 to 3; All traces are valid.

**Arguments** <value>::=<Nrf> specifies the horizontal position of the measurement pointer. Range: 0 to 20 dB.

**Measurement Views** CCDF

**Examples** :TRACe1:CCDF:X 5  
puts the measurement pointer at 5 dB on Trace 1.

**Related Commands** :TRACe<x>:CCDF:SElect, :TRACe<x>:CCDF:SHOW,  
:TRACe<x>:CCDF:Y?

**: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.

---

**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.

---

**Syntax** :TRACe<x>:CCDF:Y?

Where <x> = 1 to 3; All traces are valid.

**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.

**Measurement Views** CCDF

**Examples** :TRACe1:CCDF:Y?  
might return 14.72, indicating the CCDF is 14.72% at the measurement pointer on Trace 1.

**Related Commands** :TRACe<x>:CCDF:SElect, :TRACe<x>:CCDF:SHOW, :TRACe<x>:CCDF:X

## :TRACe:CONStE Subgroup

The :TRACe:CONStE commands control traces in the Constellation measurement.

Command Tree	Header	Parameter
	:TRACe	
	:CONStE	
	:MODE	VECTors   SYMBols

## :TRACe:CONStE:MODE(?)

Selects or queries how to display the constellation trace.

**Syntax** :TRACe:CONStE:MODE { VECTors | SYMBols }  
:TRACe:CONStE:MODE?

**Arguments** VECTors connects adjacent symbol points with the signal locus.  
SYMBols displays individual symbol points.

**Measurement Views** Constellation

**Examples** :TRACe:CONStE:MODE VECTors  
shows the constellation connecting adjacent symbol points with the signal locus.

## :TRACe<x>:DPSA Subgroup

The :TRACe<x>:DPSA commands control traces in the DPX (Digital Phosphor) Spectrum measurement.

Command Tree	Header	Parameter
	:TRACe<x>	
	:DPSA	<boolean>
	:AVERage	
	:COUnT	<numeric_value>
	:COLor	
	:INTensity	<numeric_value>
	:DOT	
	:PERsistent	<boolean>
	:TYPE	VARIable   INFinite
	:VARIable	<numeric_value>
	:FREeze	<boolean>
	:FUNCTion	NORMal   HOLD   AVERage
	:LEFToperand	TRACE1   TRACE2   TRACE3
	:RIGHTooperand	TRACE1   TRACE2   TRACE3
	:SElect	

Where <x> = 1 to 5. Each trace is defined as follows:

TRACe1: +Peak trace  
TRACe2: -Peak trace  
TRACe3: Average trace  
TRACe4: Math trace  
TRACe5: Bitmap trace

---

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

---

**:TRACe<x>:DPSA(?)**

Determines whether or not to show the specified trace in the DPX spectrum view.

**Syntax** :TRACe<x>:DPSA { OFF | ON | 0 | 1 }

:TRACe<x>:DPSA?

Where <x> = 1 to 5; All traces are valid.

**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.

**Measurement Views** DPX spectrum

**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.

**Syntax** :TRACe<x>:DPSA:AVERAge:COUNT <number>

:TRACe<x>:DPSA:AVERAge:COUNT?

Where <x> = 3; Only Trace 3 (average trace) is valid.

**Arguments** <number>::=<NR1> specifies the number of traces to combine for averaging.  
Range: 1 to 10000.

**Measurement Views** DPX spectrum

**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.

**Syntax** :TRACe<x>:DPSA:COLor:INTensity <value>

:TRACe<x>:DPSA:COLor:INTensity?

Where <x> = 1 to 5; All traces are valid.

**Arguments** <value>::=<NRF> specifies color intensity. Range: 1 to 100%.

**Measurement Views** DPX spectrum

**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.

**Syntax** :TRACe<x>:DPSA:DOT:PERSistent { OFF | ON | 0 | 1 }

:TRACe<x>:DPSA:DOT:PERSistent?

Where <x> = 5; Only Trace 5 (bitmap trace) is valid.

**Arguments** OFF or 0 disables the dot persistence.

ON or 1 enables the dot persistence.

**Measurement Views** DPX spectrum

**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.

**Syntax** :TRACe<x>:DPSA:DOT:PERsistent:TYPE { VARIable | INFinite }

:TRACe<x>:DPSA:DOT:PERsistent:TYPE?

Where <x> = 5; Only Trace 5 (bitmap trace) is valid.

**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.

**Measurement Views** DPX spectrum

**Examples** :TRACe5:DPSA:DOT:PERsistent:TYPE VARIable  
selects the variable persistence display.

**Related Commands** :TRACe<x>:DPSA:DOT:PERsistent:VARIable

**: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.

**Syntax** :TRACe<x>:DPSA:DOT:PERSistent:VARiable <number>

:TRACe<x>:DPSA:DOT:PERSistent:VARiable?

Where <x> = 5; Only Trace 5 (bitmap trace) is valid.

**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).

**Measurement Views** DPX spectrum

**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.

**Related Commands** :TRACe<x>:DPSA:DOT:PERSistent:TYPE



**:TRACe<x>:DPSA:FREEze(?)**

Determines whether or not to freeze the display of the specified trace in the DPX spectrum view.

**Syntax** :TRACe<x>:DPSA:FREEze { OFF | ON | 0 | 1 }

:TRACe<x>:DPSA:FREEze?

Where <x> = 1 to 5; All traces are valid.

**Arguments** OFF or 0 updates the display of the specified trace normally.

ON or 1 stops updating the display of the specified trace.

**Measurement Views** DPX spectrum

**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.

**Syntax** :TRACe<x>:DPSA:FUNction { NORMAl | HOLD | AVERAge }

:TRACe<x>:DPSA:FUNction?

Where <x> = 1 to 3; Trace 4 (math trace) and Trace 5 (bitmap trace) are invalid.

**Arguments** The table below shows the trace function and display.

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

For the average trace, use the :TRACe<x>:DPSA:AVERAge:COUNt command to set the average count.

**Measurement Views** DPX spectrum

**Examples** :TRACe1:DPSA:FUNction HOLD  
selects the max hold waveform for Trace 1.

**Related Commands** :TRACe<x>:DPSA:AVERAge:COUNt

**:TRACe<x>:DPSA:LEFToperand(?)**

Selects or queries the left operand for the math trace (Trace 4) in the DPX spectrum view.

**Syntax** :TRACe<x>:DPSA:LEFToperand { TRACE1 | TRACE2 | TRACE3 }  
:TRACe<x>:DPSA:LEFToperand?

Where <x> = 4; Only Trace 4 (math trace) is valid.

**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.

**Measurement Views** DPX spectrum

**Examples** :TRACe4:DPSA:LEFToperand TRACE2  
selects Trace 2 as the left operand for the math trace.

**Related Commands** :TRACe<x>:DPSA:RIGHToperand

**:TRACe<x>:DPSA:RIGHTooperand(?)**

Selects or queries the right operand for the math trace (Trace 4) in the DPX spectrum view.

**Syntax** :TRACe<x>:DPSA:RIGHTooperand { TRACE1 | TRACE2 | TRACE3 }  
:TRACe<x>:DPSA:RIGHTooperand?

Where <x> = 4; Only Trace 4 (math trace) is valid.

**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.

**Measurement Views** DPX spectrum

**Examples** :TRACe4:DPSA:RIGHTooperand TRACE1  
selects Trace 1 as the right operand for the math trace.

**Related Commands** :TRACe<x>:DPSA:LEFTooperand

**:TRACe<x>:DPSA:SElect(?)**

Selects or queries the trace to display the readout at the upper left of the DPX spectrum view.

**Syntax** :TRACe<x>:DPSA:SElect

:TRACe<x>:DPSA:SElect?

Where <x> = 1 to 5; All traces are valid.

**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.

**Measurement Views** DPX spectrum

**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:FVTime Subgroup

The :TRACe:FVTime commands control traces in the Frequency versus Time measurement.

Command Tree	Header	Parameter
	:TRACe	
	:FVTime	<boolean>
	:AVERage	
	:COUnT	<numeric_value>
	:COUnT	<numeric_value>
	:ENABle	<boolean>
	:RESet	
	:FREeze	<boolean>
	:FUNCTion	NORMal   AVERage   MAXHold   MINHold

## :TRACe:FVTime(?)

Determines whether or not to show the trace in the Frequency versus Time view.

**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.

**Measurement Views**   Frequency versus Time

**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.

**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.

**Measurement Views** Frequency versus Time

**Examples** :TRACe:FVTime:AVERAge:COUNT 64  
sets the average count to 64.

**Related Commands** :TRACe:FVTime:FUNCTion

**: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<x>:FVTime:FUNCTion is set to MAXHold or MINHold and :INITiate:CONTInuous is set to OFF.

**Syntax** :TRACe:FVTime:COUNT <number>

:TRACe:FVTime:COUNT?

**Arguments** <number>::=<NR1> specifies the count for Max/Min Hold.  
Range: 1 to 10000.

**Measurement Views** Frequency versus Time

**Examples** :TRACe:FVTime:COUNT 32  
sets the count to 32 for the Max/Min Hold trace.

**Related Commands** :INITiate:CONTInuous, :TRACe:FVTime:FUNCTion

**: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.

**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.

**Measurement Views** Frequency versus Time

**Examples** :TRACe:FVTime:COUNT:ENABLE ON  
enables the Max/Min Hold count.

**Related Commands** :TRACe:FVTime:FUNCTION

**: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.

**Syntax** :TRACe:FVTime:COUNT:RESet

**Arguments** None

**Measurement Views** Frequency versus Time

**Examples** :TRACe:FVTime:COUNT:RESet  
clears the Max/Min Hold data and counter, and restarts the process.

**Related Commands** :TRACe:FVTime:FUNCTION



## **:TRACe:FVTime:FREeze(?)**

Determines whether or not to freeze the display of the trace in the Frequency versus Time measurement.

**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.

**Measurement Views** Frequency versus Time

**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.

**Syntax** :TRACe:FVTime:FUNCTion { NORMa1 | AVERAge | MAXHold | MINHold }  
:TRACe:FVTime:FUNCTion?

**Arguments** NORMa1 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.

**Measurement Views** Frequency versus Time

**Examples** :SENSe:FVTime:FUNCTion MAXHold  
displays the Max Hold trace in the Frequency versus Time measurement.

## :TRACe:IQVTime Subgroup

The :TRACe:IQVTime commands control traces in the RF I&Q versus Time measurement.

Command Tree	Header	Parameter
	:TRACe	
	:IQVTime	
	:AVERage	
	:COUnT	<numeric_value>
	:COUnT	<numeric_value>
	:ENABle	<boolean>
	:RESEt	
	:ENABle	
	:I	<boolean>
	:Q	<boolean>
	:FREEze	<boolean>
	:FUNCTion	NORMal   AVERage   MAXHold   MINHold
	:SELEct	
	:I	<boolean>
	:Q	<boolean>

**: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.

**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.

**Measurement Views** RF I&Q versus Time

**Examples** :TRACe:IQVTime:AVERage:COUNT 64  
sets the average count to 64.

**Related Commands** :TRACe:IQVTime:FUNCTion

**: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<x>:IQVTime:FUNCTion is set to MAXHold or MINHold and :INITiate:CONTinuous is set to OFF.

**Syntax** :TRACe:IQVTime:COUNT <number>  
:TRACe:IQVTime:COUNT?

**Arguments** <number>::=<NR1> specifies the count for Max/Min Hold.  
Range: 1 to 10000.

**Measurement Views** RF I&Q versus Time

**Examples** :TRACe:IQVTime:COUNT 32  
sets the count to 32 for the Max/Min Hold trace.

**Related Commands** :INITiate:CONTinuous, :TRACe:IQVTime:FUNCTion

**: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.

**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.

**Measurement Views** RF I&Q versus Time

**Examples** :TRACe:IQVTime:COUNT:ENABle ON  
enables the Max/Min Hold count.

**Related Commands** :TRACe:IQVTime:FUNCTION

**: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.

**Syntax** :TRACe:IQVTime:COUNT:RESet

**Arguments** None

**Measurement Views** RF I&Q versus Time

**Examples** :TRACe:IQVTime:COUNT:RESet  
clears the Max/Min Hold data and counter, and restarts the process.

**Related Commands** :TRACe:IQVTime:FUNCTION

**:TRACe:IQVTime:ENABle:I(?)**

Determines whether to show or hide the I trace in the RF I&Q versus Time measurement.

**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.

**Measurement Views** RF I&Q versus Time

**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.

**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.

**Measurement Views** RF I&Q versus Time

**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.

**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.

**Measurement Views** RF I&Q versus Time

**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.

**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.

**Measurement Views** RF I&Q versus Time

**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.

**Syntax** :TRACe:IQVTime:SElect:I { OFF | ON | 0 | 1 }  
:TRACe:IQVTime:SElect:I?

**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.

**Measurement Views** RF I&Q versus Time

**Examples** :TRACe:IQVTime:SElect:I ON  
selects the I trace in the RF I&Q versus Time measurement.

**Related Commands** :TRACe:IQVTime:SElect:Q

**:TRACe:IQVTime:SElect:Q(?)**

Determines whether or not to select the Q trace to obtain the maximum and minimum measurement results.

**Syntax** :TRACe:IQVTime:SElect:Q { OFF | ON | 0 | 1 }

:TRACe:IQVTime:SElect:Q?

**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.

**Measurement Views** RF I&Q versus Time

**Examples** :TRACe:IQVTime:SElect:Q ON  
selects the Q trace in the RF I&Q versus Time measurement.

**Related Commands** :TRACe:IQVTime:SElect:I



## :TRACe:OBWidth Subgroup

The :TRACe:OBW commands control traces in the Occupied Bandwidth (OBW) measurement.

Command Tree	Header	Parameter
	:TRACe	
	:OBWidth	
	:MAXHold	<boolean>

### :TRACe:OBWidth:MAXHold(?)

Determines whether or not to perform a Max Hold on the spectrum data for the Occupied Bandwidth trace.

**Syntax** :TRACe:OBWidth:MAXHold { OFF | ON | 0 | 1 }  
:TRACe:OBWidth: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.

**Measurement Views** Occupied Bandwidth

**Examples** :TRACe:OBWidth:MAXHold ON  
performs a Max Hold on the spectrum data for the Occupied Bandwidth trace.

## :TRACe:PHVTime Subgroup

The :TRACe:PHVTime commands control traces in the Phase versus Time measurement.

Command Tree	Header	Parameter
	:TRACe	
	:PHVTime	<boolean>
	:AVERage	
	:COUnT	<numeric_value>
	:COUnT	<numeric_value>
	:ENABle	<boolean>
	:RESet	
	:FREeze	<boolean>
	:FUNCTion	NORMal   AVERage   MAXHold   MINHold

## :TRACe:PHVTime(?)

Determines whether or not to show the trace in the Phase versus Time view.

**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.

**Measurement Views**   Phase versus Time

**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.

**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.

**Measurement Views** Phase versus Time

**Examples** :TRACe:PHVTime:AVERAge:COUNT 64  
sets the average count to 64.

**Related Commands** :TRACe:PHVTime:FUNCTion

**: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<x>:PHVTime:FUNCTion is set to MAXHold or MINHold and :INITiate:CONTInuous is set to OFF.

**Syntax** :TRACe:PHVTime:COUNT <number>

:TRACe:PHVTime:COUNT?

**Arguments** <number>::=<NR1> specifies the count for Max/Min Hold. Range: 1 to 10000.

**Measurement Views** Phase versus Time

**Examples** :TRACe:PHVTime:COUNT 32  
sets the count to 32 for the Max/Min Hold trace.

**Related Commands** :INITiate:CONTInuous, :TRACe:PHVTime:FUNCTion

**: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.

**Syntax** :TRACe:PHVTime:COUNT:ENABLE { OFF | ON | 0 | 1 }  
:TRACe:PHVTime: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.

**Measurement Views** Phase versus Time

**Examples** :TRACe:PHVTime:COUNT:ENABLE ON  
enables the Max/Min Hold count.

**Related Commands** :TRACe:PHVTime:FUNCTION

**: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.

**Syntax** :TRACe:PHVTime:COUNT:RESet

**Arguments** None

**Measurement Views** Phase versus Time

**Examples** :TRACe:PHVTime:COUNT:RESet  
clears the Max/Min Hold data and counter, and restarts the process.

**Related Commands** :TRACe:PHVTime:FUNCTION

**:TRACe:PHVTime:FREeze(?)**

Determines whether to freeze the trace display in the Phase versus Time measurement.

**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.

**Measurement Views** Phase versus Time

**Examples** :TRACe:PHVTime:FREeze ON  
stops updating trace display.

**:TRACe:PHVTime:FUNction(?)**

Selects or queries the trace function in the Phase versus Time measurement.

**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.

**Measurement Views** Phase versus Time

**Examples** :TRACe:PHVTime:FUNction MAXHold  
displays the Max Hold trace in the Phase versus Time measurement.

## :TRACe<x>:PNOise Subgroup (Option 11 Only)

The :TRACe<x>:PNOise commands control traces in the Phase Noise measurement.

Command Tree	Header	Parameter
	:TRACe<x>	
	:PNOise	
	:DETEction	AVERage   POSNegative
	:FREeze	<boolean>
	:SElect	
	:SHOW	<boolean>
	:SMOothing	
	:COUnT	<numeric_value>
	:ENABle	<boolean>
	:RESet	

Where <x> = 1 and 2, representing Trace 1 and Trace 2, respectively.

## :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.

**Syntax** :TRACe<x>:PNOise:DETEction { AVERage | POSNegative }  
:TRACe<x>:PNOise:DETEction?

Where <x> = 1 and 2.

**Arguments** AVERage displays the average data value for each pixel.  
POSNegative displays the maximum and minimum data values for each pixel.

**Measurement Views** Phase noise

**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.

**Syntax** :TRACe<x>:PNOise:FREeze { OFF | ON | 0 | 1 }

:TRACe<x>:PNOise:FREeze?

Where <x> = 1 and 2.

**Arguments** OFF or 0 updates the trace display normally.

ON or 1 stops updating the trace display.

**Measurement Views** Phase noise

**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.

**Syntax** :TRACe<x>:PNOise:SElect

:TRACe<x>:PNOise:SElect?

Where <x> = 1 and 2.

**Arguments** None

**Returns** 0 (not selected) or 1 (selected).

**Measurement Views** Phase noise

**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.

**Syntax** :TRACe<x>:PNOise:SHOW { OFF | ON | 0 | 1 }

:TRACe<x>:PNOise:SHOW?

Where <x> = 1 and 2.

**Arguments** OFF or 0 hides the specified trace.

ON or 1 shows the specified trace.

**Measurement Views** Phase noise

**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.

**Syntax** :TRACe<x>:PNOise:SMOothing:COUNT <number>

:TRACe<x>:PNOise:SMOothing:COUNT?

Where <x> = 1 and 2.

**Arguments** <number>::=<NR1> specifies the number of data points to take the moving average for smoothing. Range: 3 to 50.

**Measurement Views** Phase noise

**Examples** :TRACe1:PNOise:SMOothing:COUNT 16  
sets the smoothing count to 16 for Trace 1.

**Related Commands** :TRACe<x>:PNOise:SMOothing:ENABLE



**:TRACe<x>:PNOise:SMOothing:ENABLE(?)**

Determines whether to enable or disable smoothing the specified trace in the phase noise view.

**Syntax** :TRACe<x>:PNOise:SMOothing:ENABLE { OFF | ON | 0 | 1 }

:TRACe<x>:PNOise:SMOothing:ENABLE?

Where <x> = 1 and 2.

**Arguments** OFF or 0 disables smoothing.

ON or 1 enables smoothing.

**Measurement Views** Phase noise

**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.

**Syntax** :TRACe<x>:PNOise:SMOothing:RESet

Where <x> = 1 and 2.

**Arguments** None

**Measurement Views** Phase noise

**Examples** :TRACe1:PNOise:SMOothing:RESet  
restarts the smoothing process for Trace 1.

**Related Commands** :TRACe<x>:PNOise:SMOothing:ENABLE

## :TRACe:SGRam Subgroup

The :TRACe:SGRam commands control traces in the spectrogram.

Command Tree	Header	Parameter
	:TRACe	
	:SGRam	
	:DETEction	AVERage   POSitive   NEGative   CAVerage   CPEak   QUASipeak
	:FREeze	<boolean>
	:FUNCTion	NONE   AVERage   MAXHold   MINHold
	:TIME	<numeric_value>
	:SElect	
	:LINE	<numeric_value>

## :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.

**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.

**Measurement Views** Spectrogram

**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.

**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.

**Measurement Views** 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.

**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.

**Measurement Views** Spectrogram

**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.

**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.

**Measurement Views** Spectrogram

**Examples** :TRACe1:SGRam:FUNction:TIME 1.5  
sets the time length to 1.5 minutes to combine traces for averaging.

**Related Commands** :TRACe:SGRam:FUNction

**:TRACe:SGRam:SElect:LINE(?)**

Selects or queries the number of line to send to the spectrum display.

**Syntax** :TRACe:SGRam:SElect:LINE <number>  
:TRACe:SGRam:SElect:LINE?

**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.

**Measurement Views** Spectrogram

**Examples** :TRACe:SGRam:SElect:LINE 75  
selects Line #75 in the spectrogram to send to the spectrum display.

**Related Commands** :TRACe<x>:SPEctrum

## :TRACe<x>:SPECTrum Subgroup

The :TRACe<x>:SPECTrum commands control traces in the Spectrum measurement.

Command Tree	Header	Parameter
	:TRACe<x>	
	:SPECTrum	<boolean>
	:AVERage	
	:COUNt	<numeric_value>
	:RESet	
	:COUNt	<numeric_value>
	:ENABle	<boolean>
	:RESet	
	:DETEction	AVERage   POSitive   NEGative   CAVERage   CPEak   QUASipeak
	:FREeze	<boolean>
	:FUNCTion	NONE   AVERage   MAXHold   MINHold
	:LEFTooperand	TRACE1   TRACE2   TRACE3
	:RIGHTooperand	TRACE1   TRACE2   TRACE3
	:SELEct	

Where <x> = 1 to 5. Each trace is defined as follows:

TRACe1: Trace 1  
 TRACe2: Trace 2  
 TRACe3: Trace 3  
 TRACe4: Math trace  
 TRACe5: Spectrogram

---

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

---

**:TRACe<x>:SPECTrum(?)**

Determines whether to show or hide the specified trace in the Spectrum view.

**Syntax** :TRACe<x>:SPECTrum { OFF | ON | 0 | 1 }  
:TRACe<x>:SPECTrum?

Where <x> = 1 to 5; All traces are valid.

**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.

**Measurement Views** Spectrum

**Examples** :TRACe1:SPECTrum ON  
shows Trace 1 in the Spectrum Analyzer view.

**Related Commands** :TRACe:SGRam:SElect:LINE

**: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.

**Syntax** :TRACe<x>:SPECTrum:AVERage:COUNT <number>

:TRACe<x>:SPECTrum:AVERage:COUNT?

Where <x> = 1 to 3; Trace 4 (math trace) and Trace 5 (spectrogram) are invalid.

**Arguments** <number>::=<NR1> specifies the number of traces to combine for averaging.  
Range: 1 to 10000.

**Measurement Views** Spectrum

**Examples** :TRACe1:SPECTrum:AVERage:COUNT 64  
sets the average count to 64 for Trace 1.

**Related Commands** :TRACe<x>:SPECTrum:FUNction

**: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.

**Syntax** :TRACe<x>:SPECTrum:AVERage:RESet

Where <x> = 1 to 3; Trace 4 (math trace) and Trace 5 (spectrogram) are invalid.

**Arguments** None

**Measurement Views** Spectrum

**Examples** :TRACe1:SPECTrum:AVERage:RESet  
clears average data and counter, and restarts the average process for Trace 1.

**Related Commands** :TRACe<x>:SPECTrum:FUNction



**: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.

**Syntax** :TRACe<x>:SPECTrum:COUNT <number>

:TRACe<x>:SPECTrum:COUNT?

Where <x> = 1 to 3; Trace 4 (math trace) and Trace 5 (spectrogram) are invalid.

**Arguments** <number>: :=<NR1> specifies the count for Max/Min Hold.  
Range: 1 to 10000.

**Measurement Views** Spectrum

**Examples** :TRACe1:SPECTrum:COUNT 32  
sets the count to 32 for Trace 1.

**Related Commands** :INITiate:CONTInuous, :TRACe<x>:SPECTrum:FUNCTION

**: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.

**Syntax** :TRACe<x>:SPECTrum:COUNT:ENABLE { OFF | ON | 0 | 1 }  
:TRACe<x>:SPECTrum:COUNT:ENABLE?

Where <x> = 1 to 3; Trace 4 (math trace) and Trace 5 (spectrogram) are invalid.

**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.

**Measurement Views** Spectrum

**Examples** :TRACe1:SPECTrum:COUNT:ENABLE ON  
enables the Max/Min Hold count for Trace 1.

**Related Commands** :TRACe<x>:SPECTrum:FUNCTION

**: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.

**Syntax** :TRACe<x>:SPECTrum:COUNT:RESet

Where <x> = 1 to 3; Trace 4 (math trace) and Trace 5 (spectrogram) are invalid.

**Arguments** None

**Measurement Views** Spectrum

**Examples** :TRACe1:SPECTrum:COUNT:RESet  
clears the Max/Min Hold data and counter, and restarts the process for Trace 1.

**Related Commands** :TRACe<x>:SPECTrum:FUNCTION

**: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.

**Syntax**     :TRACe<x>:SPECTrum:DETection { AVERage | POSitive | NEGative  
                  | CAVERage | CPEak | QUASipeak }

:TRACe<x>:SPECTrum:DETection?

Where <x> = 1 to 3; Trace 4 (math trace) and Trace 5 (spectrogram) are invalid.

**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.

**Measurement Views**   Spectrum

**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.

**Syntax** :TRACe<x>:SPECTrum:FREEze { OFF | ON | 0 | 1 }

:TRACe<x>:SPECTrum:FREEze?

Where <x> = 1 to 5; All traces are valid.

**Arguments** OFF or 0 updates the display of the specified trace normally.

ON or 1 stops updating the display of the specified trace.

**Measurement Views** Spectrum

**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.

**Syntax**     :TRACe<x>:SPECTrum:FUNCTion  
              { NONE | AVERAge | MAXHold | MINHold }

:TRACe<x>:SPECTrum:FUNCTion?

Where <x> = 1 to 3; Trace 4 (math trace) and Trace 5 (spectrogram) are invalid.

**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.

**Measurement Views**   Spectrum

**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.

**Syntax** :TRACe<x>:SPECTrum:LEFToperand { TRACE1 | TRACE2 | TRACE3 }  
:TRACe<x>:SPECTrum:LEFToperand?

Where <x> = 4; Only Trace 4 (math trace) is valid.

**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.

**Measurement Views** Spectrum

**Examples** :TRACe4:SPECTrum:LEFToperand TRACE1  
selects Trace 1 as the left operand for the math trace.

**Related Commands** :TRACe<x>:SPECTrum:RIGHToperand

**:TRACe<x>:SPECTrum:RIGHtooperand(?)**

Selects or queries the right operand for the math trace (Trace 4) in the Spectrum view.

**Syntax** :TRACe<x>:SPECTrum:RIGHtooperand { TRACE1 | TRACE2 | TRACE3 }

:TRACe<x>:SPECTrum:RIGHtooperand?

Where <x> = 4; Only Trace 4 (math trace) is valid.

**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.

**Measurement Views** Spectrum

**Examples** :TRACe4:SPECTrum:RIGHtooperand TRACE1  
selects Trace 1 as the right operand for the math trace.

**Related Commands** :TRACe<x>:SPECTrum:LEFTooperand



**:TRACe<x>:SPECTrum:SElect(?)**

Selects or queries the trace to display the readout at the upper left of the Spectrum view.

**Syntax** :TRACe<x>:SPECTrum:SElect  
:TRACe<x>:SPECTrum:SElect?

Where <x> = 1 to 5; All traces are valid.

---

**NOTE.** TRACe5 (spectrogram) is valid when the spectrum and spectrogram measurements are running.

---

**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.

**Measurement Views** Spectrum

**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.

## :TRACe:SPURious Subgroup

The :TRACe:SPURious commands control traces in the Spurious measurement.

Command Tree	Header	Parameter
	:TRACe	
	:SPURious	
	:COUNT	<numeric_value>
	:ENABle	<boolean>
	:RESet	
	:FREeze	<boolean>
	:FUNction	NONE   MAXHold   AVERage

## :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<x>:SPURious:FUNction is set to MAXHold or AVERage and :INITiate:CONTInuous is set to OFF.

**Syntax**     :TRACe:SPURious:COUNT <number>  
              :TRACe:SPURious:COUNT?

**Arguments**   <number>::=<NR1> specifies the count for multi-trace functions.  
                  Range: 1 to 10000.

**Measurement Views**   Spurious

**Examples**       :TRACe:SPURious:COUNT 32  
                  sets the count to 32 for multi-trace functions.

**Related Commands**   :INITiate:CONTInuous, :TRACe:SPURious:FUNction

## :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.

**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.

**Measurement Views** Spurious

**Examples** :TRACe:SPURious:COUNT:ENABle ON  
enables the count for multi-trace functions.

**Related Commands** :TRACe:SPURious:FUNCTion

## :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.

**Syntax** :TRACe:SPURious:COUNT:RESet

**Arguments** None

**Measurement Views** Spurious

**Examples** :TRACe:SPURious:COUNT:RESet  
clears the multi-function data and counter, and restarts the process.

**Related Commands** :TRACe:SPURious:FUNCTion

**:TRACe:SPURious:FREeze(?)**

Determines whether or not to freeze the display of the trace in the Spurious view.

**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.

**Measurement Views** Spurious

**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.

**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.

**Measurement Views** Spurious

**Examples** :SENSe:SPURious:FUNction MAXHold  
displays the Max Hold trace in the Spurious measurement.

# TRIGger Commands

The TRIGger commands control triggering.

## Command Tree

Header	Parameter
:TRIGger	
:MASK	
:NEW	<freq(1)>,<ampl(1)>, <freq(2)>,<ampl(2)>, ...,<freq(n)>,<ampl(n)>
:AUTO	<meas_ID>,<trace_ID>, <x_margin>,<y_margin>
:OPEN	<file_name>
:SAVE	<file_name>
[:SEQUence]	
:ADVanced	
:SWEEp	
:MODE	<boolean>
:EVENT	
:EXTFront	
:IMPedance	OHM50   OHM5K
:LEVEl	<numeric_value>
:SLOPe	RISe   FALL
:EXTRear	
:SLOPe	RISe   FALL
:GATed	HIGH   LOW
:INPut	
:FMAsk	
:VIOLation	T   F   TF   FT   TFT   FTF
:LEVEl	<numeric_value>
:SLOPe	RISe   FALL
:TDBWidth	<numeric_value>
ACTual?	
STATe	<boolean>
:TYPE	POWER   FMAsk
:SOURce	INPut   EXTFront   EXTRear   EXTGated   LINE
:FORCed	<boolean>
:IMMediate	
:STATus	<boolean>

```
:TIME  
  :DElay      <numeric_value>  
  :POSition   <numeric_value>
```

## :TRIGger:MASK:NEW (No Query Form)

Loads a new frequency mask.

**Syntax** :TRIGger:MASK:NEW <freq(1)>,<ampl(1)>,<freq(2)>,<ampl(2)>,  
...,<freq(n)>,<ampl(n)> (n = 500 maximum)

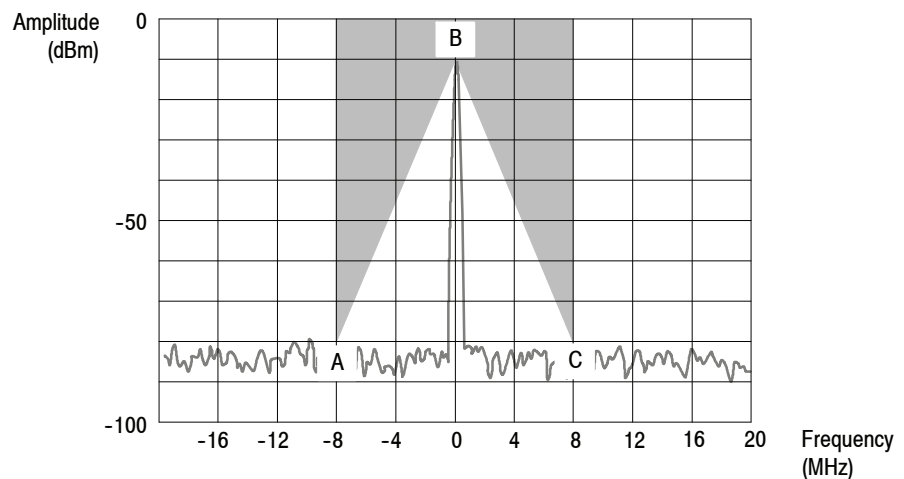
**Arguments** <freq(n)>,<ampl(n)> is a frequency (Hz) and amplitude (dBm) pair to specify a point of the mask. Up to 500 pairs can be specified with zero frequency being the center screen.

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.)

**Measurement Views** All

**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 Figure 2-5.



**Figure 2-5: Trigger mask setting example**

**Related Commands** :TRIGger[:SEquence]:EVENT:INPut:TYPE, :TRIGger[:SEquence]:STATus

**:TRIGger:MASK:NEW:AUTO (No Query Form)**

Draws a new frequency mask automatically based on a reference trace.

**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.

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)

<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 Views** All

**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.

**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* on page 2-495 in the MMEMory command section.

**Measurement Views** All

**Examples** :TRIGger:MASK:OPEN "C:\My Documents\Mask1"  
opens the trigger mask with the *Mask1* file in the *My Document* directory.

**Related Commands** :TRIGger:MASK:SAVE

## :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.

**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* on page 2-495 in the MMEMory command section.

**Measurement Views** All

**Examples** :TRIGger:MASK:SAVE "C:\My Documents\Mask1"  
saves the trigger mask to the *Mask1* file in the *My Document* directory.

**Related Commands** :TRIGger:MASK:OPEN

**:TRIGger[:SEquence]:ADVanced:SWEep:MODE(?)**

Determines whether or not to trigger each segment in the swept acquisition mode.

**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.

**Measurement Views** All

**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.

**Syntax** :TRIGger[:SEquence]:EVENT:EXTFront:IMPedance { OHM50 | OHM5K }  
:TRIGger[:SEquence]:EVENT:EXTFront:IMPedance?

**Arguments** OHM50 selects 50  $\Omega$  impedance.  
OHM5K selects 5 k $\Omega$  impedance.

**Measurement Views** All

**Examples** :TRIGger:SEquence:EVENT:EXTFront:IMPedance OHM50  
selects 50  $\Omega$  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.

**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.

**Measurement Views** All

**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.

**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.

**Measurement Views** All

**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.

**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.

**Measurement Views** All

**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.

**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.

**Measurement Views** All

**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.

**Syntax** :TRIGger[:SEQuence]:EVENT:INPut:FMASk:VIOLation  
 { T | F | TF | FT | TFT | FTF }  
 :TRIGger[:SEQuence]:EVENT:INPut:FMASk:VIOLation?

**Arguments** Table 2-50 lists the arguments which represent the trigger violations.

**Table 2-50: 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.
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.

**Measurement Views** All

**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.

**Syntax**     :TRIGger[:SEquence]:EVENT:INPut:LEVel <value>  
              :TRIGger[:SEquence]:EVENT:INPut:LEVel?

**Arguments**   <value>::=<NRf> specifies the trigger level.  
                  Range: -170 to +50 dBm.

**Measurement Views**   All

**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.

**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.

**Measurement Views**   All

**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.

**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.

**Measurement Views** All

**Examples** :TRIGger:SEQuence:EVENT:INPut:TDBWidth 5MHz  
sets the time-domain bandwidth to 5 MHz for the RF input power trigger.

**Related Commands** :TRIGger[:SEQuence]:EVENT:INPut:TDBWidth:STATe

**:TRIGger[:SEQuence]:EVENT:INPut:TDBWidth:ACTual? (Query Only)**

Queries the actual time-domain bandwidth for the RF input power trigger.

**Syntax** :TRIGger[:SEQuence]:EVENT:INPut:TDBWidth:ACTual?

**Arguments** None

**Returns** <NRf> Actual time-domain bandwidth.

**Measurement Views** All

**Examples** :TRIGger:SEQuence:EVENT:INPut:TDBWidth:ACTual?  
might return 1.000E+6, indicating that the actual time-domain bandwidth is 1 MHz.

**Related Commands** :TRIGger[:SEQuence]:EVENT:INPut:TDBWidth

**:TRIGger[:SEquence]:EVENT:INPut:TDBWidth:STATe(?)**

Determines whether to set the time-domain bandwidth automatically or manually for the RF input power trigger.

**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.

**Measurement Views** All

**Examples** :TRIGger:SEquence:EVENT:INPut:TDBWidth:STATe OFF  
sets the time-domain bandwidth automatically.

**Related Commands** :TRIGger[:SEquence]:EVENT:INPut:TDBWidth

**:TRIGger[:SEquence]:EVENT:INPut:TYPE(?)**

Selects or queries the trigger type for the source of RF input.

**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.

**Measurement Views** All

**Examples** :TRIGger:SEquence:EVENT:INPut:TYPE FMASK  
uses the frequency mask for triggering.



**:TRIGger[:SEQuence]:EVENT:SOURce(?)**

Selects or queries the trigger event source.

**Syntax** :TRIGger[:SEQuence]:EVENT:SOURce  
 { INPut | EXTFront | EXTRear | EXTGated | LINE }  
 :TRIGger[:SEQuence]:EVENT:SOURce?

**Arguments** Table 2-51 lists the arguments.

**Table 2-51: Trigger event source**

Argument	Source
INPut	RF input
EXTFront	Trigger in (front)
EXTRear	Trigger in (rear)
EXTGated	Gate in
LINE	AC line

**Measurement Views** All

**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).

**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.

**Measurement Views** All

**Examples** :TRIGger:SEQuence:FORCed ON  
causes a manual trigger if the acquisition is armed, ready and waiting for a trigger.

**Related Commands** :TRIGger[:SEQuence]:STATus

**: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).

**Syntax** :TRIGger[:SEQuence]:IMMediate

**Arguments** None

**Measurement Views** All

**Examples** :TRIGger:SEQuence:IMMediate  
causes a trigger immediately, skipping the event detection and delay.

**Related Commands** :TRIGger[:SEQuence]:STATus

**:TRIGger[:SEQuence]:STATus(?)**

Selects or queries the trigger mode (Free Run or Triggered).

**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.

**Measurement Views** All

**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).

**Syntax**     :TRIGger[:SEquence]:TIME:DELay <value>  
              :TRIGger[:SEquence]:TIME:DELay?

**Arguments**   <value>::=<NRf> specifies the trigger delay time.  
                  Range: 0 to 60 s.

**Measurement Views**   All

**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).

**Syntax**     :TRIGger[:SEquence]:TIME:POSition <value>  
              :TRIGger[:SEquence]:TIME:POSition?

**Arguments**   <value>::=<NRf> specifies the trigger position.  
                  Range: 0 to 100%.

**Measurement Views**   All

**Examples**     :TRIGger:SEquence:TIME:POSition 20  
                  sets the trigger position to 20% of the acquisition record.

# UNIT Commands

The :UNIT commands specify fundamental units for measurement.

## Command Tree

Header	Parameter
:UNIT	
:POWer	DBM   DBV   VOLTs   WATTs   DBUW   DBW   DBUV   DBMV   DBUA   DBUV_M   DBUA_M   AMPS

**:UNIT:POWer(?)**

Selects or queries the fundamental unit of power.  
This command is equivalent to [:SENSe]:POWer:UNITs.

**Syntax** :UNIT:POWer { DBM | DBV | VOLTs | WATTs | DBUW | DBW  
| DBUV | DBMV | DBUA | DBUV\_M | DBUA\_M | AMPS }  
:UNIT:POWer?

**Arguments** Table 2-52 lists the arguments.

**Table 2-52: Power units**

Argument	Power unit
DBM	dBm
DBV	dBV
VOLTs	Volts
WATTs	Watts
DBUW	dB $\mu$ W
DBW	dBW
DBUV	dB $\mu$ V
DBMV	dBmV
DBUA	dB $\mu$ A
DBUV_M	dB $\mu$ V/m
DBUA_M	dB $\mu$ A/m
AMPS	Amps

**NOTE.** Select dB $\mu$ V/m or dB $\mu$ A/m unit when using an antenna table.

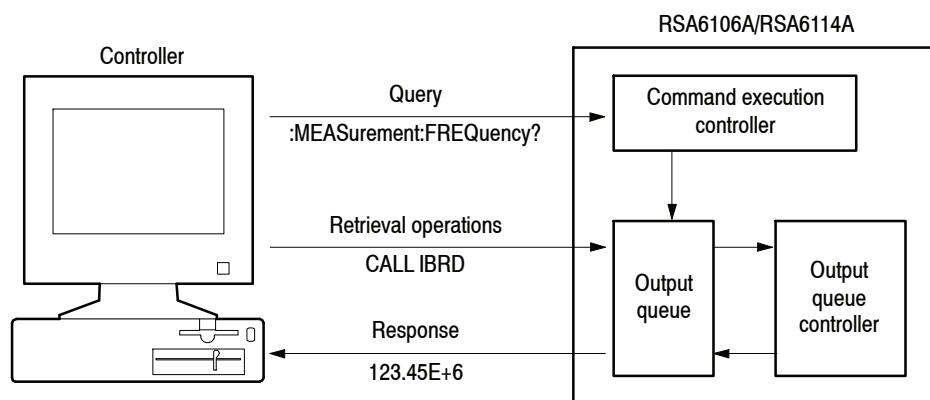
**Measurement Views** All

**Examples** :UNIT:POWer DBM  
specifies the fundamental unit of power as dBm.

**Related Commands** [:SENSe]:POWer:UNITs

## 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 2-6: 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)* on page 3-4.







# 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

Figure 3-1 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 Table 3-1.

**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.

\* The use of Bit 15 is not allowed in SCPI.  
The value of this bit is always zero.

CR: Condition Register  
TR: Transition Register  
EVR: Event Register  
ENR: Enable Register

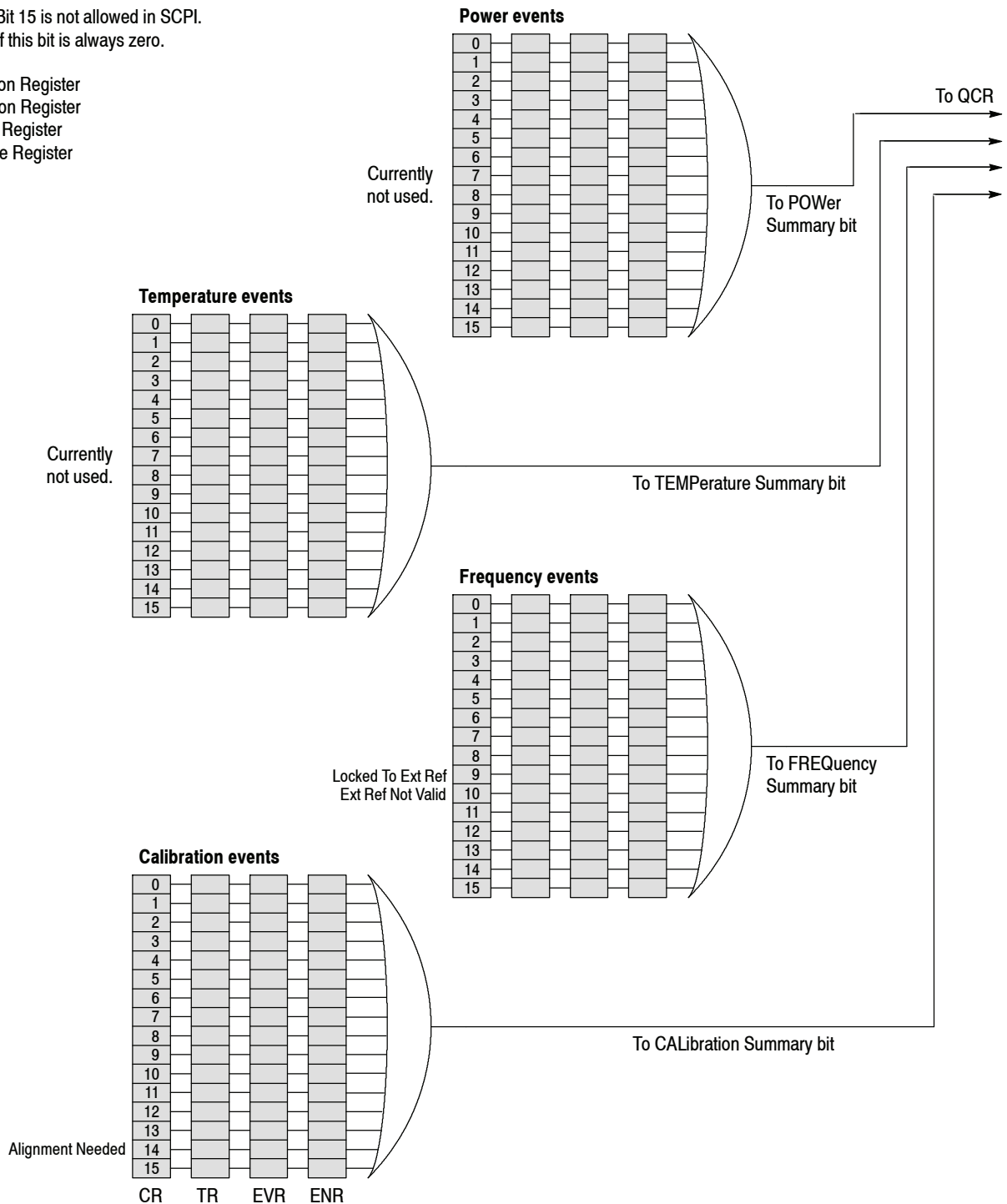


Figure 3- 1: Status/Event reporting mechanism

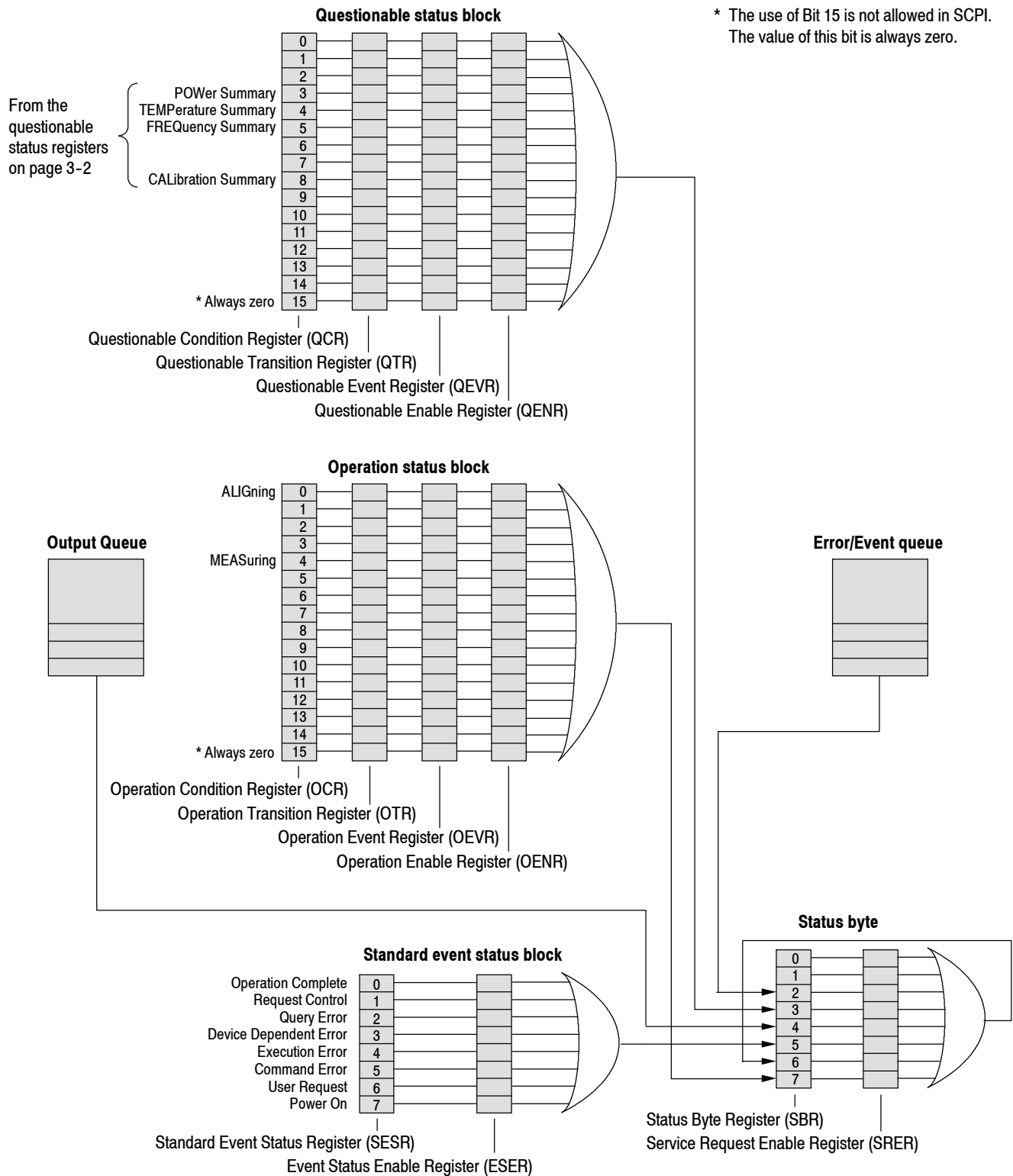


Figure 3-1: Status/Event reporting mechanism (Cont.)

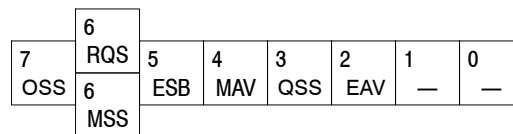
## Status Byte

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 (see Figure 3-2 and Table 3-2). 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.



**Figure 3-2: Status Byte Register (SBR)**

**Table 3-2: SBR bit functions**

Bit	Function
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 (see Figure 3-3). 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 3-3: 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 Figure 3-4 and Table 3-3. 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 3-4: 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: <ul style="list-style-type: none"> <li>■ 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</li> <li>■ When the command could not be executed properly because the conditions for execution differed from those essentially required</li> </ul>
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: <ul style="list-style-type: none"> <li>■ An attempt was made to retrieve messages from the output queue, despite the fact that the output queue is empty or in pending status.</li> <li>■ The output queue messages have been cleared despite the fact that they have not been retrieved.</li> </ul>
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 Figure 3-5 and Table 3-4. 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 3-5: 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 Figure 3-6 and Table 3-5. Use the :STATus:QUEStionable commands to access the questionable status register set.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CW						CAL			FREQ	TEMP	POW			

**Figure 3-6: 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 Figure 3-7 and Table 3-6. 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	9	8	7	6	5	4	3	2	1	0
					ERNV	LTER									

**Figure 3-7: 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 (LTER). 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 Figure 3-8 and Table 3-7. 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 3-8: 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.

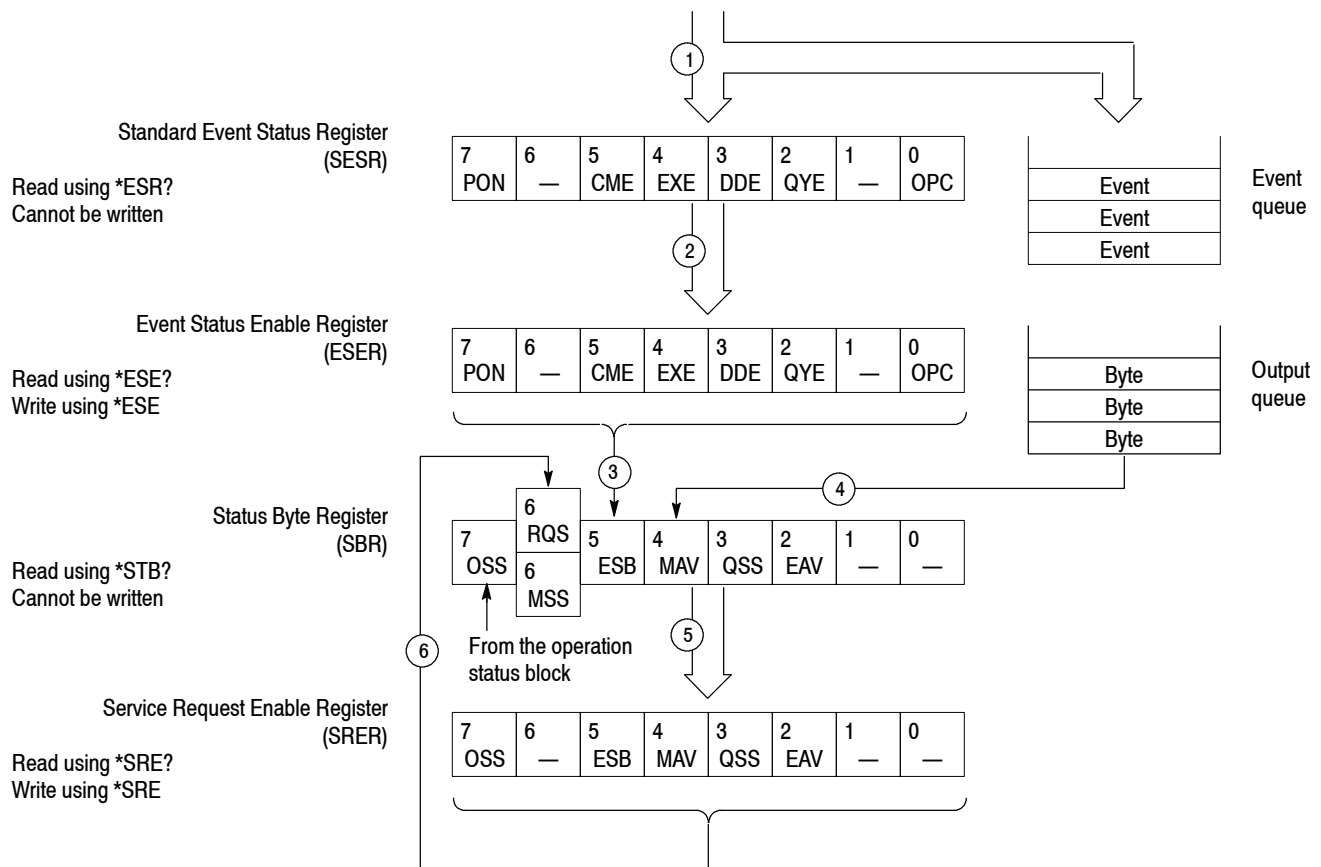
The output queue will be emptied each time a command or query is received, so the controller must read the output queue before the next command or query is issued. If this is not done, an error will occur and the output queue will be emptied; however, the operation will proceed even if an error occurs.

**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

Figure 3-9 shows an outline of the sequence for status and event processing.



**Figure 3-9: 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 Query \*OPC?** 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

**Table 3-8: Command errors (Cont.)**

<b>Error code</b>	<b>Error message</b>
-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
-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**

<b>Error code</b>	<b>Error message</b>
-200	Execution error
-201	Invalid while in local
-202	Settings lost due to RTL

**Table 3-9: Execution errors (Cont.)**

<b>Error code</b>	<b>Error message</b>
-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
-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

**Table 3-9: Execution errors (Cont.)**

<b>Error code</b>	<b>Error message</b>
-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**

<b>Error code</b>	<b>Error message</b>
-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 Tables 3-12, 3-13, and 3-14, respectively.

**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

**Table 3- 12: Device errors, global condition (cont.)**

<b>Error code</b>	<b>Error message</b>
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**

<b>Error code</b>	<b>Error message</b>
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.

**Table 3-13: Device errors, measurement condition (cont.)**

Error code	Error message
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

**Table 3- 14: Device errors, source condition (cont.)**

<b>Error code</b>	<b>Error message</b>
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.
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



**Table 3-14: Device errors, source condition (cont.)**

<b>Error code</b>	<b>Error message</b>
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



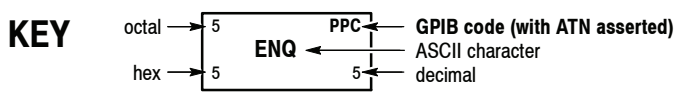


## **Appendix A: Character Charts**

The ASCII and GPIB code chart is shown in Table A-1 on page A-2.

**Table A-1: ASCII & GPIB code chart**

B7 B6 BITS B4 B3 B2 B1	0 0 0		0 0 1		0 1 0		0 1 1		1 0 0		1 0 1		1 1 0		1 1 1	
	CONTROL				NUMBERS SYMBOLS				UPPER CASE				LOWER CASE			
0 0 0 0	0 0	NUL 0	20 10	DLE 16	40 20	SP 32	60 30	0 48	100 40	@ 64	120 50	P 80	140 60	' 96	160 70	p 112
0 0 0 1	1 1	GTL SOH 1	21 11	LL0 DC1 17	41 21	LA1 ! 33	61 31	LA17 1 49	101 41	TA1 A 65	121 51	TA17 Q 81	141 61	SA1 a 97	161 71	SA17 q 113
0 0 1 0	2 2	STX 2	22 12	DC2 18	42 22	LA2 " 34	62 32	LA18 2 50	102 42	TA2 B 66	122 52	TA18 R 82	142 62	SA2 b 98	162 72	SA18 r 114
0 0 1 1	3 3	ETX 3	23 13	DC3 19	43 23	LA3 # 35	63 33	LA19 3 51	103 43	TA3 C 67	123 53	TA19 S 83	143 63	SA3 c 99	163 73	SA19 s 115
0 1 0 0	4 4	SDC EOT 4	24 14	DCL DC4 20	44 24	LA4 \$ 36	64 34	LA20 4 52	104 44	TA4 D 68	124 54	TA20 T 84	144 64	SA4 d 100	164 74	SA20 t 116
0 1 0 1	5 5	PPC ENQ 5	25 15	PPU NAK 21	45 25	LA5 % 37	65 35	LA21 5 53	105 45	TA5 E 69	125 55	TA21 U 85	145 65	SA5 e 101	165 75	SA21 u 117
0 1 1 0	6 6	ACK 6	26 16	SYN 22	46 26	LA6 & 38	66 36	LA22 6 54	106 46	TA6 F 70	126 56	TA22 V 86	146 66	SA6 f 102	166 76	SA22 v 118
0 1 1 1	7 7	BEL 7	27 17	ETB 23	47 27	LA7 , 39	67 37	LA23 7 55	107 47	TA7 G 71	127 57	TA23 W 87	147 67	SA7 g 103	167 77	SA23 w 119
1 0 0 0	8 8	GET BS 8	30 18	SPE CAN 24	50 28	LA8 ( 40	70 38	LA24 8 56	110 48	TA8 H 72	130 58	TA24 X 88	150 68	SA8 h 104	170 78	SA24 x 120
1 0 0 1	9 9	TCT HT 9	31 19	SPD EM 25	51 29	LA9 ) 41	71 39	LA25 9 57	111 49	TA9 I 73	131 59	TA25 Y 89	151 69	SA9 i 105	171 79	SA25 y 121
1 0 1 0	A A	LF 10	32 1A	SUB 26	52 2A	LA10 * 42	72 3A	LA26 : 58	112 4A	TA10 J 74	132 5A	TA26 Z 90	152 6A	SA10 j 106	172 7A	SA26 z 122
1 0 1 1	B B	VT 11	33 1B	ESC 27	53 2B	LA11 + 43	73 3B	LA27 ; 59	113 4B	TA11 K 75	133 5B	TA27 [ 91	153 6B	SA11 k 107	173 7B	SA27 { 123
1 1 0 0	C C	FF 12	34 1C	FS 28	54 2C	LA12 , 44	74 3C	LA28 < 60	114 4C	TA12 L 76	134 5C	TA28 \ 92	154 6C	SA12 l 108	174 7C	SA28   124
1 1 0 1	D D	CR 13	35 1D	GS 29	55 2D	LA13 - 45	75 3D	LA29 = 61	115 4D	TA13 M 77	135 5D	TA29 ] 93	155 6D	SA13 m 109	175 7D	SA29 } 125
1 1 1 0	E E	SO 14	36 1E	RS 30	56 2E	LA14 . 46	76 3E	LA30 > 62	116 4E	TA14 N 78	136 5E	TA30 ^ 94	156 6E	SA14 n 110	176 7E	SA30 ~ 126
1 1 1 1	F F	SI 15	37 1F	US 31	57 2F	LA15 / 47	77 3F	UNL ? 63	117 4F	TA15 O 79	137 5F	UNT - 95	157 6F	SA15 o 111	177 7F	RUBOUT (DEL) 127
		ADDRESSED COMMANDS		UNIVERSAL COMMANDS		LISTEN ADDRESSES		TALK ADDRESSES		SECONDARY ADDRESSES OR COMMANDS						



**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

Table B-1 lists the GPIB interface functions this instrument implements. Each function is briefly described on page B-2.

**Table B- 1: GPIB interface function implementation**

<b>Interface function</b>	<b>Implemented subset</b>	<b>Capability</b>
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 shows the standard interface messages that are supported by the analyzer.

**Table B-2: Standard interface messages**

<b>Message</b>	<b>Type</b>	<b>Implemented</b>
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

**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 Table C-1 through C-11.

**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<n>:LABel	Ext Gain Table n
:CALibration:CORRection:EXTErnal:EDIT<n>: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
<b>:CALCulate basic command subgroup</b>	
: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

**Table C-3: Factory initialization settings, CALCulate commands (Cont.)**

Header	Default value
:CALCulate:SEARch:LIMit:OPERation	GT
:CALCulate:SEARch:LIMit:OPERation:FEED	“Spectrum”, “Trace 1”
:CALCulate:SEARch:LIMit:OPERation:SLIMit	-20 dBm
:CALCulate:SEARch:LIMit:STATe	OFF
<b>:CALCulate:DPSA subgroup</b>	
:CALCulate:DPSA:MARKer<x>:TRACe	TRACE1
<b>:CALCulate:SPECTrum subgroup</b>	
:CALCulate:SPECTrum:MARKer<x>:TRACe	TRACE1

**Table C-4: Factory initialization settings, DISPlay commands**

Header	Default value
<b>:DISPlay:ACPower subgroup</b>	
: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
<b>:DISPlay:AVTime subgroup</b>	
: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
<b>:DISPlay:CONSte subgroup</b>	
:DISPlay:CONSte:WINDow:TRACe:GRATicule:GRID:STATe	ON

**Table C- 4: Factory initialization settings, DISPlay commands (Cont.)**

Header	Default value
<b>:DISPlay:DDEMod subgroup</b>	
:DISPlay:DDEMod:RADix	BINary
:DISPlay:DDEMod:X[:SCALe]	128 symbols
:DISPlay:DDEMod:X[:SCALe]:AUTO:STATe	ON
:DISPlay:DDEMod:X[:SCALe]:OFFSet	-1 symbol
<b>:DISPlay:DPSA subgroup</b>	
:DISPlay:DPSA:WINDow:TRACe:GRATICule:GRID:STATe	ON
:DISPlay:DPSA:Y[:SCALe]:PDIVision	10 dB/div
<b>:DISPlay:EVM subgroup</b>	
:DISPlay:EVM:Y[:SCALe]	1%
:DISPlay:EVM:Y[:SCALe]:OFFSet	0
<b>:DISPlay:FVTime subgroup</b>	
: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
<b>:DISPlay:GENeral subgroup</b>	
:DISPlay:GENeral:MEASview:SElect	SPECTrum
<b>:DISPlay:IQVTime subgroup</b>	
: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 $\mu$ V
:DISPlay:IQVTime:Y[:SCALe]:OFFSet	0
<b>:DISPlay:MCPower subgroup</b>	
: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

**Table C-4: Factory initialization settings, DISPlay commands (Cont.)**

Header	Default value
<b>:DISPlay:MERRor subgroup</b>	
:DISPlay:MERRor:Y[:SCALe]	1%
:DISPlay:MERRor:Y[:SCALe]:OFFSet	0
<b>:DISPlay:OBWidth subgroup</b>	
:DISPlay:OBWidth:MARKer:SHOW:STATe	ON
:DISPlay:OBWidth:SELected:BANDwidth	OBWidth
: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
<b>:DISPlay:PError subgroup</b>	
:DISPlay:PError:Y[:SCALe]	1°
:DISPlay:PError:Y[:SCALe]:OFFSet	0
<b>:DISPlay:PHVTime subgroup</b>	
: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
<b>:DISPlay:PNOise subgroup</b>	
: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
<b>:DISPlay:PULSe subgroup</b>	
:DISPlay:PULSe:RESult:ATX	OFF
:DISPlay:PULSe:RESult:AVERage	ON
:DISPlay:PULSe:RESult:DROop	OFF

**Table C- 4: Factory initialization settings, DISPlay commands (Cont.)**

<b>Header</b>	<b>Default value</b>
:DISPlay:PULSe:RESult:DUTPct	OFF
:DISPlay:PULSe:RESult:DUTRatio	OFF
:DISPlay:PULSe:RESult:FALL	OFF
:DISPlay:PULSe:RESult:FRDeviation	OFF
:DISPlay:PULSe:RESult:MFRqerror	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
: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

**Table C-4: Factory initialization settings, DISPlay commands (Cont.)**

<b>Header</b>	<b>Default value</b>
:DISPlay:PULSe:TRACe:X[:SCALe]:FULL	SElected
:DISPlay:PULSe:TRACe:X[:SCALe]:OFFSet	0
:DISPlay:PULSe:TRACe:X[:SCALe]:PDIVision	10 $\mu$ s
:DISPlay:PULSe:TRACe:Y[:SCALe]:FULL	100 dB
:DISPlay:PULSe:TRACe:Y[:SCALe]:OFFSet	0
<b>:DISPlay:SGRam subgroup</b>	
:DISPlay:SGRam:FREQuency:OFFSet	1.5 GHz
:DISPlay:SGRam:FREQuency:SCALe	40 MHz
:DISPlay:SGRam:TIME:OFFSet	0
:DISPlay:SGRam:TIME:OVERlap	ON
:DISPlay:SGRam:TIME:SCALe	0
<b>:DISPlay:SPECTrum subgroup</b>	
:DISPlay:SPECTrum:MARKer:NOISe:MODE	OFF
:DISPlay:SPECTrum:WINDow:TRACe:GRATicule:GRID:STATe	ON
:DISPlay:SPECTrum:Y[:SCALe]:PDIVision	10 dB/div
<b>:DISPlay:SPURious subgroup</b>	
:DISPlay:SPURious:MARKer:SHOW:STATe	ON
: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:SPURious:Y[:SCALe]	100 dB
:DISPlay:SPURious:Y[:SCALe]:OFFSet	0
<b>:DISPlay:TOVerview subgroup</b>	
: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:NOISe[:STATe]	OFF
:OUTPut:IF[:STATe]	(Option 05 only) OFF
:OUTPut:IF:BANDwidth[:BWIDth]	(Option 05 only) FLATtop
:OUTPut:IQ[:STATe]	(Option 05 only) OFF

**Table C- 8: Factory initialization settings, SENSE commands**

Header	Default value
<b>[:SENSe] basic command subgroup</b>	
[:SENSe]:ACQuisition:FFRame:Limit	1
[:SENSe]:ACQuisition:FFRame:STATe	OFF
[:SENSe]:ACQuisition:MODE(?)	AUTO
[:SENSe]:ANALysis:ADVanced:DITHer	AUTO
[:SENSe]:ANALysis:LENGth	7.44 $\mu$ s
[:SENSe]:ANALysis:LENGth:AUTO	ON
[:SENSe]:ANALysis:REFerence	ACQSTART
[:SENSe]:ANALysis:STARt	0
[:SENSe]:ANALysis:STARt:AUTO	ON

**Table C-8: Factory initialization settings, SENSE commands (Cont.)**

<b>Header</b>	<b>Default value</b>
[[:SENSe]:POWer:UNITs	DBM
[[:SENSe]:ROSCillator:SOURce	INTernal
[[:SENSe]:SPECtrum:LENGth	7.44 $\mu$ s
[[:SENSe]:SPECtrum:LENGth:AUTO	ON
[[:SENSe]:SPECtrum:STARt	0
[[:SENSe]:SPECtrum:TIME:MODE	COMMon
<b>[[:SENSe]:ACPower subgroup</b>	
[[: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
[[: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
<b>[[:SENSe]:AVTime subgroup</b>	
[[:SENSe]:AVTime:BANDwith :BWIDTH	1 MHz
[[:SENSe]:]AVTime:MAXTracepoints	HUNDredk
[[:SENSe]:AVTime:METHod	SPAN
[[:SENSe]:AVTime:SPAN	40 MHz
<b>[[:SENSe]:CCDF subgroup</b>	
[[:SENSe]:CCDF:BANDwidth :BWIDTH	40 MHz
[[:SENSe]:CCDF:TIME:AUTO	ON
[[:SENSe]:CCDF:TIME:TYPE	SINGLE

**Table C-8: Factory initialization settings, SENSE commands (Cont.)**

Header	Default value
<b>[[:SENSe]:DDEMod subgroup</b>	
[[: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
<b>[[:SENSe]:DPSA subgroup</b>	
[[: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
[[: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

**Table C-8: Factory initialization settings, SENSE commands (Cont.)**

<b>Header</b>	<b>Default value</b>
[[:SENSe]:DPSA:FREQuency:STEP	2 MHz
[[:SENSe]:DPSA:FREQuency:STEP:AUTO	ON
[[:SENSe]:DPSA:FREQuency:STOP	1.52 GHz
<b>[[:SENSe]:FVTime subgroup</b>	
[[: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
<b>[[:SENSe]:IQVTime subgroup</b>	
[[: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
<b>[[:SENSe]:MCPower subgroup</b>	
[[:SENSe]:MCPower:AVERage	OFF
[[:SENSe]:MCPower:AVERage:COUNT	10
[[:SENSe]:MCPower:BANDwidth]:BWIDth[:RESolution]	30 kHz
[[:SENSe]:MCPower:BANDwidth]:BWIDth[:RESolution]:AUTO	ON
[[:SENSe]:MCPower:BANDwidth]:BWIDth:VIDeo	300 kHz
[[:SENSe]:MCPower:BANDwidth]:BWIDth:VIDeo:STATe	OFF
[[:SENSe]:MCPower:CHANnel:BANDwidth	3.84 MHz
[[:SENSe]:MCPower:CHANnel:FILTer	RRCosine
[[:SENSe]:MCPower:CHANnel:MAIN:BANDwidth]:BWIDth	3.84 MHz
[[: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

**Table C-8: Factory initialization settings, SENSE commands (Cont.)**

<b>Header</b>	<b>Default value</b>
[: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
<b>[:SENSe]:OBWidth subgroup</b>	
[: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
<b>[:SENSe]:PHVTime subgroup</b>	
[: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
<b>[:SENSe]:PNOise subgroup</b>	
[:SENSe]:PNOise:AVERage:COUNT	10
[:SENSe]:PNOise:AVERage:ENABLE	OFF
[:SENSe]:PNOise:CARRier:FREQuency:TRACk	ON
[:SENSe]:PNOise:CARRier:THReshold	-26 dBm

**Table C-8: Factory initialization settings, SENSE commands (Cont.)**

<b>Header</b>	<b>Default value</b>
[[:SENSe]:PNOise:FREQuency:INTEgration:OFFSet:STARt	100 Hz
[[: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
<b>[[:SENSe]:PULSe subgroup</b>	
[[: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:RFAlI	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

**Table C-8: Factory initialization settings, SENSE commands (Cont.)**

Header	Default value
<b>[:SENSe]:SGRam subgroup</b>	
[: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
[:SENSe]:SGRam:BANDwidth]:BWIDth:VIDeo:STATe	OFF
[:SENSe]:SGRam:COLor	TEMPerature
[:SENSe]:SGRam:COLor:MAX	0 dBm
[:SENSe]:SGRam:COLor:MIN	-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
<b>[:SENSe]:SPECTrum subgroup</b>	
[: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

**Table C-8: Factory initialization settings, SENSE commands (Cont.)**

Header	Default value
[[:SENSe]:SPEcTrum:FREQuency:STEP	2 MHz
[[:SENSe]:SPEcTrum:FREQuency:STEP:AUTO	ON
[[:SENSe]:SPEcTrum:FREQuency:STOP	1.52 GHz
[[:SENSe]:SPEcTrum:POINts:COUNT	P801
<b>[[:SENSe]:SPURious subgroup</b>	
[[:SENSe]:SPURious:CARRier:BANDwidth	4 MHz
[[:SENSe]:SPURious:CARRier:BANDwidth:INTegration	1 MHz
[[:SENSe]:SPURious:CARRier:BANDwidth:RESolution:AUTO	ON
[[:SENSe]:SPURious:CARRier:DETEction	PEAK
[[: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[:BWIDTH	200 kHz
[[:SENSe]:SPURious:RANGe<x>:FILTer[:SHAPe]:BANDwidth[:BWIDTH :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



**Table C-8: Factory initialization settings, SENSE commands (Cont.)**

Header	Default value
<b>[:SENSe]:TOVerView subgroup</b>	
[:SENSe]:TOVerView:FREQUency:CENTer	1.5 GHz
[:SENSe:]TOVerView:MAXTracepoints	HUNDredk

**Table C-9: Factory initialization settings, TRACe commands**

Header	Default value
<b>:TRACe&lt;x&gt;:AVTime subgroup</b>	
: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
:TRACe<x>:AVTime:LEFToperand	TRACE2
:TRACe<x>:AVTime:RIGHToperand	TRACE1
<b>:TRACe&lt;x&gt;:CCDF subgroup</b>	
:TRACe<x>:CCDF:FREeze	OFF
:TRACe<x>:CCDF:SHOW	<x>=1: ON <x>=2 and 3: OFF
:TRACe<x>:CCDF:X	3 dB
<b>:TRACe:CONSte subgroup</b>	
:TRACe:CONSte:MODE	VECTors
<b>:TRACe&lt;x&gt;:DPSA subgroup</b>	
: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 %
:TRACe5:DPSA:DOT:PERSistent	ON
:TRACe5:DPSA:DOT:PERSistent:TYPE	VARiable
:TRACe5:DPSA:DOT:PERSistent:VARiable	10

**Table C-9: Factory initialization settings, TRACe commands (Cont.)**

<b>Header</b>	<b>Default value</b>
:TRACe<x>:DPSA:FReeze	OFF
:TRACe<x>:DPSA:FUNction	NORMal
:TRACe<x>:DPSA:LEFTooperand	TRACE1
:TRACe<x>:DPSA:RIghTooperand	TRACE3
<b>:TRACe:FVTime subgroup</b>	
:TRACe:FVTime:COUNt	10
:TRACe:FVTime:COUNt:ENABle	ON
:TRACe:FVTime:FReeze	OFF
:TRACe:FVTime:FUNction	NORMal
<b>:TRACe:IQVTime subgroup</b>	
: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
<b>:TRACe:OBWidth subgroup</b>	
:TRACe:OBW:MAXHold	OFF
<b>:TRACe:PHVTime subgroup</b>	
:TRACe:PHVTime:COUNt	10
:TRACe:PHVTime:COUNt:ENABle	ON
:TRACe:PHVTime:FReeze	OFF
:TRACe:PHVTime:FUNction	NORMal
<b>:TRACe&lt;x&gt;:PNOise subgroup</b>	
: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

**Table C- 9: Factory initialization settings, TRACe commands (Cont.)**

Header	Default value
<b>:TRACe&lt;x&gt;:SGRam subgroup</b>	
:TRACe<x>:SGRam:DETEction	POSitive
:TRACe<x>:SGRam:FREeze	OFF
:TRACe<x>:SGRam:FUNCTion	NONE
:TRACe<x>:SGRam:FUNCTion:TIME	0.02 minutes
:TRACe<x>:SGRam:SElect:LINE	0
<b>:TRACe&lt;x&gt;:SPECTrum subgroup</b>	
: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
<b>:TRACe:SPURious subgroup</b>	
: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 k $\Omega$
:TRIGger[:SEquence]:EVENT:EXTFront:LEVel	1.6 V
:TRIGger[:SEquence]:EVENT:EXTFront:SLOPe	RISe
:TRIGger[:SEquence]:EVENT:GATed	HIGH

**Table C- 10: Factory initialization settings, TRIGger commands (Cont.)**

<b>Header</b>	<b>Default value</b>
: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**

<b>Header</b>	<b>Default value</b>
:UNIT:POWer	dBm

# Appendix D: SCPI Conformance Information

All commands for the RSA6100A Series analyzers are based on SCPI Version 1999.0. Table D-1 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
<b>IEEE common</b>	*CAL?
	*CLS
	*ESE
	*ESR?
	*IDN?
	*OPC
	*OPT?
	*RST
	*SRE
	*STB?
	*TRG
	*WAI
<b>ABORT</b>	:ABORt
<b>INITiate</b>	:INITiate :CONTinuous
	[:IMMediate]
<b>STATus</b>	:STATus :OPERation :CONDition?
	:ENABLE
	[:EVENT]?
	:NTRansition
	:PTRansition
	:PRESet
	:QUEStionable :CONDition?
	:ENABLE
[:EVENT]?	

**Table D-1: SCPI 1999.0-defined commands (Cont.)**

Command group	Command
	:NTRansition
	:PTRansition
<b>SYSTEM</b>	:SYSTEM :COMMunicate :GPIB [:SELF] :ADDRess
	:DATE
	:ERRor :ALL?
	:CODE :ALL?
	[:NEXT]?
	:COUNt?
	[:NEXT]?
	:PRESet
	:TIME
	:VERSion?
<b>UNIT</b>	:UNIT :POWer



# **Glossary and Index**





# 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

Abbreviations, commands, queries, and parameters, 2-5  
:ABORt, 2-73  
ABORt command group, 2-14  
ABORt commands, 2-73  
ACPR, measurement mode, 2-12  
Arguments, parameters, 2-4

## B

Backus-Naur Form, 2-1  
BNF (Backus-Naur form), 2-1

## C

\*CAL, 2-62  
:CALCulate basic command subgroup, 2-77  
CALCulate command group, 2-15  
CALCulate commands, 2-75  
:CALCulate:ACPower subgroup, 2-92  
:CALCulate:ACPower:MARKer<x>:DELTA:X?, 2-93  
:CALCulate:ACPower:MARKer<x>:DELTA:Y?, 2-93  
:CALCulate:ACPower:MARKer<x>:MAXimum, 2-94  
:CALCulate:ACPower:MARKer<x>:PEAK:LEFT, 2-95  
:CALCulate:ACPower:MARKer<x>:PEAK:RIGHT, 2-95  
:CALCulate:ACPower:MARKer<x>:X(?), 2-96  
:CALCulate:ACPower:MARKer<x>:Y?, 2-96  
:CALCulate:AVTime subgroup, 2-97  
:CALCulate:AVTime:MARKer<x>:DELTA:X?, 2-98  
:CALCulate:AVTime:MARKer<x>:DELTA:Y?, 2-98  
:CALCulate:AVTime:MARKer<x>:MAXimum, 2-99  
:CALCulate:AVTime:MARKer<x>:PEAK:HIGHer, 2-100  
:CALCulate:AVTime:MARKer<x>:PEAK:LEFT, 2-100  
:CALCulate:AVTime:MARKer<x>:PEAK:LOWer, 2-101  
:CALCulate:AVTime:MARKer<x>:PEAK:RIGHT, 2-101  
:CALCulate:AVTime:MARKer<x>:TRACe(?), 2-102  
:CALCulate:AVTime:MARKer<x>:X(?), 2-103  
:CALCulate:AVTime:MARKer<x>:Y?, 2-103  
:CALCulate:CONSte subgroup, 2-104  
:CALCulate:CONSte:MARKer<x>:DELTA:X?, 2-104

:CALCulate:CONSte:MARKer<x>:MAGNitude?, 2-105  
:CALCulate:CONSte:MARKer<x>:MAXimum, 2-105  
:CALCulate:CONSte:MARKer<x>:PEAK:LEFT, 2-106  
:CALCulate:CONSte:MARKer<x>:PEAK:RIGHT, 2-106  
:CALCulate:CONSte:MARKer<x>:PHASe?, 2-107  
:CALCulate:CONSte:MARKer<x>:SYMBol?, 2-107  
:CALCulate:CONSte:MARKer<x>:VALue?, 2-108  
:CALCulate:CONSte:MARKer<x>:X(?), 2-108  
:CALCulate:DPSA subgroup, 2-109  
:CALCulate:DPSA:MARKer<x>:DELTA:X?, 2-110  
:CALCulate:DPSA:MARKer<x>:DELTA:Y?, 2-110  
:CALCulate:DPSA:MARKer<x>:MAXimum, 2-111  
:CALCulate:DPSA:MARKer<x>:PEAK:HIGHer, 2-111  
:CALCulate:DPSA:MARKer<x>:PEAK:LEFT, 2-112  
:CALCulate:DPSA:MARKer<x>:PEAK:LOWer, 2-112  
:CALCulate:DPSA:MARKer<x>:PEAK:RIGHT, 2-113  
:CALCulate:DPSA:MARKer<x>:TRACe(?), 2-114  
:CALCulate:DPSA:MARKer<x>:X(?), 2-115  
:CALCulate:DPSA:MARKer<x>:Y?, 2-115  
:CALCulate:DPSA:MARKer<x>:[SET]:CENTer, 2-113  
:CALCulate:EVM subgroup, 2-116  
:CALCulate:EVM:MARKer<x>:DELTA:X?, 2-117  
:CALCulate:EVM:MARKer<x>:DELTA:Y?, 2-117  
:CALCulate:EVM:MARKer<x>:MAXimum, 2-118  
:CALCulate:EVM:MARKer<x>:PEAK:HIGHer, 2-119  
:CALCulate:EVM:MARKer<x>:PEAK:LEFT, 2-119  
:CALCulate:EVM:MARKer<x>:PEAK:LOWer, 2-120  
:CALCulate:EVM:MARKer<x>:PEAK:RIGHT, 2-120  
:CALCulate:EVM:MARKer<x>:X(?), 2-121  
:CALCulate:EVM:MARKer<x>:Y?, 2-121  
:CALCulate:FVTime subgroup, 2-122  
:CALCulate:FVTime:MARKer<x>:DELTA:X?, 2-123  
:CALCulate:FVTime:MARKer<x>:DELTA:Y?, 2-123  
:CALCulate:FVTime:MARKer<x>:MAXimum, 2-124  
:CALCulate:FVTime:MARKer<x>:PEAK:HIGHer, 2-125  
:CALCulate:FVTime:MARKer<x>:PEAK:LEFT, 2-125  
:CALCulate:FVTime:MARKer<x>:PEAK:LOWer, 2-126  
:CALCulate:FVTime:MARKer<x>:PEAK:RIGHT, 2-126  
:CALCulate:FVTime:MARKer<x>:X(?), 2-127  
:CALCulate:FVTime:MARKer<x>:Y?, 2-127

- :CALCulate:IQVTime subgroup, 2-128
- :CALCulate:IQVTime:MARKer<x>:DELTA:X?, 2-129
- :CALCulate:IQVTime:MARKer<x>:DELTA:Y?, 2-129
- :CALCulate:IQVTime:MARKer<x>:MAXimum, 2-130
- :CALCulate:IQVTime:MARKer<x>:PEAK:HIGHer, 2-131
- :CALCulate:IQVTime:MARKer<x>:PEAK:LEFT, 2-131
- :CALCulate:IQVTime:MARKer<x>:PEAK:LOWer, 2-132
- :CALCulate:IQVTime:MARKer<x>:PEAK:RIGHT, 2-132
- :CALCulate:IQVTime:MARKer<x>:TRACe(?), 2-133
- :CALCulate:IQVTime:MARKer<x>:X(?), 2-134
- :CALCulate:IQVTime:MARKer<x>:Y?, 2-134
- :CALCulate:MARKer:ADD, 2-78
- :CALCulate:MARKer:AOff, 2-78
- :CALCulate:MARKer:DELeTe, 2-79
- :CALCulate:MARKer:MODE(?), 2-79
- :CALCulate:MARKer:PEAK:EXCursion(?), 2-80
- :CALCulate:MARKer:PEAK:THReShold(?), 2-80
- :CALCulate:MCPower subgroup, 2-135
- :CALCulate:MCPower:MARKer<x>:DELTA:X?, 2-136
- :CALCulate:MCPower:MARKer<x>:DELTA:Y?, 2-136
- :CALCulate:MCPower:MARKer<x>:MAXimum, 2-137
- :CALCulate:MCPower:MARKer<x>:PEAK:LEFT, 2-138
- :CALCulate:MCPower:MARKer<x>:PEAK:RIGHT, 2-138
- :CALCulate:MCPower:MARKer<x>:X(?), 2-139
- :CALCulate:MCPower:MARKer<x>:Y?, 2-139
- :CALCulate:MERRor subgroup (option 21 only), 2-140
- :CALCulate:MERRor:MARKer<x>:DELTA:X?, 2-141
- :CALCulate:MERRor:MARKer<x>:DELTA:Y?, 2-141
- :CALCulate:MERRor:MARKer<x>:MAXimum, 2-142
- :CALCulate:MERRor:MARKer<x>:PEAK:HIGHer, 2-143
- :CALCulate:MERRor:MARKer<x>:PEAK:LEFT, 2-143
- :CALCulate:MERRor:MARKer<x>:PEAK:LOWer, 2-144
- :CALCulate:MERRor:MARKer<x>:PEAK:RIGHT, 2-144
- :CALCulate:MERRor:MARKer<x>:X(?), 2-145
- :CALCulate:MERRor:MARKer<x>:Y?, 2-145
- :CALCulate:OBWidth subgroup, 2-146
- :CALCulate:OBWidth:MARKer<x>:DELTA:X?, 2-147
- :CALCulate:OBWidth:MARKer<x>:DELTA:Y?, 2-147
- :CALCulate:OBWidth:MARKer<x>:MAXimum, 2-148
- :CALCulate:OBWidth:MARKer<x>:PEAK:HIGHer, 2-148
- :CALCulate:OBWidth:MARKer<x>:PEAK:LEFT, 2-149
- :CALCulate:OBWidth:MARKer<x>:PEAK:LOWer, 2-149
- :CALCulate:OBWidth:MARKer<x>:PEAK:RIGHT, 2-150
- :CALCulate:OBWidth:MARKer<x>:X(?), 2-151
- :CALCulate:OBWidth:MARKer<x>:Y?, 2-151
- :CALCulate:OBWidth:MARKer<x>[:SET]:CENTer, 2-150
- :CALCulate:PERRor subgroup (option 21 only), 2-152
- :CALCulate:PERRor:MARKer<x>:DELTA:X?, 2-153
- :CALCulate:PERRor:MARKer<x>:DELTA:Y?, 2-153
- :CALCulate:PERRor:MARKer<x>:MAXimum, 2-154
- :CALCulate:PERRor:MARKer<x>:PEAK:HIGHer, 2-155
- :CALCulate:PERRor:MARKer<x>:PEAK:LEFT, 2-155
- :CALCulate:PERRor:MARKer<x>:PEAK:LOWer, 2-156
- :CALCulate:PERRor:MARKer<x>:PEAK:RIGHT, 2-156
- :CALCulate:PERRor:MARKer<x>:X(?), 2-157
- :CALCulate:PERRor:MARKer<x>:Y?, 2-157
- :CALCulate:PHVTime subgroup, 2-158
- :CALCulate:PHVTime:MARKer<x>:DELTA:X?, 2-159
- :CALCulate:PHVTime:MARKer<x>:DELTA:Y?, 2-159
- :CALCulate:PHVTime:MARKer<x>:MAXimum, 2-160
- :CALCulate:PHVTime:MARKer<x>:PEAK:HIGHer, 2-161
- :CALCulate:PHVTime:MARKer<x>:PEAK:LEFT, 2-161
- :CALCulate:PHVTime:MARKer<x>:PEAK:LOWer, 2-162
- :CALCulate:PHVTime:MARKer<x>:PEAK:RIGHT, 2-162
- :CALCulate:PHVTime:MARKer<x>:X(?), 2-163
- :CALCulate:PHVTime:MARKer<x>:Y?, 2-163
- :CALCulate:PULSe subgroup (Option 20 only), 2-164
- :CALCulate:PULSe:STATic:MARKer<x>:MAXimum, 2-166
- :CALCulate:PULSe:STATic:MARKer<x>:X(?), 2-169
- :CALCulate:PULSe:STATic:MARKer<x>:Y?, 2-169
- :CALCulate:PULSe:STATistics:MARKer<x>:DELTA:X?, 2-165

- :CALCulate:PULSe:STATistics:MARKer<x>  
:DELTA:Y?, 2-165
- :CALCulate:PULSe:STATistics:MARKer<x>:PEAK  
:HIGHer, 2-166
- :CALCulate:PULSe:STATistics:MARKer<x>:PEAK  
:LEFT, 2-167
- :CALCulate:PULSe:STATistics:MARKer<x>:PEAK  
:LOWer, 2-167
- :CALCulate:PULSe:STATistics:MARKer<x>:PEAK  
:RIGHT, 2-168
- :CALCulate:PULSe:TRACe:MARKer<x>:MAXimum,  
2-171
- :CALCulate:PULSe:TRACe:MARKer<x>:PEAK  
:HIGHer, 2-171
- :CALCulate:PULSe:TRACe:MARKer<x>:PEAK  
:LEFT, 2-172
- :CALCulate:PULSe:TRACe:MARKer<x>:PEAK  
:LOWer, 2-172
- :CALCulate:PULSe:TRACe:MARKer<x>:PEAK  
:RIGHT, 2-173
- :CALCulate:PULSe:TRACe:MARKer<x>:X(?), 2-174
- :CALCulate:PULSe:TRACe:MARKer<x>:Y?, 2-174
- :CALCulate:SEARCh:LIMit:FAIL?, 2-81
- :CALCulate:SEARCh:LIMit:MATCh:BEEP[:STATe]  
(?), 2-82
- :CALCulate:SEARCh:LIMit:MATCh:SACQuire  
[:STATe](?), 2-82
- :CALCulate:SEARCh:LIMit:MATCh:SDATa[:STATe]  
(?), 2-83
- :CALCulate:SEARCh:LIMit:MATCh:SPICture[:STATe]  
(?), 2-84
- :CALCulate:SEARCh:LIMit:MATCh:STRace[:STATe]  
(?), 2-85
- :CALCulate:SEARCh:LIMit:OPERation:FEED(?),  
2-87
- :CALCulate:SEARCh:LIMit:OPERation:MASK  
:LOAD, 2-88
- :CALCulate:SEARCh:LIMit:OPERation:MASK  
:STORE, 2-88
- :CALCulate:SEARCh:LIMit:OPERation:SLIMit(?),  
2-89
- :CALCulate:SEARCh:LIMit:OPERation(?), 2-86
- :CALCulate:SEARCh:LIMit:REPort:DATA?, 2-90
- :CALCulate:SEARCh:LIMit:REPort:POINts?, 2-90
- :CALCulate:SEARCh:LIMit:STATe(?), 2-91
- :CALCulate:SGRam subgroup, 2-175
- :CALCulate:SGRam:MARKer<x>:DELTA:X  
:FREQuency?, 2-176
- :CALCulate:SGRam:MARKer<x>:DELTA:X[:TIME]?,  
2-176
- :CALCulate:SGRam:MARKer<x>:DELTA:Y?, 2-177
- :CALCulate:SGRam:MARKer<x>:MAXimum, 2-177
- :CALCulate:SGRam:MARKer<x>:PEAK:HIGHer,  
2-178
- :CALCulate:SGRam:MARKer<x>:PEAK:LEFT, 2-178
- :CALCulate:SGRam:MARKer<x>:PEAK:LOWer,  
2-179
- :CALCulate:SGRam:MARKer<x>:PEAK:RIGHT,  
2-179
- :CALCulate:SGRam:MARKer<x>:X:FREQuency(?),  
2-180
- :CALCulate:SGRam:MARKer<x>:X[:TIME](?), 2-181
- :CALCulate:SGRam:MARKer<x>:Y?, 2-181
- :CALCulate:SGRam:MARKer<x>[:SET]:CENTer,  
2-180
- :CALCulate:SPECtrum subgroup, 2-182
- :CALCulate:SPECtrum:MARKer<x>:DELTA:X?,  
2-183
- :CALCulate:SPECtrum:MARKer<x>:DELTA:Y?,  
2-183
- :CALCulate:SPECtrum:MARKer<x>:MAXimum,  
2-184
- :CALCulate:SPECtrum:MARKer<x>:PEAK:HIGHer,  
2-184
- :CALCulate:SPECtrum:MARKer<x>:PEAK:LEFT,  
2-185
- :CALCulate:SPECtrum:MARKer<x>:PEAK:LOWer,  
2-185
- :CALCulate:SPECtrum:MARKer<x>:PEAK:RIGHT,  
2-186
- :CALCulate:SPECtrum:MARKer<x>:TRACe(?),  
2-187
- :CALCulate:SPECtrum:MARKer<x>:X(?), 2-188
- :CALCulate:SPECtrum:MARKer<x>:Y?, 2-188
- :CALCulate:SPECtrum:MARKer<x>[:SET]:CENTer,  
2-186
- :CALCulate:SPURious subgroup, 2-189
- :CALCulate:SPURious:MARKer<x>:DELTA:X?,  
2-190
- :CALCulate:SPURious:MARKer<x>:DELTA:Y?,  
2-190
- :CALCulate:SPURious:MARKer<x>:MAXimum,  
2-191
- :CALCulate:SPURious:MARKer<x>:PEAK:HIGHer,  
2-191
- :CALCulate:SPURious:MARKer<x>:PEAK:LEFT,  
2-192
- :CALCulate:SPURious:MARKer<x>:PEAK:LOWer,  
2-192
- :CALCulate:SPURious:MARKer<x>:PEAK:RIGHT,  
2-193
- :CALCulate:SPURious:MARKer<x>:X(?), 2-194
- :CALCulate:SPURious:MARKer<x>:Y?, 2-194

- :CALCulate:SPURious:MARKer<x>[:SET]:CENTer, 2-193
- :CALCulate:TOVerview subgroup, 2-195
- :CALCulate:TOVerview:MARKer<x>:DELTA:X?, 2-196
- :CALCulate:TOVerview:MARKer<x>:DELTA:Y?, 2-196
- :CALCulate:TOVerview:MARKer<x>:MAXimum, 2-197
- :CALCulate:TOVerview:MARKer<x>:PEAK:HIGHer, 2-198
- :CALCulate:TOVerview:MARKer<x>:PEAK:LEFT, 2-198
- :CALCulate:TOVerview:MARKer<x>:PEAK:LOWer, 2-199
- :CALCulate:TOVerview:MARKer<x>:PEAK:RIGHT, 2-199
- :CALCulate:TOVerview:MARKer<x>:X(?), 2-200
- :CALCulate:TOVerview:MARKer<x>:Y?, 2-200
- CALibration command group, 2-21
- CALibration commands, 2-201
- :CALibration:ABORt, 2-202
- :CALibration:AUTO(?), 2-202
- :CALibration:CORREction:EXTernal:EDIT<x>:LABel(?), 2-203
- :CALibration:CORREction:EXTernal:EDIT<x>:STATe(?), 2-203
- :CALibration:CORREction:EXTernal:GAIN:STATe(?), 2-204
- :CALibration:CORREction:EXTernal:GAIN[:MAGNitude]?, 2-204
- :CALibration:CORREction:EXTernal:PROBE:CONNect?, 2-205
- :CALibration:CORREction:EXTernal:PROBE:STATe(?), 2-206
- :CALibration:CORREction:EXTernal:PROBE[:MAGNitude]?, 2-205
- Case sensitivity, 2-8
- Character chart, A-1
- \*CLS, 2-63
- Command group
  - ABORt, 2-14
  - CALCulate, 2-15
  - CALibration, 2-21
  - DISPlay, 2-22
  - FETCh, 2-29
  - IEEE common, 2-14
  - INITiate, 2-35
  - INPut, 2-35
  - MMEMory, 2-36
  - OUTPut, 2-37
  - READ, 2-38

- SENSe, 2-44
- STATus, 2-53
- SYSTem, 2-54
- TRACe, 2-55
- TRIGger, 2-58
- UNIT, 2-59
- Commands
  - chaining, 2-6
  - rules for forming, 2-1
  - structure of IEEE 488.2 commands, 2-9
  - syntax, 2-1
- Conformance information, D-1
- Creating commands, 2-3

## D

- :DISPlay basic command subgroup, 2-208
- :DISPlay command group, 2-22
- DISPlay commands, 2-207
- :DISPlay:ACPower subgroup, 2-211
- :DISPlay:ACPower:MARKer:SHOW:STATe(?), 2-212
- :DISPlay:ACPower:PLEVel:SHOW:STATe(?), 2-212
- :DISPlay:ACPower:RESet:SCALE, 2-213
- :DISPlay:ACPower:WINDow:TRACe:GRATICule:GRID:STATe(?), 2-213
- :DISPlay:ACPower:X[:SCALE]:AUTO, 2-214
- :DISPlay:ACPower:X[:SCALE]:OFFSet(?), 2-215
- :DISPlay:ACPower:X[:SCALE](?), 2-214
- :DISPlay:ACPower:Y[:SCALE]:AUTO, 2-216
- :DISPlay:ACPower:Y[:SCALE]:OFFSet(?), 2-216
- :DISPlay:ACPower:Y[:SCALE](?), 2-215
- :DISPlay:AVTime subgroup, 2-217
- :DISPlay:AVTime:MARKer:SHOW:STATe(?), 2-218
- :DISPlay:AVTime:RESet, 2-218
- :DISPlay:AVTime:TRIGger:LEVel:STATe(?), 2-219
- :DISPlay:AVTime:WINDow:TRACe:GRATICule:GRID:STATe(?), 2-219
- :DISPlay:AVTime:X:RSCale, 2-220
- :DISPlay:AVTime:X[:SCALE]:AUTO, 2-221
- :DISPlay:AVTime:X[:SCALE]:AUTO:STATe(?), 2-221
- :DISPlay:AVTime:X[:SCALE]:FULL(?), 2-222
- :DISPlay:AVTime:X[:SCALE]:MAXimum?, 2-223
- :DISPlay:AVTime:X[:SCALE]:MINimum?, 2-223
- :DISPlay:AVTime:X[:SCALE]:OFFSet:MAXimum?, 2-225
- :DISPlay:AVTime:X[:SCALE]:OFFSet:MINimum?, 2-225
- :DISPlay:AVTime:X[:SCALE]:OFFSet(?), 2-224
- :DISPlay:AVTime:Y:RSCale, 2-226
- :DISPlay:AVTime:Y[:SCALE]:FULL(?), 2-227
- :DISPlay:AVTime:Y[:SCALE]:OFFSet(?), 2-227
- :DISPlay:CONSte subgroup, 2-228



- :DISPlay:CONStE:WINDow:TRACe:GRATicule:GRID:STATe(?), 2-228
- :DISPlay:DDEMod subgroup, 2-229
- :DISPlay:DDEMod:MEASview:DELete, 2-230
- :DISPlay:DDEMod:MEASview:NEw, 2-231
- :DISPlay:DDEMod:MEASview:SElect(?), 2-231
- :DISPlay:DDEMod:RADix(?), 2-232
- :DISPlay:DDEMod:X[:SCALe]:AUTo, 2-234
- :DISPlay:DDEMod:X[:SCALe]:AUTo:STATe(?), 2-234
- :DISPlay:DDEMod:X[:SCALe]:MAXimum?, 2-235
- :DISPlay:DDEMod:X[:SCALe]:MINimum?, 2-235
- :DISPlay:DDEMod:X[:SCALe]:OFFSet:MAXimum?, 2-237
- :DISPlay:DDEMod:X[:SCALe]:OFFSet:MINimum?, 2-237
- :DISPlay:DDEMod:X[:SCALe]:OFFSet(?), 2-236
- :DISPlay:DDEMod:X[:SCALe]:RESet, 2-238
- :DISPlay:DDEMod:X[:SCALe](?), 2-233
- :DISPlay:DPSA subgroup, 2-239
- :DISPlay:DPSA:WINDow:TRACe:GRATicule:GRID:STATe(?), 2-239
- :DISPlay:DPSA:Y[:SCALe]:PDIVision(?), 2-240
- :DISPlay:EVM subgroup, 2-241
- :DISPlay:EVM:Y[:SCALe]:AUTo, 2-242
- :DISPlay:EVM:Y[:SCALe]:OFFSet(?), 2-242
- :DISPlay:EVM:Y[:SCALe](?), 2-241
- :DISPlay:FVTime subgroup, 2-243
- :DISPlay:FVTime:WINDow:TRACe:GRATicule:GRID:STATe(?), 2-244
- :DISPlay:FVTime:X[:SCALe]:AUTo, 2-246
- :DISPlay:FVTime:X[:SCALe]:AUTo:STATe(?), 2-246
- :DISPlay:FVTime:X[:SCALe]:MAXimum?, 2-247
- :DISPlay:FVTime:X[:SCALe]:MINimum?, 2-247
- :DISPlay:FVTime:X[:SCALe]:OFFSet:MAXimum?, 2-249
- :DISPlay:FVTime:X[:SCALe]:OFFSet:MINimum?, 2-249
- :DISPlay:FVTime:X[:SCALe]:OFFSet(?), 2-248
- :DISPlay:FVTime:X[:SCALe](?), 2-245
- :DISPlay:FVTime:Y[:SCALe]:AUTo, 2-250
- :DISPlay:FVTime:Y[:SCALe]:OFFSet(?), 2-251
- :DISPlay:FVTime:Y[:SCALe](?), 2-250
- :DISPlay:GENeral subgroup, 2-252
- :DISPlay:GENeral:MEASview:DELete, 2-253
- :DISPlay:GENeral:MEASview:NEw, 2-254
- :DISPlay:GENeral:MEASview:SElect(?), 2-255
- :DISPlay:GPRF subgroup, 2-256
- :DISPlay:GPRF:MEASview:DELete, 2-256
- :DISPlay:GPRF:MEASview:NEw, 2-257
- :DISPlay:GPRF:MEASview:SElect(?), 2-258
- :DISPlay:IQVTime subgroup, 2-259
- :DISPlay:IQVTime:WINDow:TRACe:GRATicule:GRID:STATe(?), 2-260
- :DISPlay:IQVTime:X[:SCALe]:AUTo, 2-262
- :DISPlay:IQVTime:X[:SCALe]:AUTo:STATe(?), 2-262
- :DISPlay:IQVTime:X[:SCALe]:MAXimum?, 2-263
- :DISPlay:IQVTime:X[:SCALe]:MINimum?, 2-263
- :DISPlay:IQVTime:X[:SCALe]:OFFSet:MAXimum?, 2-265
- :DISPlay:IQVTime:X[:SCALe]:OFFSet:MINimum?, 2-265
- :DISPlay:IQVTime:X[:SCALe]:OFFSet(?), 2-264
- :DISPlay:IQVTime:X[:SCALe](?), 2-261
- :DISPlay:IQVTime:Y[:SCALe]:AUTo, 2-266
- :DISPlay:IQVTime:Y[:SCALe]:OFFSet(?), 2-267
- :DISPlay:IQVTime:Y[:SCALe]:RESCale, 2-267
- :DISPlay:IQVTime:Y[:SCALe](?), 2-266
- :DISPlay:MCPower subgroup, 2-268
- :DISPlay:MCPower:MARKer:SHOW:STATe(?), 2-269
- :DISPlay:MCPower:PLEVel:SHOW:STATe(?), 2-269
- :DISPlay:MCPower:RESet:SCALe, 2-270
- :DISPlay:MCPower:WINDow:TRACe:GRATicule:GRID:STATe(?), 2-270
- :DISPlay:MCPower:X[:SCALe]:AUTo, 2-271
- :DISPlay:MCPower:X[:SCALe]:OFFSet(?), 2-272
- :DISPlay:MCPower:X[:SCALe](?), 2-271
- :DISPlay:MCPower:Y[:SCALe]:AUTo, 2-273
- :DISPlay:MCPower:Y[:SCALe]:OFFSet(?), 2-273
- :DISPlay:MCPower:Y[:SCALe](?), 2-272
- :DISPlay:MERRor subgroup, 2-274
- :DISPlay:MERRor:Y[:SCALe]:AUTo, 2-275
- :DISPlay:MERRor:Y[:SCALe]:OFFSet(?), 2-275
- :DISPlay:MERRor:Y[:SCALe](?), 2-274
- :DISPlay:OBWidth subgroup, 2-276
- :DISPlay:OBWidth:MARKer:SHOW:STATe(?), 2-277
- :DISPlay:OBWidth:RESet:SCALe, 2-277
- :DISPlay:OBWidth:SElecteD:BANdwidth(?), 2-278
- :DISPlay:OBWidth:WINDow:TRACe:GRATicule:GRID:STATe(?), 2-278
- :DISPlay:OBWidth:X[:SCALe]:AUTo, 2-279
- :DISPlay:OBWidth:X[:SCALe]:OFFSet(?), 2-280
- :DISPlay:OBWidth:X[:SCALe](?), 2-279
- :DISPlay:OBWidth:Y[:SCALe]:AUTo, 2-281
- :DISPlay:OBWidth:Y[:SCALe]:OFFSet(?), 2-281
- :DISPlay:OBWidth:Y[:SCALe](?), 2-280
- :DISPlay:PERRor subgroup, 2-282
- :DISPlay:PERRor:Y[:SCALe]:AUTo, 2-283
- :DISPlay:PERRor:Y[:SCALe]:OFFSet(?), 2-283
- :DISPlay:PERRor:Y[:SCALe](?), 2-282
- :DISPlay:PHVTime subgroup, 2-284
- :DISPlay:PHVTime:WINDow:TRACe:GRATicule:GRID:STATe(?), 2-285

- :DISPlay:PHVTime:X[:SCALE]:AUTO, 2-287
- :DISPlay:PHVTime:X[:SCALE]:AUTO:STATe(?), 2-287
- :DISPlay:PHVTime:X[:SCALE]:MAXimum?, 2-288
- :DISPlay:PHVTime:X[:SCALE]:MINimum?, 2-288
- :DISPlay:PHVTime:X[:SCALE]:OFFSet:MAXimum?, 2-290
- :DISPlay:PHVTime:X[:SCALE]:OFFSet:MINimum?, 2-290
- :DISPlay:PHVTime:X[:SCALE]:OFFSet(?), 2-289
- :DISPlay:PHVTime:X[:SCALE](?), 2-286
- :DISPlay:PHVTime:Y[:SCALE]:AUTO, 2-291
- :DISPlay:PHVTime:Y[:SCALE]:AXIS:REFerence(?), 2-292
- :DISPlay:PHVTime:Y[:SCALE]:AXIS(?), 2-292
- :DISPlay:PHVTime:Y[:SCALE]:OFFSet(?), 2-293
- :DISPlay:PHVTime:Y[:SCALE]:RESCale, 2-293
- :DISPlay:PHVTime:Y[:SCALE](?), 2-291
- :DISPlay:PNOise subgroup (Option 11 only), 2-294
- :DISPlay:PNOise:MARKer:SHOW:STATe(?), 2-295
- :DISPlay:PNOise:RESet:SCALE, 2-295
- :DISPlay:PNOise:WINDow:TRACe:GRATICule:GRID:STATe(?), 2-296
- :DISPlay:PNOise:X[:SCALE]:AUTO, 2-296
- :DISPlay:PNOise:X[:SCALE]:STARt(?), 2-297
- :DISPlay:PNOise:X[:SCALE]:STOP(?), 2-297
- :DISPlay:PNOise:Y[:SCALE]:AUTO, 2-298
- :DISPlay:PNOise:Y[:SCALE]:OFFSet(?), 2-299
- :DISPlay:PNOise:Y[:SCALE]:PDIVision(?), 2-299
- :DISPlay:PNOise:Y[:SCALE](?), 2-298
- :DISPlay:PULSe subgroup (Option 20 only), 2-300
- :DISPlay:PULSe:MEASview:DELeTe, 2-302
- :DISPlay:PULSe:MEASview:NEW, 2-302
- :DISPlay:PULSe:MEASview:SElect(?), 2-303
- :DISPlay:PULSe:RESult:ATX(?), 2-303
- :DISPlay:PULSe:RESult:AVERAge(?), 2-304
- :DISPlay:PULSe:RESult:DRooP(?), 2-304
- :DISPlay:PULSe:RESult:DUTPct(?), 2-305
- :DISPlay:PULSe:RESult:DUTRatio(?), 2-305
- :DISPlay:PULSe:RESult:FALL(?), 2-306
- :DISPlay:PULSe:RESult:FRDeviAtion(?), 2-306
- :DISPlay:PULSe:RESult:MFReqqerror(?), 2-307
- :DISPlay:PULSe:RESult:MPHerror(?), 2-307
- :DISPlay:PULSe:RESult:PHDeviAtion(?), 2-308
- :DISPlay:PULSe:RESult:PPFFreqency(?), 2-308
- :DISPlay:PULSe:RESult:PPoWer(?), 2-309
- :DISPlay:PULSe:RESult:PPPHase(?), 2-309
- :DISPlay:PULSe:RESult:RINTerval(?), 2-310
- :DISPlay:PULSe:RESult:RIPlle(?), 2-310
- :DISPlay:PULSe:RESult:RISE(?), 2-311
- :DISPlay:PULSe:RESult:RMSFreqerror(?), 2-311
- :DISPlay:PULSe:RESult:RMSPherror(?), 2-312
- :DISPlay:PULSe:RESult:RRAtE(?), 2-312
- :DISPlay:PULSe:RESult:TIME(?), 2-313
- :DISPlay:PULSe:RESult:WIDTh(?), 2-313
- :DISPlay:PULSe:SElect:NUMBer(?), 2-314
- :DISPlay:PULSe:SElect:RESult(?), 2-315
- :DISPlay:PULSe:STATistics:MARKer:SHOW:STATe(?), 2-316
- :DISPlay:PULSe:STATistics:PLOT(?), 2-317
- :DISPlay:PULSe:STATistics:WINDow:TRACe:GRATICule:GRID:STATe(?), 2-317
- :DISPlay:PULSe:STATistics:X:RSCale, 2-318
- :DISPlay:PULSe:STATistics:X[:SCALE]:NUMBer(?), 2-318
- :DISPlay:PULSe:STATistics:X[:SCALE]:OFFSet(?), 2-319
- :DISPlay:PULSe:STATistics:Y:RSCale, 2-319
- :DISPlay:PULSe:STATistics:Y[:SCALE]:FULL(?), 2-320
- :DISPlay:PULSe:STATistics:Y[:SCALE]:OFFSet(?), 2-321
- :DISPlay:PULSe:STATistics:Y[:SCALE]:STOP?, 2-322
- :DISPlay:PULSe:TRACe:MARKer:SHOW:STATe(?), 2-322
- :DISPlay:PULSe:TRACe:POINt:SHOW(?), 2-323
- :DISPlay:PULSe:TRACe:WINDow:TRACe:GRATICule:GRID:STATe(?), 2-323
- :DISPlay:PULSe:TRACe:X:RSCale, 2-324
- :DISPlay:PULSe:TRACe:X[:SCALE]:FULL(?), 2-325
- :DISPlay:PULSe:TRACe:X[:SCALE]:OFFSet(?), 2-326
- :DISPlay:PULSe:TRACe:X[:SCALE]:PDIVision(?), 2-326
- :DISPlay:PULSe:TRACe:X[:SCALE](?), 2-324
- :DISPlay:PULSe:TRACe:Y:RSCale, 2-327
- :DISPlay:PULSe:TRACe:Y[:SCALE]:FULL(?), 2-327
- :DISPlay:PULSe:TRACe:Y[:SCALE]:OFFSet(?), 2-328
- :DISPlay:PULSe:TRACe:Y[:SCALE]:STOP?, 2-328
- :DISPlay:SGRam subgroup, 2-329
- :DISPlay:SGRam:FREQuency:AUTO, 2-329
- :DISPlay:SGRam:FREQuency:OFFSet(?), 2-330
- :DISPlay:SGRam:FREQuency:SCALE(?), 2-330
- :DISPlay:SGRam:TIME:AUTO, 2-331
- :DISPlay:SGRam:TIME:OFFSet(?), 2-331
- :DISPlay:SGRam:TIME:OVERlap(?), 2-332
- :DISPlay:SGRam:TIME:SCALE(?), 2-332
- :DISPlay:SPECTrum subgroup, 2-333
- :DISPlay:SPECTrum:MARKer:NOISE:MODE(?), 2-334
- :DISPlay:SPECTrum:WINDow:TRACe:GRATICule:GRID:STATe(?), 2-335
- :DISPlay:SPECTrum:Y[:SCALE]:PDIVision(?), 2-335
- :DISPlay:SPURious subgroup, 2-336



:DISPlay:SPURious:MARKer:SHOW:STATe(?), 2-337  
 :DISPlay:SPURious:RESet:SCALe, 2-337  
 :DISPlay:SPURious:SElect:NUMBer(?), 2-338  
 :DISPlay:SPURious:SHOW:LIMit(?), 2-338  
 :DISPlay:SPURious:WINDow:TRACe:GRATICule  
   :GRID:STATe(?), 2-339  
 :DISPlay:SPURious:X[:SCALe]:AUTO, 2-339  
 :DISPlay:SPURious:X[:SCALe]:STARt(?), 2-340  
 :DISPlay:SPURious:X[:SCALe]:STOP(?), 2-340  
 :DISPlay:SPURious:Y[:SCALe]:AUTO, 2-341  
 :DISPlay:SPURious:Y[:SCALe]:OFFSet(?), 2-342  
 :DISPlay:SPURious:Y[:SCALe](?), 2-341  
 :DISPlay:TOVerview subgroup, 2-343  
 :DISPlay:TOVerview:WINDow:TRACe:GRATICule  
   :GRID:STATe(?), 2-343  
 :DISPlay:TOVerview:X[:SCALe]:AUTO, 2-345  
 :DISPlay:TOVerview:X[:SCALe]:OFFSet(?), 2-345  
 :DISPlay:TOVerview:X[:SCALe](?), 2-209, 2-344  
 :DISPlay:TOVerview:Y[:SCALe]:AUTO, 2-346  
 :DISPlay:TOVerview:Y[:SCALe]:OFFSet(?), 2-347  
 :DISPlay:TOVerview:Y[:SCALe]:RESCale, 2-347  
 :DISPlay:TOVerview:Y[:SCALe](?), 2-346  
 :DISPlay:WINDow:OPTimized:MEASurement?, 2-210  
 DPX spectrum, measurement mode, 2-12

## E

Error codes, 3-15  
   commands, 3-15  
   device specific, 3-18  
   execution, 3-16  
   hardware, 3-18, 3-19  
   query, 3-19  
   unique, 3-19  
 \*ESE(?), 2-64  
 \*ESR?, 2-65

## F

:FETCh basic command subgroup, 2-350  
 FETCh command group, 2-29  
 FETCh commands, 2-349  
 :FETCh:ACPower subgroup, 2-355  
 :FETCh:ACPower?, 2-356  
 :FETCh:ACPower:CHANnel:POWER?, 2-357  
 :FETCh:ACPower:SPECTrum?, 2-358  
 :FETCh:AVTime subgroup, 2-359  
 :FETCh:AVTime:AVERAge?, 2-359  
 :FETCh:AVTime:FIRSt|:SECond|:THIRd|:FOURth?,  
   2-360  
 :FETCh:AVTime:MAXimum?, 2-361  
 :FETCh:AVTime:MAXLocation?, 2-361

:FETCh:AVTime:MINimum?, 2-362  
 :FETCh:AVTime:MINLocation?, 2-362  
 :FETCh:AVTime:RESult?, 2-363  
 :FETCh:CCDF subgroup, 2-364  
 :FETCh:CCDF?, 2-365  
 :FETCh:CCDF:FIRSt|:SECond|:THIRd:X?, 2-366  
 :FETCh:CCDF:FIRSt|:SECond|:THIRd:XY?, 2-367,  
   2-538  
 :FETCh:CCDF:FIRSt|:SECond|:THIRd[:Y]?, 2-368  
 :FETCh:CONSte subgroup (Option 21 only), 2-369  
 :FETCh:CONSte:RESults?, 2-369  
 :FETCh:CONSte:TRACe?, 2-370  
 :FETCh:DDEMod subgroup, 2-371  
 :FETCh:DDEMod:STABle?, 2-371  
 :FETCh:DDEMod:SYNCh:WORD:LENGth?, 2-372  
 :FETCh:DDEMod:SYNCh:WORD:POSition?, 2-372  
 :FETCh:DPSA subgroup, 2-373  
 :FETCh:DPSA:TRACe:AVERAge?, 2-373  
 :FETCh:DPSA:TRACe:MATH?, 2-374  
 :FETCh:DPSA:TRACe:MAXimum?, 2-375  
 :FETCh:DPSA:TRACe:MINimum?, 2-376  
 :FETCh:EVM subgroup, 2-377  
 :FETCh:EVM:FERRor?, 2-377  
 :FETCh:EVM:PEAK?, 2-378  
 :FETCh:EVM:PINDex?, 2-378  
 :FETCh:EVM:RMS?, 2-379  
 :FETCh:EVM:TRACe?, 2-379  
 :FETCh:FVTime subgroup, 2-380  
 :FETCh:FVTime?, 2-380  
 :FETCh:FVTime:MAX?, 2-381  
 :FETCh:FVTime:MAXLocation?, 2-381  
 :FETCh:FVTime:MIN?, 2-382  
 :FETCh:FVTime:MINLocation?, 2-382  
 :FETCh:FVTime:RESult?, 2-383  
 :FETCh:IQVTime subgroup, 2-384  
 :FETCh:IQVTime:I?, 2-384  
 :FETCh:IQVTime:MAX?, 2-385  
 :FETCh:IQVTime:MAXLocation?, 2-385  
 :FETCh:IQVTime:MIN?, 2-386  
 :FETCh:IQVTime:MINLocation?, 2-386  
 :FETCh:IQVTime:Q?, 2-387  
 :FETCh:IQVTime:RESult?, 2-388  
 :FETCh:MCPower subgroup, 2-389  
 :FETCh:MCPower:ADJacent:CHANnels?, 2-390  
 :FETCh:MCPower:CHANnel:POWER?, 2-391  
 :FETCh:MCPower:MAIN:CHANnels?, 2-392  
 :FETCh:MCPower:SPECTrum?, 2-393  
 :FETCh:MERRor subgroup, 2-394  
 :FETCh:MERRor:FERRor?, 2-394  
 :FETCh:MERRor:PEAK?, 2-395  
 :FETCh:MERRor:PINDex?, 2-395  
 :FETCh:MERRor:RMS?, 2-396  
 :FETCh:MERRor:TRACe?, 2-396

- :FETCh:OBWidth subgroup, 2-397
- :FETCh:OBWidth:FREQuency:ERRor?, 2-398
- :FETCh:OBWidth:OBWidth:BA NDwidth?, 2-398
- :FETCh:OBWidth:OBWidth:LEFT:FREQuency?, 2-399
- :FETCh:OBWidth:OBWidth:LEFT:LEVel?, 2-399
- :FETCh:OBWidth:OBWidth:POWer?, 2-400
- :FETCh:OBWidth:OBWidth:RI GHt:FREQuency?, 2-401
- :FETCh:OBWidth:OBWidth:RI GHt:LEVel?, 2-401
- :FETCh:OBWidth:SPECTrum?, 2-402
- :FETCh:OBWidth:XDBBAndwidth:BA NDwidth?, 2-402
- :FETCh:OBWidth:XDBBAndwidth:LEFT:FREQuency?, 2-403
- :FETCh:OBWidth:XDBBAndwidth:LEFT:LEVel?, 2-403
- :FETCh:OBWidth:XDBBAndwidth:POWer?, 2-404
- :FETCh:OBWidth:XDBBAndwidth:RI GHt:FREQuency?, 2-405
- :FETCh:OBWidth:XDBBAndwidth:RI GHt:LEVel?, 2-405
- :FETCh:PERRor subgroup, 2-406
- :FETCh:PERRor:FERRor?, 2-406
- :FETCh:PERRor:PEAK?, 2-407
- :FETCh:PERRor:PINdex?, 2-407
- :FETCh:PERRor:RMS?, 2-408
- :FETCh:PERRor:TRACe?, 2-408
- :FETCh:PHVTime subgroup, 2-409
- :FETCh:PHVTime?, 2-409
- :FETCh:PHVTime:MAX?, 2-410
- :FETCh:PHVTime:MAXLocation?, 2-410
- :FETCh:PHVTime:MIN?, 2-411
- :FETCh:PHVTime:MINLocation?, 2-411
- :FETCh:PHVTime:RESult?, 2-412
- :FETCh:PNOise subgroup (Option 11 only), 2-413
- :FETCh:PNOise:ALL?, 2-414
- :FETCh:PNOise:CARRier:FERRor?, 2-415
- :FETCh:PNOise:CARRier:POWer?, 2-415
- :FETCh:PNOise:JITTer?, 2-416
- :FETCh:PNOise:RESidual:FM?, 2-416
- :FETCh:PNOise:RMS:PNOise?, 2-417
- :FETCh:PNOise:SPECTrum<x>:X?, 2-417
- :FETCh:PNOise:SPECTrum<x>:XY?, 2-418
- :FETCh:PNOise:SPECTrum<x>[:Y]?, 2-419
- :FETCh:PULSe subgroup (Option 20 only), 2-420
- :FETCh:PULSe:STATic:ATX?, 2-438
- :FETCh:PULSe:STATic:AVERAge?, 2-439
- :FETCh:PULSe:STATic:DROOp?, 2-440
- :FETCh:PULSe:STATic:DUTPet?, 2-441
- :FETCh:PULSe:STATic:DUTRatio?, 2-442
- :FETCh:PULSe:STATic:FALL?, 2-443
- :FETCh:PULSe:STATic:PPOWer?, 2-449
- :FETCh:PULSe:STATic:PPPHase?, 2-450
- :FETCh:PULSe:STATic:RINTerval?, 2-451
- :FETCh:PULSe:STATic:RI PPlE?, 2-452
- :FETCh:PULSe:STATic:RI SE?, 2-453
- :FETCh:PULSe:STATic:RRATe?, 2-456
- :FETCh:PULSe:STATic:WI DTh?, 2-457
- :FETCh:PULSe:STATistics?, 2-437
- :FETCh:PULSe:STATistics:FRDeViation?, 2-444
- :FETCh:PULSe:STATistics:MFReQerror?, 2-445
- :FETCh:PULSe:STATistics:MPHerror?, 2-446
- :FETCh:PULSe:STATistics:PHDeViation?, 2-447
- :FETCh:PULSe:STATistics:PPFRequency?, 2-448
- :FETCh:PULSe:STATistics:RMSFRerror?, 2-454
- :FETCh:PULSe:STATistics:RMSPherror?, 2-455
- :FETCh:PULSe:TRACe:X?, 2-458
- :FETCh:PULSe:TRACe:XY?, 2-459
- :FETCh:PULSe:TRACe:Y]?, 2-460
- :FETCh:PULSe[:RESult]:ATX?, 2-421
- :FETCh:PULSe[:RESult]:AVERAge?, 2-422
- :FETCh:PULSe[:RESult]:DROOp?, 2-423
- :FETCh:PULSe[:RESult]:DUTPet?, 2-423
- :FETCh:PULSe[:RESult]:DUTRatio?, 2-424
- :FETCh:PULSe[:RESult]:FALL?, 2-424
- :FETCh:PULSe[:RESult]:FRDeViation?, 2-425
- :FETCh:PULSe[:RESult]:MFReQerror?, 2-426
- :FETCh:PULSe[:RESult]:MPHerror?, 2-427
- :FETCh:PULSe[:RESult]:PHDeViation?, 2-428
- :FETCh:PULSe[:RESult]:PPFRequency?, 2-429
- :FETCh:PULSe[:RESult]:PPOWer?, 2-430
- :FETCh:PULSe[:RESult]:PPPHase?, 2-431
- :FETCh:PULSe[:RESult]:RINTerval?, 2-432
- :FETCh:PULSe[:RESult]:RI PPlE?, 2-432
- :FETCh:PULSe[:RESult]:RI SE?, 2-433
- :FETCh:PULSe[:RESult]:RMSFRerror?, 2-433
- :FETCh:PULSe[:RESult]:RMSPherror?, 2-434
- :FETCh:PULSe[:RESult]:RRATe?, 2-435
- :FETCh:PULSe[:RESult]:TIME?, 2-436
- :FETCh:PULSe[:RESult]:WI DTh?, 2-436
- :FETCh:RFIN:IQ?, 2-351
- :FETCh:RFIN:IQ:HEADer?, 2-352
- :FETCh:RFIN:IQ:SCALE?, 2-353
- :FETCh:RFIN:RECOrd:IDS?, 2-354
- :FETCh:SGRam subgroup, 2-461
- :FETCh:SGRam?, 2-461
- :FETCh:SPECTrum subgroup, 2-462
- :FETCh:SPECTrum:TRACe<x>?, 2-463
- :FETCh:SPURious subgroup, 2-464
- :FETCh:SPURious:CARRier:POWer?, 2-465
- :FETCh:SPURious:COUNt?, 2-465
- :FETCh:SPURious:PASS?, 2-466
- :FETCh:SPURious:SPECTrum:X?, 2-466

:FETCh:SPURious:SPEctrum:XY?, 2-467  
 :FETCh:SPURious:SPEctrum[:Y]?, 2-468  
 :FETCh:SPURious:SPUR<x>:AMPLitude:ABSolute?,  
 2-469  
 :FETCh:SPURious:SPUR<x>:AMPLitude:RELative?,  
 2-470  
 :FETCh:SPURious:SPUR<x>:FREQuency:ABSolute?,  
 2-471  
 :FETCh:SPURious:SPUR<x>:FREQuency:RELative?,  
 2-471  
 :FETCh:SPURious:SPUR<x>:LIMit:ABSolute?, 2-472  
 :FETCh:SPURious:SPUR<x>:LIMit:RELative?, 2-473  
 :FETCh:SPURious:SPUR<x>:LIMit:VIOLation?,  
 2-474  
 :FETCh:SPURious:SPUR<x>:RANGe?, 2-474  
 :FETCh:SQUAL subgroup, 2-475  
 :FETCh:SQUALity:FREQuency:ERRor?, 2-476  
 :FETCh:SQUALity:GAIN:IMBalance?, 2-476  
 :FETCh:SQUALity:ORIGin:OFFSet?, 2-477  
 :FETCh:SQUALity:PEAK:EVM?, 2-477  
 :FETCh:SQUALity:PEAK:EVM:DB?, 2-478  
 :FETCh:SQUALity:PEAK:EVM:LOCation?, 2-478  
 :FETCh:SQUALity:PEAK:MERRor?, 2-479  
 :FETCh:SQUALity:PEAK:MERRor:DB?, 2-479  
 :FETCh:SQUALity:PEAK:MERRor:LOCation?, 2-480  
 :FETCh:SQUALity:PEAK:PERRor?, 2-481  
 :FETCh:SQUALity:PEAK:PERRor:LOCation?, 2-481  
 :FETCh:SQUALity:QUADrature:ERRor?, 2-482  
 :FETCh:SQUALity:RHO?, 2-482  
 :FETCh:SQUALity:RMS:EVM?, 2-483  
 :FETCh:SQUALity:RMS:EVM:DB?, 2-483  
 :FETCh:SQUALity:RMS:MER:DB?, 2-484  
 :FETCh:SQUALity:RMS:MERRor?, 2-485  
 :FETCh:SQUALity:RMS:MERRor:DB?, 2-485  
 :FETCh:SQUALity:RMS:PERRor?, 2-486  
 :FETCh:TOVerView subgroup, 2-487  
 :FETCh:TOVerView?, 2-487  
 File, specifying, 2-495  
 Functional groups, 2-13

## G

### GPiB

- configurations, 1-4
- connection rules, 1-4
- interface specification, B-1
- setting GPIB address, 1-5

## H

Hierarchy tree, 2-2

## I

\*IDN?, 2-66  
 IEEE 488.2 common commands, 2-9  
 IEEE common command group, 2-14  
 IEEE common commands, 2-61  
 IEEE Std 488.2-1987, 2-1  
 Initialization settings, C-1  
 INITiate command group, 2-35  
 INITiate commands, 2-489  
 :INITiate:CONTinuous(?), 2-489  
 :INITiate[:IMMEDIATE], 2-490  
 INPut command group, 2-35  
 INPut commands, 2-491  
 :INPut:MLEVel|RLEVel(?), 2-494  
 :INPut[:RF]:ATTenuation:AUTO(?), 2-492  
 :INPut[:RF]:ATTenuation:MONitor:STATe(?), 2-493  
 :INPut[:RF]:ATTenuation(?), 2-492  
 :INPut[:RF]:GAIN:STATe(?), 2-494  
 Interface message, B-3

## M

Marker mnemonics, 2-76  
 MCPR, measurement mode, 2-12  
 Measurement views, description, 2-11  
 :MMEMory basic command subgroup, 2-496  
 MMEMory command group, 2-36  
 MMEMory commands, 2-495  
 :MMEMory:AVTime subgroup, 2-504  
 :MMEMory:AVTime:LOAD:TRACe<x>, 2-505  
 :MMEMory:AVTime:STORE:TRACe<x>, 2-505  
 :MMEMory:CALibration:LOAD:CORRection  
 :EXternal:EDIT<x>, 2-497  
 :MMEMory:CALibration:STORE:CORRection  
 :EXternal:EDIT<x>, 2-497  
 :MMEMory:CCDF subgroup, 2-506  
 :MMEMory:CCDF:LOAD:TRACe<x>, 2-507  
 :MMEMory:CCDF:STORE:TRACe<x>, 2-507  
 :MMEMory:DPSA Subgroup, 2-508  
 :MMEMory:DPSA:LOAD:TRACe<x>, 2-509  
 :MMEMory:DPSA:STORE:TRACe<x>, 2-509  
 :MMEMory:FVTime subgroup, 2-510  
 :MMEMory:FVTime:LOAD:TRACe, 2-511  
 :MMEMory:FVTime:STORE:TRACe, 2-511  
 :MMEMory:IQVTime subgroup, 2-512  
 :MMEMory:IQVTime:LOAD:TRACe:I, 2-513  
 :MMEMory:IQVTime:LOAD:TRACe:Q, 2-513  
 :MMEMory:IQVTime:STORE:TRACe:I, 2-514  
 :MMEMory:IQVTime:STORE:TRACe:Q, 2-514  
 :MMEMory:LOAD:IQ, 2-498

:MMEMory:LOAD:STATe, 2-498  
 :MMEMory:LOAD:TRACe, 2-499  
 :MMEMory:PHVTime subgroup, 2-515  
 :MMEMory:PHVTime:LOAD:TRACe, 2-516  
 :MMEMory:PHVTime:STORE:TRACe, 2-516  
 :MMEMory:SPECtrum subgroup, 2-517  
 :MMEMory:SPECtrum:LOAD:TRACe<x>, 2-518  
 :MMEMory:SPECtrum:STORE:TRACe<x>, 2-518  
 :MMEMory:SPURious subgroup, 2-519  
 :MMEMory:SPURious:LOAD:TABLE, 2-520  
 :MMEMory:SPURious:STORE:TABLE, 2-520  
 :MMEMory:STORE:IQ, 2-500  
 :MMEMory:STORE:IQ:CSV, 2-501  
 :MMEMory:STORE:IQ:MAT, 2-501  
 :MMEMory:STORE:MSTate, 2-502  
 :MMEMory:STORE:RESults, 2-502  
 :MMEMory:STORE:STATe, 2-503  
 :MMEMory:STORE:TRACe, 2-503  
 Mnemonics, constructed, 2-10

## O

\*OPC(?), 2-66  
 \*OPT?, 2-67  
 OUTPut command group, 2-37  
 OUTPut commands, 2-521  
 :OUTPut:IF:BANDwidth(?), 2-522  
 :OUTPut:IF:STATe(?), 2-522  
 :OUTPut:NOISe[:STATe](?), 2-523

## P

Parameter Types Used in Syntax Descriptions, 2-4

## Q

Queries, 2-3  
 Query Responses, 2-3  
 Queues  
   event, 3-10  
   output, 3-10  
 Quotes, 2-8

## R

READ command group, 2-38  
 READ commands, 2-525  
 :READ:ACPower subgroup, 2-526  
 :READ:ACPower?, 2-527  
 :READ:ACPower:CHANnel:POWer?, 2-528  
 :READ:ACPower:SPECtrum?, 2-529

:READ:AVTime subgroup, 2-530  
 :READ:AVTime:AVERAge?, 2-530  
 :READ:AVTime:FIRSt[:SECond]:THIRd[:FOURth?],  
   2-531  
 :READ:AVTime:MAXimum?, 2-532  
 :READ:AVTime:MAXLocation?, 2-532  
 :READ:AVTime:MINimum?, 2-533  
 :READ:AVTime:MINLocation?, 2-533  
 :READ:AVTime:RESult?, 2-534  
 :READ:CCDF subgroup, 2-535  
 :READ:CCDF?, 2-536  
 :READ:CCDF:FIRSt[:SECond]:THIRd:X?, 2-537  
 :READ:CCDF:FIRSt[:SECond]:THIRd[:Y]?, 2-539  
 :READ:CONSte subgroup (Option 21 only), 2-540  
 :READ:CONSte:RESults?, 2-540  
 :READ:CONSte:TRACe?, 2-541  
 :READ:DDEMod subgroup (Option 21 only), 2-542  
 :READ:DDEMod:STABLE?, 2-542  
 :READ:DPSA subgroup, 2-543  
 :READ:DPSA:TRACe:AVERAge?, 2-543  
 :READ:DPSA:TRACe:MATH?, 2-544  
 :READ:DPSA:TRACe:MAXimum?, 2-545  
 :READ:DPSA:TRACe:MINimum?, 2-546  
 :READ:EVM subgroup, 2-547  
 :READ:EVM:FERRor?, 2-547  
 :READ:EVM:PEAK?, 2-548  
 :READ:EVM:PINDEX?, 2-548  
 :READ:EVM:RMS?, 2-549  
 :READ:EVM:TRACe?, 2-549  
 :READ:FVTime subgroup, 2-550  
 :READ:FVTime?, 2-550  
 :READ:FVTime:MAX?, 2-551  
 :READ:FVTime:MAXLocation?, 2-551  
 :READ:FVTime:MIN?, 2-552  
 :READ:FVTime:MINLocation?, 2-552  
 :READ:FVTime:RESult?, 2-553  
 :READ:IQVTime subgroup, 2-554  
 :READ:IQVTime:I?, 2-554  
 :READ:IQVTime:MAX?, 2-555  
 :READ:IQVTime:MAXLocation?, 2-555  
 :READ:IQVTime:MIN?, 2-556  
 :READ:IQVTime:MINLocation?, 2-556  
 :READ:IQVTime:Q?, 2-557  
 :READ:IQVTime:RESult?, 2-558  
 :READ:MCPowe:MAIN:CHANnels?, 2-562  
 :READ:MCPower subgroup, 2-559  
 :READ:MCPower:ADJacent:CHANnels?, 2-560  
 :READ:MCPower:CHANnel:POWer?, 2-561  
 :READ:MCPower:SPECtrum?, 2-563  
 :READ:MERRor subgroup, 2-564  
 :READ:MERRor:FERRor?, 2-564  
 :READ:MERRor:PEAK?, 2-565  
 :READ:MERRor:PINDEX?, 2-565



- :READ:MERRor:RMS?, 2-566
- :READ:MERRor:TRACe?, 2-566
- :READ:OBWidth subgroup, 2-567
- :READ:OBWidth:FREQuency:ERRor?, 2-568
- :READ:OBWidth:OBWidth:BANDwidth?, 2-568
- :READ:OBWidth:OBWidth:LEFT:FREQuency?, 2-569
- :READ:OBWidth:OBWidth:LEFT:LEVel?, 2-569
- :READ:OBWidth:OBWidth:POWer?, 2-570
- :READ:OBWidth:OBWidth:RIGHT:FREQuency?, 2-571
- :READ:OBWidth:OBWidth:RIGHT:LEVel?, 2-571
- :READ:OBWidth:SPECtrum?, 2-572
- :READ:OBWidth:XDBBANDwidth:BANDwidth?, 2-572
- :READ:OBWidth:XDBBANDwidth:LEFT:FREQuency?, 2-573
- :READ:OBWidth:XDBBANDwidth:LEFT:LEVel?, 2-573
- :READ:OBWidth:XDBBANDwidth:POWer?, 2-574
- :READ:OBWidth:XDBBANDwidth:RIGHT:FREQuency?, 2-575
- :READ:OBWidth:XDBBANDwidth:RIGHT:LEVel?, 2-575
- :READ:PERRor subgroup, 2-576
- :READ:PERRor:FERRor?, 2-576
- :READ:PERRor:PEAK?, 2-577
- :READ:PERRor:PINDex?, 2-577
- :READ:PERRor:RMS?, 2-578
- :READ:PERRor:TRACe?, 2-578
- :READ:PHVTime subgroup, 2-579
- :READ:PHVTime?, 2-579
- :READ:PHVTime:MAX?, 2-580
- :READ:PHVTime:MAXLocation?, 2-580
- :READ:PHVTime:MIN?, 2-581
- :READ:PHVTime:MINLocation?, 2-581
- :READ:PHVTime:RESult?, 2-582
- :READ:PNOise subgroup (Option 11 only), 2-583
- :READ:PNOise:ALL?, 2-584
- :READ:PNOise:CARRier:FERRor?, 2-585
- :READ:PNOise:CARRier:POWer?, 2-585
- :READ:PNOise:JITTer?, 2-586
- :READ:PNOise:RESidual:FM?, 2-586
- :READ:PNOise:RMS:PNOise?, 2-587
- :READ:PNOise:SPECtrum<x>:X?, 2-587
- :READ:PNOise:SPECtrum<x>:XY?, 2-588
- :READ:PNOise:SPECtrum<x>[:Y]?, 2-589
- :READ:PULSe subgroup (Option 20 only), 2-590
- :READ:PULSe:STATic:ATX?, 2-608
- :READ:PULSe:STATic:AVERAge?, 2-609
- :READ:PULSe:STATic:DROOp?, 2-610
- :READ:PULSe:STATic:DUTPct?, 2-611
- :READ:PULSe:STATic:DUTRatio?, 2-612
- :READ:PULSe:STATic:FALL?, 2-613
- :READ:PULSe:STATic:PPOWer?, 2-619
- :READ:PULSe:STATic:PPHase?, 2-620
- :READ:PULSe:STATic:RINTerval?, 2-621
- :READ:PULSe:STATic:RIPple?, 2-622
- :READ:PULSe:STATic:RISE?, 2-623
- :READ:PULSe:STATic:RRATe?, 2-626
- :READ:PULSe:STATic:WIDTHh?, 2-627
- :READ:PULSe:STATistics?, 2-607
- :READ:PULSe:STATistics:FRDeviation?, 2-614
- :READ:PULSe:STATistics:MFRreqerror?, 2-615
- :READ:PULSe:STATistics:MPHerror?, 2-616
- :READ:PULSe:STATistics:PHDeviation?, 2-617
- :READ:PULSe:STATistics:PPFRequency?, 2-618
- :READ:PULSe:STATistics:RMSFrequeror?, 2-624
- :READ:PULSe:STATistics:RMSPherror?, 2-625
- :READ:PULSe:TRACe:X?, 2-628
- :READ:PULSe:TRACe:XY?, 2-629
- :READ:PULSe:TRACe[:Y]?, 2-630
- :READ:PULSe[:RESult]:ATX?, 2-591
- :READ:PULSe[:RESult]:AVERAge?, 2-592
- :READ:PULSe[:RESult]:DROOp?, 2-593
- :READ:PULSe[:RESult]:DUTPct?, 2-593
- :READ:PULSe[:RESult]:DUTRatio?, 2-594
- :READ:PULSe[:RESult]:FALL?, 2-594
- :READ:PULSe[:RESult]:FRDeviation?, 2-595
- :READ:PULSe[:RESult]:MFRreqerror?, 2-596
- :READ:PULSe[:RESult]:MPHerror?, 2-597
- :READ:PULSe[:RESult]:PHDeviation?, 2-598
- :READ:PULSe[:RESult]:PPFRequency?, 2-599
- :READ:PULSe[:RESult]:PPOWer?, 2-600
- :READ:PULSe[:RESult]:PPPHase?, 2-601
- :READ:PULSe[:RESult]:RINTerval?, 2-602
- :READ:PULSe[:RESult]:RIPple?, 2-602
- :READ:PULSe[:RESult]:RISE?, 2-603
- :READ:PULSe[:RESult]:RMSFrequeror?, 2-603
- :READ:PULSe[:RESult]:RMSPherror?, 2-604
- :READ:PULSe[:RESult]:RRATe?, 2-605
- :READ:PULSe[:RESult]:TIME?, 2-606
- :READ:PULSe[:RESult]:WIDTHh?, 2-606
- :READ:SGRam subgroup, 2-631
- :READ:SGRam?, 2-631
- :READ:SPECtrum subgroup, 2-632
- :READ:SPECtrum:TRACe<x>?, 2-633
- :READ:SPURious subgroup, 2-634
- :READ:SPURious:CARRier:POWer?, 2-635
- :READ:SPURious:COUNt?, 2-635
- :READ:SPURious:PASS?, 2-636
- :READ:SPURious:SPECtrum:X?, 2-636
- :READ:SPURious:SPECtrum:XY?, 2-637
- :READ:SPURious:SPECtrum[:Y]?, 2-638

- :READ:SPURious:SPUR<x>:AMPLitude:ABSolute?, 2-639
  - :READ:SPURious:SPUR<x>:AMPLitude:RELative?, 2-640
  - :READ:SPURious:SPUR<x>:FREQuency:ABSolute?, 2-641
  - :READ:SPURious:SPUR<x>:FREQuency:RELative?, 2-641
  - :READ:SPURious:SPUR<x>:LIMit:ABSolute?, 2-642
  - :READ:SPURious:SPUR<x>:LIMit:RELative?, 2-643
  - :READ:SPURious:SPUR<x>:LIMit:VIOLation?, 2-644
  - :READ:SPURious:SPUR<x>:RANGe?, 2-644
  - :READ:SQUAL subgroup, 2-645
  - :READ:SQUALity:FREQuency:ERRor, 2-646
  - :READ:SQUALity:GAIN:IMBalance?, 2-646
  - :READ:SQUALity:ORIGin:OFFSet?, 2-647
  - :READ:SQUALity:PEAK:EVM?, 2-647
  - :READ:SQUALity:PEAK:EVM:DB?, 2-648
  - :READ:SQUALity:PEAK:EVM:LOCation?, 2-648
  - :READ:SQUALity:PEAK:MERRor?, 2-649
  - :READ:SQUALity:PEAK:MERRor:DB?, 2-649
  - :READ:SQUALity:PEAK:MERRor:LOCation?, 2-650
  - :READ:SQUALity:PEAK:PERRor?, 2-651
  - :READ:SQUALity:PEAK:PERRor:LOCation?, 2-651
  - :READ:SQUALity:QUADrature:ERRor?, 2-652
  - :READ:SQUALity:RHO?, 2-652
  - :READ:SQUALity:RMS:EVM?, 2-653
  - :READ:SQUALity:RMS:EVM:DB?, 2-653
  - :READ:SQUALity:RMS:MERRor:DB?, 2-654
  - :READ:SQUALity:RMS:MERRor?, 2-655
  - :READ:SQUALity:RMS:MERRor:DB?, 2-655
  - :READ:SQUALity:RMS:PERRor?, 2-656
  - :READ:TOverview subgroup, 2-657
  - :READ:TOverview?, 2-657
  - Registers, Service Request Enable Register (SRER), 3-5
  - Retrieving response message, 2-979
  - \*RST, 2-68
  - Rules
    - command forming, 2-1
    - for using SCPI commands, 2-8
- ## S
- SCPI
    - abbreviating, 2-5
    - chaining commands, 2-6
    - commands, 2-2
    - conformance information, D-1
    - general rules, 2-8
    - parameter types, 2-4
    - subsystem hierarchy tree, 2-2
  - SCPI commands and queries syntax, 2-2
    - creating commands, 2-3
    - creating queries, 2-3
  - SENSE command group, 2-44
  - SENSE commands, 2-659
    - [:SENSe]:ACPower subgroup, 2-680
    - [:SENSe]:ACPower:AVERage:COUNt(?), 2-681
    - [:SENSe]:ACPower:AVERage(?), 2-681
    - [:SENSe]:ACPower:BANDwidth|:BWiDth:ViDeo:STATe(?), 2-684
    - [:SENSe]:ACPower:BANDwidth|:BWiDth:ViDeo(?), 2-684
    - [:SENSe]:ACPower:BANDwidth|:BWiDth[:RESolution]:ACTual?, 2-682
    - [:SENSe]:ACPower:BANDwidth|:BWiDth[:RESolution]:AUTO(?), 2-683
    - [:SENSe]:ACPower:BANDwidth|:BWiDth[:RESolution](?), 2-682
    - [:SENSe]:ACPower:CHANnel:BANDwidth|:BWiDth(?), 2-685
    - [:SENSe]:ACPower:CHANnel:FiLTeR(?), 2-685
    - [:SENSe]:ACPower:CHANnel:PAIRs(?), 2-686
    - [:SENSe]:ACPower:CHANnel:SPACing(?), 2-686
    - [:SENSe]:ACPower:CHiPrate(?), 2-687
    - [:SENSe]:ACPower:CLear:RESults, 2-687
    - [:SENSe]:ACPower:FREQuency:STEP:AUTO(?), 2-689
    - [:SENSe]:ACPower:FREQuency:STEP(?), 2-689
    - [:SENSe]:ACPower:FREQuency(?), 2-688
    - [:SENSe]:ACPower:NFLoor:STATe(?), 2-690
    - [:SENSe]:ACPower:OPTimize:SPAN(?), 2-690
    - [:SENSe]:ACPower:RRCRolloff(?), 2-691
    - [:SENSe]:ACQuisition:BANDwidth|:BWiDth(?), 2-662
    - [:SENSe]:ACQuisition:FFRame:ACTual?, 2-662
    - [:SENSe]:ACQuisition:FFRame:LIMit(?), 2-663
    - [:SENSe]:ACQuisition:FFRame:STATe(?), 2-663
    - [:SENSe]:ACQuisition:MEMory:AVAILable:SAMPles?, 2-664
    - [:SENSe]:ACQuisition:MEMory:CAPacity[:TiME]?, 2-664
    - [:SENSe]:ACQuisition:MEMory:USED[:PERCent]?, 2-665
    - [:SENSe]:ACQuisition:MODE(?), 2-665
    - [:SENSe]:ACQuisition:SAMPles(?), 2-666
    - [:SENSe]:ACQuisition:SEConds(?), 2-666
    - [:SENSe]:ANALysis subgroup, 2-660
    - [:SENSe]:ANALysis:ADVanced:DITHer:HWARe:STATus?, 2-668
    - [:SENSe]:ANALysis:ADVanced:DITHer(?), 2-667
    - [:SENSe]:ANALysis:LENGth:ACTual?, 2-670

- [:SENSe]:ANALysis:LENGth:AUTO(?), 2-670
- [:SENSe]:ANALysis:LENGth(?), 2-669
- [:SENSe]:ANALysis:REFErence(?), 2-671
- [:SENSe]:ANALysis:STARt:AUTO(?), 2-672
- [:SENSe]:ANALysis:STARt(?), 2-671
- [:SENSe]:AVTime subgroup, 2-692
- [:SENSe]:AVTime:BANDwidth|:BWIDth:ACTual?, 2-693
- [:SENSe]:AVTime:BANDwith|:BWIDth(?), 2-693
- [:SENSe]:AVTime:CLEar:RESults, 2-694
- [:SENSe]:AVTime:MAXTracepoints(?), 2-694
- [:SENSe]:AVTime:METHod(?), 2-695
- [:SENSe]:AVTime:SPAN(?), 2-696
- [:SENSe]:CCDF subgroup, 2-697
- [:SENSe]:CCDF:BANDwidth|:BWIDth(?), 2-697
- [:SENSe]:CCDF:CLEar, 2-698
- [:SENSe]:CCDF:TIME:AUTO(?), 2-699
- [:SENSe]:CCDF:TIME:LENGth(?), 2-698
- [:SENSe]:DDEMod subgroup, 2-700
- [:SENSe]:DDEMod:ANALysis:LENGth:ACTual?, 2-702
- [:SENSe]:DDEMod:ANALysis:LENGth:AUTO(?), 2-702
- [:SENSe]:DDEMod:ANALysis:LENGth(?), 2-701
- [:SENSe]:DDEMod:BURSt:DETECT(?), 2-703
- [:SENSe]:DDEMod:BURSt:THREShold(?), 2-704
- [:SENSe]:DDEMod:CARRier:OFFSet(?), 2-704
- [:SENSe]:DDEMod:FILTer:ALPHa(?), 2-705
- [:SENSe]:DDEMod:FILTer:MEASurement(?), 2-706
- [:SENSe]:DDEMod:FILTer:REFErence(?), 2-707
- [:SENSe]:DDEMod:MAGNitude:NORMAlize(?), 2-707
- [:SENSe]:DDEMod:MODulation:TYPE(?), 2-708
- [:SENSe]:DDEMod:SRATE(?), 2-709
- [:SENSe]:DDEMod:SWAP:IQ(?), 2-709
- [:SENSe]:DDEMod:SYMBol:POINts(?), 2-710
- [:SENSe]:DDEMod:SYNCh:WORD:SYMBol(?), 2-712
- [:SENSe]:DDEMod:SYNCh:WORD(?), 2-711
- [:SENSe]:DDEMod:TIME:UNITs(?), 2-712
- [:SENSe]:DPSA subgroup, 2-713
- [:SENSe]:DPSA:AUDio:DEMod:GAIN(?), 2-714
- [:SENSe]:DPSA:AUDio:DEMod:RXBWidth(?), 2-714
- [:SENSe]:DPSA:AUDio:DEMod:RXFRequency?, 2-715
- [:SENSe]:DPSA:AUDio:DEMod:STATe(?), 2-715
- [:SENSe]:DPSA:AUDio:DEMod:TUNE(?), 2-716
- [:SENSe]:DPSA:AUDio:DEMod:TYPE(?), 2-717
- [:SENSe]:DPSA:BANDwidth|:BWIDth[:RESolution]:AUTO(?), 2-718
- [:SENSe]:DPSA:BANDwidth|:BWIDth[:RESolution](?), 2-717
- [:SENSe]:DPSA:CLEar:RESults, 2-718
- [:SENSe]:DPSA:COLor:MAXimum(?), 2-720
- [:SENSe]:DPSA:COLor:MINimum(?), 2-720
- [:SENSe]:DPSA:COLor(?), 2-719
- [:SENSe]:DPSA:FREQuency:CENTer(?), 2-721
- [:SENSe]:DPSA:FREQuency:SPAN(?), 2-721
- [:SENSe]:DPSA:FREQuency:STARt(?), 2-722
- [:SENSe]:DPSA:FREQuency:STEP:AUTO(?), 2-723
- [:SENSe]:DPSA:FREQuency:STEP(?), 2-723
- [:SENSe]:DPSA:FREQuency:STOP(?), 2-724
- [:SENSe]:FVTime subgroup, 2-725
- [:SENSe]:FVTime:CLEar:RESults, 2-725
- [:SENSe]:FVTime:FREQuency:CENTer(?), 2-726
- [:SENSe]:FVTime:FREQuency:SPAN(?), 2-727
- [:SENSe]:FVTime:FREQuency:STARt(?), 2-727
- [:SENSe]:FVTime:FREQuency:STEP:AUTO(?), 2-728
- [:SENSe]:FVTime:FREQuency:STEP(?), 2-728
- [:SENSe]:FVTime:FREQuency:STOP(?), 2-729
- [:SENSe]:FVTime:MAXTracepoints(?), 2-730
- [:SENSe]:IQVTime subgroup, 2-731
- [:SENSe]:IQVTime:CLEar:RESults, 2-731
- [:SENSe]:IQVTime:FREQuency:CENTer(?), 2-732
- [:SENSe]:IQVTime:FREQuency:SPAN(?), 2-733
- [:SENSe]:IQVTime:FREQuency:STARt(?), 2-733
- [:SENSe]:IQVTime:FREQuency:STEP:AUTO(?), 2-734
- [:SENSe]:IQVTime:FREQuency:STEP(?), 2-734
- [:SENSe]:IQVTime:FREQuency:STOP(?), 2-735
- [:SENSe]:IQVTime:MAXTracepoints(?), 2-736
- [:SENSe]:MCPower subgroup, 2-737
- [:SENSe]:MCPower:AVERage:COUNT(?), 2-738
- [:SENSe]:MCPower:AVERage(?), 2-738
- [:SENSe]:MCPower:BANDwidth|:BWIDth:VIDeo:STATe(?), 2-741
- [:SENSe]:MCPower:BANDwidth|:BWIDth:VIDeo(?), 2-741
- [:SENSe]:MCPower:BANDwidth|:BWIDth[:RESolution]:ACTual?, 2-739
- [:SENSe]:MCPower:BANDwidth|:BWIDth[:RESolution]:AUTO(?), 2-740
- [:SENSe]:MCPower:BANDwidth|:BWIDth[:RESolution](?), 2-739
- [:SENSe]:MCPower:CHANnel:ADJacent:ADD, 2-742
- [:SENSe]:MCPower:CHANnel:ADJacent:DELeTe, 2-742
- [:SENSe]:MCPower:CHANnel:FILTer(?), 2-743
- [:SENSe]:MCPower:CHANnel:MAIN:BANDwidth|:BWIDth(?), 2-743
- [:SENSe]:MCPower:CHANnel:MAIN:COUNT(?), 2-744
- [:SENSe]:MCPower:CHANnel:MAIN:INACTive(?), 2-744

- [:SENSe]:MCPower:CHANnel:MAIN:SPACing(?),  
2-745
- [:SENSe]:MCPower:CHIPrate(?), 2-745
- [:SENSe]:MCPower:CLear:RESults, 2-746
- [:SENSe]:MCPower:FREQuency:STEP:AUTO(?),  
2-747
- [:SENSe]:MCPower:FREQuency:STEP(?), 2-747
- [:SENSe]:MCPower:FREQuency(?), 2-746
- [:SENSe]:MCPower:NFLoor:STATe(?), 2-748
- [:SENSe]:MCPower:OPTimize(?), 2-748
- [:SENSe]:MCPower:RCHannels?, 2-749
- [:SENSe]:MCPower:RCHannels:MAIN<x>, 2-749
- [:SENSe]:MCPower:RCHannels:TOTal, 2-750
- [:SENSe]:MCPower:RRCRolloff(?), 2-750
- [:SENSe]:MEASurement:FREQuency(?), 2-672
- [:SENSe]:OBWidth subgroup, 2-751
- [:SENSe]:OBWidth:AVERage:COUNt(?), 2-752
- [:SENSe]:OBWidth:AVERage(?), 2-752
- [:SENSe]:OBWidth:BANDwidth|:BWIDth  
:MEASurement(?), 2-753
- [:SENSe]:OBWidth:BANDwidth|:BWIDth:VIDeo  
:STATe(?), 2-755
- [:SENSe]:OBWidth:BANDwidth|:BWIDth:VIDeo(?),  
2-755
- [:SENSe]:OBWidth:BANDwidth|:BWIDth  
[:RESolution]:ACTual?, 2-754
- [:SENSe]:OBWidth:BANDwidth|:BWIDth  
[:RESolution]:AUTO(?), 2-754
- [:SENSe]:OBWidth:BANDwidth|:BWIDth  
[:RESolution](?), 2-753
- [:SENSe]:OBWidth:CLear:RESults, 2-756
- [:SENSe]:OBWidth:FREQuency:CENTer(?), 2-756
- [:SENSe]:OBWidth:FREQuency:STEP:AUTO(?),  
2-757
- [:SENSe]:OBWidth:FREQuency:STEP(?), 2-757
- [:SENSe]:OBWidth:PERCent(?), 2-758
- [:SENSe]:OBWidth:XDBLevel(?), 2-758
- [:SENSe]:PHVTime subgroup, 2-759
- [:SENSe]:PHVTime:CLear:RESults, 2-759
- [:SENSe]:PHVTime:FREQuency:CENTer(?), 2-760
- [:SENSe]:PHVTime:FREQuency:SPAN(?), 2-761
- [:SENSe]:PHVTime:FREQuency:START(?), 2-761
- [:SENSe]:PHVTime:FREQuency:STEP:AUTO(?),  
2-762
- [:SENSe]:PHVTime:FREQuency:STEP(?), 2-762
- [:SENSe]:PHVTime:FREQuency:STOP(?), 2-763
- [:SENSe]:PHVTime:MAXTracepoints(?), 2-764
- [:SENSe]:PNOise subgroup (Option 11 only), 2-765
- [:SENSe]:PNOise:AVERage:COUNt(?), 2-766
- [:SENSe]:PNOise:AVERage:ENABle(?), 2-766
- [:SENSe]:PNOise:CARRier:FREQuency:TRACk(?),  
2-767
- [:SENSe]:PNOise:CARRier:THReshold(?), 2-767
- [:SENSe]:PNOise:CLear:RESults, 2-768
- [:SENSe]:PNOise:FREQuency:INTegration:OFFSet  
:START(?), 2-769
- [:SENSe]:PNOise:FREQuency:INTegration:OFFSet  
:STOP(?), 2-769
- [:SENSe]:PNOise:FREQuency:PLOT:OFFSet:START  
(?), 2-770
- [:SENSe]:PNOise:FREQuency:PLOT:OFFSet:STOP  
(?), 2-770
- [:SENSe]:PNOise:OPTimization(?), 2-771
- [:SENSe]:POWER:UNITs(?), 2-673
- [:SENSe]:PULSe subgroup (Option 20 only), 2-772
- [:SENSe]:PULSe:ANALyze:LEVel:FIFTy(?), 2-773
- [:SENSe]:PULSe:ANALyze:LEVel:HUNDred(?),  
2-774
- [:SENSe]:PULSe:ANALyze:LEVel(?), 2-773
- [:SENSe]:PULSe:ANALyze:MEASurement:TIME  
:AUTO(?), 2-775
- [:SENSe]:PULSe:ANALyze:MEASurement:TIME  
:START(?), 2-776
- [:SENSe]:PULSe:ANALyze:MEASurement:TIME  
:STOP(?), 2-776
- [:SENSe]:PULSe:ANALyze:PMLocation(?), 2-777
- [:SENSe]:PULSe:ANALyze:POINT:LOCation(?),  
2-777
- [:SENSe]:PULSe:ANALyze:RFALI(?), 2-778
- [:SENSe]:PULSe:ANALyze:RIPple(?), 2-778
- [:SENSe]:PULSe:CARRier:OFFSet(?), 2-779
- [:SENSe]:PULSe:CARRier:SEARch(?), 2-779
- [:SENSe]:PULSe:DETECT:MEASurement(?), 2-780
- [:SENSe]:PULSe:DETECT:NUMBER(?), 2-781
- [:SENSe]:PULSe:DETECT:POWER[:THReshold](?),  
2-782
- [:SENSe]:PULSe:DETECT:TIME[:THReshold](?),  
2-782
- [:SENSe]:PULSe:FILTer:BANDwidth|:BWIDth(?),  
2-783
- [:SENSe]:PULSe:FILTer:MEASurement(?), 2-784
- [:SENSe]:PULSe:FREFerence:AUTO(?), 2-785
- [:SENSe]:PULSe:FREFerence:CHIRpbw(?), 2-786
- [:SENSe]:PULSe:FREFerence:OFFSet(?), 2-786
- [:SENSe]:PULSe:MODulation:TYPE(?), 2-787
- [:SENSe]:PULSe:SIGNAL:TYPE(?), 2-787
- [:SENSe]:REANalyze, 2-674
- [:SENSe]:ROSCillator:SOURce(?), 2-675
- [:SENSe]:SGRam subgroup, 2-788
- [:SENSe]:SGRam:BANDwidth|:BWIDth:OPTimization  
(?), 2-789
- [:SENSe]:SGRam:BANDwidth|:BWIDth:RESolution  
(?), 2-790



- [:SENSe]:SGRam:BANDwidth|:BWIDth:VIDeo:STATe(?), 2-793
- [:SENSe]:SGRam:BANDwidth|:BWIDth:VIDeo(?), 2-793
- [:SENSe]:SGRam:BANDwidth|:BWIDth[:RESolution]:ACTual?, 2-790
- [:SENSe]:SGRam:BANDwidth|:BWIDth[:RESolution]:AUTO(?), 2-791
- [:SENSe]:SGRam:BANDwidth|:BWIDth[:RESolution]:MODE(?), 2-792
- [:SENSe]:SGRam:COLor:MAX(?), 2-795
- [:SENSe]:SGRam:COLor:MIN(?), 2-795
- [:SENSe]:SGRam:COLor(?), 2-794
- [:SENSe]:SGRam:FFT:WINDow(?), 2-796
- [:SENSe]:SGRam:FILTer[:SHAPE](?), 2-797
- [:SENSe]:SGRam:FREQuency:CENTer(?), 2-798
- [:SENSe]:SGRam:FREQuency:SPAN:BANDwidth[:RESolution]:RATIo(?), 2-799
- [:SENSe]:SGRam:FREQuency:SPAN:MAXimum, 2-800
- [:SENSe]:SGRam:FREQuency:SPAN(?), 2-799
- [:SENSe]:SGRam:FREQuency:STARt(?), 2-800
- [:SENSe]:SGRam:FREQuency:STEP:AUTO(?), 2-801
- [:SENSe]:SGRam:FREQuency:STEP(?), 2-801
- [:SENSe]:SGRam:FREQuency:STOP(?), 2-802
- [:SENSe]:SPECtrum subgroup, 2-803
- [:SENSe]:SPECtrum:BANDwidth|:BWIDth:OPTimization(?), 2-804
- [:SENSe]:SPECtrum:BANDwidth|:BWIDth:VIDeo:STATe(?), 2-808
- [:SENSe]:SPECtrum:BANDwidth|:BWIDth:VIDeo(?), 2-808
- [:SENSe]:SPECtrum:BANDwidth|:BWIDth[:RESolution]:ACTual?, 2-805
- [:SENSe]:SPECtrum:BANDwidth|:BWIDth[:RESolution]:AUTO(?), 2-806
- [:SENSe]:SPECtrum:BANDwidth|:BWIDth[:RESolution]:MODE(?), 2-807
- [:SENSe]:SPECtrum:BANDwidth|:BWIDth[:RESolution](?), 2-805
- [:SENSe]:SPECtrum:CLear:RESults, 2-809
- [:SENSe]:SPECtrum:FFT:WINDow(?), 2-810
- [:SENSe]:SPECtrum:FILTer[:SHAPE](?), 2-811
- [:SENSe]:SPECtrum:FREQuency:CENTer(?), 2-812
- [:SENSe]:SPECtrum:FREQuency:SPAN:BANDwidth[:RESolution]:RATIo(?), 2-813
- [:SENSe]:SPECtrum:FREQuency:SPAN(?), 2-813
- [:SENSe]:SPECtrum:FREQuency:STARt(?), 2-814
- [:SENSe]:SPECtrum:FREQuency:STEP:AUTO(?), 2-815
- [:SENSe]:SPECtrum:FREQuency:STEP(?), 2-815
- [:SENSe]:SPECtrum:FREQuency:STOP(?), 2-816
- [:SENSe]:SPECtrum:LENGth:ACTual?, 2-677
- [:SENSe]:SPECtrum:LENGth:AUTO(?), 2-677
- [:SENSe]:SPECtrum:LENGth(?), 2-676
- [:SENSe]:SPECtrum:MAX:SPAN, 2-817
- [:SENSe]:SPECtrum:POINts:COUNt(?), 2-817
- [:SENSe]:SPECtrum:STARt(?), 2-678
- [:SENSe]:SPECtrum:TIME:MODE(?), 2-679
- [:SENSe]:SPURious subgroup, 2-818
- [:SENSe]:SPURious:CARRier:BANDwidth|:BWIDth:INTEgration(?), 2-820
- [:SENSe]:SPURious:CARRier:BANDwidth|:BWIDth(?), 2-820
- [:SENSe]:SPURious:CARRier:BANDwidth|:BWIDth[:RESolution]:AUTO(?), 2-822
- [:SENSe]:SPURious:CARRier:BANDwidth|:BWIDth[:RESolution](?), 2-821
- [:SENSe]:SPURious:CARRier:DETEction(?), 2-823
- [:SENSe]:SPURious:CARRier:FREQuency(?), 2-823
- [:SENSe]:SPURious:CARRier:THReshold(?), 2-824
- [:SENSe]:SPURious:CLear:RESults, 2-824
- [:SENSe]:SPURious:LIST(?), 2-826
- [:SENSe]:SPURious:MODE(?), 2-827
- [:SENSe]:SPURious:OPTimization(?), 2-827
- [:SENSe]:SPURious:POINts:COUNt(?), 2-828
- [:SENSe]:SPURious:RANGe<x>:BANDwidth:VIDeo:STATe(?), 2-829
- [:SENSe]:SPURious:RANGe<x>:BANDwidth:VIDeo(?), 2-829
- [:SENSe]:SPURious:RANGe<x>:DETEction(?), 2-830
- [:SENSe]:SPURious:RANGe<x>:EXCURsion(?), 2-831
- [:SENSe]:SPURious:RANGe<x>:FILTer[:SHAPE]:BANDwidth:AUTO(?), 2-834
- [:SENSe]:SPURious:RANGe<x>:FILTer[:SHAPE]:BANDwidth(?), 2-833
- [:SENSe]:SPURious:RANGe<x>:FILTer[:SHAPE](?), 2-832
- [:SENSe]:SPURious:RANGe<x>:FREQuency:STARt(?), 2-835
- [:SENSe]:SPURious:RANGe<x>:FREQuency:STOP(?), 2-836
- [:SENSe]:SPURious:RANGe<x>:LIMit:ABSolute:STARt(?), 2-837
- [:SENSe]:SPURious:RANGe<x>:LIMit:ABSolute:STOP(?), 2-838
- [:SENSe]:SPURious:RANGe<x>:LIMit:MASK(?), 2-839
- [:SENSe]:SPURious:RANGe<x>:LIMit:RELative:STARt(?), 2-840
- [:SENSe]:SPURious:RANGe<x>:LIMit:RELative:STOP(?), 2-841
- [:SENSe]:SPURious:RANGe<x>:STATe(?), 2-842
- [:SENSe]:SPURious:RANGe<x>:THReshold(?), 2-843

- [:SENSe]:SPURious:REfERENCE:MANual:POWer(?),  
2-845
  - [:SENSe]:SPURious:REfERENCE(?), 2-844
  - [:SENSe]:SPURious[:FREQuency]:OVerlap?, 2-825
  - [:SENSe]:TOVerview subgroup, 2-846
  - [:SENSe]:TOVerview:FREQuency:CENTer(?), 2-847
  - [:SENSe]:TOVerview:MAXTracepoints(?), 2-847
  - [:SENSe]:USETtings, 2-679
  - SI prefix and unit, 2-7
  - Special characters, 2-5
  - Specifying the file, 2-495
  - \*SRE(?), 2-69
  - STATus command group, 2-53
  - STATus commands, 2-849
  - :STATus:ACPower:EVENTs?, 2-851
  - :STATus:AVTime:EVENTs?, 2-852
  - :STATus:CCDF:EVENTs?, 2-853
  - :STATus:CONSte:EVENTs?, 2-854
  - :STATus:DPSA:EVENTs?, 2-855
  - :STATus:EVM:EVENTs?, 2-856
  - :STATus:FVTime:EVENTs?, 2-857
  - :STATus:IQVTime:EVENTs?, 2-858
  - :STATus:MCPower:EVENTs?, 2-859
  - :STATus:MERRor:EVENTs?, 2-860
  - :STATus:OBWidth:EVENTs?, 2-861
  - :STATus:OPERation:CONDition?, 2-862
  - :STATus:OPERation:ENABle(?), 2-862
  - :STATus:OPERation:NTRansition(?), 2-863
  - :STATus:OPERation:PTRansition(?), 2-864
  - :STATus:OPERation[:EVENT]?, 2-863
  - :STATus:PERRor:EVENTs?, 2-865
  - :STATus:PHVTime:EVENTs?, 2-866
  - :STATus:PNOise:EVENTs?, 2-867
  - :STATus:PRESet, 2-867
  - :STATus:PULSe:RESult:EVENTs?, 2-868
  - :STATus:PULSe:STATistics:EVENTs?, 2-869
  - :STATus:PULSe:TRACe:EVENTs?, 2-870
  - :STATus:QUEStionable:CALibration:CONDition?,  
2-873
  - :STATus:QUEStionable:CALibration:ENABle(?),  
2-874
  - :STATus:QUEStionable:CALibration:NTRansition(?),  
2-876
  - :STATus:QUEStionable:CALibration:PTRansition(?),  
2-877
  - :STATus:QUEStionable:CALibration[:EVENT]?, 2-875
  - :STATus:QUEStionable:CONDition?, 2-871
  - :STATus:QUEStionable:ENABle(?), 2-871
  - :STATus:QUEStionable:FREQuency:CONDition?,  
2-877
  - :STATus:QUEStionable:FREQuency:ENABle(?),  
2-878
  - :STATus:QUEStionable:FREQuency:NTRansition(?),  
2-880
  - :STATus:QUEStionable:FREQuency:PTRansition(?),  
2-881
  - :STATus:QUEStionable:FREQuency[:EVENT]?, 2-879
  - :STATus:QUEStionable:NTRansition(?), 2-872
  - :STATus:QUEStionable:PTRansition(?), 2-873
  - :STATus:QUEStionable[:EVENT]?, 2-872
  - :STATus:SGRAM:EVENTs?, 2-882
  - :STATus:SPECTrum:EVENTs?, 2-883
  - :STATus:SPURious:EVENTs?, 2-884
  - :STATus:SQUALity:EVENTs?, 2-885
  - \*STB?, 2-70
  - Syntax, command, 2-1
  - SYSTem command group, 2-54
  - SYSTem commands, 2-887
  - :SYSTem:COMMunicate:GPIB[:SELF]:ADDReSS(?),  
2-888
  - :SYSTem:DATE(?), 2-888
  - :SYSTem:ERRor:ALL?, 2-889
  - :SYSTem:ERRor:CODE:ALL?, 2-890
  - :SYSTem:ERRor:CODE[:NEXT]?, 2-891
  - :SYSTem:ERRor:COUNt?, 2-891
  - :SYSTem:ERRor[:NEXT]?, 2-892
  - :SYSTem:OPTions?, 2-893
  - :SYSTem:PRESet, 2-893
  - :SYSTem:TIME(?), 2-894
  - :SYSTem:VERSion?, 2-894
- ## T
- TekVISA, 1-6
  - TRACe command group, 2-55
  - TRACe commands, 2-895
  - :TRACe:CONSte subgroup, 2-911
  - :TRACe:CONSte:MODE(?), 2-911
  - :TRACe:FVTime subgroup, 2-922
  - :TRACe:FVTime:AVErage:COUNt(?), 2-923
  - :TRACe:FVTime:FREeze(?), 2-925
  - :TRACe:FVTime:FUNcTION(?), 2-925
  - :TRACe:FVTime(?), 2-922
  - :TRACe:IQVTime subgroup, 2-926
  - :TRACe:IQVTime:AVErage:COUNt(?), 2-927
  - :TRACe:IQVTime:COUNt:ENABle(?), 2-928
  - :TRACe:IQVTime:COUNt:RESet, 2-928
  - :TRACe:IQVTime:COUNt(?), 2-927
  - :TRACe:IQVTime:ENABle:I(?), 2-929
  - :TRACe:IQVTime:ENABle:Q(?), 2-929
  - :TRACe:IQVTime:FREeze(?), 2-930
  - :TRACe:IQVTime:FUNcTION(?), 2-930
  - :TRACe:IQVTime:SELect:I(?), 2-931

- :TRACe:IQVTime:SElect:Q(?), 2-932
- :TRACe:OBW subgroup, 2-933
- :TRACe:OBW:MAXHold(?), 2-933
- :TRACe:PHVTime subgroup, 2-934
- :TRACe:PHVTime:COUNt:ENABle(?), 2-936
- :TRACe:PHVTime:COUNt:RESet, 2-936
- :TRACe:PHVTime:COUNt(?), 2-935
- :TRACe:PHVTime:FREeze(?), 2-937
- :TRACe:PHVTime:FUNcTION(?), 2-937
- :TRACe:PVTime:AVERAge:COUNt(?), 2-935
- :TRACe:PVTime(?), 2-934
- :TRACe:SGRAM subgroup, 2-942
- :TRACe:SGRam:DETEction(?), 2-943
- :TRACe:SGRam:FREeze(?), 2-944
- :TRACe:SGRam:FUNcTION:TIME(?), 2-945
- :TRACe:SGRam:FUNcTION(?), 2-944
- :TRACe:SGRam:SElect:LINE(?), 2-945
- :TRACe:SPURious subgroup, 2-958
- :TRACe:SPURious:COUNt:ENABle(?), 2-959
- :TRACe:SPURious:COUNt:RESet, 2-959
- :TRACe:SPURious:COUNt(?), 2-958
- :TRACe:SPURious:FREeze(?), 2-960
- :TRACe:SPURious:FUNcTION(?), 2-960
- :TRACe<x>:AVTime, 2-896
- :TRACe<x>:AVTime:AVERAge:COUNt(?), 2-897
- :TRACe<x>:AVTime:AVERAge:RESet, 2-898
- :TRACe<x>:AVTime:COUNt:ENABle(?), 2-900
- :TRACe<x>:AVTime:COUNt:RESet, 2-901
- :TRACe<x>:AVTime:COUNt(?), 2-899
- :TRACe<x>:AVTime:FREeze(?), 2-901
- :TRACe<x>:AVTime:FUNcTION(?), 2-902
- :TRACe<x>:AVTime:LEFToperand(?), 2-903
- :TRACe<x>:AVTime:RIGHtooperand(?), 2-904
- :TRACe<x>:AVTime:SElect(?), 2-905
- :TRACe<x>:AVTime(?), 2-897
- :TRACe<x>:CCDF subgroup, 2-906
- :TRACe<x>:CCDF:FREeze(?), 2-906
- :TRACe<x>:CCDF:SElect(?), 2-907
- :TRACe<x>:CCDF:SHOW(?), 2-908
- :TRACe<x>:CCDF:X(?), 2-909
- :TRACe<x>:CCDF:Y?, 2-910
- :TRACe<x>:DPSA subgroup, 2-912
- :TRACe<x>:DPSA:AVERAge:COUNt(?), 2-913
- :TRACe<x>:DPSA:COLor:INTensity(?), 2-914
- :TRACe<x>:DPSA:DOT:PERsistent:TYPE(?), 2-915
- :TRACe<x>:DPSA:DOT:PERsistent:VARiable(?), 2-916
- :TRACe<x>:DPSA:DOT:PERsistent(?), 2-914
- :TRACe<x>:DPSA:FREeze(?), 2-917
- :TRACe<x>:DPSA:FUNcTION(?), 2-918
- :TRACe<x>:DPSA:LEFToperand(?), 2-919
- :TRACe<x>:DPSA:RIGHtooperand(?), 2-920
- :TRACe<x>:DPSA:SElect(?), 2-921
- :TRACe<x>:DPSA(?), 2-913
- :TRACe<x>:FVTime:COUNt:ENABle(?), 2-924
- :TRACe<x>:FVTime:COUNt:RESet, 2-924
- :TRACe<x>:FVTime:COUNt(?), 2-923
- :TRACe<x>:PNOise subgroup (Option 11 only), 2-938
- :TRACe<x>:PNOise:DETEction(?), 2-938
- :TRACe<x>:PNOise:FREeze(?), 2-939
- :TRACe<x>:PNOise:SElect(?), 2-939
- :TRACe<x>:PNOise:SHOW(?), 2-940
- :TRACe<x>:PNOise:SMOothing:COUNt(?), 2-940
- :TRACe<x>:PNOise:SMOothing:ENABle(?), 2-941
- :TRACe<x>:PNOise:SMOothing:RESet, 2-941
- :TRACe<x>:SPECtrum subgroup, 2-946
- :TRACe<x>:SPECtrum:AVERAge:COUNt(?), 2-948
- :TRACe<x>:SPECtrum:AVERAge:RESet, 2-948
- :TRACe<x>:SPECtrum:COUNt:ENABle(?), 2-950, 2-951
- :TRACe<x>:SPECtrum:COUNt(?), 2-949
- :TRACe<x>:SPECtrum:DETEction(?), 2-952
- :TRACe<x>:SPECtrum:FREeze(?), 2-953
- :TRACe<x>:SPECtrum:FUNcTION(?), 2-954
- :TRACe<x>:SPECtrum:LEFToperand(?), 2-955
- :TRACe<x>:SPECtrum:RIGHtooperand(?), 2-956
- :TRACe<x>:SPECtrum:SElect(?), 2-957
- :TRACe<x>:SPECtrum(?), 2-947
- \*TRG, 2-71
- TRIGger command group, 2-58
- TRIGger commands, 2-961
- :TRIGger:MASK:NEW, 2-963
- :TRIGger:MASK:NEW:AUTO, 2-964
- :TRIGger:MASK:OPEN, 2-965
- :TRIGger:MASK:SAVE, 2-965
- :TRIGger[:SEquence]:ADVanced:SWEEp:MODE(?), 2-966
- :TRIGger[:SEquence]:EVENT:EXTFront:IMPedance(?), 2-966
- :TRIGger[:SEquence]:EVENT:EXTFront:LEVel(?), 2-967
- :TRIGger[:SEquence]:EVENT:EXTFront:SLOPe(?), 2-967
- :TRIGger[:SEquence]:EVENT:EXTRear:SLOPe(?), 2-968
- :TRIGger[:SEquence]:EVENT:GATed(?), 2-968
- :TRIGger[:SEquence]:EVENT:INPut:FMASK:VIOLation(?), 2-969
- :TRIGger[:SEquence]:EVENT:INPut:LEVel(?), 2-970
- :TRIGger[:SEquence]:EVENT:INPut:SLOPe(?), 2-970
- :TRIGger[:SEquence]:EVENT:INPut:TDBWidth:ACTual?, 2-971
- :TRIGger[:SEquence]:EVENT:INPut:TDBWidth:STATe(?), 2-972

:TRIGger[:SEquence]:EVENT:INPut:TDBWidth(?),  
2-971  
:TRIGger[:SEquence]:EVENT:INPut:TYPE(?), 2-972  
:TRIGger[:SEquence]:EVENT:SOURce(?), 2-973  
:TRIGger[:SEquence]:FORCed(?), 2-974  
:TRIGger[:SEquence]:IMMediate, 2-974  
:TRIGger[:SEquence]:STATus(?), 2-975  
:TRIGger[:SEquence]:TIME:DELay(?), 2-976  
:TRIGger[:SEquence]:TIME:POSition(?), 2-976

## U

Unit and SI prefix, 2-7  
UNIT command group, 2-59  
UNIT commands, 2-977  
:UNIT:POWer(?), 2-978

## W

\*WAI, 2-71