Agilent 86140B Series Optical Spectrum Analyzer

Programming Guide

Agilent Technologies

Notices

Agilent Technologies Deutschland GmbH
Herrenberger Str. 130
71034 Böblingen
Germany

Manual Part Number

86140-90069

Edition

Second edition, January 2005

First edition, January 2002

Warranty

This Agilent Technologies instrument product is warranted against defects in material and workmanship for a period of one year from date of shipment. During the warranty period, Agilent will, at its option, either repair or replace products that prove to be defective.
For warranty service or repair, this product must be returned to a service facility designated by Agilent. Buyer shall prepay shipping charges to Agilent and Agilent shall pay shipping charges to return the product to Buyer. However, Buyer shall pay all shipping charges, duties, and taxes for products returned to Agilent from another country.

Agilent warrants that its software and firmware designated by Agilent for use with an instrument will execute its programming instructions when properly installed on that instrument. Agilent does not warrant that the operation of the instrument, software, or firmware will be uninterrupted or error free.

Limitation of Warranty

The foregoing warranty shall not apply to defects resulting from improper or inadequate maintenance by Buyer, Buyer-supplied software or interfacing, unauthorized modification or misuse, operation outside of the environmental specifications for the product, or improper site preparation or maintenance.
No other warranty is expressed or implied. Agilent Technologies specifically disclaims the implied warranties of Merchantability and Fitness for a Particular Purpose.

Exclusive Remedies

The remedies provided herein are Buyer's sole and exclusive remedies. Agilent Technologies shall not be liable for any direct, indirect, special, incidental, or consequential damages whether based on contract, tort, or any other legal theory.

Assistance

Product maintenance agreements and other customer assistance agreements are available for Agilent Technologies products. For any assistance contact your nearest Agilent Technologies Sales and Service Office.

Certification

Agilent Technologies Inc. certifies that this product met its published specifications at the time of shipment from the factory.
Agilent Technologies further certifies that its calibration measurements are traceable to the United States National Institute of Standards and Technology, NIST (formerly the United States National Bureau of Standards, NBS) to the extent allowed by the Institutes's calibration facility, and to the calibration facilities of other International Standards Organization members.

ISO 9001 Certification

Produced to ISO 9001 international quality system standard as part of our objective of continually increasing customer satisfaction through improved process control.

Safety Notices

CAUTION

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

WARNING

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

Warnings and Notices

WARNING

To avoid the possibility of injury or death, you must observe the fol lowing precautions before switching on the instrument.
Insert the power cable plug only into a socket outlet provided with a protective earth contact. Do not negate this protective action by the using an extension cord without a protective conductor.

WARNING

Never look directly into the end of a fiber or a connector, unless you are absolutely certain that there is no signal in the fiber.

Agilent Technologies Sales and Service Offices

For more information about Agilent Technologies test and measurement products, applications, services, and for a current sales office listing, viesit our web site:

http://www.agilent.com/comms/lightwave

You can also contact one of the following centers and ask for a test and measurement sales representative.

United States:	1 800 452 4844 1 888 900 8921 (FAX)
Canada:	1 877 894 4414 (905) 282 6495 (FAX)
Europe:	(31 20) 547 2111 (43 20) 547 2190 (FAX)
Japan:	0120 421 345 0120 421 678 (FAX)
Mexico	(52 55) 5081 9469 (52 55) 5081 9467 (FAX)
Australia:	1 800 629 485 1 800 142 134 (FAX)
Asia-Pacific:	(852) 3197 7889 (852) 2506 9233 (FAX)
Brazil	(55 11) 4197 3600 (55 11) 4197 3800 (FAX)

Table of Contents

1
Remote Operation

General Safety Considerations

This product has been designed and tested in accordance with the standards listed on the Manufacturer's Declaration of Conformity, and has been supplied in a safe condition. The documentation contains information and warnings that must be followed by the user to ensure safe operation and to maintain the product in a safe condition.

Install the instrument according to the enclosure protection provided.
This instrument does not protect against the ingress of water.
This instrument protects against finger access to hazardous parts within the enclosure.

Safety Symbols

CAUTION

The *caution* sign denotes a hazard. It calls attention to a procedure which, if not correctly performed or adhered to, could result in damage to or destruction of the product. Do not proceed beyond a caution sign until the indicated conditions are fully understood and met.

WARNING

The *warning* sign denotes a hazard. It calls attention to a procedure which, if not correctly performed or adhered to, could result in injury or loss of life. Do not proceed beyond a warning sign until the indicated conditions are fully understood and met.

WARNING **If this product is not used as specified, the protection provided by the equipment could be impaired. This product must be used in a normal condition (in which all means for protection are intact) only.**

WARNING **No operator serviceable parts inside. Refer servicing to qualified service personnel. To prevent electrical shock do not remove covers.**

WARNING **This is a Safety Class 1 Product (provided with a protective earthing ground incorporated in the power cord). The mains plug shall only be inserted in a socket outlet provided with a protective earth contact. Any interruption of the protective conductor inside or outside of the instrument is likely to make the instrument dangerous. Intentional interruption is prohibited.**

WARNING **To prevent electrical shock, disconnect the instrument from mains before cleaning. Use a dry cloth or one slightly dampened with water to clean the external case parts. Do not attempt to clean internally.**

CAUTION Fiber-optic connectors are easily damaged when connected to dirty or damaged cables and accessories. The Agilent 86140B series's front-panel INPUT connector is no exception. When you use improper cleaning and handling techniques, you risk expensive instrument repairs, damaged cables, and compromised measurements. Before you connect any fiber-optic cable to the Agilent 86140B series, refer to "Cleaning Connectors for Accurate Measurements" in the usesr's guide.

CAUTION This product is designed for use in Installation Category II and Pollution Degree 2 per IEC 61010-1C and 664 respectively.

CAUTION Do not use too much liquid in cleaning the Optical Spectrum Analyzer. Water can enter the front-panel keyboard, damaging sensitive electronic components.

CAUTION VENTILATION REQUIREMENTS: When installing the product in a cabinet, the convection into and out of the product must not be restricted. The ambient temperature (outside the cabinet) must be less than the maximum operating temperature of the product by 4 C for every 100 watts dissipated in the cabinet. If the total power dissipated in the cabinet is greater than 800 watts, then forced convection must be used.

CAUTION Install the instrument so that the detachable power cord is readily identifiable and is easily reached by the operator. The detachable power cord is the instrument disconnecting device. It disconnects the mains circuit from the mains supply before other parts of the instrument. The front panel switch is only a standby switch and is not a LINE switch. Alternatively, an externally installed switch or circuit breaker (which is readily identifiable and is easily reached by the operator) may be used as a disconnecting device.

CAUTION Always use the three-prong AC power cord supplied with this instrument. Failure to ensure adequate earth grounding by not using this cord may cause instrument damage.

CAUTION This instrument has autoranging line voltage input. Be sure the supply voltage is within the specified range.

CAUTION The Agilent 86140B, 86141B, and 86142B Option 004/005 EELED sources contain an IEC Class 1 LED, according to IEC 60825.

CAUTION Use of controls or adjustment or performance of procedures other than those specified herein may result in hazardous radiation exposure.

Summary for Experienced GPIB Programmers

This section is intended for people who are familiar with GPIB instrument programming. Terms like EOI, command synchronization and SCPI should be familiar to you. If not, review the following sections in this chapter. This section is a summary to get your OSA application operating quickly, focusing on ways in which the OSA differs from other instruments you may have encountered.

Auto Align

The optical spectrum analyzer contains optics that can drift out of alignment with environmental variations such as air pressure, temperature and humidity. To compensate for this, the Auto Align routine should be run when the instrument has been turned on, warmed up and stabilized, and whenever the temperature changes more than 1 degree centigrade. Since temperature is difficult to monitor, the rule of thumb is to auto align every 8 hours for normal applications, and once an hour for applications that require high amplitude stability or accuracy.

To perform the Auto Align routine, connect a stable input signal and then send a CALIBRATE:ALIGN command. The preferred method is to position marker 1 on the desired signal and use the marker align command CALIBRATE:ALIGN:MARKER 1. Using the marker allows you to select the appropriate signal.

EOI required

Before sending remote commands to the instrument, set your GPIB interface to assert the GPIB END message (also called EOI) with the last character of the transmission. There is a separate GPIB signal line dedicated to EOI, and when the OSA sees the EOI, it closes the buffer and begins processing the commands. The OSA will not accept any more commands until it has parsed and launched (but not necessarily completed) all the commands in the buffer.

Most interfaces use EOI by default. If, for example, you are using a National Instruments GPIB card with Visual C or Visual Basic, EOI is probably on. Similarly, if you are using an Agilent GPIB card in a HP-UX workstation, programming in C or C++, EOI is probably on. If you are using Basic for Windows, you will need to enable EOI each time you open a session to the instrument. This is done using the ASSIGN @OSA key words, and associating the END keyword with the session.

Turn the input buffer on

To provide compatibility with earlier firmware revisions, the input buffer is disabled in the OSA default state. In order to get the faster and more regular behavior, turn the buffer on each time you preset the instrument using:

```
sendCommand SYST:COMM:GPIB:BUFF ON
```

Note that front panel preset or remote *RST will reset the OSA to default conditions and also turns the input buffer off. You may also wish to turn the buffer off by sending:

```
sendCommand SYST:COMM:GPIB:BUFF OFF
```

Use device clear

The OSA uses the Selected Device Clear interface message to free the instrument from error conditions. Clearing the instrument at the start of any control application will avoid confusing the instrument. You may find it useful to write a small program that does nothing but clear the instrument, so you can clear it when you need to as you debug your application.

Note that Selected Device Clear is not a text message like *CLS. It is an interface level command. For VISA, use viClear(). For NI 488.2, use ibclr(). For Agilent SICL, use iclear(). For Basic for windows, use CLEAR 723 or CLEAR @Osa.

Use single sweep mode

When using remote programming, you will want the instrument to take a sweep only after it is set up and you want the sweep to occur. Allowing the instrument to sweep while you are sending commands significantly slows the instrument down. In general, unless an operator is tuning a device, it is best to turn continuous sweep off with:

```
INITiate:CONTinuous OFF
```

The *RST command automatically turns continuous sweep off.

Command synchronization

The OSA may start executing the next command before finishing the previous one. This is common among GPIB controlled instruments - it allows the instrument's multi-tasking operating system to execute commands as quickly as possible. Because of this, you will need to use *OPC, *OPC? or *WAI to find out when a sweep is finished before attempting to read data from the instrument. Most commonly, you will send an INITiate command to take a single sweep, then wait for the sweep

to finish before executing the next command. You also need to wait for any command that starts with "AUTO", such as automeasure or autoalign, and for any commands that perform a calibration.

The recommended technique for waiting is to use the query form of *OPC (*OPC?). This way, after initiating the sweep, you can do other things while waiting for the instrument. When you do wait, make sure your GPIB time-out is longer than what the instrument needs to finish the command. A typical sequence would be:

```
SENDMESSAGE INIT;*OPC?          // take a sweep
                                //perform other tasks while waiting
RECEIVEINTEGER                  // wait for "1" indicating end of sweep
```

Other alternatives involve polling the instrument for status, and that in turn slows the instrument down by making it respond to the queries. If you must poll for status, it is recommended that you send *ESE 1 to enable the OPC bit to set bit 5 of the main status register, then use serial poll to query the main status register.

Serial poll is a GPIB interface level operation, not a command like *STB?. It is accessed using special commands. For VISA, use VIREADSTB. For NI 488.2, use IBRSP. For SICL, use IREADSTB(). For Basic for Windows, use SPOLL 723 or SPOLL @OSA.

Once the OPC bit is set, you will need to manually clear the status register. The easiest way to do this is to send *CLS. You can also clear it by reading the register with *ESR?

If you don't need an explicit response from the instrument, the *WAI command will cause the instrument to finish processing all prior commands before continuing.

Maximum query rate

If you are waiting for a bit in a status byte, avoid writing a tight loop that continuously queries the instrument. This causes the instrument firmware to spend an excessive amount of time responding to your queries rather than running the instrument. Do not query the instrument more than once every 5 ms. For Visual Basic and Visual C, use the SLEEP command to put 5 ms delay in your query loop. For posix compliant platforms, use the NANOSLEEP() function to delay the loop. For Basic for Windows, use WAIT 0.005.

Watching for error messages

Whenever an error occurs, it appears briefly on the instrument display and is added to the error queue. Whenever there are messages in the error queue, bit 2 of the status byte is set. The recommended practice is to use serial poll to read the status byte and then use a bitwise 'and' against the

value 4 to see if the bit is set. If it is set, you should use SYSTEM:ERROR? to read all the error messages until you get a response that begins with "+0", indicating no further errors.

As you are developing your application, you should check the error status every time you read from or write to the instrument.

Most commonly used commands

The most commonly used commands are listed here. For commands that are listed twice, the first listing is the short form, the second is the long form. Most SCPI commands have a long form that is easier to understand but takes longer to send. The long form is listed only for your understanding and for cross referencing elsewhere in the manual. In practice, you should use the short form.

Table 1 Most commonly used GPIB commands

Command	Usage
SELECTED DEVICE CLEAR	This is not a text command - it is a GPIB hardware message you should use to start off your application.
*RST	The second thing you should do in your application to return the instrument to a known state and set single sweep mode.
SYST:COMM:GPIB:BUFF ON SYSTEM:COMM:GPIB:BUFFER ON | OFF	Turn the GPIB input buffer on and allow parallel command execution.
CENT 1550 NM [SENSE:][WAVELENGTH:]CENTER	Program the center wavelength, in nm.
SPAN, START, STOP	Other wavelength oriented commands.
BAND 2 NM [SENSE:]BANDWIDTH:[RESOLUTION]	Set the resolution bandwidth to 2 nm.
POW:RANG:LOW -45 DBM [SENSE:]POWER:[DC:]RANGE:LOWER	Tell the instrument the lowest power you expect to see, causing the instrument to adjust its sensitivity. Impacts sweep time.
POW:RANG:AUTO OFF [SENSE:]POWER:[DC:]RANGE:AUTO ON | OFF	Turn off auto-ranging - the automatic changing of gain range during a sweep. Use POW:RANG:LOW and DISP:TRAC:Y:RLEV to select the minimum and maximum powers expected.
DISP:TRAC:Y:RLEV-10 DBM DISPLAY:[WINDOW[1]:]TRACE:Y[1,2]:[SCALE:]RLEVEL	Set the display reference level - the maximum power expected to display.
INIT INITIATE:[IMMEDIATE]	Take a sweep.
*OPC?	Instrument returns a "1" when all GPIB commands received prior to the *OPC? are complete.
*WAI	Instrument stops processing new GPIB commands until all GPIB commands receiver prior to *WAI are complete.

Table 1 Most commonly used GPIB commands

Command	Usage
CALC:MARK1:MAX CALCULATE:MARKER[1|2|3|4]:MAXIMUM	Positions marker 1 to the maximum amplitude point.
CALC:MARK1:X? CALCULATE:MARKER[1|2|3|4]:X?	Read back marker 1 wavelength.
CALC:MARK1:Y? CALCULATE:MARKER[1|2|3|4]:Y?	Read back marker 1 amplitude.
CALC:MARK1:FUNC:NOISE ON CALCULATE:MARKER[1|2|3|4]:FUNCTION:NOISE:[STATE:] ON|OFF	Turn noise marker on for marker 1.
CALC:MARK1:FUNC:NOIS:RES? CALCULATE:MARKER[1|2|3|4]:FUNCTION:NOISE:RESULT?	Read back noise marker value, noise power normalized to 1 nm (or 0.1 nm if specified with noise marker bandwidth function).
TRAC? TRA TRACE:[DATA:][Y]? TRA|TRB|TRC|TRD|TRE|TRF	Transfer entire trace, according to format set with the FORM command.
*IDN?	Return string describing model number, serial number and software version.
SYST:ERR?	Read error at top of queue.
DISP OFF DISPLAY:[WINDOW[1]] ON|OFF	Turn the display off, without affecting the instrument state. Takes about 12 seconds.

Working with Older OSA Applications

This user's guide applies to OSA's with firmware revision B.04.00 and later. If you have an application written for OSA's with earlier firmware, your application will still run. The instrument defaults to a backward-compatible mode. There will be a small speed improvement (about 10% for typical applications) and you may notice that there is more error checking.

You can upgrade the firmware to the latest version by obtaining the upgrade from http://www.agilent.com/comms/osaupgrade. See "Firmware Upgrade" on page 3-28 in the User's Guide for instructions on how to install the upgrade.

To take advantage of the new capabilities, you need to:

- Turn the GPIB input buffer on after every *RST by sending SYSTEM:COMM:GPIB:BUFFER ON.
- Use *OPC? whenever you take a sweep, do an automeasure, do an autoalign, or perform a calibration. You can also use *OPC and *WAI.
- Closely monitor error messages and error conditions.

The older firmware has no input buffer, and all commands are executed serially. That is, the first command is parsed and fully executed before the characters for the next command are accepted from GPIB. This has the benefit of not needing *OPC to synchronize a sweep, but holds up the GPIB interface, making it very difficult to get status information when the instrument is busy.

The new firmware has an optional software input buffer. If you turn it on using SYSTEM:COMM:GPIB:BUFFER ON, the instrument will handshake in characters as fast as possible until the END (also known as EOI) message is received. At that point, the instrument will parse the buffer and launch the commands. Once the commands have begun execution, the instrument will go back and look for more incoming commands, even though the prior commands may not have finished executing. To upgrade an application to take full advantage of the new firmware you will need to use *OPC, *OPC? or *WAI to synchronize sweeps.

For example, if you initiate a sweep and then immediately read a marker value, the OSA may return the marker value before the new sweep has finished. You will have no way of knowing whether the marker data came from the previous sweep or the new sweep.

To avoid this problem, use *OPC?. This causes the instrument to return a "1" when all commands received prior to the *OPC? have completed.

```
sendMessage INITiate;*OPC?
receiveInteger
```

Or, use *WAI inside the message to force synchronization:

```
sendMessage INITiate;*WAI;:CALCulate1:MARKer1:MAXimum
```

This command takes a sweep and finds the highest peak once the sweep is complete.

If you are writing an application that works with older and newer OSA firmware, avoid turning the input buffer on. Turning on the input buffer places the responsibility on the programmer to control command execution order.

Introduction to Controlling an OSA

One of the easiest ways to learn how to write programs to control the instrument is to look at simple examples. In "Example Programs" on page 43, you'll find several useful example programs. Although they are written using the BASIC language, you can easily convert them to the language that you are using. The Agilent 86140B series' GPIB address is configured at the factory to a value of 23. You must set the output and input functions of your programming language to send the commands to this address. Pressing the green PRESET key does not change the GPIB address.

To change the GPIB address

1 Press the front-panel SYSTEM key.

2 Press the MORE SYSTEM FUNCTIONS.... softkey.

3 Press the REMOTE SETUP.... softkey, and change the GPIB address.

Remote mode and front-panel lockout

Whenever the instrument is in Remote mode, the RMT message is displayed on the instrument's screen and all keys are disabled except for the front-panel LOCAL key. This key can be pressed by the user to restore front-panel control of the instrument.

You can specify a local lockout mode that deactivates the front-panel LOCAL key. If the instrument is in local lockout mode, all the front-panel keys are disabled.

Consult the documentation for your programming environment to determine which commands are used to put an instrument in the remote and local lockout modes. These are not Agilent 86140B series commands; they are GPIB interface control level commands.

Remote command buffering

The OSA has an optional GPIB input buffer. With this buffer turned off, (the default state), the OSA accepts command data via GPIB, testing each byte. Once a complete command is received and interpreted, the GPIB handshake is held until the command operation is completed. Once completed, the next command byte is read by the instrument. If several commands are included in a single output statement, the computer will not be able to complete the controller output operation until the OSA has executed all of the commands. This process can hold the GPIB interface, or program control, past a timeout cycle.

Using the Input Buffer

If the command :SYSTEM:COMM:GPIB:BUFFER ON is sent, the 4 kbyte input buffer is activated.

With the presence of the software buffer:

- An entire GPIB message up to 4 kbyte can be read in a single operation.
- The commands within the buffer are executed faster.
- The instrument does not necessarily block other commands until the buffer is finished, allowing status query during long operations such as sweeps, calibrations, auto-align and auto-measure.

When the OSA recognizes that GPIB commands are available on the bus, it reads all the available characters and releases the bus while it processes the commands. You can then talk to other instruments or use serial poll to query the status of the OSA. It is possible to send additional commands to the OSA, and the OSA will respond to new commands while it is still processing the original commands. It is recommended that you use *OPC and serial poll to determine when the instrument has finished processing the commands.

EOI (End or Identify) is required to separate command strings. EOI is part of the command structure suggested by IEEE 488.2 and SCPI programming standards. You must use the proper command terminator to verify that the end of message command is sent with EOI enabled. Check the definition of the command terminator in your programming language. Without EOI present, the buffer will be waiting for EOI to begin execution and a pause will be generated while the OSA buffer is waiting for execution instructions.

It is important to remember that *RST, or front panel reset, disables the input buffer operation.

Synchronization Commands

If you turn the input buffer on, the OSA executes commands faster and is easier to synchronize with the controller. However, the presence of the buffer means that the OSA will not necessarily execute the commands in the order they are received. That is, it will not necessarily wait until one command completes before starting the next one. Two synchronization commands are provided, *OPC (operation complete) and *WAI (wait), that allow you to tell the instrument when to execute as fast as possible and when to stop and wait for a command to complete. In particular, it is important to wait for a sweep to complete before reading markers or trace data.

*OPC Command

The more frequently used command, operation complete, or *OPC, has two command forms:

- *OPC tells the instrument to set the OPC bit (bit zero of extended status register) when all the commands received prior to OPC have completed.
- *OPC? tells the instrument to respond with a "1" when all the commands receiver prior to OPC have completed. The status bit is not set in this case.

*OPC works for any command, as in taking sweeps, doing calibrations, auto measure, and so on. Of the two command forms, Agilent recommends that you use *OPC? in most cases. Using Visual Basic, a typical transaction may be:

```
SENDCOMMAND INST, ":SENSE:START 1320 NM;:SENSE:STOP
1325 NM;:INITIATE:IMMEDIATE; *OPC?"
VALUE = READINTREPLY(INST)
SENDCOMMAND INST,
":CALCULATE:MARKER1:MAXIMUM;:CALCULATE:MARKER1:X?"
VALUE = READDOUBLEREPLY(INST)
```

where:

- SENDCOMMAND is a routine in your favorite language using your favorite interface to send a command to the OSA.
- INST is a handle to the OSA appropriate to the interface you are using.
- READINTREPLY is a routine in your favorite language using your favorite interface to read a reply from the OSA and convert it to an integer.
- READDOUBLEREPLY is a routine in your favorite language using your favorite interface to read a reply from the OSA and convert it to a double precision real number.

In the above example, the OSA is set up for a measurement, a sweep is initiated, then the OSA pauses until the sweep is completed. Once the sweep is complete, a marker is set to the maximum peak and then the marker wavelength is read. If *OPC? was not used to force the instrument to wait, the OSA would have started the sweep, and then moved the marker to the maximum peak on the partially completed sweep. In general, it is necessary to use OPC synchronization before any command that moves markers or reads data back into the controller.

Status byte command form

If you wish to use the status byte form of the command, the most efficient approach is to enable the extended status byte mask to reflect the OPC bit through to the main status register. The main status register can be read with a serial poll, which is a bus-level operation. Serial poll executes far faster than a command transaction with the OSA.

An example using serial poll is:

```
' enable the OPC bit to reflect through to main status byte
sendCommand inst, "*ESE 1"
sendCommand inst, ":SENSe:STARt 1320 nm;:SENSe:STOP 1325 nm;:INI-
TIate:IMMediate; *OPC"
' Talk to other instruments, and so on, until you want to check on sweep
value = serialPoll(inst)
' if bit 5 is set, the sweep is done
   if (value & 32) then
   'sweep is done, read status register to clear condition
   sendCommand inst, "*ESR?")
   value = readIntReply(inst)
end if
```

where:

- serialPoll is a routine written in your favorite language using your favorite interface to perform a serial poll.

*WAI Command

The other synchronization command provided is *WAI. This command is identical to *OPC, but there is no external indication that the command is complete. It forces the instrument to finish the commands received prior to the *WAI command before executing the subsequent commands.

To take a sweep and transfer data as quickly as possible, you could:

```
' configure ascii binary transfer. FORM BIN,32 is faster, but harder to convert
sendCommand inst, "FORM ASCII"
sendCommand inst,
":SENSe:START 1320 nm;:SENSe:STOP 1325 nm;:INITiate:IMMediate;*WAI;TR
ACe:DATA:Y? TRA"
Call readAsciiTrace(inst, trace)
```

where:

- readAsciiTrace is a routine written in your favorite language using your favorite interface to read the ascii data and convert it into an array.

Controlling the sweep

Placing the optical spectrum analyzer in remote mode and sending the DISPlay:WINDow:TRACe:ALL:SCALe:AUTO command finds the largest signal and optimizes the instrument settings. This command also sets single sweep mode on the instrument. If the DISPlay:WINDow:TRACe:ALL:SCALe:AUTO command is not used, single sweep can be set using the INITiate:CONTinuous OFF command. The trace data present in the instrument must be updated by taking a sweep when appropriate using the INITiate:IMMediate command. Use this command to update the sweep after changing settings.

This mode of operation allows the program to control the sweep and ensure that data read from, or operated on in the instrument, is updated correctly. Controlling the sweep also minimizes the amount of time the instrument spends sweeping. At high sensitivity and high resolution settings, sweeps can take a significant amount of time. Controlling the sweep ensures that the amount of time spent acquiring data is optimized and that the data being displayed is valid for the current settings.

SCPI Syntax Rules

The following information applies to the common and instrument-specific commands. All measurement values and parameters are sent and received only as ASCII strings with the exception of the following commands. These commands can send and receive floating point binary data in IEEE 488.2 indefinite or definite length blocks:

HCOPY:DATA?
MMEMORY:DATA
TRACE:DATA:Y:POWER
TRACE:DATA:Y:RATIO
TRACE:DATA:Y?
MEMORY:STATE:EXTENDED?

Table 2 Syntax Notation Conventions

Convention	Description
::=	Means *is defined as.*
|	Indicates a choice of one element from a list. For example, A | B indicates A or B, but not both.
[ ]	Indicates the enclosed item is optional.
{ }	Indicates the enclosed item can be incorporated in the command several times, once, or not at all.
<file_name>	File names must conform to standard MS-DOS®[a] file naming conventions.
<trace_name>	TRA, TRB, TRC, TRD, TRE, TRF
<data_block>	This parameter represents the arbitrary block program data as defined by IEEE 488.2. Arbitrary block program data allows any 8-bit bytes to be transmitted. This includes extended ASCII control codes and symbols. Two types of data blocks are defined: definite-length blocks and indefinite-length blocks. The definite-length block consists of a "#" character, followed by one digit (in ASCII) specifying the number of length bytes to follow, followed by the length (in ASCII), followed by length bytes of binary data. For example, two bytes of binary data would be sent as follows: #12<8 bit data byte><8 bit data byte> The indefinite-length block consists of a "#" character, followed by a "0" character (in ASCII), followed by any number of bytes of binary data. The data stream is terminated by a new line character with EOI set. For example, two bytes of binary data would be sent as follows: #0<8 bit data byte><8 bit data byte>NL^EOI

[a] MS-DOS is a U.S. registered trademark of Microsoft Corporation.

SCPI commands are grouped in subsytems

In accordance with IEEE 488.2, the instrument's commands are grouped into "subsystems." Commands in each subsystem perform similar tasks. The first page of Chapter , "Programming Commands" lists where each subsystem is documented.

Sending a command

To send a command to the instrument, create a command string from the commands listed in this book, and place the string in your program language's output statement. For example, the following string places marker1 on the peak of the active trace:

```
OUTPUT 723;"CALCULATE:MARKER1:MAXIMUM"
```

Use either short or long forms

Commands and queries may be sent in either long form (complete spelling) or short form (abbreviated spelling). The description of each command in this manual shows both versions; the extra characters for the long form are shown in lowercase. The following is a long form of a command:

```
OUTPUT 723;":SENSe:WAVelength:STARt?"
```

And this is the short form of the same command:

```
OUTPUT 723;":SENS:WAV:STAR?"
```

Programs written in long form are easily read and are almost self-documenting. Using short form commands conserves the amount of controller memory needed for program storage and reduces the amount of I/O activity.

The rules for creating short forms from the long form are as follows:

The mnemonic is the first four characters of the keyword unless the fourth character is a vowel, in which case the mnemonic is the first three characters of the keyword.

This rule is *not* used if the length of the keyword is exactly four characters.

Table 3 Examples of Short Forms

Long Form	Equivalent Short Form
DISPLAY	DISP
MODE	MODE
SYSTEM	SYST
ERROR	ERR

You can use upper or lowercase letters

Program headers can be sent using any combination of uppercase or lowercase ASCII characters. In commands, uppercase lettering indicates that the uppercase portion of the command is the short form of the command. For example, in the command WAVELENGTH, WAV is the short form.

Instrument responses, however, are always returned in uppercase.

Combine commands in the same subsystem

You can combine commands from the same subsystem provided that they are both on the same level in the subsystem's hierarchy. Commands are separated with a semi-colon (;). For example, the following two lines,

```
OUTPUT 723;":SENSE:WAVELENGTH:STARt 1300NM"
OUTPUT 723;":SENSE:WAVELENGTH:STOP 1400NM"
```

can be combined into one line:

```
OUTPUT 723;":SENSE:WAVELENGTH:STARt 1300NM;STOP 1400NM"
```

The semicolon separates the two functions.

Combine commands from different subsystems

You can send commands and program queries from different subsystems on the same line by preceding the new subsystem by a semicolon followed by a colon. In the following example, the colon and semicolon pair before CALCULATE allows you to send a command from another subsystem.

```
OUTPUT 723;":SENSE:WAVELENGTH:SPAN:FULL;:CALCULATE:MARKER1:
MAXIMUM"
```

Sending common commands

If a subsystem has been selected and a common command is received by the instrument, the instrument remains in the selected subsystem. For example, if the command

```
OUTPUT 723;":SENSE:WAVELENGTH:STARt 1300NM;*CLS;STOP 1400NM"
```

is sent to the instrument, the Sense subsystem remains selected. If some other type of command is received within a program message, you must reenter the original subsystem after the command.

Adding parameters to a command

Many commands have parameters that specify an option. Use a space character to separate the parameter from the command, as shown in the following line:

```
OUTPUT 723;":SENSE:BWIDTH:RES 0.1NM"
```

Separate multiple parameters with a comma (,). Spaces can be added around the commas to improve readability.

```
OUTPUT 723;":DISPLAY:WINDOW:TRACE:STATE TRB, ON"
```

White space

White space is defined to be one or more characters from the ASCII set of 0 through 32 decimal, excluding 10 (NL). White space is usually optional, and can be used to increase the readability of a program.

Numbers

All numbers are expected to be strings of ASCII characters. Thus, when sending the number 9, you would send a byte representing the ASCII code for the character "9" (which is 57). A three-digit number like 102 would require three bytes (ASCII codes 49, 48, and 50). This is taken care of automatically when you include the entire instruction in a string. Several representations of a number are possible. For example, the following numbers are all equal:
28, 0.28E2 and 280E-1. The table below lists the available multipliers.

Table 4 Multipliers

Multiplier	Mnemonic	Multiplier	Mnemonic
1E18	EX	1E-3	M
1E15	PE	1E-6	U
1E12	T	1E-9	N
1E9	G	1E-12	P
1E6	MA	1E-15	F
1E3	K	1E-18	A

If a measurement cannot be made, no numerical response is given and an error is placed into the error queue. For example,

```
*RST
:CALCULATE1:MARKER1:X?
```

will timeout the controller and place a *Settings conflict* error in the error queue because no marker has been enabled.

Program message terminator

The string of instructions sent to the instrument is executed after the instruction terminator is received. The terminator may be either a new-line (NL) character, the End-Or-Identify (EOI) line asserted, or a combination of the two. All three ways are equivalent. Asserting the EOI sets the EOI control line low on the last byte of the data message. The NL character is an ASCII linefeed (decimal 10). The NL terminator has the same function as an EOS (End Of String) and EOT (End Of Text) terminator. Note that if the input buffer is activated, EOI is required.

Querying data

Data is requested from the instrument using a query. Queries can be used to find out how the instrument is currently configured. They are also used to obtain results of measurements made by the instrument, with the query actually activating the measurement. String responses are returned as uppercase letters.

Queries take the form of a command followed by a question mark (?). After receiving a query, the instrument places the answer in its output queue. The answer remains in the output queue until it is read or another command is issued. For example, the query

```
OUTPUT 723;":CALCULATE:MARKER1:X?"
```

places the wavelength of marker 1 in the output queue. In BASIC, the controller input statement

```
ENTER 723;RANGE
```

passes the value across the bus to the controller and places it in the variable "Range". Sending another command or query before reading the result of a query causes the output queue to be cleared and the current response to be lost. This generates an error in the error queue. The output of the instrument may be numeric or character data depending on what is queried. Refer to the specific commands for the formats and types of data returned from queries. You can send multiple queries to the instrument within a single program message, but you must also read them back within a single program message. This can be accomplished by either reading them back into a string variable or into multiple numeric variables. When you read the result of multiple queries into string variables, each response is separated by a semicolon.

Monitoring the Instrument

Your programs can monitor the Agilent 86140B series for its operating status, including querying execution or command errors and determining whether or not measurements have been completed. Several status registers and queues are provided to accomplish these tasks as shown in Figure on page 30. The status structures shown in the figure consist of condition registers, event registers, event enable registers, and, in the case of the Operation Status Structure, transition filters. For example, there exists the Standard Status *Condition* Register, the Standard Status *Event* Register, and the Standard Status *Event Enable* Register. Condition registers show the *current* condition of the status lines. Event registers show that an event has occurred. Once latched, these registers stay set until cleared. Event enable registers are masks that you can use to enable or disable the reporting of individual bits from an event register.

Querying a register always returns the value as a weighted sum of all set bits. Refer to Table 5. For example, if the value returned was 528, this would indicate that bits 4 and 9 were set. Mask registers are set using these same values. For example, the *ESE 60 command sets bits 2 through 5 of the Standard Status Event Enable Register. Whenever any one of bits 2 through 5 of the Standard Status Event Register goes high, bit 5 of the status byte will be set.

Table 5 Decimal Values of Event Enable Register Bits

Bit	Decimal Value	Bit	Decimal Value	Bit	Decimal Value	Bit	Decimal Value
0	1	4	16	8	256	12	4096
1	2	5	32	9	512	13	8192
2	4	6	64	10	1024	14	16,384
3	8	7	128	11	2048	15	32,768

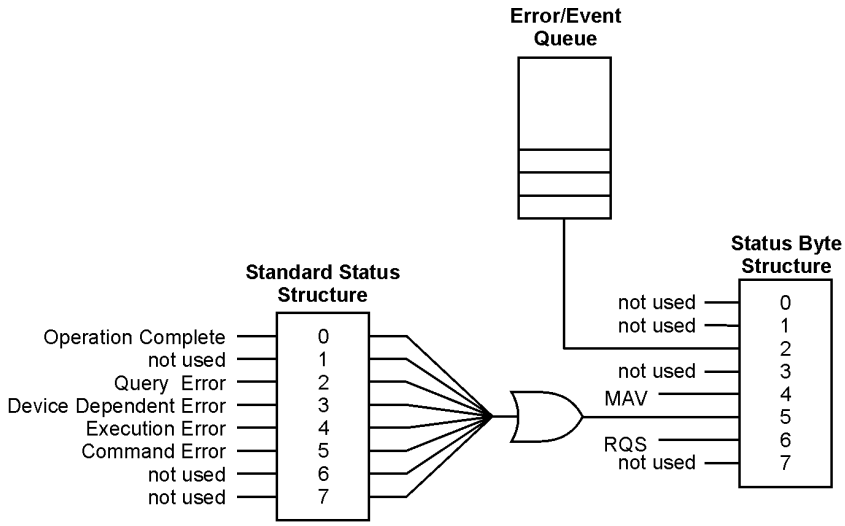

Figure 1 Status Registers

The STATus:PRESet command clears all event registers and sets all bits in the event enable registers. Use the *CLS common command to clear all event registers and all queues except the output queue. If *CLS is sent immediately following a program message terminator, the output queue is also cleared. In addition, the request for the *OPC bit is also cleared.

For an example program using the status registers, refer to "Example 9. Monitoring the status registers" on page 56.

Status Byte

The Status Byte contains summary bits that monitor activity in the other status registers and queues. The register's bits are set and cleared by summary bits from other registers or queues. If a bit in the Status Byte goes high, query the value of the source register to determine the cause.

Command	Use
GPIB serial poll command	Returns the status byte value. Reads bit 6 as the Request Service (RQS) bit and clears the bit which clears the SRQ interrupt.
*STB? common command	Returns the status byte value. Reads bit 6 as the Master Summary Status (MSS) and does not clear the bit or have any effect on the SRQ inter rupt.
*SRE common command	Sets or reads the event enable register value (mask).

Standard Status Structure

The Standard Status Structure monitors the following instrument status events: operation complete, query error, device dependent error, execution error, and command error. When one of these events occurs, the event sets the corresponding bit in the register.

Command	Use
*ESR? common command	Returns and clears the value of the event register.
*OPC common command	When all operations have finished, sets bit 0 of the event register. The query returns a 1 when all operations have finished.
*ESE common command	Sets or returns the value of the event enable reg ister (mask).

Output Queue

The output queue stores the instrument responses that are generated by certain commands and queries that you send to the instrument. The output queue generates the Message Available Summary bit when the output queue contains one or more bytes. This summary bit sets the MAV bit (bit 4) in the Status Byte. The method used to read the output queue depends upon the programming language and environment. For example, with BASIC, the output queue may be read using the ENTER statement.

Error Queue

As errors are detected, they are placed in an error queue. Instrument specific errors are indicated by positive values. General errors have negative values. You can clear the error queue by reading its contents, sending the *CLS command, or by cycling the instrument's power. The error queue is first in, first out. If the error queue overflows, the last error in the queue is replaced with error -350, "Queue overflow." Any time the queue overflows, the least recent errors remain in the queue, and the most recent error is discarded. The length of the instrument's error queue is 30 (29 positions for the error messages, and 1 position for the "Queue overflow" message). Querying errors removes the oldest error from the head of the queue, which opens a position at the tail of the queue for a new error. When all the errors have been read from the queue, subsequent error queries return 0, "No error."

Command	Use
*CLS common command	Clears the error queue (and all event registers).
SYSTEM:ERROR?	Returns and removes the oldest error from the head of the queue.

Amplitude Correction Remote Commands

Multi-Point Amplitude Correction (AMPCOR) gives you the ability to compensate for amplitude errors introduced by optical connectors or cable losses at the input port of a device. For example, if an optical connector or cable is placed between the OSA and an optical component, AMPCOR can compensate for the connector or cable loss. With AMPCOR, amplitude levels (including markers) indicate the power at the optical component's output. The true characteristics of the device are revealed by moving the measurement (or reference) point beyond the connector or cable to the optical component. Higher measurement accuracy is achieved by eliminating optical connector or cable loss errors.

Correction factors are entered in wavelength and amplitude pairs. The wavelength values entered must be equal or increasing in value to prevent an error condition. If more than one offset is entered for a given wavelength, a step in amplitude can be created. For example, if the following data set was entered into correction set 1:

```
:SENSe:CORRection:CSET1
:SENSe:CORRection:DATA 1200E-9,0,1300E-9,0, 1300E-9,2,1400E-9,4
```

wavelengths up to and including 1300 nm will have a 0 dB offset. Wavelengths between 1300 nm and 1400 nm will be interpolated between 2 dB and 4 dB (logarithmically or linearly, depending on the setting selection). If more than two offsets are entered for a given wavelength, the first and last offsets entered will be used to correct the trace. Any intermediate points will be disregarded. For example, if the following correction data was entered into correction set 2:

```
:SENSe:CORRection:CSET 2
:SENSe:CORRection:DATA 1200E-9,0,1300E-9,0,1300E-9,5,1300E-9,2,1400E-9,4
```

the OSA would use the 0 dB offset and the 2 dB offset as correction points and interpolation endpoints, and disregard the 5 dB offset. In normal usage, correction offsets should be continuous, and the wavelengths in increasing order.

Four sets of correction factors are available and stored in memory. Only one set may be selected at a time.

When AMPCOR is turned on, the correction points are applied across the active measurement range and added to all measurement results. Between points, the correction values are interpolated linearly or logarithmically. When measuring at wavelengths outside the first and last correction points, the first or last value (as appropriate) is used as the

correction value. Refer to page 32 for a listing of the AMPCOR remote commands. Since AMPCOR consumes processing power, you should only use AMPCOR if necessary.

Tip: You can turn on amplitude corrections and select a correction set by pressing AMPLITUDE > AMPLITUDE SETUP from the OSA front panel. Refer to "Amplitude Setup" on page 3-5 in the User's Guide for Amplitude Setup panel information.

The following commands will select correction set number 1, set the correction factor, and turn AMPCOR ON.

```
:SENSE:CORRECTION:CSET 1
:SENSE:CORRECTION:DATA 1200E-9,0,1201E-9,2, 1202E-9,3,1203E-9,0
:SENSE:CORRECTION:STATE ON
```

The following commands will enter values to correction set number 2, 3, and 4.

```
:SENSE:CORRECTION:CSET 2
:SENSE:CORRECTION:DATA 1200E-9,0,1201E-9,1, 1202E-9,2,1203E-9,0

:SENSE:CORRECTION:CSET 3
:SENSE:CORRECTION:DATA 1200E-9,0,1201E-9,2, 1202E-9,3,1203E-9,0

:SENSE:CORRECTION:CSET 4
:SENSE:CORRECTION:DATA 1200E-9,0,1201E-9,1, 1202E-9,3,1203E-9,0
```

While AMPCOR is ON, selecting a different correction set will immediately correct the measured data to the new correction set. For example, to select correction set number three use:

```
:SENS:CORR:CSET 3
```

NOTE If AMPCOR is used when auto-ranging is turned OFF, the signal may get clipped. To avoid signal clipping, limit the correction factor to a range from –3dB to +12dB.

Techniques to Improve Throughput

This section provides a series of suggestions and rules of thumb that are useful in increasing measurement throughput.

Take control of the sweep

Put the instrument in single sweep mode using INIT:CONT:OFF, and sweep only to acquire data. The sweep is the most time consuming operation in the OSA. You should send all your sweep related settings and then take the sweep, taking as few sweeps as possible.

Turn the display off

The single most effective thing you can do to increase measurement throughput is turn the display off. If the operator is interacting with the computer display rather than the instrument display, consider turning the display off. This does not make the internal motors or the data acquisition system operate any faster. Instead, it removes all the OSA system overhead associated with maintaining a real time display. This makes software intensive operations much faster, but sweep associated operations won't change very much. As a result, the speed improvement you will see depends on your application. With the display off, you can expect a typical application that is taking sweeps, using markers and transferring traces to run about twice as fast.

To turn the display off:

```
sendMessage DISP OFF; *OPC?
receiveInteger
```

This takes about 12 seconds and does not affect the instrument state. You can turn the display back on whenever you want to, either by sending DISP ON or by pressing the local key on the instrument front panel. It again will take about 12 seconds. Note that you cannot turn off the display from the front panel.

The instrument behavior with the display off is identical to behavior with display on, only faster. You can develop your application with the display on for easy debugging, then turn the display off once everything is working.

The DISP OFF command is unusual in that a *RST or a SYST:PRES does not turn the display back on. Once off, the display stays off until you want it to come back on.

Combine commands into a single message

The instrument will execute commands faster if it receives them all in a single message. For example:

```
SENDMESSAGE INIT              // take a sweep
SENDMESSAGE *OPC?             // instrument tells us when done
RECEIVEINTEGER                // wait for reply
```

This is inefficient because the instrument will receive and parse the INIT command, and actually launch the sweep before returning to GPIB to receive and parse the *OPC?. The instrument will hold off GPIB for several hundred milliseconds before accepting the *OPC? message. Instead, combine them into a single message:

```
SENDMESSAGE INIT;OPC?         // take a sweep, instrument tells us when
                                 done

RECEIVERINTEGER               // wait for reply
```

A good rule of thumb is to combine closely related commands. For example, sending the OPC with the command it waits for makes good sense. If you are only changing one or two sweep parameters, you might include that in the message also. Avoid putting too much in a single message, as this makes it difficult to read and rearrange your application program contents.

Transfer traces with more than 2 marker operations

Modern controllers have plenty of processing power to calculate trace statistics, and can often calculate faster than the instrument. The rule of thumb is that if you perform more than two marker functions (and associated value read back), it is more efficient to transfer the trace and do the post-processing in your computer. It takes about 1/4 second to transfer a 1000 point trace using a 32 bit binary transfer.

The primary exception to this rule is the noise markers. The noise markers use the actual resolution bandwidth (measured on an instrument-by-instrument basis) to normalize the measured power to a 1 or 0.1 nm span. This calculation is difficult to perform in the controller because it doesn't have direct access to the actual resolution bandwidth values.

Use 32 bit binary trace transfers

The most convenient trace transfer is an ASCII trace transfer, but downloading the data in REAL32 format is substantially faster. Binary can take half as long as ASCII, which saves about 100 ms for every 1000 points transferred. ASCII format sends the trace value reformatted as 12 character strings, whereas a 32 bit binary transfer sends the data in IEEE 32 bit format. This is the same representation used inside the instrument and takes only 4 8-bit data bytes. There are two binary transfer formats

available - IEEE 32 bit and IEEE 64 bit formats. The instrument does not carry more than 32 bits of data resolution internally, so there is no benefit to doing a 64 bit transfer. To configure for a 32 bit binary transfer:

```
SENDMESSAGE FORMAT REAL,32
```

The advantage of binary over ASCII can be even greater for automation projects using Visual Basic and C++. With short binary transfer times, the data transfer process is faster than the hardware. This makes it possible to devise ways to process the data faster to keep up with the hardware. Note that in Visual C++ it is possible to download and convert the trace in one step by changing the format string.

The following table and chart compare the different times it takes to transfer trace data. The data was taken using Visual C++ with National Instruments VISA. Since the data must be converted to be useful, both transfer and conversion times are listed.

Table 6 Binary vs. ASCII Downloads

Trace Points	Binary Transfer	Binary Conversion	Binary Total	ASCII Transfer	ASCII Conversion	ACSII Total	Binary Advantage
1000	47	78	125	219	16	235	110
2000	79	171	250	422	16	438	188
3000	110	250	360	640	31	671	311
4000	140	344	484	844	31	875	391
5000	172	437	609	1063	31	1094	485
6000	203	515	718	1282	31	1313	595
7000	234	594	828	1484	47	1531	703
8000	266	687	953	1703	63	1766	813
9000	297	781	1078	1906	63	1969	891

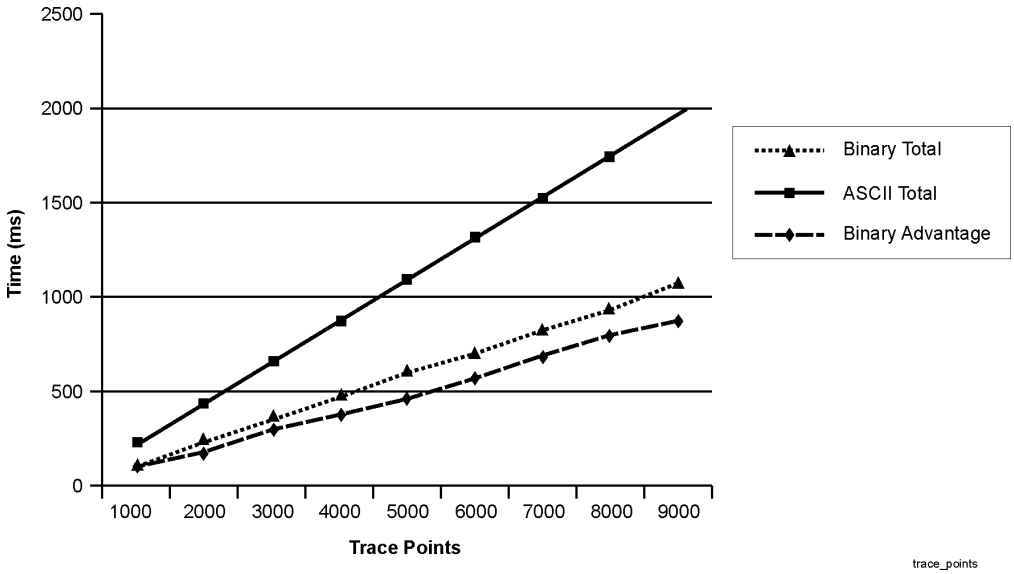

Figure 2 Binary vs. ASCII Downloads

One of the biggest challenges to transferring binary data on a Windows platform is the necessity to swap the byte order. For a 32 bit floating point number, you have 4 bytes. If we number them 0, 1, 2, and 3, we need to swap 0 and 3 and then swap 1 and 2. The C algorithm below does this:

```
INT CNT;
CHAR TEMPBYTE;
CHAR *TRACEBYTES;
// allocate buffer and transfer trace into traceBytes

FOR (CNT=0;CNT<NUMBYTES;CNT+=4)
{

    // swap bytes 3 and 0
    TEMPBYTE       = TRACEBYTES[CNT];
    TRACEBYTES[CNT]   = TRACEBYTES[CNT + 3];
    TRACEBYTES[CNT+3] = TEMPBYTE;

    // swap bytes 2 and 1
    TEMPBYTE       = TRACEBYTES[CNT+1];
    TRACEBYTES[CNT+1] = TRACEBYTES[CNT+2];
    TRACEBYTES[CNT+2] = TEMPBYTE;

}
```

Because of the type casting and byte manipulation involved, it is very difficult to do this in Visual Basic. It is possible to write a DLL in Visual C to do the conversion.

Optimize Sweep Parameters

It is common practice to use conservative sweep parameters when first developing a test, perhaps sweeping broader or using narrower resolution bandwidth than necessary, to get reliable data. Once the application is running reliably, evaluate your sweep settings:

- Consider a wider resolution bandwidth.
- Consider a narrower span.
- Consider taking one larger sweep instead of several small ones.
- Set sensitivity at a higher level.
- Experiment with sweep time.
- Take as few sweeps as possible to obtain test data.

Sending Remote Commands over the LAN

The Standard Instrument Control Library (SICL) Local Area Network (LAN) enables you to control your remote analyzer over the LAN as if your local analyzer is connected directly to the controller with the GPIB. SICL provides control of your analyzer over the LAN, using a variety of computing platforms, I/O interfaces, and operating systems.

Your analyzer implements a SICL LAN *server*. To control the analyzer, you need a SICL LAN *client* application running on a computer or workstation that is connected to the analyzer over a LAN. Typical applications implementing a SICL LAN client include:

- Agilent VEE
- HT BASIC
- National Instrument's LabView with Agilent VISA/SICL client drivers

SICL LAN can be used with Windows 95, Windows 98, Windows NT, and HP-UX.

SICL LAN Set-up Information

Before setting up your controller as a SICL LAN client, you will need to collect the following information:

GPIB name	The GPIB name is the name given to a device used to communicate with the analyzer. Your analyzer is shipped with hpib7 as the GPIB name.
GPIB logical unit	The logical unit number is a unique integer assigned to the device to be controlled using SICL LAN. Your analyzer is shipped with the logical unit number set to 7.
GPIB device address	The device address is the GPIB device address (bus address) assigned to the device to be controlled using SICL LAN. Your analyzer is shipped with the GPIB device address set to 23.

The SICL LAN client uses the GPIB name, GPIB logical unit number, and GPIB address to communicate with the server. You must match these parameters exactly when you set up the SICL LAN client and server.

Setting Up Your Analyzer as a SICL LAN Server

Your analyzer becomes an active SICL LAN server when configured for networking. Please refer to “SICL LAN Set-up Information” on page 40 for the network setup information. See “To change the GPIB address´´ on page 18 to change the GPIB address from the front panel.

Setting Up Your Controller as a SICL LAN Client

For Windows 95, Windows 98, and Windows NT:

1 Install Agilent I/O Libraries.

2 Run I/0 configuration.

3 Select **LAN Client** from the Available Interface Types.

4 Click **Configure**.

5 Click **OK**.

6 Select **VISA LAN Client** from the Available Interface Types.

7 Click **Configure**.

8 Enter the hostname or IP address of your analyzer.

9 Enter hpib7 in the remote SICL interface name field.

10 Click **OK**.

11 Select the new **VISA LAN Client** then click **Edit**.

12 Click **Edit VISA Config**.

13 Clear the **Unselect Identify devices at run-time** check box.

14 Click **Add Device**.

15 Enter the GPIB address of your analyzer, then Click **OK** three times to exit the application.

For HPUX:

1 Install Agilent SICL Libraries.

2 Run iosetup.

3 Select **LAN Client f**rom the Available Interface Types.

4 Click **Configure**.

5 Click **OK**.

Limitations:

1 There may be a period of up to 5 seconds when you will be unable to open a LAN interface with the OSA. It is common for LAN clients to open an Interface Session for device actions. If your client application opens an interface session, there will be a delay between the time the interface is closed and when you will be able to open a new interface.

2 You will receive separate responses to SCPI queries that are combined on a single line. For example if you send SENS:START?STOP? you will have to read twice to get both responses.

```
iprintf (osa, "SENS:START?;STOP?\n");
iscanf (osa, "%t", startWavelength);                    /* +6.00000000E-007 */
iscanf (osa, "%t", stopWavelength);                     /* +1.70000000E-006 */
```

Normally (with GPIB) you would receive both responses separated with a semicolon.

```
iprintf (osa, "SENS:START?;STOP?\n");
iscanf (osa, "%t", startStop);                          /* +6.00000000E-007;+1.70000000E-006
*/
```

3 You must not set a time-out of less then 1-second.

Example Programs

These programs are provided to give you examples of using Agilent 86140B series remote programming commands in typical applications. They are not meant to teach general programming techniques or provide ready-to-use solutions. They should allow you to see how measurements are performed and how to return data to the computer. The programs are written in BASIC for Windows.

The following example programs are provided in this section:

Example 1. Initialization and a simple measurement

Description

This program provides the basic building block for beginning development of a measurement routine. The *RST common command resets the instrument to predetermined settings to provide a common starting point. The automeasure function locates the largest signal in the spectrum and optimizes the display of the signal. The maximum signal is located and a marker placed on the signal. This signal is then used for the autoalign function. Autoalign aligns the internal components of the OSA to compensate for any effects of handling, temperature, and humidity. This operation should be performed whenever the instrument is moved or the environmental conditions change. It should be performed after the instrument is at operating temperature. Periodic use of autoalign assures optimum performance. The program sets the start and stop wavelength and the amplitude sensitivity.

Program

```
! Initialization and a simple measurement
!
PRINT "Single Measurement Example"
CLEAR 7                                          ! Clear GPIB interface
ASSIGN @Osa TO 723;EOL CHR$(10) END               ! Turn on END message
CLEAR @Osa                                       ! Selected device clear
OUTPUT @Osa;"disp:wind:text:data 'Single Measurement'"
!
!******************************Initialization Routine******************************
!
PRINT
PRINT "Presetting the instrument"
OUTPUT @Osa;"*rst;*opc?"
ENTER @Osa;Temp
!
PRINT "Turning buffer on"
OUTPUT @Osa;"syst:comm:gpib:buff on"             ! Turn OSA GPIB input buffer on
!
PRINT
PRINT "Performing AutoMeasure"                   ! Automeasure
OUTPUT @Osa;"disp:wind:trac:all:scal:auto;*Opc?"
ENTER @Osa;Temp
!
PRINT "Performing Autoalign"                     ! Autoalign
OUTPUT @Osa;"cal:alig:mark1;*opc?"
ENTER @Osa;Temp
!
!*******************************************************************************
!
PRINT
PRINT "...measurement begins"
!
OUTPUT @Osa;"sens:wav:star 1314nm"               ! Set start wavelength
OUTPUT @Osa;"sens:wav:stop 1316nm"               ! Set stop wavelength
OUTPUT @Osa;"sens:pow:dc:rang:low -70dbm"        ! Set amplitude sensitivity
OUTPUT @Osa;"sens:bwid:res 0.1nm"                ! Set the resolution bandwidth
OUTPUT @Osa;"init:imm;*opc?"                     ! Take a sweep
ENTER @Osa;Temp
OUTPUT @Osa;"calc:mark1:max"                     ! Locate max signal
```

```
OUTPUT @Osa;"calc:mark1:scen"          ! Marker to center
OUTPUT @Osa;"init:imm;*opc?"           ! Take a sweep
ENTER @Osa;Temp
!
LOCAL @Osa                             ! Return to local operation
END
```

Example 2. Locating the largest signal

Description

This program finds the largest signal, zooms to a narrow span, and then uses markers to return signal wavelength and amplitude to the computer.

Program

```
! Locating the Largest Signal
!
PRINT "OSA Zoom Example"
CLEAR 7                                          ! Clear GPIB interface
ASSIGN @Osa TO 723;EOL CHR$(10) END               ! Turn on END message
CLEAR @Osa                                       ! Selected device clear
OUTPUT @Osa;"disp:wind:text:data 'Display the largest Signal'"
!
PRINT
PRINT "Presetting the instrument"
OUTPUT @Osa;"*rst;*opc?"                         ! Preset the instrument
ENTER @Osa;Temp
PRINT
!
PRINT "Turning buffer on"
OUTPUT @Osa;"syst:comm:gpib:buff on"             ! Turn OSA GPIB input buffer on
!
PRINT
PRINT "Performing AutoMeasure"                   ! Automeasure
OUTPUT @Osa;"disp:wind:trac:all:scal:auto;*opc?"
ENTER @Osa;Temp
!
PRINT "Performing Autoalign"                     ! Autoalign
OUTPUT @Osa;"cal:alig:mark1;*opc?"
ENTER @Osa;Temp
!
PRINT "...Measurement begins"
!
OUTPUT @Osa;"init:imm;*opc?"                     ! Take a sweep, wait for completion
ENTER @Osa;Temp
OUTPUT @Osa;"calc1:mark1:max"                    ! Marker to peak
OUTPUT @Osa;"calc1:mark1:scen"                   ! Marker to center
OUTPUT @Osa;"sens:wav:span 10nm"                 ! Set span
!
OUTPUT @Osa;"init:imm;*opc?"                     ! Take a sweep, wait for completion
ENTER @Osa;Temp
OUTPUT @Osa;"calc1:mark1:max"                    ! Marker to peak
OUTPUT @Osa;"calc1:mark1:scen"                   ! Marker to center
!
OUTPUT @Osa;"init:imm;*opc?"                     !Take a sweep, wait for completion
ENTER @Osa;Temp
!
OUTPUT @Osa;"calc1:mark1:x?"                     ! Read marker wavelength
ENTER @Osa;Markwl
!
OUTPUT @Osa;"calc1:mark1:y?"                     ! Read marker amplitude
ENTER @Osa;Markamp
!
PRINT
PRINT "Marker values:"
Markwl=Markwl*1.E+9                              ! Convert to nm
PRINT "wavelength: ",Markwl;"nm"
PRINT "amplitude: ",Markamp;"dBm"
!
LOCAL @Osa                                       ! Return OSA to local operation
END
```

Example 3. Bandwidth

Description

The 20 dB marker BW function is used to determine the bandwidth of the signal. The program assumes a narrowband signal as an input.

Program

```
! Bandwidth
!
CLEAR 7                                          ! Clear GPIB interface
ASSIGN @Osa TO 723;EOL CHR$(10) END               ! Turn on END message
CLEAR @Osa                                       ! Selected device clear
PRINT "Signal Bandwidth Measurement"
OUTPUT @Osa;"disp:wind:text:data 'Signal Bandwidth Measurement'"
!
PRINT
PRINT "Presetting the instrument"
OUTPUT @Osa;"*rst;*opc?"                         ! Preset the instrument
ENTER @Osa;Temp
!
PRINT "Turning buffer on"
OUTPUT @Osa;"syst:comm:gpib:buff on"             ! Turn OSA GPIB input buffer on
!
PRINT "...Measurement Begins"
!
OUTPUT @Osa;"sens:wav:star 1530nm"               ! Set start wavelength
OUTPUT @Osa;"sens:wav:stop 1570nm"               ! Set stop wavelength
OUTPUT @Osa;"sens:pow:dc:rang:low -60dBm"        ! Set sensitivity
OUTPUT @Osa;"init:imm;*opc?"                     ! Take a sweep
ENTER @Osa;Temp
!
OUTPUT @Osa;"calc1:mark1:max"                    ! Marker to peak
OUTPUT @Osa;"calc1:mark1:srl"                    ! Marker to reference level
OUTPUT @Osa;"calc1:mark1:scen"                   ! Marker to center
OUTPUT @Osa;"sens:wav:span 10nm"                 ! Set span
!
OUTPUT @Osa;"calc1:mark1:x?"                     ! Read marker wavelength
ENTER @Osa;Markwl
OUTPUT @Osa;"calc1:mark1:y?"                     ! Read marker amplitude
ENTER @Osa;Markamp
Markwl=Markwl*1.E+9!Convert to standard measurement units (nm)
PRINT "Marker wavelength";Markwl;"nm"
PRINT "Marker amplitude";Markamp;"dBm"
PRINT
!
OUTPUT @Osa;"sens:bwid:res 0.1 nm"               ! Set resolution bandwidth to min
OUTPUT @Osa;"sens:wav:span 2nm"                  ! Set span
OUTPUT @Osa;"init:imm;*opc?"                     ! Take a sweep
ENTER @Osa;Temp
!
OUTPUT @Osa;"calc1:mark1:max"                    ! Marker to peak for reference point
OUTPUT @Osa;"calc1:mark1:func:bwid:ndb -20.0 db" ! Select db down where bw is
calculated
OUTPUT @Osa;"calc1:mark1:func:bwid:int on"       ! Enable bw marker interpolation
OUTPUT @Osa;"calc1:mark1:func:bwid:read wav"     ! Set the BW unit of measurement to
WL
OUTPUT @Osa;"calc1:mark1:func:bwid:stat on"      ! Enable bandwidth marker
!
OUTPUT @Osa;"calc1:mark1:func:bwid:res?"         ! Return axis values between markers
ENTER @Osa;Rbw
IF Rbw<9.E+37 THEN Cnt! test for valid result
PRINT "BW not found"
STOP
```

```
Cnt:! label
!
OUTPUT @Osa;"calc1:mark1:func:bwid:x:left?"          ! Read left BW marker X axis value
ENTER @Osa;Markleft
OUTPUT @Osa;"calc1:mark1:func:bwid:x:righ?"          ! Read right BW marker X axis value
ENTER @Osa;Markright
OUTPUT @Osa;"calc1:mark1:func:bwid:x:cent?"          ! Read center BW marker X axis value
ENTER @Osa;Markcent
!
! Convert to standard measurement units (nm)
Rbw=Rbw*1.E+9
Markleft=Markleft*1.E+9
Markright=Markright*1.E+9
Markcent=Markcent*1.E+9
!
PRINT "-20 db Marker bandwidth",Rbw;"nm"
PRINT "Left marker";Markleft;"nm"
PRINT "Right marker";Markright;"nm"
PRINT "Center";Markcent;"nm"

LOCAL @Osa
END
```

Example 4. Maximum and minimum amplitude values

Description

The marker delta function is used to find the maximum and minimum (peak and pit) values of the signal.

Program

```
! Maximum and Minimum amplitude values
!
CLEAR 7                                          ! Clear GPIB interface
ASSIGN @Osa TO 723;EOL CHR$(10) END               ! Turn on END message
CLEAR @Osa                                       ! Selected device clear
!
PRINT "Peak to Pit Example"
OUTPUT @Osa;"disp:wind:text:data 'Minimum and Maximum Signals'"
!
PRINT
PRINT "Presetting the instrument"
OUTPUT @Osa;"*rst;*opc?"                         ! Preset the instrument
ENTER @Osa;Temp
!
PRINT "Turning buffer on"
OUTPUT @Osa;"syst:comm:gpib:buff on"             ! Turn OSA GPIB input buffer on
!
OUTPUT @Osa;"sens:wav:star 1530nm"               ! Set start wavelength
OUTPUT @Osa;"sens:wav:stop 1570nm"               ! Set stop Wavelength
OUTPUT @Osa;"init:imm;*opc?"                     ! Take a sweep
ENTER @Osa;Temp
OUTPUT @Osa;"calc1:mark1:max"                    ! Marker to peak
OUTPUT @Osa;"calc1:mark1:scen"                   ! Marker to center
OUTPUT @Osa;"calc:mark1:srl"                     ! Marker to reference level
OUTPUT @Osa;"sens:wav:span 10nm"                 ! Set span
!
PRINT "Measurement begins"
PRINT
!
OUTPUT @Osa;"sens:pow:dc:rang:low -60dBm"        ! Set sensitivity
OUTPUT @Osa;"init:imm;*opc?"                     ! Take a sweep
ENTER @Osa;Temp
!
OUTPUT @Osa;"calc1:mark1:max"                    ! Marker to peak
OUTPUT @Osa;"calc1:mark1:x?"                     ! Read marker wavelength
ENTER @Osa;Markwl
OUTPUT @Osa;"calc1:mark1:y?"                     ! Read marker amplitude
ENTER @Osa;Markamp
!
OUTPUT @Osa;"calc1:mark1:func:delt:stat on"      !Turn delta marker on
!
OUTPUT @Osa;"calc1:mark1:min"                    ! Marker to pit
!
OUTPUT @Osa;"calc1:mark1:func:delt:y:offs?"      ! Read Delta Y marker
ENTER @Osa;Markdelty
OUTPUT @Osa;"calc1:mark1:func:delt:x:offs?"
ENTER @Osa;Markdeltx
!
PRINT "   Marker values"
PRINT
Markwl=Markwl*1.E+9                              ! Convert to nm
PRINT Markwl;"nm",Markamp;"dBm"
!
Markdeltx=Markdeltx*1.E+9                        ! Convert to nm
```

```
!
PRINT
PRINT "Marker Delta Values"
PRINT
PRINT Markdeltx;"nm",Markdelty;"dBm"
!
LOCAL @Osa                                    ! Return control to local
END
```

Example 5. Maximum and minimum values over time

Description

This program locates the largest signal using automeasure, and adjusts the center wavelength, span, and sensitivity settings. Trace B is then viewed and updated and set to maximum hold. Trace C is then viewed, updated, and set to minimum hold. Signal variations with time can now be monitored.

Program

```
! Maximum and minimum values over time
!
CLEAR 7                                          ! Clear GPIB interface
ASSIGN @Osa TO 723;EOL CHR$(10) END               ! Turn on END message
CLEAR @Osa                                       ! Selected device clear
!
PRINT "Min/Max Hold Example"
OUTPUT @Osa;"disp:wind:text:data 'Minimum and Maximum Signals'"
!
PRINT
PRINT "Presetting the instrument"
OUTPUT @Osa;"*rst;*opc?"                         ! Preset the instrument
ENTER @Osa;Temp
!
PRINT "Turning buffer on"
OUTPUT @Osa;"syst:comm:gpib:buff on"             ! Turn OSA GPIB input buffer on
!
PRINT "...measurement begins"
!
OUTPUT @Osa;"sens:wav:star 1530nm"               ! Set start wavelength
OUTPUT @Osa;"sens:wav:stop 1570nm"               ! Set stop Wavelength
OUTPUT @Osa;"init:imm;*opc?"                     ! Take a sweep
ENTER @Osa;Temp
OUTPUT @Osa;"calc1:mark1:max"                    ! Marker to peak
OUTPUT @Osa;"calc1:mark1:scen"                   ! Marker to center
OUTPUT @Osa;"calc:mark1:srl"
OUTPUT @Osa;"sens:wav:span 20nm"                 ! Set span
!
OUTPUT @Osa;"sens:pow:dc:rang:low -65dBm"        ! Set the sensitivity
!
! Update and view trace B and set it to max hold
!
OUTPUT @Osa;"disp:wind:trac:stat trB, on"        ! View trace B
OUTPUT @Osa;"trac:feed:cont trB, Alw"            ! Update trace B
OUTPUT @Osa;"calc2:max:stat on"                  ! Set trace B to max hold
!
! Update and view trace C and set it to min hold
!
OUTPUT @Osa;"disp:wind:trac:stat trC, on"        ! View trace C
OUTPUT @Osa;"trac:feed:cont trC, Alw"            ! Update trace C
OUTPUT @Osa;"calc3:min:stat on"                  ! Set trace C to min hold
!
OUTPUT @Osa;"init:cont on"                       ! Turn continuous sweep on
!
LOCAL @Osa                                       ! Return to local control
END
```

Example 6. Returning trace data

Description

This program locates the largest signal and then zooms to a narrow span. The trace length is changed to 101 points and the entire trace data is read in and printed on the display.

Program

```
! Returning trace data
!
CLEAR 7                                    ! Clear GPIB interface
ASSIGN @Osa TO 723;EOL CHR$(10) END         ! Turn on END message
CLEAR @Osa                                 ! Selected device clear
!
PRINT "OSA Trace Example"
OUTPUT @Osa;"disp:wind:text:data 'Trace Readout'"
!
DIM Tdata(1:101)                           ! Create a trace array
DIM Wdata(1:101)                           ! Wavelength data
!
PRINT
PRINT "Presetting the instrument"
OUTPUT @Osa;"*rst;*opc?"                   ! Preset the instrument
ENTER @Osa;Temp
!
PRINT "Turning buffer on"
OUTPUT @Osa;"syst:comm:gpib:buff on"       ! Turn OSA GPIB input buffer on
!
OUTPUT @Osa;"sens:wav:star 1530nm"         ! Set start wavelength
OUTPUT @Osa;"sens:wav:stop 1570nm"         ! Set stop Wavelength
OUTPUT @Osa;"init:imm;*opc?"               ! Take a sweep
ENTER @Osa;Temp
OUTPUT @Osa;"calc1:mark1:max"              ! Marker to peak
OUTPUT @Osa;"calc1:mark1:scen"             ! Marker to center
OUTPUT @Osa;"calc:mark1:srl"               ! Marker to reference level
!
PRINT "Measurement begins"
!
OUTPUT @Osa;"sens:wav:span 10nm"           ! Set span
OUTPUT @Osa;"init:imm;*opc?"
ENTER @Osa;Temp
OUTPUT @Osa;"calc1:mark1:max"              ! Marker to peak
OUTPUT @Osa;"calc1:mark1:scen"             ! Marker to center
OUTPUT @Osa;"init:imm;*opc?"               ! Take a sweep
ENTER @Osa;Temp
!
! Read in the trace data
!
OUTPUT @Osa;"sens:swe:poin 101"            ! Set trace length to 101
OUTPUT @Osa;"init:imm;*Opc?"               ! Take a sweep
ENTER @Osa;Temp
!
!
!OUTPUT @Osa;"form ascii"                  ! Set data format to ASCII
OUTPUT @Osa;"trac:data:y? tra"             ! Request data
ENTER @Osa;Tdata(*)                        ! Read trace data
!
! Read start, stop and trace length
!
OUTPUT @Osa;"sens:wav:star?"               ! Read start wavelength
ENTER @Osa;Startw
OUTPUT @Osa;"sens:wav:stop?"               ! Read stop wavelength
```

```
ENTER @Osa;Stopw
OUTPUT @Osa;"sens:swe:poin?"                    ! Read trace length
ENTER @Osa;Tlength
Bucket=(Stop-Start)/(Tlength-1)                 ! Calculate bucket length
PRINT "Data Point Size",Bucket;"nm"
!
PRINT "Point","Wavelength","Amplitude"
!
! The following lines calculate the wavelength value of each point of the trace.
! Note that wavelength values of a trace cannot be directly queried.
!
FOR I=1 TO 101
     Wlength=Startw+(Bucket*(I-1))! Calculate point wavelength
     Wlength=Wlength*1.E+9                      ! Convert to nm
     PRINT I,Wlength,Tdata(I)
NEXT I
!
LOCAL @Osa                                      ! Return to local control
END
```

Example 7. Trace normalization

Description

This program demonstrates trace normalization. Normalization is used to observe changes to a displayed response. For example, run the program and then bend the fiber to observe the change in signal level across the spectrum. Trace C displays the difference between trace A and trace B.

Program

```
! Trace Normalization
!
CLEAR 7                                          ! Clear GPIB interface
ASSIGN @Osa TO 723;EOL CHR$(10) END               ! Turn on END message
CLEAR @Osa                                       ! Selected device clear
!
PRINT "OSA Normalization Example"
OUTPUT @Osa;"disp:wind:text:data 'OSA Normalization Example'"
!
PRINT
PRINT "Presetting the instrument"
OUTPUT @Osa;"*rst;*opc?"                         ! Preset the instrument
ENTER @Osa;Temp
!
PRINT "Turning buffer on"
OUTPUT @Osa;"syst:comm:gpib:buff on"             ! Turn OSA GPIB input buffer on
!
OUTPUT @Osa;"init:imm;*opc?"
ENTER @Osa;Temp
OUTPUT @Osa;"calc1:mark1:max"                    ! Marker to peak
OUTPUT @Osa;"calc1:mark1:scen"                   ! Marker to center
OUTPUT @Osa;"calc:mark1:srl"                     ! Marker to reference level
OUTPUT @Osa;"calc:mark1:stat off"                ! Turn marker 1 off
!
PRINT "...Measurement begins"
!
OUTPUT @Osa;"Sens:bwid:res 10nm"                 ! Fix resolution bw
!
OUTPUT @Osa;"init:imm;*opc?"                     ! Take a sweep
ENTER @Osa;Temp
!
! Trace A is the active trace
OUTPUT @Osa;"Disp:Wind:Trac:Stat TrA,On"         ! Turn on Trace A
OUTPUT @Osa;"Trac:Feed:Cont TrA, Alw"
!
! Trace B is the reference trace
OUTPUT @Osa;"disp:Wind:Trac:Stat TrB,ON"         ! Turn on Trace B
OUTPUT @Osa;"Trac:Feed:Cont TrB, Alw"
!
! Trace C displays the difference between A & B
OUTPUT @Osa;"disp:Wind:Trac:Stat TrC,ON"         ! Turn on Trace C
OUTPUT @Osa;"Trac:Feed:Cont TrC, Alw"!
!
OUTPUT @Osa;"Trac:Feed:Cont TrB,Nev"             ! Stop updating B
!
OUTPUT @Osa;"init:cont on"                       ! Set up continuous sweep
!
! Trace math function Log Math C=A-B
OUTPUT @Osa;"Calc3:Math:Expr ( TRA/TRB )"        ! Normalize Trace A to B
OUTPUT @Osa;"Calc3:Math:Stat On"                 ! Turn on normalization
!
LOCAL @Osa                                       ! Return to local control
END
```

Example 8. Total power measurement

Description

This program demonstrates the total power function. The ASE broadband noise power of an EDFA source is measured. Two sweeps are taken, one of the entire trace and one, using line markers, of just the noise hump. The total power of the two different traces are displayed.

Program

```
! Total power measurement
!
CLEAR 7                                          ! Clear GPIB interface
ASSIGN @Osa TO 723;EOL CHR$(10) END               ! Turn on END message
CLEAR @Osa                                       ! Selected device clear
!
PRINT
PRINT "Presetting the instrument"
OUTPUT @Osa;"*rst;*opc?"                         ! Preset the instrument
ENTER @Osa;Temp
!
PRINT "Turning buffer on"
OUTPUT @Osa;"syst:comm:gpib:buff on"             ! Turn OSA GPIB input buffer on
!
OUTPUT @Osa;"sens:wav:cent 1550nm"               ! Set center wavelength of signal
OUTPUT @Osa;"sens:wav:span 10nm"                 ! Set the span
OUTPUT @Osa;"sens:bwid:res .1nm"                 ! Set resolution bandwidth
OUTPUT @Osa;"sens:pow:rang:low -60dbm"           ! Set sensitivity to -60 dBm
!
OUTPUT @Osa;"init:imm;*opc?"                     ! Take a sweep
ENTER @Osa;Temp
!
OUTPUT @Osa;"calc1:tpow:stat 1"                  ! Turn the power state on
OUTPUT @Osa;"calc1:tpow:data?"                   ! Query the total power
ENTER @Osa;Tpower
PRINT "Entire trace:";Tpower                     ! Print the total power
PRINT
!
! Select portion of trace
OUTPUT @Osa;"calc1:tpow:iran:low 1547.6nm"       ! Set the upper & lower
OUTPUT @Osa;"calc1:tpow:iran:upp 1552.6nm"       !     limits for the calculation
!
OUTPUT @Osa;"calc1:tpow:data?"                   ! Query the total power
ENTER @Osa;Tpower
PRINT "Portion of trace:";Tpower                 ! Print the total power
!
LOCAL @Osa                                       ! Return to local operation
END
```

Example 9. Monitoring the status registers

Description

This program presets the instrument and then selects the largest signal in the span. This program demonstrates the use of status registers to detect programming errors. A serial poll is performed to read the instrument status byte. The same status byte is read with *STB?. The internal error register is also read and displayed. The error queue is queried to display the error condition.

Program

```
! Monitoring the status registers
!
CLEAR 7                                          ! Clear GPIB interface
ASSIGN @Osa TO 723;EOL CHR$(10) END               ! Turn on END message
CLEAR @Osa                                       ! Selected device clear
!
PRINT "OSA Status Byte example"
OUTPUT @Osa;"disp:wind:text:data 'OSA Status Byte Example'"
PRINT
!*****************************Initialization routine*****************************
!
PRINT "Presetting the instrument"
OUTPUT @Osa;"*rst;*opc?"                         ! Preset the instrument
ENTER @Osa;Temp
!
PRINT "Turning buffer on"
OUTPUT @Osa;"syst:comm:gpib:buff on"             ! Turn OSA GPIB input buffer on
!
OUTPUT @Osa;"init:imm;*opc?"
ENTER @Osa;Temp
OUTPUT @Osa;"calc1:mark1:max"                    ! Marker to peak
OUTPUT @Osa;"calc1:mark1:scen"                   ! Marker to center
OUTPUT @Osa;"calc:mark1:srl"                     ! Marker to reference level
OUTPUT @Osa;"sens:wav:span 20nm"                 ! Set the Span
OUTPUT @Osa;"calc:mark1:stat off"                ! Turn marker 1 off
!
!*****************************End Initialization*****************************
!
OUTPUT @Osa;"*CLS"                               ! Clear Error Queue
!
OUTPUT @Osa;"*ESE 32"                            ! Set Standard Event Enable bit 5 (32)
!
OUTPUT @Osa;"init:imm;*opc?"                     ! Take a sweep
ENTER @Osa;Temp
!
Sbyte=SPOLL(@Osa)                                ! Read serial poll status byte
PRINT
PRINT "Serial Poll Status Byte";Sbyte
PRINT
!
OUTPUT @Osa;"*stb?"                              ! Read the Status Byte Register
ENTER @Osa;Stat
PRINT "Status Byte Register:";Stat
PRINT
!
OUTPUT @Osa;"*esr?"                              ! Read and clear the Std Event Status
Register
ENTER @Osa;Stat
PRINT "Standard Event Status Register:";Stat
PRINT
```

```
!
REPEAT
OUTPUT @Osa;"system:error?"                    ! Query error queue entries
ENTER @Osa;Errno;Error$
PRINT "Error Queue";Errno;Error$
UNTIL Errno=0                                  !Error=0 is end of error queue
!
LOCAL @Osa                                     ! Return to local operation
END
```

Example 10 SMSR calculation with display off

Description

This program calculates the side mode suppression ratio with the display off.

Program

```
! SMSR calculation with display off
!
CLEAR 7                                          ! Clear GPIB interface
ASSIGN @Osa TO 723;EOL CHR$(10) END               ! Turn on END message
CLEAR @Osa                                       ! Selected device clear
!
PRINT "SMSR example with display off"
OUTPUT @Osa;"disp:wind:text:data 'SMSR Measurement'"
!
!*****************************Initialization Routine*******************************
PRINT "Turning display off"
OUTPUT @Osa;"disp:wind off;*opc?"                ! Turn display off
ENTER @Osa;Temp
!
PRINT
PRINT "Presetting the instrument"
OUTPUT @Osa;"*rst;*opc?"
ENTER @Osa;Temp
!
PRINT "Turning buffer on"
OUTPUT @Osa;"syst:comm:gpib:buff on"             ! Turn OSA GPIB input buffer on
!
! assumes a light source at approx 1550 nm
OUTPUT @Osa;"SENS:WAV:CENT 1550nm"               ! Set center to 1550 nm
OUTPUT @Osa;"SENS:WAV:SPAN 20nm"                 ! Set span to 20 nm
OUTPUT @Osa;"INIT:IMM;*opc?"                     ! Take a sweep
ENTER @Osa;Temp
!
OUTPUT @Osa;"CALC:MARK:MAX"                      ! Place marker one at Max
OUTPUT @Osa;"CALC:MARK:SCEN"                     ! Set marker to center
OUTPUT @Osa;"CALC:MARK:SRL"                      ! Set marker to reference level
OUTPUT @Osa;"SENS:POW:DC:RANGe:LOW -61DBM"       ! Set sensitivity to -61 dBm
OUTPUT @Osa;"INIT:IMM;*opc?"                     ! Take a sweep
ENTER @Osa;Temp
OUTPUT @Osa;"CALC:MARK1:MAX"                     ! Set marker 1 to Max
!
OUTPUT @Osa;"CALC:MARK1:Y?"                      ! Get the peak amplitude
ENTER @Osa;Peakamp
OUTPUT @Osa;"CALC:MARK1:MAX:NEXT"                ! Set marker 1 to the next highest peak
OUTPUT @Osa;"CALC:MARK1:Y?"                      ! Get amplitude of the next highest
peak
ENTER @Osa;Nextpeakamp
!
PRINT
PRINT "Peak Amplitude",Peakamp;"dBm"
Smsr=Peakamp-Nextpeakamp                         ! Calculate Side Mode Suppression
mode
PRINT "SMSR = ",Smsr
!
PRINT
PRINT "Turning the display back on"
OUTPUT @Osa;"disp:wind on;*Opc?"                 ! Turn the display back on
ENTER @Osa;Temp
PRINT "*******Measurement complete*******"
```

```
LOCAL @Osa                    ! Return control to OSA
END
```

Front Panel Functions to Remote Commands

This is a table of the front-panel functions of the Agilent 86140B series and the corresponding remote commands.

Table 7 Front Panel Function to Remote Command for the Agilent 86140B Series (Sheet 1 of 7)

Front Panel Function	Remote Command
Amplitude	
Reference Level	DISPlay:WINDow:TRACe:Y:SCALe:RLEVel
Scale/Div	DISPlay:WINDow:TRACe:Y:SCALe:PDIVision
Display Mode	DISPlay:WINDow:TRACe:Y:SCALe:SPACing
Sensitivity	Automatic: SENSe:POWer:DC:RANGe:LOWer:AUTO Manual: SENSe:POWer:DC:RANGe:LOWer
Peak to Reference Level	CALCulate:MARKer:MAXimum CALCulate:MARKer:SRLevel
Trace Integ	CALCulate:TPOWer:STATe
Amplitude Setup	
Reference Level Position	DISPlay:WINDow:TRACe:Y:SCALe:RPOSition
Amplitude Units	UNITs:POWer
Auto Ranging	SENSe:POWer:DC:RANGe:AUTO
Auto Zero	CALibration:ZERO:AUTO
Auto Chop Mode[a]	SENSe:CHOP:STATe
Power Calibration	CALibration:POWer:STATe
Amplitude Correction Set	SENSe:CORRection:CSET
Amplitude Correction Mode	SENSe:CORRection:STATe
Applications	
Get a list of measurement modes and applications	INSTrument:CATalog? INSTrument:CATalog:FULL?
Select mode and application	INSTrument:SELect
Select the instrument mode or application using a numeric value	INSTrument:NSELect
Auto Align	CALibration:ALIGn CALibration:ALIGn:MARKer
Auto Measure	DISPlay:WINDow:TRACe:ALL:SCALe:AUTO

Table 7 Front Panel Function to Remote Command for the Agilent 86140B Series (Sheet 2 of 7)

Front Panel Function	Remote Command
Bandwidth/Sweep	
Res BW	SENSe:BANDwidth:RESolution:AUTO SENSe:BANDwidth:RESolution
Video BW	SENSe:BANDwidth:VIDeo:AUTO SENSe:BANDwidth:VIDeo
Sweep Time	SENSe:SWEep:TIME:AUTO SENSe:SWEep:TIME
Repeat Sweep	INITiate:CONTinuous
Single Sweep	INITiate:IMMediate
Select Path	ROUTe:Path[b]
More BW/Sweep Functions	
Trigger Mode	
Internal	TRIGger[:SEQuence]:SOURce
Gated	
External	TRIGger[:SEQuence]:SOURce
ADC+	TRIGger[:SEQuence]:SLOPe
ADC-	TRIGger[:SEQuence]:SLOPe
ADC AC	TRIGger[:SEQuence]:SLOPe
Trigger Delay	TRIGger[:SEQuence]:DELay
ADC Trig Sync	TRIGger[:SEQuence]:OUTPut
ADC Sync Out	TRIGger[:SEQuence]:OUTPut:PULSe:STATe
ADC Sync Out Duty Cycle	TRIGger[:SEQuence]:OUTPut:PULSe:DCYCle
ADC Sync Out Pulse Width	TRIGger[:SEQuence]:OUTPut:PULSe:WIDTh
Local	GPIB GoToLocal Command
Marker	
Active Marker	CALCulate:MARKer:STATe CALCulate:MARKer:AOFF
Active Trace	CALCulate:MARKer:TRACe
Peak Search	CALCulate:MARKer:MAXimum
Marker to CENTER	CALCulate:MARKer:SCENter
Marker to REF LEVEL	CALCulate:MARKer:SRLevel
Marker Setup	
Normal Marker Units	CALCulate:MARKer:X:READout
BW Marker Units	CALCulate:MARKer:FUNCtion:BANDwidth:READout
Delta Marker Units	CALCulate:MARKer:FUNCtion:DELTa:X:READout

Table 7 Front Panel Function to Remote Command for the Agilent 86140B Series (Sheet 3 of 7)

Front Panel Function	Remote Command
Normal/Delta Marker Interpolation	CALCulate:MARKer:INTerpolation
Bandwidth/Marker Interpolation	CALCulate:MARKer:FUNCtion:BANDwidth:INTerpolation
Peak Excursion	CALCulate:MARKer:PEXCursion:PEAK
Pit Excursion	CALCulate:MARKer:PEXCursion:PIT
Use Marker Search Threshold	CALCulate:THReshold
Noise Marker Reference Bandwidth	CALCulate:MARKer:FUNCtion:NOISe:BANDwidth
Peak Search at End of Each Sweep	
OSNR Noise	CALCulate:MARKer:FUNCtion:OSNR:MODE CALCulate:MARKer:FUNCtion:OSNR:OFFSet
More Marker Functions	
Marker Search Menu	
Search Mode Peak	
Peak Search	CALCulate:MARKer:MAXimum
Next Peak Down ↓	CALCulate:MARKer:MAXimum:NEXT
Next Peak Left ←	CALCulate:MARKer:MAXimum:LEFT
Next Peak Right →	CALCulate:MARKer:MAXimum:RIGHt
Active Marker	CALCulate:MARKer:STATe CALCulate:MARKer:AOFF
Search Mode Pit	
Pit Search	CALCulate:MARKer:MINimum
Next Pit Up –	CALCulate:MARKer:MINimum:NEXT
Next Pit Left ←	CALCulate:MARKer:MINimum:LEFT
Next Pit Right →	CALCulate:MARKer:MINimum:RIGHt
Active Marker	CALCulate:MARKer:STATe CALCulate:MARKer:AOFF
Marker BW	CALCulate:MARKer:FUNCtion:BANDwidth:STATe
Noise Marker	CALCulate:MARKer:FUNCtion:NOISe:STATe
Delta Marker	CALCulate:MARKer:FUNCtion:DELTa:STATe
OSNR Marker	CALCulate:MARKer[1|2|3|4]:FUNCtion:OSNR[:STATe]
Line Marker Menu	
Wavelength Marker 1	Integration Limit: CALCulate:TPOWer:IRANge:LOWer Search Limit: CALCulate:MARKer:SRANge:LOWer Sweep Limit: SENSe:WAVelength:SRANge:LOWer

Table 7 Front Panel Function to Remote Command for the Agilent 86140B Series (Sheet 4 of 7)

Front Panel Function	Remote Command
Wavelength Marker 2	Integration Limit: CALCulate:TPOWer:IRANge:UPPer Search Limit: CALCulate:MARKer:SRANge:UPPer Sweep Limit: SENSe:WAVelength:SRANge:UPPer
Line Markers Off	CALCulate:MARKer:SRANge Off
Advanced Line Marker Functions	
Integrate Limit	CALCulate:TPOWer:IRANge:STATe
Search Limit	CALCulate:MARKer:SRANge:STATe
Sweep Limit	SENSe:WAVelength:SRANge:STATe
Trace Integ	CALCulate:TPOWer:STATe
Preset	SYSTem:PRESet
Print	HCOPy:IMMediate
Save/Recall	
Save Menu	
Measurement	*SAV
Trace Only	MMEMory:STORe:TRACe
Save Graphics	
Graphic Format	HCOPy:DEVice:LANGuage
Save To	
File Name	
Recall Menu	
Measurement	*RCL
Trace	MMEMory:LOAD:TRACe
Recall From	
Delete Menu	MMEMory:DELete
Format Floppy Disk	MMEMory:INITialize
Backup/Restore Menu	
Backup Internal Memory	
Restore Internal Memory	
Fast Meas SAVE	
Fast Meas RECALL	
System	
Help	
Show Critical Errors	SYSTem:ERRor?
Show BW Errors	SYSTem:ERRor?

Table 7 Front Panel Function to Remote Command for the Agilent 86140B Series (Sheet 5 of 7)

Front Panel Function	Remote Command
Show Notices	
Show Warnings	SYSTem:ERRor?
Revision	*IDN?
Set Title	DISPlay:WINDow:TEXT:DATA
Options	
Current Source Setup	SOURce:CATalog? SOURce:CURRent:PULSe:STATe SOURce:CURRent:LIMit SOURce:STATe
Light Source Setup	SOURce:STATe
Printer Setup	
Printer Location	HCOPy:DESTination
Calibration	
Power Cal Setup	
Factory Power Cal Date	CALibration:DATE?
User Power Cal Date	CALibration:POWer:DATE?
Set Calibration Power	CALibration:POWer:VALue
Execute Power Calibration	CALibration:POWer
Calibrate Second Path[c]	CALibration:POWer:PATh[1|2] on|off
Wavelength Calibration Setup	
Factory Wavelength Cal Date	CALibration:DATE?
User Wavelength Cal Date	CALibration:WAVelength:DATE?
Signal Source Wavelengths Referenced In	SENSe:CORRection:RVELocity:MEDium
Set Calibration Wavelength	CALibration:WAVelength:VALue
Perform Calibration	CALibration:WAVelength:EXTernal CALibration:WAVelength:INTernal
Calibrate Second Path[c]	CALibration:WAV:PATh[1|2] on|off
Move Active Area	No command
More System Functions	
OSA State	
Display Setup	
Agilent Logo	
Date/Time	
Title	
Active Function Area Assist	

Table 7 Front Panel Function to Remote Command for the Agilent 86140B Series (Sheet 6 of 7)

Front Panel Function	Remote Command
Set Time/ Date	
Current Time	
Current Date	
Time Zone	
Service Menu	
Power State	SYSTem:PON:TYPE
Factory Preset	
Configure Network	
Firmware Upgrade	
Adv Service Functions	
Zero Now	CALibration:ZERO:AUTO
Grating Order	SENSe:GORDer:AUTO
Wavelength Limit	SENSe:WAVelength:LIMit
More Adv Service Functions	
TransZ 2 - 3 Lock	SENSe:POWer:DC:RANGe:LOCK
Multipoint Align	CALibration:ALIGn:EXTernal
Enhanced WL Cal Setup	
Perform EWC	CALibration:WAVelength:EXTernal
Calibration Range	CALibration:WAVelength:EWC
OSA Extended State	
Auto Measure Setup	
Span	DISPlay:WINDow:TRACe:X:SCALe:AUTO:SPAN DISPlay:WINDow:TRACe:X:SCALe:AUTO:SPAN:AUTO
Scale/div	DISPlay:WINDow:TRACe:Y:SCALe:AUTO:PDIVision DISPlay:WINDow:TRACe:Y:SCALe:AUTO:PDIVision:AUTO
Auto Meas at Marker	DISPlay:WINDow:TRACe:ALL:SCALe:AUTO:MARKer
Optimize Sensitivity	DISPlay:WINDow:TRACe:ALL:SCALe:AUTO:OPTimize
GPIB and Network Setup	
User Share Identity	
User Name	SYSTem:COMMunicate:NETWork:USERname
Password	SYSTem:COMMunicate:NETWork:PASSword
Workgroup	SYSTem:COMMunicate:NETWork:WORKgroup
File Shares	
1 through 4	MMEMory:FSHare:PATH

Table 7 Front Panel Function to Remote Command for the Agilent 86140B Series (Sheet 7 of 7)

Front Panel Function	Remote Command
Printer Shares	
1 through 4	HCOPy:DEVice:PSHare:ADDRess
Traces	
Active Trace Update and View commands below will affect active trace.	
Update A...F	TRACe:FEED:CONTrol
View A...F	DISPlay:WINDow:TRACe:STATe
Hold A...F	CALCulate[1 | 2 | 3 | 4 | 5 | 6]:MAXimum:STATe CALCulate[1 | 2 | 3 | 4 | 5 | 6]:MINimum:STATe
Reset Min/Max Hold	CALCulate[1 | 2 | 3 | 4 | 5 | 6]:MAXimum:CLEar CALCulate[1 | 2 | 3 | 4 | 5 | 6]:MINimum:CLEar
Trace Math	CALCulate:MATH:STATe
Default Math...	CALCulate:MATH:EXPRession:DEFine
Exchange Menu	TRACe:EXCHange
Offset	CALCulate[1 | 2 | 3 | 4 | 5 | 6]:OFFSet
All Math Off	CALCulate[1 | 2 | 3 | 4 | 5 | 6]:MATH:STATe
Averaging On | Off	CALCulate:AVERage:STATe
Trace Setup...	
Sweep Points	SENSe:SWEep:POINts
Wavelength	
Center Wavelength	SENSe:WAVelength:CENTer
Span	SENSe:WAVelength:SPAN
Start WL	SENSe:WAVelength:STARt
Stop WL	SENSe:WAVelength:STOP
Peak to Center	CALCulate:MARKer:MAXimum CALCulate:MARK:SCENter
Wavelength Setup	
Wavelength Units	
Wavelength Calibration	CALibrate:WAVelength:STATe
Wavelength Offset	SENSe:WAVelength:OFFSet
Center Wavelength Step Size	SENse:WAVelength:CENTer:STEP:INCRement
Wavelength Referenced In	SENSe:CORRection:RVELocity:MEDium

[a] Not available on the 86144B or 86146B.
[b] Available on the 86144B and 86146A only.
[c] For 86146B only.

2
Programming Commands

Introduction

This chapter is the reference for all Agilent 86140B commands. In accordance with IEEE 488.2, the instrument's commands are grouped into subsystems. Commands in each subsystem perform similar tasks. The following subsystems are provided:

Common commands
CALCulate
CALibration
DISPlay
FORMat
HCOPy
INITiate
INPut
INSTrument
MEMory
MMEMory
ROUTe
SENSe
SOURce
STATus
SYSTem
TRACe
TRIGger
UNIT

Command Conventions

Table 8 Notation Conventions and Definitions

Convention	Description
< >	Angle brackets indicate text strings entered by the developer.
[ ]	Square brackets indicate that the keyword DEFAULT can be used instead of a value or a variable for that parameter. Refer to the actual command description for the behavior when the DEFAULT keyword is used for a parameter.
|	Indicates a choice of one element from a list.
{ }	Braces indicate a group of constants to select from. Each constant is separated by the | character.
<string>	The <string> is used to represent quoted strings and allows any character in the ASCII 7 bit code (including nonprintable characters) to be transmitted as a message. This data field is particularly useful where text is to be displayed (for example, on a printer or CRT type device). The OSA accepts either single or double quotes.
name	Indicates the variable for which you provide a descriptive name. Any letter (Aa-Zz) followed by letters, digits (0-9) and underscore (_). Only the first 32 characters are significant.
<file_name>	File names must conform to standard MS-DOS®[a] file naming conventions.
<trace_name>	TRA, TRB, TRC, TRD, TRE, TRF
<data_block>	This parameter represents the arbitrary block program data as defined by IEEE 488.2. Arbitrary block program data allows any 8-bit bytes to be transmitted. This includes extended ASCII control codes and symbols. Two types of data blocks are defined: definite-length blocks and indefinite-length blocks. The definite-length block consists of a "#" character, followed by one digit (in ASCII) specifying the number of length bytes to follow, followed by the length (in ASCII), followed by length bytes of binary data. For example, two bytes of binary data would be sent as follows: #12<8 bit data byte><8 bit data byte> The indefinite-length block consists of a "#" character, followed by a "0" character (in ASCII), followed by any number of bytes of binary data. The data stream is terminated by a new line character with EOI set. For example, two bytes of binary data would be sent as follows: #0<8 bit data byte><8 bit data byte>NL^EOI
spec_min	–infinity. The parameter ***spec_min*** cannot be a variable, only a constant or DEFAULT.
spec_max	+infinity. The parameter ***spec_max*** cannot be a variable, only a constant or DEFAULT.
from	Start wavelength or frequency of trace in nm (default) or THz.

Table 8 Notation Conventions and Definitions

Convention	Description
to	Stop wavelength or frequency of trace in nm (default) or THz.
excursion	+excursion: means excursion dBs up (for example, from a pit). -excursion: means excursion dBs down (for example, from a peak).
ref_pt	The reference point to be used for a measurement keyword.

[a] MS-DOS is a U.S. registered trademark of Microsoft Corporation.

Command Trees

Common Commands

```
*CLS
*ESE <numeric_value>
*ESR?
*IDN?
*OPC
*OPT?
*RCL <numeric_value>|<filename>[INTernal|FLOPpy]
*RST
*SAV <numeric_value>|<filename>[INTernal|FLOPpy]
*SRE <numeric_value>
*STB?
*TST?
*WAI
```

CALCulate

```
:AVERage
    :CLEar
    :COUNt <numeric_value>
    [:STATe] ON|OFF|1|0
:CENTermass
    [:DATA]
    :STATe ON|OFF|1|0
:FWHM
    [:DATA]
    :STATe ON|OFF|1|0
:MARKer [1|2|3|4]
    :AOFF
    :FUNCtion
        :BWIDth|BANDwidth
            :INTerpolate ON|OFF|1|0
            :NDB <numeric_value>
            :READout FREQuency|WAVelength|TIME
            :RESult?
            [:STATe] ON|OFF|1|0
            :X
                :CENTer?
                :LEFT?
                :RIGHt?
        :DELTa
            :RESet
            [:STATe]
            :X
                :OFFSet?
                    :FREQuency <numeric_value>
                    :TIME <numeric_value>
                    [:WAVelength] <numeric_value>
                :READout FREQuency|WAVelength|TIME
                :REFerence?
            :Y
                :OFFSet?
```

```
                :REFerence?
            :NOISe
                :BWIDth|BANDwidth
                :RESult?
                [:STATe] ON|OFF|1|0
            :OSNR
                :MODE PIT|MANual|AUTO
                :OFFSet <numeric_value> M|UM|NM|A
                :RESult?
                [:STATe] ON|OFF|1|0
                :X
                    :CENTer?
                    :LEFT?
                    :RIGHt?
                :Y
                    :CENTer?
                    :LEFT?
                    :RIGHt?
            :PRESet
        :INTerpolate ON|OFF|1|0
        :MAXimum
            :LEFT
            :NEXT
            :RIGHt
        :MINimum
            :LEFT
            :NEXT
            :RIGHt
        :PEXCursion
            [:PEAK] <numeric_value>
            :PIT <numeric_value>
        :SCENter
        :SRANge
            :LOWer?
                :FREQuency <param>
                :TIME <param>
                [:WAVelength] <param>
            [:STATe] ON|OFF|1|0
            :UPPer?
                :FREQuency <param>
                :TIME <param>
                [:WAVelength] <param>
        :SRLevel
        [:STATe] ON|OFF|1|0
        :TRACe TRA|TRB|TRC|TRD|TRE|TRF
        :X?
            :FREQuency <numeric_value>
            :READout FREQuency|WAVelength|TIME
            :TIME <numeric_value>
            [:WAVelength] <param>
        :Y?
    :MATH
        [:EXPRession][:DEFine] (<expression>)
        :STATe ON|OFF|1|0
    :MAXimum
        :CLEar
```

```
    [:STATe] ON|OFF|1|0
:MEAN
    [:DATA]?
    :RANGe
        :LOWer?
            :FREQuency <numeric_value>[HZ|KHZ|MHZ|GHZ|THZ]
            :TIME <numeric_value>[NS|US|MS|S]
            [:WAVelength] <numeric_value>[M|UM|NM|A]
        [:STATe] ON|OFF|1|0
        :UPPer?
            :FREQuency <numeric_value>[HZ|KHZ|MHZ|GHZ|THZ]
            :TIME <numeric_value>[NS|US|MS|S]
            [:WAVelength] <numeric_value>[M|UM|NM|A]
    :STATe ON|OFF|1|0
:MINimum
    :CLEar
    [:STATe] ON|OFF|1|0
:OFFset <param>
:POWer[:DATA]?
:SIGMa[:DATA]?
:THReshold <param>[W|MW|UW|DBM]
    :STATe ON|OFF|1|0
:TPOWer
    [:DATA]?
    :IRANge
        :LOWer
        [:STATe]
        :UPPer
    :STATe
```

CALibration

```
:ALIGn
    [:AUTO]:MARKer
    EXTernal
    :FILTER
    :INTernal
    :MARKer
    :PRESet
    :TADD
:DATE?
:POWer
    :DATE?
    :PATH OFF|ON|0|1
    :EXTernal
    :STATe ON|OFF|1|0
    :VALue <param>
    :WAVelength <numeric_value>[M|UM|NM|A|HZ|KHZ|MHZ|GHZ]
:PRESet
:STATe OFF|ON|0|1
:WAVelength
    :DATE?
    :EWC
        :FUNCtion OFF|ON|0|1
        :RANGe FULL|TELEcom
    [:EXTernal]
```

```
        :MULTipoint
            :DATA
            :DELete
            :MARKer
        [:NORMal]
            :MARKer
        :VALue
    :INTernal
        :MULTipoint
        [:NORMal]
    :MODE NORMal|MULTipoint
    :PATH OFF|ON|0|1
    :STATe OFF|ON|0|1
    :USER:DATA <string>
:ZERO[:AUTO] OFF|ON|0|1|ONCE
```

DISPlay

```
[:WINDow]
[:WINDow[1]]
    :ANNotation[:ALL] OFF|ON|0|1
    :POPup[1|2|3|4][:ALL] OFF|ON|0|1
    :TEXT
        :CLEar
        :DATA <string> | <data_block>
    :TRACe
        :ALL [:SCALe][:AUTO]
            :MARKer OFF|ON|0|1
            :OPTimize OFF|ON|0|1
        :GRATicule:GRID[:STATe] OFF|ON|0|1
        [:STATe] <trace_name>, OFF|ON|0|1
        :X[:SCALe]:AUTO:SPAN <numeric_value> [M|NM|UM]
            ;AUTO OFF|ON|0|1
        :Y[SCALe]
            :AUTO:PDIVision <numeric_value>[DB]
            :SPACing LINear|LOGarithmic
        :Y[1|2][SCALe]
            :AUTO:PDIVision:AUTO OFF|ON|0|1
            :LINear OFF|ON|0|1
            :PDIVision <numeric_value>[DB]
            :RLEVel <numeric_value>[DBM|W|MW|UM|NW|DB]
            :RPOSition <numeric_value>
```

FORMat

```
[:DATA] REAL[32,64]|ASCii
```

HCOPy

```
:DATA?

:DESTination
“SYSTem:COMMunicate:CENTronics”|”SYSTem:COMMunicate:FSHare[1|2|3|
4]”|
”SYSTem:COMMunicate:INTernal”

:DEVice
```

```
    :LANGuage <PCL|CGM>
    :PSHare[1|2|3|4]
        :ADDRess
        [:PATH]<param>
:IMMediate
```

INITiate

```
:CONTinuous OFF|ON|0|1
:IMMediate]
```

INPut

```
:FILTer
    :MAXimum
        :LEFT
        :NEXT
            :RIGHt
    :MINimum
        :LEFT
        :NEXT
            :RIGHt
:   :SCENT
    :SRLevel
    :X
    :Y?
```

INSTrument

```
:CATalog?
    :FULL?
:NSELect <numeric value>
:SELect
```

MEMory

```
:STATe[:EXTended]?
```

MMEMory

```
:CATalog? FSHare[1|2|3|4]|INTernal|FLOPpy
:DATA <file_name>, <data_block>
:DELete <file_name> FSHare[1|2|3|4]|INTernal|FLOPpy
:FSHAre[1|2|3|4]
    :ADDRess
    [:PATH]
:INITialize [:INTernal|FLOPpy]
:LOAD:TRACe <trace_name>, <file_name>
FSHare[1|2|3|4]|INTernal|FLOPpy
:STORe:TRACe <trace_name>, <file_name>
FSHare[1|2|3|4]|INTernal|FLOPpy
```

ROUte

```
:PATH <INTernal|EXTernal>
```

SENSe

```
:BANDwidth|BWIDth
    [:RESolution]
        :AUTO OFF|ON|0|1
        :RATio <numeric_value>
    :VIDeo <numeric_value> [HZ|KHZ|MHZ|GHZ]
        :AUTO OFF|ON|0|1
:CHOP[:STATe] OFF|ON|0|1
:CORRection
    :CSET[:SELect] 1|2|3|4
    :DATA wvl,ratio{wvl,ratio...}
    :RVELocity:MEDium
    :STATe OFF|ON|0|1
    :X:SPACing LOG|LIN
:GORDer[:AUTO] OFF|ON|0|1
:POWer[:DC]:RANGe
    :AUTO OFF|ON|0|1
    :LOCK OFF|ON|0|1
    :LOWer <numeric_value> [DBM|W|MW|UW|NW]
        :AUTO OFF|ON|0|1
:SWEep
    :POINts <numeric_value>
    :TIME
        :AUTO OFF|ON|0|1
[:WAVelength]
    :CENTer <numeric_value> [M|UM|NM|PM|A|HZ|KHZ|MHZ|GHZ|THZ]
        :STEP
            :AUTO OFF|ON|0|1
            [:INCRement] <numeric_value> [M|UM|NM|A]
    :LIMit OFF|ON|0|1
    :OFFSet <numeric_value> [M|UM|NM|A]
    :SPAN <numeric_value> [M|UM|NM|A]
        :FULL
    :SRANge
        :LOWer <numeric_value>
[M|UM|NM|PM|A|HZ|KHZ|MHZ|GHZ|THZ]
        [:STATe] OFF|ON|0|1
        :UPPer <numeric_value>
[M|UM|NM|PM|A|HZ|KHZ|MHZ|GHZ|THZ]
    :STARt <numeric_value> [M|UM|NM|PM|A|HZ|KHZ|MHZ|GHZ|THZ]
    :STOP <numeric_value> [M|UM|NM|PM|A|HZ|KHZ|MHZ|GHZ|THZ]
```

SOURce[n]

```
:CATalog?
:CURRent
    [:LEVel][:IMMediate][:AMPLitude] <param>
    :LIMit[:AMPLitude] <param>
    :PULSe:STATe OFF|ON|0|1
:PULSe
    :DCYCle <param>
    :WIDTh <param>
:STATe <identifier> OFF|ON|0|1
```

STATus

```
:OPERation
    :CONDition?
    ENABle
    [:EVENt]?
    :NTRansition <int_value>
    :PTRansition <int_value>
:PRESet
:QUEStionable
    :CONDition?
    :ENABle <int_value>
    [:EVENt]?
```

SYSTem

```
:COMMunicate
    :GPIB[:SELF]:BUFFer OFF|ON|0|1
    :NETWork
        :PASSword
        :USERname
        :WORKgroup <oaran>
:DATE?
:ERRor:[:NEXT]?
:HELP:HEADers
:PON[:TYPE] PRESet|LAST
:PRESet
:TIME?
:TZONe:NAME?
:VERSion?
```

TRACe

```
[:DATA]
    :X
        :STARt? <trace_name>
        :STOP? <trace_name>
        :TIMe:SSTop <trace_name>,<numeric_value>,<numeric_value>
        :TYPE? <trace_name>
        [:WAVelength]:SSTop <trace_name>,<numeric_value>,<numeric_value>
    [:Y]? <trace_name>
        :[:POWER]
<trace_name>,<data_block>|<numeric_value>{,<numeric_value>}
        :RATio
<trace_name>,<data_block>|<numeric_value>{,<numeric_value>}
:EXCHange <trace_1> | <trace_2>
:FEED:CONTrol <trace_name>, ALWays | NEVer
:POINts <trace_name>,<numeric_value>
```

TRIGger

```
[:SEQuence]
    :DELay <numeric_value> [S|MS|US|NS]
    :OUTPut OFF|ON|0|1
        :PULSe
            :DCYCle <numeric_value>
```

```
            :STATe OFF|ON|0|1
            :WIDTh <numeric_value> [S|MS|US|NS]
        :SLOPe POSitive|NEGative|EITHer
        :SOURce IMMEdiate|EXTernal|INTernal
```

UNIT

```
:POWer DBM|W
:RATio DB|LINear|AUTO
```

Common Commands

Common commands control generic device functions that are common among many different types of instruments. They can be received and processed by the instrument whether they are sent over the GPIB as separate program messages of within other program messages.

Common Commands

*CLS
*ESE <numeric_value>
*ESR?
*IDN?
*OPC
*OPT?
*RCL <numeric_value>|<filename>[INTernal|FLOPpy]
*RST
*SAV <numeric_value>|<filename>[INTernal|FLOPpy]
*SRE <numeric_value>
*STB?
*TST?
*WAI

*CLS (Clear Status)

Clears all the event status registers summarized in the status byte register. This command resets the status data structure. It does this by emptying the error queue and clearing all bits in all of the event registers.

Syntax

```
*CLS
```

Example

This example clears the status data structures of the Agilent 86140B series.

```
10 OUTPUT 723;"*CLS"
20 END
```

*ESE (Event Status Enable)

Sets the bits in the Standard Event Enable register.
The Standard Event register monitors GPIB errors and synchronization conditions such as operation complete, request control, query error, device dependent error, execution error, command error, and power on. The parameter is rounded to an integer value and interpreted as a binary number, representing the bit values of the register.

The Standard Event Status Enable Register contains a mask value for the bits to be enabled in the Standard Event Status Register. A "1" in the Standard Event Status Enable Register enables the corresponding bit in the Standard Event Status Register. A "0" in the enable register disables the corresponding bit.

The *ESE? query returns the current contents of the Standard Event Status Enable Register as an integer from 0 to 255.

Table 9 Standard Event Status Enable Register Bits

Bit	Weight	Enables	Definition
7	128	PON - Power On	Indicates power is turned on.
6	64		Not used. Permanently set to zero.
5	32	CME - Command Error	Indicates whether the parser detected an error.
4	16	EXE - Execution Error	Indicates whether a parameter was out-of-range, or was inconsistent with the current settings.
3	8	DDE - Device Dependent Error	Indicates whether the device was unable to complete an operation for device-dependent reasons.
2	4	QYE - Query Error	Indicates if the protocol for queries has been violated.
1	2	RQC - Request Control	Indicates whether the device is requesting control.
0	1	OPC - Operation Complete	Indicates whether the device has completed all pending operations.

Syntax

```
*ESE <integer>
*ESE?
```

<integer> is a mask from 0 to 255.

Example

This example enables the User Request (URQ) bit of the Standard Event Status Enable Register. When this bit is enabled and a front-panel key is pressed, the Event Summary bit (ESB) in the Status Byte Register is also set.

```
10 OUTPUT 723;"*ESE 64"
20 END
```

*ESR? (Event Status Register)

Reads and clears the Standard Event Status register.

The register is cleared when it is read. The response value is an integer, to be interpreted as a binary number, representing the bit values of the register.

When the standard event status register is read, the value returned is the total of the weights of all the bits that are set to one at the time the byte is read. The following table shows each bit in the event status register and its bit weight. The register is cleared when it is read.

The query response value is an integer, to be interpreted as a binary number, representing the bit values of the register. The integer ranges from 0 to 255.

Bit	Bit Weight	Bit Name	Condition
7	128	PON	1 = OFF to ON transition has occurred.
6	64		Not Used. Permanently set to zero.
5	32	CME	0 = no command errors. 1 = a command error has been detected.
4	16	EXE	0 = no execution error. 1 = an execution error has been detected.
3	8	DDE	0 = no device-dependent errors. 1 = a device-dependent error has been detected.
2	4	QYE	0 = no query errors. 1 = a query error has been detected.
1	2	RQC	0 = request control - NOT used - always 0.
0	1	OPC	0 = operation is not complete. 1 = operation is complete.
	0 = False = Low		1 = True = High

Syntax

*ESR?

*IDN? (Identification Number)

The *IDN? query returns a string value which identifies the instrument type and firmware revision.

An *IDN? query must be the last query in a program message. Any queries after the *IDN? query in a program are ignored.

The maximum length of the identification string is 50 bytes.

The return string is a comma-separated list consisting of Manufacturer, Model Number, Serial Number, and Firmware Revision.

Syntax

*IDN?

Example

This example places the Agilent 86140B series's identification information in the string variable, Identify$, then prints the identification information to the computer screen.

```
10 DIM Identify$[50]!Dimension variable
20 OUTPUT 723;"*IDN?"
30 ENTER 723;Identify$
40 PRINT Identify$
50 END
```

Related Key

REVISION

*OPC (Operation Complete)

Sets bit 0 in the Standard Event Status register when all pending operations have finished.

This command is useful when the computer is sending commands to other instruments. The computer can poll the event status register to check when the instrument has completed the operation. Use the *OPC? query to ensure all operations have completed before continuing the program. By following a command with an *OPC? query and an ENTER statement, the program will pause until the response (ASCII 1) is returned by the instrument.

Be sure the computer's time-out limit is at least two seconds, since some of the instrument commands take approximately one second to complete.

The *OPC query allows synchronization between the computer and the Agilent 86140B series by using the message available (MAV) bit in the

Status Byte, or by reading the output queue. Unlike the *OPC command, the *OPC query does not affect the OPC Event bit in the Standard Event Status Register.

The *OPC? query places an ASCII character "1" in the Agilent 86140B series's output queue when all pending selected device operations have finished.

Syntax

```
*OPC
*OPC?
```

Example

This example sets the operation complete bit in the Standard Event Status Register when the PRINT operation is complete.

```
10 OUTPUT 723;":PRINT;*OPC"
20 END
```

*OPT? (Option)

The OPT? query returns a string with a list of installed options. If no options are installed, the string will have a 0 as the first character.

The return string is a comma-separated list that identifies the optical spectrum analyzer's option configuration. A 0 indicates no options are present.

The length of the returned string may increase as options become available in the future. Once implemented, an option name will be appended to the end of the returned string, delimited by a comma.

Syntax

```
*OPT?
```

Example

This example places all options into the string variable, Options$, then prints the option model and serial numbers to the computer's screen.

```
10 DIM Options$[100]
20 OUTPUT 723;"*OPT?"
30 ENTER 723;Options$
40 PRINT Options$
50 END
```

*RCL (Recall)

Recalls previously saved instrument settings from the requested register or file.

The *RCL command restores the state of the Agilent 86140B series to a setup previously stored in the specified save/recall register. A Agilent 86140B series setup must have been stored previously in the specified register. Registers 0 through 9 are general-purpose registers and can be used by the *RCL command. When using the *SAV command, you can save states using an 8 character, caps insensitive, name.

An integer, 0 through 9, specifies the save/recall register that contains the Agilent 86140B series setup you want to recall.

NOTE The auto span value will not be saved with the measurement.

Syntax

*RCL 0-9| "NAME"
*RCL <filename>, [INTernal| FLOPPY|FSHARE1| FSHARE2| FSHARE3| FSHARE4]

Example

This example restores the Agilent 86140B series to the Agilent 86140B series setup stored in register 3.

```
10 OUTPUT 723;"*RCL 3"
20 END
```

Related Commands

*SAV. An error message appears on the Agilent 86140B series display if nothing has been previously saved in the specified register.

Related Key

(RECALL MENU) MEASUREMENT (TRACE AND STATE)|TRACE (DATA ONLY)

*RST (Reset)

The *RST command returns the instrument to a known condition.

Executes a device reset for single sweep mode and returns the instrument to a known state.

Syntax

*RST

Related Commands

SYSTem:PRESet (for continuous sweep mode)

*SAV (Save)

Saves instrument settings to the designated register or file. An integer, 0 through 9, specifies which register to save the current Agilent 86140B series setup to.

Syntax

*SAV 0-9| "NAME", [INT| FLOP| FSH1|FSH2|FSH3|FSH4]

Example

This example stores the current Agilent 86140B series setup to register 3.

```
10 OUTPUT 723;"*SAV 3"
20 END
```

This example stores the current OSA setup onto the floppy disk as data.dat

```
10 OUTPUT 723;"*SAV 'data', FLOP"
20 END
```

Related Commands

*RCL (Recall)

Related Key

(SAVE MENU) MEASUREMENT (TRACE AND STATE)|TRACE (DATA ONLY)

*SRE (Service Request Enable)

The *SRE command sets the bits in the Service Request Enable register.

This parameter is rounded to an integer value and interpreted as a binary number, representing the bit values of the register. The Service Request Enable register serves as a mask for the Status Byte. When a bit in the Status Byte goes to 1, if the corresponding bit in the Service Request Enable register is a 1, the instrument asserts the Service Request line on the GPIB.

An integer, 0 to 255, represents a mask value for the bits enabled in the Service Request Enable Register.

The *SRE? query returns the current contents of the Service Request Enable Register.

Syntax

```
*SRE
*SRE?
```

Example

This example enables a service request to be generated when a message is available in the output queue. When a message is available, the MAV bit is high.

```
10 OUTPUT 723;"*SRE 16"
20 END
```

*STB? (Status Byte)

Returns the current value of the instrument's Status Byte.

The master summary status (MSS) bit 6 indicates whether or not the device has at least one reason for requesting service.

This will not change the Status Byte register. The response value is an integer, to be interpreted as a binary number, representing the bit values of the register. Performing a serial poll on the instrument also reads the Status Byte register, except that bit 6 indicates whether there is a service request that has not been serviced. The most convenient way to clear the Status Byte register is to send a *CLS command. The Status Byte register summarizes the states of the other register sets. It is also responsible for generating service requests.

*STB? returns an integer, 0 to 255, representing a mask value for the bits enabled in the Status Byte.

Bit	Bit Weight	Bit Name	Condition
7	128	OPER	0 = no enabled operation status conditions have occurred 1 = an enabled operation status condition has occurred
6	64	RQS/MSS	0 = Agilent 86140B series has no reason for service 1 = Agilent 86140B series is requesting service
5	32	ESB	0 = no event status conditions have occurred 1 = an enabled event status condition occurred
4	16	MAV	0 = no output messages are ready 1 = an output message is ready
3	8	—	0 = not used
2	4	MSG	0 = no message has been displayed 1 = message has been displayed
1	2	USR	0 = no enabled user event conditions have occurred 1 = an enabled user event condition has occurred
0	1	TRG	0 = no trigger has occurred 1 = a trigger occurred
0 = False = Low			1 = True = High

Syntax

*STB?

Example

This example reads the contents of the Status Byte into the numeric variable, Value, then prints the value of the variable to the computer's screen.

```
10 OUTPUT 723;"*STB?"
20 ENTER 723;Value
30 PRINT Value
40 END
```

Related Command

*CLS

*TST? (Test)

Tests the analyzer interface hardware and returns 0 if the interface is funcitonal.

Syntax

*TST?

Example

This example performs a self-test on the OSA and places the results in the numeric variable, Results. The program then prints the results to the computer screen.

```
10 OUTPUT 723;"*TST?"
20 ENTER 723;Results
30 PRINT Value
40 END
```

*WAI (Wait-to-Continue)

Prevents the instrument from executing any further commands until the current command has finished executing.

The *WAI command ensures that overlapped commands are completely processed before subsequent commands, those sent after the *WAI command, are processed. *This command is not needed by the Agilent 86140B series in non-buffered mode, since all commands are non-overlapped.*

Syntax

*WAI

Example

This example executes a single acquisition, and causes the instrument to wait until acquisition is complete before executing any additional commands.

```
10 OUTPUT 723;"SINGle;*WAI"
20 END
```

CALCulate Subsystem Commands

The CALCulate subsystem performs post-acquisition data processing. The CALCulate subsystem operates on data acquired by a SENSe function.

NOTE CALC: is interpreted as CALC1:. CALC1 controls TRA, CALC2 controls TRB, CALC3 controls TRC, CALC4 controls TRD, CALC5 controls TRE, and CALC6 controls TRF.

The commands in this subsystem have the following command hierarchy:

CALCulate

```
:AVERage
    :CLEar
    :COUNt <numeric_value>
    [:STATe] ON|OFF|1|0
:CENTermass
    [:DATA]
    :STATe ON|OFF|1|0
:FWHM
    [:DATA]
    :STATe ON|OFF|1|0
:MARKer [1|2|3|4]
    :AOFF
    :FUNCtion
        :BWIDth|BANDwidth
            :INTerpolate ON|OFF|1|0
            :NDB <numeric_value>
            :READout FREQuency|WAVelength|TIME
            :RESult?
            [:STATe] ON|OFF|1|0
            :X
                :CENTer?
                :LEFT?
                :RIGHt?
        :DELTa
            :RESet
            [:STATe]
            :X
                :OFFSet?
                    :FREQuency <numeric_value>
                    :TIME <numeric_value>
                    [:WAVelength] <numeric_value>
                :READout FREQuency|WAVelength|TIME
                :REFerence?
            :Y
```

```
                :OFFSet?
                :REFerence?
            :NOISe
                :BWIDth|BANDwidth
                :RESult?
                [:STATe] ON|OFF|1|0
            :OSNR
                :MODE PIT|MANual|AUTO
                :OFFSet <numeric_value> M|UM|NM|A
                :RESult?
                [:STATe] ON|OFF|1|0
                :X
                    :CENTer?
                    :LEFT?
                    :RIGHt?
                :Y
                    :CENTer?
                    :LEFT?
                    :RIGHt?
            :PRESet
        :INTerpolate ON|OFF|1|0
        :MAXimum
            :LEFT
            :NEXT
            :RIGHt
        :MINimum
            :LEFT
            :NEXT
            :RIGHt
        :PEXCursion
            [:PEAK] <numeric_value>
            :PIT <numeric_value>
        :SCENter
        :SRANge
            :LOWer?
                :FREQuency <param>
                :TIME <param>
                [:WAVelength] <param>
            [:STATe] ON|OFF|1|0
            :UPPer?
                :FREQuency <param>
                :TIME <param>
                [:WAVelength] <param>
        :SRLevel
        [:STATe] ON|OFF|1|0
        :TRACe TRA|TRB|TRC|TRD|TRE|TRF
        :X?
            :FREQuency <numeric_value>
            :READout FREQuency|WAVelength|TIME
            :TIME <numeric_value>
            [:WAVelength] <param>
        :Y?
    :MATH
        [:EXPRession][:DEFine] (<expression>)
        :STATe ON|OFF|1|0
    :MAXimum
```

```
    :CLEar
    [:STATe] ON|OFF|1|0
:MEAN
    [:DATA]?
    :RANGe
        :LOWer?
            :FREQuency <numeric_value>[HZ|KHZ|MHZ|GHZ|THZ]
            :TIME <numeric_value>[NS|US|MS|S]
            [:WAVelength] <numeric_value>[M|UM|NM|A]
        [:STATe] ON|OFF|1|0
        :UPPer?
            :FREQuency <numeric_value>[HZ|KHZ|MHZ|GHZ|THZ]
            :TIME <numeric_value>[NS|US|MS|S]
            [:WAVelength] <numeric_value>[M|UM|NM|A]
    :STATe ON|OFF|1|0
:MINimum
    :CLEar
    [:STATe] ON|OFF|1|0
:OFFset <param>
:POWer[:DATA]?
:SIGMa[:DATA]?
:THReshold <param>[W|MW|UW|DBM]
    :STATe ON|OFF|1|0
:TPOWer
    [:DATA]?
    :IRANge
        :LOWer
        [:STATe]
        :UPPer
    :STATe
```

CALCulate[1|2|3|4|5|6]:AVERage:CLEar

Causes the average data to be cleared and the average counter to be reset to zero.

Syntax

```
CALC[1|2|3|4|5|6]:AVER:CLE
```

CALCulate[1|2|3|4|5|6]:AVERage:COUnt

Sets the number of measurements to be averaged. When the number of measurements taken is less than the count, the following formula is used to calculate the data.

$$\text{AVG} = \frac{\text{sum of all measurements}}{\text{number of measurements}}$$

If the number of measurements is greater than or equal to the count, the following formula is used to calculate the data:

$$\text{New average} = \frac{\text{count} - 1}{\text{count}} \times \text{last average} + \frac{\text{new measurement}}{\text{count}}$$

Syntax

CALC[1|2|3|4|5|6]:AVER:COUN <numeric_value>

CALC[1|2|3|4|5|6]:AVER:COUN?

Related Key

AVERAGING ON|OFF

CALCulate[1|2|3|4|5|6]:AVERage[:STATe]

Turns trace averaging on and off. CACLulate[1|2|3|4|5|6] refers to traces A, B, C, D, E, and F respectively.

Each trace can be set to one of four mutually exclusive states: normal, average (CALCulate[1|2|3|4|5|6]:AVERage[:STATe]), maximum hold (CALCulate[1|2|3|4|5|6]:MAXimum[:STATe]), or minimum hold (CALCulate[1|2|3|4|5|6]:MINimum[:STATe]). Enabling one state will disable any previous setting. For example, turning on trace averaging for trace B (CALCulate2:AVERage ON) would turn off a minimum hold state (CALCulate2:MINimum OFF).

Trace averaging, maximum, and minimum holds can be used in conjunction with trace math. If the trace that you wish to average is the result of a math expression (C=log A–B), the result of the math expression will be averaged. TRC will be the averaged difference of TRA and TRB.

Trace averaging is a point-by-point average of the current data with previous data.

Syntax

```
CALC[1|2|3|4|5|6]:AVER[:STAT] OFF|ON|0|1

CALC[1|2|3|4|5|6]:AVER[:STAT]?
```

Example

OUTPUT @OSA;"INIT:CONT off"	! go to single sweep mode
OUTPUT @OSA;"CALC1:AVER:STAT on"	! turns averaging on trace A
OUTPUT @OSA;"CALC1:AVER:COUN 10"	! set averaging for 10 sweep
OUTPUT @OSA;"INIT:IMM;*OPC?"	! take sweep
ENTER @OSA; Temp	! trace A is averaged over 10 sweeps
OUTPUT @OSA; "CALC1:AVER:CLEAR"	! clears average data, averaging remains on
OUTPUT @OSA; "CALC1:AVER:STAT off"	! turn averaging off

Related Key

AVERAGING ON|OFF

CALCulate[1|2|3|4|5|6]:CENTermass:[DATA]?

Returns the Center of Mass calculation results in meters.

Trace A corresponds to CALCulate1 and so on. Corrections to all calculations are made for the slope and variation of the resolution bandwidth filter over the wavelength range of the trace. When CALCulate[1|2|3|4|5|6]:TPOWer:IRANge is on, the calculation is performed over the upper and lower range limits. All calculations (SOURce, CENTermass, FWHM, SIGMA, and TPOWer) share the same line marker limits. Sending this query when the CALCulate[1|2|3|4|5|6]:CENTermass :STATe is off will generate an error.

Syntax

```
CALC:CENT:[DATA]?
```

Example

OUTPUT @Osa;"calc1:cent:state on"	! turn center of mass calculation on for trace a
OUTPUT @Osa;"calc1:cent:data?"	! query for center of mass calculation results
ENTER @Osa;Data	! place center of mass data into variable
PRINT Data	! print Center of Mass value

CALCulate[1|2|3|4|5|6]:CENTermass:STATe

Turns the Center of Mass (Mean Wavelength) calculation on.

Trace A corresponds to CALCulate1 and so on. Only one Center of Mass calculation may be ON. For instance, if Center of Mass is ON for Trace A, turning Center of Mass ON for Trace B will disable the Center of Mass calculation for Trace A. The data is returned in either ASCII or binary form as determined by the FORMat:DATA command.

The Center of Mass of the trace points is normalized by the ratio of the trace point spacing and the resolution bandwidth. The power and wavelength of each trace point are used to calculate the Center of Mass. The formula used is:

$$Center\ of\ Mass = \sum_{i=1}^{n} \frac{P_i}{P_o} \left(\frac{trace\ point\ spacing}{resolution\ bandwidth} \right) \lambda_i$$

where:

λ_i is the wavelength of a single trace point

P_i is the power of a single trace point

P_o is total power as defined below

Total Power is the summation of the power at each trace point, normalized by the ratio of the trace point spacing and the resolution bandwidth. The formula used is:

$$Total\ Power = \sum_{i=1}^{n} P_i \left(\frac{trace\ point\ spacing}{resolution\ bandwidth} \right)$$

where:

P_i is the power of a single trace point

Syntax

```
CALC:CENT:STAT OFF|ON|0|1
```

Example

OUTPUT @Osa;"calc1:cent:state on"	! turn center of mass calculation on for trace a
OUTPUT @Osa;"calc1:cent:data?"	! query for center of mass calculation results
ENTER @Osa;Data	! place center of mass data into variable
PRINT Data	! print Center of Mass value

CALCulate[1|2|3|4|5|6]:FWHM[:DATA]?

Returns the FWHM calculation results in meters.

Trace A corresponds to CALCulate 1 and so on. Corrections to all calculations are made for the slope and variation of the resolution bandwidth filter over the wavelength range of the trace. When CALCulate[1|2|3|4|5|6]:TPOWer:IRANge is on, the calculation is performed over the upper and lower range limits. All calculations (SOURce, CENTermass, FWHM, SIGMa, and TPOWer) share the same line marker limits. Sending this query when the CALCulate[1|2|3|4|5|6]:FWHM:STATe is off will generate an error.

Syntax

```
CALC:FWHM: [DATA]?
```

Example

OUTPUT @OSA;"calc1:fwhm:state on"	! turn full width half max on for trace a
OUTPUT @OSA;"calc1:fwhm:data?"	! query for fwhm calculation results
ENTER @OSA; Data	! place fwhm data into variable
PRINT Data	! print Full Width Half Max value

CALCulate[1|2|3|4|5|6]:FWHM:STATe

Turns the Full Width Half Maximum (FWHM) calculation on.

Trace A corresponds to CALCulate 1 and so on. Only one of each calculation may be on. For instance if FWHM is on for Trace A, turning FWHM on for Trace B will disable the FWHM calculation for Trace A. The data is returned in either ASCII or binary form as determined by the FORMat:DATA command.

FWHM (Full Width at Half Maximum) describes the spectral width of the half-power (-3 dB) points of the trace, assuming a continuous, Gaussian power distribution. The half-power points are those where the power spectral density is one-half that of the peak amplitude. The formula used is:

$$FWHM = 2{,}355\sigma$$

where: σ is sigma as defined in CALCulate[1|2|3|4|5|6]:SIGMa

Syntax

CALC:FWHM:STAT OFF|ON|0|1

Example

OUTPUT @OSA;"calc1:fwhm:state on"	! turn full width half max on for trace a
OUTPUT @OSA;"calc1:fwhm:data?"	! query for fwhm calculation results
ENTER @OSA; Data	! place fwhm data into variable
PRINT Data	! print Full Width Half Max value

CALCulate:MARKer:AOFF

This command is not available for the filter mode application in the Agilent 86144B/86146B, because this command turns off all markers, including filter mode tune marker.

Turns off all markers and marker functions.

NOTE

Going to zero span will turn off all markers. This is because markers are referenced to a particular time or wavelength *not* a particular display position. Going out of zero span will restore the markers to the state they were in before going to zero span. Changing to or from zero span changes the fundamental units for the X-axis.

If no marker number is given in the following marker commands, the command is interpreted as referring to marker number 1. For example, CALC:MARK ON is equivalent to CALC:MARK1 ON.

Syntax

CALC:MARK:AOFF

Example

OUTPUT @Osa;"calc:mark:aoff"	! all markers turned off

Related Keys

ACTIVE MARKER 1|2|3|4|OFF

CALCulate:MARKer[1|2|3|4]:FUNCtion:BWIDth|BANDwidth :INTerpolate

Turns the bandwidth marker interpolation on or off.

When interpolation is on, the bandwidth markers will be placed at the exact number of dB (NDB) setting from the normal marker if the trace data allows. The position of the marker will be linearly interpolated between two true trace data points. The default state is on. If interpolate is off, for negative NDB values, the bandwidth markers will be at values closest to and more negative than the NDB value. For positive NDB values, the bandwidth markers will be at values closest to and more positive than the NDB values. This will typically result in a wider bandwidth measurement.

This is a global setting and controls the interpolation state for all four bandwidth markers.

The query response indicates whether interpolation is turned on or off.

Syntax

```
CALC:MARK[1|2|3|4]:FUNC:BWID|BAND:INT OFF|ON|0|1

CALC:MARK[1|2|3|4]:FUNC:BWID|BAND:INT?
```

Example

OUTPUT @Osa;"calc:mark1:func:band:stat on"	! turn bandwidth marker 1 on trace a
OUTPUT @Osa;"calc:mark1:func:band:int on"	! turn bandwidth interpolation on

Related Key

(MARKER SETUP) BANDWIDTH INTERPOLATION ON|OFF

CALCulate:MARKer[1|2|3|4]:FUNCtion:BWIDth|BANDwidth:NDB

Sets the desired vertical offset from the numbered marker of the bandwidth markers.The parameter units are as specified in the UNIT:RATio command.

This value can be set or queried anytime. The marker does not have to be on or in the bandwidth function.

Syntax

```
CALC:MARK[1|2|3|4]:FUNC:BWID|BAND:NDB <numeric_value>

CALC:MARK[1|2|3|4]:FUNC:BWID|BAND:NDB?
```

Example

OUTPUT @Osa;"calc:mark1:func:band:stat on"	! turn bandwidth marker 1 on trace a
OUTPUT @Osa;"calc:mark1:func:band:ndb -3db"	! set the vertical offset of the bandwidth markers

Related Key

(MARKER BW ON OFF) -0.5 dB | -3 dB | -6 dB | -10 dB | -20 dB

CALCulate:MARKer[1|2|3|4]:FUNCtion:BWIDth|BANDwidth:READout

Sets the X-axis readout for frequency or wavelength when the instrument is in a non-zero span.

This setting controls only the bandwidth marker X-axis readouts and the X:Left? and X:Right? queries. The delta markers have their own setting. This setting controls all four bandwidth markers.

Trying to set the readout to TIME when in a non-zero span generates a "Settings conflict" error. Trying to set the readout to FREQuency or WAVelength when in zero span also generates a "Settings conflict" error. When the instrument is set to zero span, the readout will automatically change to TIME. This command is primarily useful for non-zero spans.

Syntax

```
CALC:MARK[1|2|3|4]:FUNC:BWID|BAND:READ FREQ
```

| WAV | TIME

:CALC:MARK[1 | 2 | 3 | 4]:FUNC:BWID | BAND:READ?

Example

OUTPUT @Osa;"calc:mark1:func:band:stat on"	! turn bandwidth marker 1 on trace a
OUTPUT @Osa;"calc:mark1:func:band:read FREQ"	! sets bandwidth marker x query values to !FREQuency

Related Key

(MARKER SETUP) BW MARKER UNITS

CALCulate:MARKer[1 | 2 | 3 | 4]:FUNCtion:BWIDth | BANDwidth:RESult?

Returns the difference in the X-axis values between the left and right bandwidth markers. The units returned are determined by the CALCulate:MARKer:FUNCtion:BWIDth | BANDwidth:X:READout state.

This query generates a "Settings conflict" error if the bandwidth function is OFF for the specified marker.

For READout of FREQuency, the result is returned in Hertz. For READout of WAVelength, the result is returned in meters. If the bandwidth markers cannot find the desired number of dB (NDB) setting relative to the normal marker, the result returned will be 9.91e37. This value is defined by the SCPI standard to represent NaN (not a number).

Syntax

CALC:MARK[1 | 2 | 3 | 4]:FUNC:BWID | BAND:RES?

Example

OUTPUT @Osa;"calc:mark1:func:band:stat on"	! turn bandwidth marker 1 on trace a
OUTPUT @Osa;"calc:mark1:max"	! set bandwidth marker to strongest signal
OUTPUT @Osa;"calc:mark1:func:bwid:res?"	! query the difference in X values between
ENTER @Osa;Res	!left and right bandwidth markers
PRINT Res	

CALCulate:MARKer[1|2|3|4]:FUNCtion:BWIDth|BANDwidth[:STATe] OFF|ON|0|1

Turns the bandwidth marker function on or off for a particular marker.

Each marker may be in one of five mutually exclusive states:

Normal (set by CALCulate:MARKer[1|2|3|4]:FUNCtion:PRESet),

Bandwidth
(set by CALCulate:MARKer[1|2|3|4]:BWIDth}BANDwidth[:STATe])

Noise (set by CALCulate:MARKer[1|2|3|4]:FUNCtion:NOISe[:STATe])

Delta (set by CALCulate:MARKer[1|2|3|4]:FUNCtion:DELTa[:STATe])

OSNR (set by CALCulate:MARKer[1|2|3|4]:FUNCtion:OSNR[:STATe])

Enabling one state for a marker will disable any previous state.

If the bandwidth function is turned on for a marker that is off, the marker will be turned on, placed at the center wavelength, and then the bandwidth marker will measure the bandwidth relative to this marker.

Syntax

```
CALC:MARK[1|2|3|4]:FUNC:BWID|BAND[:STAT] OFF|ON|0|1

CALC:MARK[1|2|3|4]:FUNC:BWID|BAND[:STAT]?
```

Example

OUTPUT @Osa;"calc:mark1:func:band:stat on"	! turn bandwidth marker 1 on trace a

Related Commands

```
CALC:MARK:FUNC:BWID|BAND:INT
CALC:MARK:FUNC:BWID|BAND:NDB
CALC:MARK:FUNC:BWID|BAND:READ
CALC:MARK:FUNC:BWID|BAND:RES
CALC:MARK:FUNC:BWID|BAND:X:CENT?
CALC:MARK:FUNC:BWID|BAND:X:LEFT?
CALC:MARK:FUNC:BWID|BAND:X:RIGH?
```

Related Key

MARKER BW ON OFF

CALCulate:MARKer[1|2|3|4]:FUNCtion:BWIDth|BANDwidth:X :CENTer?

Returns the absolute X-axis value from the center of the bandwidth marker (mean of the left and right markers).

The units returned are determined by the CALCulate:MARKer:FUNCtion :BWIDth|BANDwidth:X:READout state. For READout of FREQuency, the X value is returned in Hertz. For READout of WAVelength, the X value is returned in meters.

This query generates a "Settings conflict" error if the bandwidth function is off for the specified marker.

Syntax

```
CALC:MARK[1|2|3|4]:FUNC:BWID|BAND:X:CENT?
```

Example

OUTPUT @Osa;"calc:mark1:func:band:stat on"	! turn bandwidth marker 1 on trace a
OUTPUT @Osa;"calc:mark1:max"	! set bandwidth marker to peak signal
OUTPUT @Osa;"calc:mark1:func:band:x:Cent?"	! query for wavelength of the normal marker
ENTER @Osa;Cent	! left and right bandwidth markers
PRINT Cent	

CALCulate:MARKer[1|2|3|4]:FUNCtion:BWIDth|BANDwidth:X:LEFT?

Returns the absolute X-axis value of the left bandwidth marker.

The units returned are determined by the CALCulate:MARKer:FUNCtion :BWIDth|BANDwidth:X:READout state. For READout of FREQuency, the X value is returned in Hertz. For READout of WAVelength, the X value is returned in meters.

This query generates a "Settings conflict" error if the bandwidth function is OFF for the specified marker.

Syntax

```
CALC:MARK[1|2|3|4]:FUNC:BWID|BAND:X:LEFT?
```

Example

OUTPUT @Osa;"calc:mark1:func:band:stat on"	! turn bandwidth marker 1 on trace a
OUTPUT @Osa;"calc:mark1:max"	! set bandwidth marker to peak signal
OUTPUT @Osa;"calc:mark1:func:band:x:left?"	! query the wavelength of the left bandwidth !marker
ENTER @Osa;Left	
PRINT Left	

CALCulate:MARKer[1|2|3|4]:FUNCtion:BWIDth|BANDwidth:X:RIGHt?

Returns the absolute X-axis value of the right bandwidth marker.

The units returned are determined by the CALCulate:MARKer[1|2|3|4] :FUNCtion:BWIDth|BANDwidth:X:READout state. For READout of FREQuency, the X value is returned in Hertz. For READout of WAVelength, the X value is returned in meters.

This query generates a "Settings conflict" error if the bandwidth function is OFF for the specified marker.

Syntax

```
CALC:MARK[1|2|3|4]:FUNC:BWID|BAND:X:RIGH?
```

Example

OUTPUT @Osa;"calc:mark1:func:band:stat on"	! turn bandwidth marker 1 on trace a
OUTPUT @Osa;"calc:mark1:max"	! set bandwidth marker to peak signal
OUTPUT @Osa;"calc:mark1:func:band:x:right?"	! query the frequency of the right bandwidth !marker
ENTER @Osa;Right	
PRINT Right	

CALCulate:MARKer[1|2|3|4]:FUNCtion:DELTa:RESet

Sets the reference for the delta marker to the current position of the delta marker.

Syntax

```
CALC:MARK[1|2|3|4]:FUNC:DELT:RES
```

Example

OUTPUT @Osa;"calc:mark1:max"	! turn on marker one, set to highest peak
OUTPUT @Osa;"calc:mark1:func:delt:stat on"	! activate delta mode for marker 1
OUTPUT @Osa;"calc:mark1:func:delt:res"	! set reference for delta marker to current marker position

Related Commands

```
CALCulate:MARKer[1|2|3|4]:FUNCtion:DELTa ON|OFF
```

CALCulate:MARKer[1|2|3|4]:FUNCtion:DELta[:STATe]

Turns the delta marker function on or off for a particular marker.

Each marker may be in one of five mutually exclusive states:

Normal (set by CALCulate:MARKer[1|2|3|4]:FUNCtion:PRESet),

Bandwidth
(set by CALCulate:MARKer[1|2|3|4]:BWIDth}BANDwidth[:STATe])

Noise (set by CALCulate:MARKer[1|2|3|4]:FUNCtion:NOISe[:STATe])

Delta (set by CALCulate:MARKer[1|2|3|4]:FUNCtion:DELTa[:STATe])

OSNR (set by CALCulate:MARKer[1|2|3|4]:FUNCtion:OSNR[:STATe])

Enabling one state for a marker will disable any previous state.

If the delta function is turned on for a marker that is off, the marker will be turned on, placed at the center wavelength, and the delta function will be turned on.

Read the offset: CALCulate:MARKer[1|2|3|4]:FUNCtion:DELTa:X:OFFSet?

Syntax

CALC:MARK[1|2|3|4]:FUNC:DELT[:STAT] OFF|ON|0|1

CALC:MARK[1|2|3|4]:FUNC:DELT[:STAT]?

Example

OUTPUT @Osa;"calc:mark1:func:delt:stat on"	! activate delta mode for marker 1

Related Key

DELTA MARKER ON|OFF

CALCulate:MARKer[1|2|3|4]:FUNCtion:DELTa:X:OFFSet?

Returns the difference between the absolute X-axis value of the delta marker and the X-axis value of the reference marker.

The units of the value returned by the query are determined by the CALCulate:MARKer:FUNCtion:DELTa:X:READout state. For READout of FREQuency, the units are Hertz. For READout of WAVelength, the units are meters. For READout of TIME, the units are seconds.

This query generates a "Settings conflict" error if the delta function is OFF for the specified marker.

Syntax

CALC:MARK[1|2|3|4]:FUNC:DELT:X:OFFS?

Example

OUTPUT @Osa;"calc:mark1:max"	! turn on marker one, set to highest peak
OUTPUT @Osa;"calc:mark1:func:delt:stat on"	! activate delta mode for marker 1
OUTPUT @Osa;"calc:mark1:func:delt:y:offs?"	! query the delta
ENTER @Osa;Delty	

CALCulate:MARKer[1|2|3|4]:FUNCtion:DELTa:X:OFFSet:FREQuency

Allows you to set the marker offset in frequency units.

The marker X-axis value corresponds to *the reference X value + the offset value*. The default units of the parameter for this command are Hertz.

This query generates a "Settings conflict" error if the delta function is off for the specified marker.

Syntax

```
CALC:MARK[1|2|3|4]:FUNC:DELT:X:OFFS:FREQ
<numeric_value>

CALC:MARK[1|2|3|4]:FUNC:DELT:X:OFFS:FREQ?
```

CALCulate:MARKer[1|2|3|4]:FUNCtion:DELTa:X:OFFSet:TIME

Allows you to set the marker offset when the instrument is in zero span.

The marker X-axis value corresponds to *the reference X value + the offset value*. The default units of the parameter are seconds.

This query generates a "Settings conflict" error if the delta function is off for the specified marker.

NOTE This command functions only in zero span.

Example

	! put the instrument in 0 span
CALCulate:MARKer1:FUNCtion:DELTa:STATe ON	! turn on the delta marker with the command
CALCulate:MARKer1:FUNCtion:DELTa:X:OFFSet: TIME 30 ms	! set the time value
CALCulate:MARKer1:FUNCtion:DELTa:X:OFFSet?	! read the offset

Syntax

```
CALC:MARK[1|2|3|4]:FUNC:DELT:X:OFFS:TIME
<numeric_value>
```

CALCulate:MARKer[1|2|3|4]:FUNCtion:DELTa:X:OFFSet [:WAVelength]

Allows you to set the marker offset in wavelength units.

The marker X-axis value corresponds to *the reference X value + the offset value*. The default units of the parameter are meters.

Even though the offset READout may be FREQuency, this command can still be used to specify the offset using wavelength units.

For example:

```
CALCulate:MARKer[1|2|3|4]:FUNCtion:DELTa:X
:OFFSet:WAVelength 10NM when readout is WAVElength
CALCulate:MARKer[1|2|3|4]:FUNCtion:DELTa:X
:OFFSet:FREQuency 10THZ when readout is FREQuency
CALCulate:MARKer[1|2|3|4]:FUNCtion:DELTa:X
:OFFSet:WAVelength 1E-8M when readout is FREQuency
```

The query generates a “Settings conflict” error if the delta function is off for the specified marker.

Syntax

```
CALC:MARK[1|2|3|4]:FUNC:DELT:X:OFFS[:WAV]
<numeric_value>

CALC:MARK[1|2|3|4]:FUNC:DELT:X:OFFS?
```

Example

OUTPUT @Osa;"calc:mark1:max"	! turn on marker one, set to highest peak
OUTPUT @Osa;"calc:mark1:func:delt:stat on"	! activate delta mode for marker 1
OUTPUT @Osa;"calc:mark1:func:delt:x:offs 1nm"	! sets delta marker offset to 1nm

CALCulate:MARKer[1|2|3|4]:FUNCtion:DELTa:X:READout

Sets the X-axis readout for frequency or wavelength when the instrument is in a non-zero span.

This setting controls only the delta offset and the delta reference X-axis readouts. The bandwidth markers have their own setting. This setting controls all four delta markers.

Trying to set the readout to TIME when in a non-zero span generates a "Settings conflict" error. Trying to set the readout to FREQuency or WAVelength when in a zero span also generates a "Settings conflict" error. When the instrument is set to zero span, the readout will automatically change to TIME. If the delta marker is off a "Settings conflict" error is generated. This command is primarily useful for non-zero spans.

Syntax

```
CALC:MARK[1|2|3|4]:FUNC:DELT:X:READ FREQ|WAV|TIM

CALC:MARK[1|2|3|4]:FUNC:DELT:X:READ?
```

Example

OUTPUT @Osa;"calc:mark1:max"	! turn on marker one, set to highest peak
OUTPUT @Osa;"calc:mark1:func:delt:stat on"	! activate delta mode for marker 1
OUTPUT @Osa;"calc:mark1:func:delt:X:read WAV"	! sets delta marker x query values to !WAVelength units

CALCulate:MARKer[1|2|3|4]:FUNCtion:DELTa:X:REFerence?

Returns the X-axis value of the reference marker.

The units of the returned value are determined by the CALCulate:MARKer :FUNCtion:DELTa:X:READout setting. For a READout of FREQuency, the return value is in Hertz. For a READout of WAVelength, the return value is in meters. For READout of TIME, the X value is returned in seconds.

This query generates a "Settings conflict" error if the delta function is off for the specified marker.

Syntax

```
CALC:MARK[1|2|3|4]:FUNC:DELT:X:REF?
```

Example

OUTPUT @Osa;"calc:mark1:max"	! turn on marker one, set to highest peak
OUTPUT @Osa;"calc:mark1:func:delt:stat on"	! activate delta mode for marker 1
OUTPUT @Osa;"calc:mark1:func:delt:X:ref?"	! query the x value of the reference marker
ENTER @Osa;Refx	
PRINT Refx	

CALCulate:MARKer[1|2|3|4]:FUNCtion:DELTa:Y:OFFset?

Returns the difference between the delta marker absolute Y value and the reference Y value.

This query generates a “Settings conflict” error if the delta function is off for the specified marker.

Syntax

```
CALC:MARK[1|2|3|4]:FUNC:DELT:Y:OFFS?
```

Example

OUTPUT @Osa;"calc:mark1:max"	! turn on marker one, set to highest peak
OUTPUT @Osa;"calc:mark1:func:delt:stat on"	! activate delta mode for marker 1
OUTPUT @Osa;"calc:mark1:func:delt:y:offs?"	! query the delta
ENTER @Osa;Delty	
PRINT Delty	

CALCulate:MARKer[1|2|3|4]:FUNCtion:DELTa:Y:REFerence?

Returns the Y-axis value of the reference marker.

This query generates a “Settings conflict” error if the delta function is off for the specified marker.

Syntax

```
CALC:MARK[1|2|3|4]:FUNC:DELT:Y:REF?
```

Example

OUTPUT @Osa;"calc:mark1:max"	! turn on marker one, set to highest peak
OUTPUT @Osa;"calc:mark1:func:delt:stat on"	! activate delta mode for marker 1
OUTPUT @Osa;"calc:mark1:func:Delt:Y:Ref?"	! query the amplitude of the ref marker
ENTER @Osa;Refy	
PRINT Refy	

CALCulate:MARKer[1|2|3|4]:FUNCtion:NOISe:BWIDth|BANDwidth

Sets the normalization bandwidth for the marker noise result query and the LED Source Test peak density calculation. The default units for the parameter are meters. There are only two allowable settings: 1.0 nm and 0.1 nm. Sending any value outside this range will generate a "Data out of range" error. Sending a value within this range will set the bandwidth to whichever of the two possible settings is closest to the specified value.

Syntax

```
CALC:MARK[1|2|3|4]:FUNC:NOIS:BWID|BAND
<numeric_value>

CALC:MARK[1|2|3|4]:FUNC:NOIS:BWID|BAND?
```

Example

OUTPUT @Osa;"calc:mark1:func:nois:stat on"	! activate noise mode for marker 1
OUTPUT @Osa;"calc:mark1:func:nois:band 1 nm"	! set noise marker bandwidth to 1 nm
OUTPUT @Osa;"calc:mark1:func:nois:res?"	! query the noise marker
ENTER @Osa;Noise	
PRINT Noise	

Related Key

(MARKER SETUP) NOISE MARKER REFERENCE BANDWIDTH

CALCulate:MARKer[1|2|3|4]:FUNCtion:NOISe:RESult?

Returns the noise marker value normalized to 1.0 nm or 0.1 nm. The normalization bandwidth (selection of 1.0 nm or 0.1 nm) is controlled by the CALCulate:MARKer:FUNCtion:NOISe:BWIDth command.

A "Settings conflict" error is generated if the noise function is off for the specified marker.

Syntax

CALC:MARK[1|2|3|4]:FUNC:NOIS:RES?

Example

OUTPUT @Osa;"calc:mark1:func:nois:stat on"	! activate noise mode for marker 1
OUTPUT @Osa;"calc:mark1:func:nois:band 1 nm"	! set noise marker bandwidth to 1 nm
OUTPUT @Osa;"calc:mark1:func:nois:res?"	! query the noise marker
ENTER @Osa;Noise	
PRINT Noise	

Related Commands

CALC:MARK:FUNC:NOIS:STAT

CALC:MARK:FUNC:NOIS:BAND

CALCulate:MARKer[1|2|3|4]:FUNCtion:NOISe[:STATe]

Turns the marker noise function on or off. Use the CALCulate:MARKer:X command to position the noise marker.

Each marker may be in one of five mutually exclusive states:

Normal (set by CALCulate:MARKer[1|2|3|4]:FUNCtion:PRESet),

Bandwidth
(set by CALCulate:MARKer[1|2|3|4]:BWIDth}BANDwidth[:STATe])

Noise (set by CALCulate:MARKer[1|2|3|4]:FUNCtion:NOISe[:STATe])

Delta (set by CALCulate:MARKer[1|2|3|4]:FUNCtion:DELTa[:STATe])

OSNR (set by CALCulate:MARKer[1|2|3|4]:FUNCtion:OSNR[:STATe])

Enabling one state for a marker will disable any previous state.

Syntax

```
CALC:MARK[1|2|3|4]:FUNC:NOIS[:STAT] OFF|ON|0|1

CALC:MARK[1|2|3|4]:FUNC:NOIS[:STAT]?
```

Example

OUTPUT @Osa;"calc:mark1:func:nois:stat on"	! activate noise mode for marker 1
OUTPUT @Osa;"calc:mark1:func:nois:band 1 nm"	! set noise marker bandwidth to 1 nm
OUTPUT @Osa;"calc:mark1:func:nois:res?"	! query the noise marker
ENTER @Osa;Noise	
PRINT Noise	

Related Commands

```
CALCulate:MARKer:FUNCtion:NOISe:BWIDth|BANDwidth
CALCulate:MARKer:FUNCtion:NOISe:RESult?
```

Related Key

NOISE MARKER ON OFF

CALCulate:MARKer[1|2|3|4]:FUNCtion:OSNR:MODE

Changes between pit, manual, or auto mode. The mode determines where the noise measurements are calculated.

Syntax

```
CALC:MARK[1|2|3|4]:FUNC:OSNR:MODE:PIT|MAN|AUTO

CALC:MARK[1|2|3|4]:FUNC:OSNR:MODE?
```

Example

OUTPUT @Osa;"calc:mark1:func:osnr:state on"	! activate osnr mode for marker 1
OUTPUT @Osa;"calc:mark1:func:osnr:mode MAN"	! set MANual mode for osnr marker 1

Related Key

(MARKER SETUP) OSNR NOISE

CALCulate:MARKer[1|2|3|4]:FUNCtion:OSNR:OFFSet

Sets the noise marker offset (the fixed distance between the center marker and noise marker). This parameter is used only in manual mode.

Syntax

```
CALC:MARK[1|2|3|4]:FUNC:OSNR:OFFS <numeric value>
M|UM|NM|A

CALC:MARK[1|2|3|4]:FUNC:OSNR:OFFS?
```

Example

OUTPUT @Osa;"calc:mark1:func:osnr:state on"	! activate osnr mode for marker 1
OUTPUT @Osa;"calc:mark1:func:osnr:mode MAN"	! set MANual mode for osnr marker 1
OUTPUT @Osa;"calc:mark1:func:osnr:offs 2nm"	! set osnr offset

Related Key

(MARKER SETUP) OSNR NOISE

CALCulate:MARKer[1|2|3|4]:FUNCtion:OSNR:RESult?

Reads the OSNR value for the specified markers. If the OSNR result is invalid, then the value, 9.91e+37, is displayed. This error will occur if no peak is found or if the noise level is above the channel peak. The units may be dB/1.0 nm or dB/0.1 nm (logarithmic) or unit-less/1.0 nm or unit-less/0.1 nm (linear value) depending on the Y axis units. To control the noise marker bandwidth, use CALCulate:MARKer:FUNCtion:NOISe:BWIDth | BANDwidth 0.1 | 1 NM.

The command for selecting logarithmic or linear units is: DISPlay[:WINDow]:TRACe:Y [:SCALe]:SPACing LINear|LOGarithmic.

Syntax

```
CALC:MARK[1|2|3|4]:FUNC:OSNR:RES?
```

Example

OUTPUT @Osa;"calc:mark1:func:osnr:state on"	! activate osnr mode for marker 1
OUTPUT @Osa;"calc:mark1:func:osnr:res?"	! query osnr value
ENTER @Osa;Result	
PRINT Result	

CALCulate:MARKer[1|2|3|4]:FUNCtion:OSNR[:STATe]

Turns the OSNR marker on or off.

Each marker may be in one of five mutually exclusive states:

Normal (set by CALCulate:MARKer[1|2|3|4]:FUNCtion:PRESet),

Bandwidth
(set by CALCulate:MARKer[1|2|3|4]:BWIDth}BANDwidth[:STATe])

Noise (set by CALCulate:MARKer[1|2|3|4]:FUNCtion:NOISe[:STATe])

Delta (set by CALCulate:MARKer[1|2|3|4]:FUNCtion:DELTa[:STATe])

OSNR (set by CALCulate:MARKer[1|2|3|4]:FUNCtion:OSNR[:STATe])

Enabling one state for a marker will disable any previous state.

Syntax

CALC:MARK[1|2|3|4]:FUNC:OSNR[:STAT] OFF|ON|0|1

CALC:MARK[1|2|3|4]:FUNC:OSNR[:STAT]?

Example

OUTPUT @Osa;"calc:mark1:func:osnr:state on"	! activate osnr mode for marker 1

Related Key

OSNR MARKER ON OFF

CALCulate:MARKer[1|2|3|4]:FUNCtion:OSNR:X:CENTer?

Reads the X value of the center noise marker. A "Setting conflict" error is generated if the OSNR is off for the specified marker. If the OSNR result is invalid, then the value, 9.91e+37, is displayed when unable to locate a peak. The units are determined by CALCulate:MARKer:X:READout FREQency|WAVelength.

Syntax

CALC:MARK[1|2|3|4]:FUNC:OSNR:X:CENT?

Example

OUTPUT @Osa;"calc:mark1:func:osnr:state on"	! activate osnr mode for marker 1
OUTPUT @Osa;"calc:mark1:func:osnr:x:cent?"	! query osnr value
ENTER @Osa;Xcent	
PRINT Xcent	

CALCulate:MARKer[1|2|3|4]:FUNCtion:OSNR:X:LEFT?

Reads the X value of the left noise marker. A "Setting conflict" error is generated if the OSNR is off for the specified marker. If the OSNR result is invalid, then the value, 9.91e+37, is displayed when unable to locate a peak. The units are determined by CALCulate:MARKer:X:READout FREQency|WAVelength.

Syntax

```
CALC:MARK[1|2|3|4]:FUNC:OSNR:X:LEFT?
```

Example

OUTPUT @Osa;"calc:mark1:func:osnr:state on"	! activate osnr mode for marker 1
OUTPUT @Osa;"calc:mark1:func:osnr:x:left?"	! query osnr value
ENTER @Osa;Xleft	
PRINT Xleft	

CALCulate:MARKer[1|2|3|4]:FUNCtion:OSNR:X:RIGHt?

Reads the X value of the right noise marker. A "Setting conflict" error is generated if the OSNR is off for the specified marker. If the OSNR result is invalid, then the value, 9.91e+37, is displayed when unable to locate a peak. The units are determined by CALCulate:MARKer:X:READout FREQency|WAVelength.

Syntax

```
CALC:MARK[1|2|3|4]:FUNC:OSNR:X:RIGH?
```

Example

OUTPUT @Osa;"calc:mark1:func:osnr:state on"	! activate osnr mode for marker 1
OUTPUT @Osa;"calc:mark1:func:osnr:x:right?"	! query osnr value
ENTER @Osa;Xright	
PRINT Xright	

CALCulate:MARKer[1|2|3|4]:FUNCtion:OSNR:Y:CENTer?

Reads the Y value of the center noise marker. A "Setting conflict" error is generated if the OSNR is OFF for the specified marker. If the OSNR result is invalid, then the value, 9.91e+37, is displayed. The units may be dB (logarithmic) or unit-less (linear value) depending on the Y axis units.

The command for selecting logarithmic or linear units is: DISPlay[:WINDow]:TRACe:Y [:SCALe]:SPACing LINear|LOGarithmic.

Syntax

```
CALC:MARK[1|2|3|4]:FUNC:OSNR:Y:CENT?
```

Example

OUTPUT @Osa;"calc:mark1:func:osnr:state on"	! activate osnr mode for marker 1
OUTPUT @Osa;"calc:mark1:func:osnr:y:cent?"	! query osnr value
ENTER @Osa;Ycent	

CALCulate:MARKer[1|2|3|4]:FUNCtion:OSNR:Y:LEFT?

Reads the Y value of the left noise marker. A "Setting conflict" error is generated if the OSNR is off for the specified marker. If the OSNR result is invalid, then the value, 9.91e+37, is displayed when unable to locate a peak. The units may be dB/1.0 nm or dB/0.1 nm (logarithmic) or unit-less/1.0 nm or unit-less/0.1 nm (linear value) depending on the Y axis units and noise reference bandwidth.

The command for selecting logarithmic or linear units is: DISPlay[:WINDow]:TRACe:Y [:SCALe]:SPACing LINear|LOGarithmic.

Syntax

```
CALC:MARK[1|2|3|4]:FUNC:OSNR:Y:LEFT?
```

Example

OUTPUT @Osa;"calc:mark1:func:osnr:state on"	! activate osnr mode for marker 1
OUTPUT @Osa;"calc:mark1:func:osnr:y:left?"	! query osnr value
ENTER @Osa;Yleft	

CALCulate:MARKer[1|2|3|4]:FUNCtion:OSNR:Y:RIGHt?

Reads the Y value of the right noise marker. A "Setting conflict" error is generated if the OSNR is off for the specified marker. If the OSNR result is invalid, then the value, 9.91e+37, is displayed when unable to locate a peak. The units may be dB/1.0 nm or dB/0.1 nm (logarithmic) or unit-less/1.0 nm or unit-less/0.1 nm (linear value) depending on the Y axis units and noise reference bandwidth.

The command for selecting logarithmic or linear units is: DISPlay[:WINDow]:TRACe:Y [:SCALe]:SPACing LINear|LOGarithmic.

Syntax

```
CALC:MARK[1|2|3|4]:FUNC:OSNR:Y:RIGH?
```

Example

OUTPUT @Osa;"calc:mark1:func:osnr:state on"	! activate osnr mode for marker 1
OUTPUT @Osa;"calc:mark1:func:osnr:y:right?"	! query osnr value
ENTER @Osa;Yright	

CALCulate:MARKer[1|2|3|4]:FUNCtion:PRESet

Turns off all marker functions for the specified marker. This command is provided as a convenient way to turn all marker functions off without having to check the state of each individual marker function.

Each marker may be in one of five mutually exclusive states:

Normal (set by CALCulate:MARKer[1|2|3|4]:FUNCtion:PRESet),

Bandwidth
(set by CALCulate:MARKer[1|2|3|4]:BWIDth}BANDwidth[:STATe])

Noise (set by CALCulate:MARKer[1|2|3|4]:FUNCtion:NOISe[:STATe])

Delta (set by CALCulate:MARKer[1|2|3|4]:FUNCtion:DELTa[:STATe])

OSNR (set by CALCulate:MARKer[1|2|3|4]:FUNCtion:OSNR[:STATe])

Enabling one state for a marker will disable any previous state.

Syntax

```
CALC:MARK[1|2|3|4]:FUNC:PRES
```

Example

OUTPUT @Osa;"calc:mark1:FUNC:Pres"	! turn marker function on off

CALCulate:MARKer[1|2|3|4]:INTerpolate

Turns the normal/delta and the filter mode marker interpolation on or off.

When interpolation is on, the normal/delta and filter mode markers will be placed at the exact X setting, if the trace data allows. The marker will linearly interpolate between two true trace data points. The default state is off.

This setting controls the interpolation state for all four markers, except for the bandwidth markers.

Syntax

CALC:MARK[1|2|3|4]:INT OFF|ON|0|1

CALC:MARK[1|2|3|4]:INT?

Example

OUTPUT @Osa;"calc:mark1:int on"	! turn marker interpolation on

Related Key

NORMAL/DELTA MARKER INTERPOLATION ON|OFF

CALCulate:MARKer[1|2|3|4]:MAXimum

Places the specified marker on the highest point of the trace.

The point does *not* have to meet the peak excursion and threshold criteria. The marker trace is determined by the CALCulate:MARKer:TRACe command. If the specified marker is off, it will be turned on and placed on the highest point of the trace.

Syntax

CALC:MARK[1|2|3|4]:MAX

Example

OUTPUT @Osa;"calc:mark1:stat on"	! turn marker one on
OUTPUT @Osa;"calc:mark1:max"	! set marker one to max

Related Commands

CALCulate:MARKer[1|2|3|4]:MAXimum:LEFT
CALCulate:MARKer[1|2|3|4]:MAXimum:NEXT
CALCulate:MARKer[1|2|3|4]:MAXimum:RIGHt

Related Key

Peak Search

CALCulate:MARKer[1|2|3|4]:MAXimum:LEFT

Tunes the marker and grating position to the next peak to the left, using the previously defined peak search definition. If no peak is found, no action is taken.

Syntax

CALC:MARK[1|2|3|4]:MAX:LEFT

Example

OUTPUT @Osa;"calc:mark1:stat on"	! turn marker one on
OUTPUT @Osa;"calc:mark1:max:left"	! set marker one to next peak left

Related Commands

CALCulate:MARKer[1|2|3|4]:MAXimum
CALCulate:MARKer[1|2|3|4]:MAXimum:NEXT
CALCulate:MARKer[1|2|3|4]:MAXimum:RIGHt

Related Key

NEXT PEAK LEFT ←

CALCulate:MARKer[1|2|3|4]:MAXimum:NEXT

Places the marker on the next highest peak from the current marker amplitude.

This next highest peak must meet the peak excursion and threshold criteria. If the specified marker is off, it will be turned on, placed at the center wavelength, and the search for the next maximum will begin from that point.

Syntax

```
CALC:MARK[1|2|3|4]:MAX:NEXT
```

Example

OUTPUT @Osa;"calc:mark1:stat on"	! turn marker one on
OUTPUT @Osa;"calc:mark1:max:next"	! set marker one to next peak

Related Commands

```
CALCulate:MARKer[1|2|3|4]:MAXimum
CALCulate:MARKer[1|2|3|4]:MAXimum:LEFT
CALCulate:MARKer[1|2|3|4]:MAXimum:RIGHt
```

Related Key

NEXT PEAK DOWN ↓

CALCulate:MARKer[1|2|3|4]:MAXimum:RIGHt

Tunes the marker and grating position to the next peak to the right, using the previously defined peak search definition. If no peak is found, no action is taken.

Syntax

```
CALC:MARK[1|2|3|4]:MAX:RIGH
```

Example

OUTPUT @Osa;"calc:mark1:stat on"	! turn marker one on
OUTPUT @Osa;"calc:mark1:max:right"	! set marker one to next peak right

Related Commands

CALCulate:MARKer[1|2|3|4]:MAXimum
CALCulate:MARKer[1|2|3|4]:MAXimum:LEFT
CALCulate:MARKer[1|2|3|4]:MAXimum:NEXT

Related Key

NEXT PEAK RIGHT →

CALCulate:MARKer[1|2|3|4]:MINimum

Places the specified marker on the lowest point of the trace.

The point does *not* have to meet the pit excursion and threshold criteria. The marker trace is determined by the CALCulate:MARKer:TRACe command. If the specified marker is off, it will be turned on and placed on the lowest point of the trace.

Syntax

CALC:MARK[1|2|3|4]:MIN

Example

OUTPUT @Osa;"calc:mark1:stat on"	! turn marker one on
OUTPUT @Osa;"calc:mark1:min"	! set marker one to lowest point

Related Commands

CALCulate:MARKer[1|2|3|4]:MINimum:LEFT
CALCulate:MARKer[1|2|3|4]:MINimum:NEXT
CALCulate:MARKer[1|2|3|4]:MINimum:RIGHt

Related Key

PIT SEARCH

CALCulate:MARKer[1|2|3|4]:MINimum:LEFT

Places the marker on the next pit located at a shorter wavelength than the current marker wavelength position.

This next pit must meet the pit excursion and threshold criteria. If the specified marker is off, it will be turned on, placed at the center wavelength, and the search to the left will begin from that point.

Syntax

```
CALC:MARK[1|2|3|4]:MIN:LEFT
```

Example

OUTPUT @Osa;"calc:mark1:min:left"	! set marker to next pit, left

Related Commands

```
CALCulate:MARKer[1|2|3|4]:MINimum
CALCulate:MARKer[1|2|3|4]:MINimum:NEXT
CALCulate:MARKer[1|2|3|4]:MINimum:RIGHt
```

Related Key

NEXT PIT LEFT ←

CALCulate:MARKer[1|2|3|4]:MINimum:NEXT

Places the marker on the next lowest pit from the current marker amplitude.

This next lowest pit must meet the pit excursion and threshold criteria. If the specified marker is off, it will be turned on, placed at the center wavelength, and the search for the next minimum will begin from that point.

Syntax

```
CALC:MARK[1|2|3|4]:MIN:NEXT
```

Example

OUTPUT @Osa;"calc:mark1:stat on"	! turn marker one on
OUTPUT @Osa;"calc:mark1:min:next"	! set marker to next pit

Related Commands

```
CALCulate:MARKer[1|2|3|4]:MINimum
CALCulate:MARKer[1|2|3|4]:MINimum:LEFT
CALCulate:MARKer[1|2|3|4]:MINimum:RIGHt
```

Related Key

NEXT PIT UP –

CALCulate:MARKer[1|2|3|4]:MINimum:RIGHt

Places the marker on the next pit located at a longer wavelength than the current marker wavelength position.

This next pit must meet the pit excursion and threshold criteria. If the specified marker is off, it will be turned on, placed at the center wavelength, and the search to the right will begin from that point.

Syntax

```
CALC:MARK[1|2|3|4]:MIN:RIGH
```

Example

OUTPUT @Osa;"calc:mark1:stat on"	! turn marker one on
OUTPUT @Osa;"calc:mark1:min:right"	! set marker to next pit, right

Related Commands

```
CALCulate:MARKer[1|2|3|4]:MINimum
CALCulate:MARKer[1|2|3|4]:MINimum:LEFT
CALCulate:MARKer[1|2|3|4]:MINimum:NEXT
```

Related Key

NEXT PIT RIGHT →

CALCulate:MARKer[1|2|3|4]:PEXCursion[:PEAK]

Sets the peak excursion value for the marker search routines.

The peak excursion value is used to determine whether or not a local maximum in the trace is to be considered a peak. To qualify as a peak, both sides of the local maximum must fall by at least the peak excursion value.

Syntax

```
CALC:MARK[1|2|3|4]:PEXC[:PEAK]<numeric_value>

CALC:MARK[1|2|3|4]:PEXC[:PEAK]?
```

Example

OUTPUT @Osa;"calc:mark1:stat on"	! turn marker one on
OUTPUT @Osa;"calc:mark1:pexc:Peak 5db"	! set peak excursion to 5db

Related Commands

```
CALCulate:MARKer[1|2|3|4]:PEXCursion:PIT
```

Related Key

PEAK EXCURSION

CALCulate:MARKer[1|2|3|4]:PEXCursion:PIT

Sets the pit excursion value for the marker search routines.

The pit excursion value is used to determine whether or not a local minimum in the trace is to be considered a pit. To qualify as a pit, both sides of the local minimum must rise by at least the pit excursion value.

Syntax

```
CALC:MARK[1|2|3|4]:PEXC:PIT<numeric_value>

CALC:MARK[1|2|3|4]:PEXC:PIT?
```

Example

OUTPUT @Osa;"calc:mark1:stat on"	! turn marker one on
OUTPUT @Osa;"calc:mark1:pexc:pit 5db"	! set pit excursion to 5db

Related Commands

CALCulate:MARKer[1|2|3|4]:PEXCursion:[PEAK]

Related Key

PIT EXCURSION

CALCulate:MARKer[1|2|3|4]:SCENter

Sets the center wavelength to the wavelength value of the marker.

Syntax

CALC:MARK[1|2|3|4]:SCEN

Example

OUTPUT @Osa;"calc:mark1:stat on"	! turn marker one on
OUTPUT @Osa;"calc:mark1:scen"	! set center to marker one wavelength

Related Commands

CALCulate:MARKer:[1|2|3|4]:MAXimum:SENSe:WAVelength:CENTer

Related Key

MARKER TO CENTER

ALCulate:MARKer[1|2|3|4]:SRANge:LOWer?

Returns the lower limit for the marker search range.

The range used for the marker search range is the same range used for the total power calculation, trace mean range, and wavelength sweep range. The return value is in meters, unless span is set to zero, in which case the return value is in seconds.

Syntax

```
CALC:MARK[1|2|3|4]:SRAN:LOW?
```

CALCulate:MARKer[1|2|3|4]:SRANge:LOWer:FREQuency

Sets the lower limit for the marker search range.

Setting this value when :CALCuate:MARKer:SRANge:STATe is off will automatically turn CALCulate:MARKer:SRANge:STATe on. The range used for the marker search range is the same range used for the total power calculation, trace mean range, and wavelength sweep range. Changing the range with this command will change all four ranges. Sending the command when the instrument is in a zero span will generate a “Settings conflict” error. The default units for the parameter is in Hertz.

Syntax

```
CALC:MARK[1|2|3|4]:SRAN:LOW:FREQ <param>
```

Example

OUTPUT @Osa;"calc:mark1:sran:low:freq 196thz"	! set lower bound for marker search using !frequency units

CALCulate:MARKer[1|2|3|4]:SRANge:LOWer:TIME

Sets the lower limit for the marker search range.

Setting this value when CALCulate:MARKer:SRANge:STATe is off will automatically turn CALCulate:MARKer:SRANge:STATe on. The range used for the marker search range is the same range used for the total power calculation, trace mean range, and wavelength sweep range. Changing the range with this command will change all four ranges. Sending this command while span is not set to zero will results in a "Settings conflict" error. Default units for the parameter is in seconds.

Syntax

```
CALC:MARK[1|2|3|4]:SRAN:LOW:TIME <param>
```

CALCulate:MARKer[1|2|3|4]:SRANge:LOWer[:WAVelength]

Sets the lower limit for the marker search range.

Setting this value when CALCulate:MARKer:SRANge:STATe is off will automatically turn CALCulate:MARKer:SRANge:STATe on. The range used for the marker search range is the same range used for the total power calculation, trace mean range, and wavelength sweep range. Changing the range with this command will change all four ranges. Sending the command when the instrument is in a zero span will generate a "Settings conflict" error. Default units for the parameter is in meters; frequency units are allowed.

Syntax

```
CALC:MARK[1|2|3|4]:SRAN:LOW[:WAV] <param>
```

Related Key

WAVELENGTH MARKER 1
Search Limit On Off
Line Markers Off

CALCulate:MARKer[1|2|3|4]:SRANge[:STATe] OFF|ON|0|1

Turns the search range on or off for all the markers. When the search range is on, all the marker maximum/minimum searches will be within the upper and lower wavelength range. Although there is a single range controlling the total power integration, marker search range, mean calculation, and wavelength sweep range, there are four independent state settings for limiting them to the range. If all four states are off, setting CALCulate:Marker:SRANge:STATe to on will initialize the lower range to

$$start + \frac{span}{3}$$

and the upper range to

$$start + \left(2 \times \frac{span}{3}\right) \quad .$$

Syntax

```
CALC:MARK[1|2|3|4]:SRAN[:STAT] OFF|ON|0|1
```

Example

OUTPUT @Osa;"calc:mark1:stat on"	! turn marker one on
OUTPUT @Osa;"calc:mark1:sran:stat on"	! turn marker search range limits on

Related Key

SEARCH LIMIT ON OFF

CALCulate:MARKer[1|2|3|4]:SRANge:UPPer?

Returns the upper limit for the marker search range.

The range used for the marker search range is the same range used for the total power calculation, trace mean range, and wavelength sweep range. The return value is in meters, unless span is set to zero, in which case the return value is in seconds.

Syntax

```
CALC:MARK[1|2|3|4]:SRAN:UPP?
```

Related Key

WAVELENGTH LINE MKR 2
Search Limit On Off
Line Markers Off

CALCulate:MARKer[1|2|3|4]:SRANge:UPPer:FREQuency

Sets the upper limit for the marker search range.

Setting this value when CALCulate:Marker:SRANge:STATe is off will automatically turn CALCulate:Marker:SRANge:STATe on. The range used for the marker search range is the same range used for the total power calculation, trace mean range, and wavelength sweep range. Changing the range with this command will change all four ranges. Sending the command when the instrument is in a zero span will generate a "Settings conflict" error. The default units for the parameter is in Hertz.

Syntax

```
CALC:MARK[1|2|3|4]:SRAN:UPP:FREQ <param>
```

Example

OUTPUT @Osa;"calc:mark1:sran:stat on"	! turn marker search range limits on
OUTPUT @Osa;"calc:mark1:sran:upp:freq 196thz"	! set upper bound for marker search using !frequency units

CALCulate:MARKer[1|2|3|4]:SRANge:UPPer:TIME

Sets the upper limit for the marker search range.

Setting this value when CALCulate:Marker:SRANge:STATe is off will automatically turn CALCulate:Marker:SRANge:STATe on. The range used for the marker search range is the same range used for the total power calculation, trace mean range, and wavelength sweep range. Changing the range with this command will change all four ranges. Sending this command while span is not set to zero will result in a "Settings conflict" error. Default units for the parameter is in seconds.

Syntax

```
CALC:MARK[1|2|3|4]:SRAN:UPP:TIME <param>
```

CALCulate:MARKer[1|2|3|4]:SRANge:UPPer[:WAVelength]

Sets the upper limit for the marker search range.

Setting this value when CALCulate:Marker:SRANge:STATe is off will automatically turn CALCulate:Marker:SRANge:STATe on. The range used for the marker search range is the same range used the for total power calculation, trace mean range, and wavelength sweep range. Changing the range with this command will change all four ranges. Sending the command when the instrument is in a zero span will generate a "Settings conflict" error. Default units for the parameter is in meters; frequency units are allowed.

Syntax

```
CALC:MARK[1|2|3|4]:SRAN:UPP:WAV] <param>
```

Example

OUTPUT @Osa;"calc:mark1:stat on"	! turn marker one on
OUTPUT @Osa;"calc:mark1:sran:stat on"	! turn marker search range limits on
OUTPUT @Osa;"calc:mark1:sran:upp:wav 1555nm"	! set upper bound for marker search using !wavelength units

Related Key

WAVELENGTH MARKER 2

CALCulate:MARKer[1|2|3|4]:SRLevel

Sets the reference level to the amplitude of the marker.

Syntax

```
CALC:MARK[1|2|3|4]:SRL
```

Example

OUTPUT @Osa;"calc:mark1:stat on"	! turn marker one on
OUTPUT @Osa;"calc:mark1:max"	! set marker one to max
OUTPUT @Osa;"calc:mark1:SRL"	! set the reference level to the marker amplitude

Related Key

MARKER TO REF LEVEL

CALCulate:MARKer[1|2|3|4]:[:STATe]

Turns a particular marker on or off.

If no number is given for the MARKer node, 1 is assumed. (For example, CALCulate:MARKer on will turn marker 1 on.) The marker will be placed on the trace determined by the CALCulate:MARKer:TRACe command. If no trace is specified, the default trace is trace A. The marker will be placed at the center wavelength. Turning a marker off will turn off any marker function that was on for that particular marker. When the marker is turned on again, all the marker functions for that marker will be off.

Syntax

CALC:MARK[1|2|3|4][:STAT] OFF|ON|0|1

CALC:MARK[1|2|3|4][:STAT]?

Related Keys

ACTIVE MARKER 1|2|3|4|OFF

CALCulate:MARKer[1|2|3|4]:TRACe

Places the marker on a particular trace.

Syntax

CALC:MARK[1|2|3|4]:TRAC TRA|TRB|TRC|TRD|TRE|TRF

CALC:MARK[1|2|3|4]:TRAC?

Example

OUTPUT @Osa;"calc:mark1:stat on"	! turn marker one on
OUTPUT @Osa;"calc:mark1:trac tra"	! set marker one to trace a (default)

Related Key

(MARKER) ACTIVETRACE A|B|C|D|E|F

CALCulate:MARKer[1|2|3|4]:X?

Returns the X-axis value of the normal marker.

When the delta function is on, the absolute X-axis value of the delta marker is returned. When the bandwidth function is ON, the X-axis value of the center marker is returned.

The units of the value returned by the query is determined by the CALCulate:MARKer:X:READout state. For READout of FREQuency, the units returned are in Hertz. For READout of WAVelength, the units returned are meters. For READout of TIME, the units are in seconds.

Sending the query when the specified marker is off will generate a "Settings conflict" error.

Syntax

CALC:MARK[1|2|3|4]:X?

Example

OUTPUT @Osa;"calc:mark1:stat on"	! turn marker one on
OUTPUT @Osa;"calc:mark1:x?"	! query for marker 1 wavelength
ENTER @Osa;Markx	
OUTPUT @Osa;"calc:mark1:y?"	! query for marker 1 amplitude
PRINT "marker 1 wavelength: "	
PRINT Markx	

CALCulate:MARKer[1|2|3|4]:X:FREQuency

Sets the X-axis value of the normal marker.

When the delta function is on, the absolute X-axis value of the delta marker is controlled. When the bandwidth function is on, the X-axis value of the center marker is controlled. Sending the command when the specified marker is off will turn the marker on and place the marker at the desired position. Sending the command when the instrument is in a zero span will generate a “Settings conflict” error.

Syntax

```
CALC:MARK[1|2|3|4]:X:FREQ <numeric_value>
```

Example

OUTPUT @Osa;"calc:mark1:stat on"	! turn marker one on
OUTPUT @Osa;"calc:mark1:x:freq 195 thz"	! set marker one position using frequency units

CALCulate:MARKer[1|2|3|4]:X:READout

Sets the X-axis readout for frequency or wavelength when the instrument is in a non-zero span.

This setting controls only the normal marker X-axis and the delta reference readout. The bandwidth and delta offset markers have their own settings. This setting controls all four normal markers.

Trying to set the READout to TIME when in a non-zero span generates a “Settings conflict” error. Trying to set the READout to FREQuency or WAVelength when in zero span also generate a "Settings conflict" error. When the instrument is set to zero span, the readout will automatically change to TIME. This command is primarily useful for non-zero spans.

Syntax

```
CALC:MARK[1|2|3|4]:X:READ FREQ|WAV|TIME

CALC:MARK[1|2|3|4]:X:READ?
```

Example

OUTPUT @Osa;"calc:mark1:stat on"	! turn marker one on
OUTPUT @Osa;"calc:mark:x:read WAV"	! set all marker x values to return in WAVelength units

Related Key

NORMAL MARKER UNITS

CALCulate:MARKer[1|2|3|4]:X:TIME

Sets the X-axis value of the normal marker when the instrument is in zero span.

When the delta function is on, the absolute X-axis value of the delta marker is controlled. When the bandwidth function is ON, the X-axis value of the center marker is controlled. The default units of the parameter is seconds.

Sending the command when the specified marker is off will turn the marker on and place the marker at the desired position. Sending the command when the instrument is in a non-zero span will generate a "Settings conflict" error.

Syntax

```
CALC:MARK[1|2|3|4]:X:TIME <numeric_value>
```

CALCulate:MARKer[1|2|3|4]:X[:WAVelength]

Tunes the marker to the supplied value.

Syntax

```
CALC:MARK[1|2|3|4]:X:WAV <param>[M|UM|NM|PM|A]
```

Example

OUTPUT @Osa;"calc:mark1:stat on"	! turn marker one on
OUTPUT @Osa;"calc:mark1:x:wav 1541nm"	! set the x position using wavelength units

CALCulate:MARKer[1|2|3|4]:Y?

Returns the Y-axis value of the normal marker.

When the delta function is on, the value returned is the absolute Y-axis value of the delta marker. When the bandwidth function is on, the value returned is the Y-axis value of the center marker.

Sending the command when the specified marker is off will generate a "Settings conflict" error.

Syntax

```
CALC:MARK[1|2|3|4]:Y?
```

Example

OUTPUT @Osa;"calc:mark1:stat on"	! turn marker one on
OUTPUT @Osa;"calc:mark1:max"	! set marker one to max
ENTER @Osa;Markx	
OUTPUT @Osa;"calc:mark1:y?"	! query for mark1 amplitude
ENTER @Osa;Marky	

CALCulate[1|2|3|4|5|6]:MATH[:EXPRession][:DEFine]

Defines a math expression to be used when the math operations are turned on.

The <expression> can contain a <trace_name> as operands. The math operations will be performed in linear units. If, for example, the desired operation is TRA – TRB in log units, the expression should be defined as TRA / TRB. Each CALCulate subsystem can have one expression defined. Recursive expressions are not allowed.

NOTE CALCulate1 controls TRA, CALCulate2 controls TRB, CALCulate3 controls TRC, CALCulate4 controls TRD, CALCulate5 controls TRE, and CALCulate6 controls TRF

Syntax

```
CALC:MATH[:EXPR][:DEF](<expression>)

CALC[1|2|3|4|5|6]:MATH[:EXPR][:DEF]?
<expression> := <trace_name><operator><trace_name>
<operator>::= +|–|*|/
```

Example

OUTPUT @Osa "CALCulate3:MATH:EXPRession (TRA / TRB)"	! C = Alog – B
OUTPUT @Osa "CALCulate3:MATH:EXPRession (TRA * TRB)"	! C = Alog + B
OUTPUT @Osa "CALCulate3:MATH:EXPRession (TRA – TRB)"	! C = Alin – B
OUTPUT @Osa "CALCulate3:MATH:EXPRession (TRA + TRB)"	! C = Alin + B
OUTPUT @Osa "CALCulate:MATH:EXPRession (TRC / TRD)"	! F = Clog – D

Related Key

DEFAULT MATH...

CALCulate[1|2|3|4|5|6]:MATH:STATe

Determines whether or not math processing is done.

Syntax

```
CALC:MATH:STAT OFF|ON|0|1

CALC[1|2|3|4|5|6]:MATH:STAT
```

Related Key

TRACE MATH OFF

CALCulate[1|2|3|4|5|6]:MAXimum:CLEar

Clears the current maximum hold values for the trace and allows a new maximum hold to occur.

The trace will be initialized to a very negative dBm value (–300 dBm). If the specified trace is not in the maximum hold state, this command will have no effect.

Syntax

```
CALC:MAX:CLE
```

Example

OUTPUT @Osa;"calc1:max:state on"	! turn on trace mean function for trace A
OUTPUT @Osa;"calc1:max:clear"	! clear max data for trace A
OUTPUT @Osa;"init:cont on"	! begin sweeping to acquire max hold data for trace !A

Related Key

RESET MIN/MAX HOLD

CALCulate[1|2|3|4|5|6]:MAXimum[:STATe]

Turns maximum hold on a trace on or off.

The maximum hold operation compares the current amplitude value of each point on a trace in the current sweep to the corresponding point detected during the previous sweep, then stores the maximum value. CACLulate[1|2|3|4|5|6] refers to traces A, B, C, D, E, and F respectively.

Each trace can be set to one of four mutually exclusive states: normal, average (CALCulate[1|2|3|4|5|6]:AVERage[:STATe]), maximum hold (CALCulate[1|2|3|4|5|6]:MAXimum[:STATe]), or minimum hold (CALCulate[1|2|3|4|5|6]:MINimum[:STATe]). Enabling one state will disable any previous setting. For example, turning on trace averaging for trace B (CALCulate2:AVERage ON) would turn off a minimum hold state (CALCulate2:MINimum OFF).

Trace averaging, maximum, and minimum holds can be used in conjuction with trace math. If the trace that you wish to place a maximum hold on is the result of a math expression (C=log A–B), the maximum result of the math expression will be held. TRC will be the maximum difference of TRA and TRB.

Syntax

```
CALC:MAX[:STAT] OFF|ON|0|1

CALC[1|2|3|4|5|6]:MAX[:STAT]?
```

Example

OUTPUT @Osa;"calc1:max:state on"	! turn on trace mean function for trace A
OUTPUT @Osa;"calc1:max:clear"	! clear max data for trace A
OUTPUT @Osa;"init:cont on"	! begin sweeping to acquire max hold data for trace !A

Related Key

HOLD A NONE|MAX|MIN

CALCulate[1|2|3|4|5|6]:MEAN[:DATA]?

Returns the arithmetic mean of the trace.

The points of the trace are summed in linear units and the sum is divided by the number of points. When the CALCulate[1|2|3|4|5|6]:MEAN:RANge is on, the mean is calculated over the upper and lower X-axis range limits. If the CALCulate[1|2|3|4|5|6]:MEAN:RANge is off, the mean is calculated over the entire trace. Sending this query when the CALCulate[1|2|3|4|5|6]:MEAN:STATe is off will generate a "Settings conflict" error. The MEAN calculation is performed at the end of sweep. Sending this query when the instrument is in the middle of a sweep will return the MEAN calculated for the *previous* sweep.

Syntax

```
CALC:MEAN[:DATA]?
```

Example

OUTPUT @Osa;"calc1:mean:state on"	! turn on trace mean function for trace A
OUTPUT @Osa;"calc1:mean:data?"	! query for mean trace data, trace a
ENTER @Osa;Mean	
PRINT Mean	

CALCulate[1|2|3|4|5|6]:MEAN:RANGe:LOWer?

Returns the lower X-axis limit for the trace mean range calculation. The range used for the trace mean range is the same range used for the marker search range.

The return value is in meters, unless the span is set to zero, in which case the return value is in seconds.

Syntax

```
CALC:MEAN:RANG:LOW?
```

CALCulate[1|2|3|4|5|6]:MEAN:RANGe:LOWer:FREQuency

Sets the lower X-axis limit for the trace mean range calculation.

Setting this value when CALCulate[1|2|3|4|5|6]:MEAN:RANGe:STATe is off will automatically turn CALCulate[1|2|3|4|5|6]:MEAN:RANGe:STATe on.

The range used for the trace mean range is the same range used for the total power calculation, marker search range, and wavelength sweep range. Changing the range with this command will change all four ranges. Default units for the parameters are Hertz.

Sending this command when the instrument is in a zero span will generate a “Settings conflict” error.

Syntax

```
CALC:MEAN:RANG:LOW:FREQ <numeric_value>
[HZ|KHZ|MHZ|GHZ|THZ]
```

Example

OUTPUT @Osa;"calc1:mean:state on"	! turn on trace mean function for trace A
OUTPUT @Osa;"calc1:mean:rang:stat on"	! turn on range for trace a mean function
OUTPUT @Osa;"calc1:mean:rang:low:freq 196 thz"	! set low range for mean trace function using !frequency units

CALCulate[1|2|3|4|5|6]:MEAN:RANGe:LOWer:TIME

Sets the lower X-axis limit for the trace mean range calculation.

Setting this value when CALCulate[1|2|3|4|5|6]:MEAN:RANGe:STATe is off will automatically turn CALCulate[1|2|3|4|5|6]:MEAN:RANGe:STATe on.

The range used for the trace mean range is the same range used for the total power calculation, marker search range, and wavelength sweep range. Changing the range with this command will change all four ranges. Default units for the parameters are seconds.

Sending this command while span is not set to zero will result in a "Settings conflict" error.

Syntax

```
CALC:MEAN:RANG:LOW:TIME <numeric_value>[NS|US|MS|S]
```

CALCulate[1|2|3|4|5|6]:MEAN:RANGe:LOWer[:WAVelength]

This command sets the lower X-axis limit for the trace mean range calculation.

Setting this value when CALCulate[1|2|3|4|5|6]:MEAN:RANGe:STATe is off will automatically turn CALCulate[1|2|3|4|5|6]:MEAN:RANGe:STATe on.

The range used for the trace mean range is the same range used for the total power calculation, marker search range, and wavelength sweep range. Changing the range with this command will change all four ranges.

Sending this command when the instrument is in a zero span will generate a "Settings conflict" error. Default units for the parameter are meters. Frequency units are also allowed.

Syntax

```
CALC:MEAN:RANG:LOW[:WAV] <numeric_value>
[M|UM|NM|PM|A]
```

Example

OUTPUT @Osa;"calc1:mean:state on"	! turn on trace mean function for trace A
OUTPUT @Osa;"calc1:mean:rang:stat on"	! turn on range for trace a mean function
OUTPUT @Osa;"calc1:mean:rang:low:wav 1550nm"	! set high range for trace mean function !using Wavelength units

CALCulate[1|2|3|4|5|6]:MEAN:RANGe[:STATe]

Turns the trace mean calculation range on or off for all traces.

Turning the calculation range on will also turn the CALCulate[1|2|3|4|5|6]:MEAN:STATe on for the specified trace (the trace is specified via its subopcode). There is a single range controlling the total power integration, trace mean range, marker search range, and wavelength sweep range, but there are four independent state settings for limiting them to the range.

Syntax

```
CALC[1|2|3|4|5|6]:MEAN:RANG[:STAT] OFF|ON|0|1

CALC[1|2|3|4|5|6]:MEAN:RANG[:STAT]?
```

Example

OUTPUT @Osa;"calc1:mean:state on"	! turn on trace mean function for trace A
OUTPUT @Osa;"calc1:mean:rang:stat on"	! turn on range for trace a mean function

CALCulate[1|2|3|4|5|6]:MEAN:RANGe:UPPer?

Returns the upper X-axis limit for the trace mean range calculation.

The range used for the trace mean range is the same range used for the total power calculation, marker search range, and wavelength sweep range

The returned value is in meters, unless the span is set to zero, in which case the returned value is in seconds.

Syntax

```
CALC:MEAN:RANG:UPP?
```

CALCulate[1|2|3|4|5|6]:MEAN:RANGe:UPPer:FREQuency

Sets the upper X-axis limit for the trace mean range calculation.

Setting this value when CALCulate[1|2|3|4|5|6]:MEAN:RANGe:STATe is off will automatically turn CALCulate[1|2|3|4|5|6]:MEAN:RANGe:STATe on. The range used for the trace mean range is the same range used for the total power calculation, marker search range, and wavelength sweep range. Changing the range with this command will change all four ranges. Default units for the parameter are Hertz.

Syntax

```
CALC:MEAN:RANG:UPP:FREQ <numeric_value>
[HZ|KHZ|MHZ|GHZ|THZ]
```

Example

OUTPUT @Osa;"calc1:mean:state on"	! turn on trace mean function for trace A
OUTPUT @Osa;"calc1:mean:rang:upp:freq 196 thz"	! set upper range for mean trace function !using frequency units

CALCulate[1|2|3|4|5|6]:MEAN:RANGe:UPPer:TIME

Sets the upper X-axis limit for the trace mean range calculation.

Setting this value when CALCulate[1|2|3|4|5|6]:MEAN:RANGe:STATe is off will automatically turn CALCulate[1|2|3|4|5|6]:MEAN:RANGe:STATe on. The range used for the trace mean range is the same range used for the total power calculation, marker search range, and wavelength sweep range. Changing the range with this command will change all four ranges. Default units for the parameter are seconds.

Sending this command while span is not set to zero will result in a "Settings conflict" error.

Syntax

```
CALC:MEAN:RANG:UPP:TIME <numeric_value>[NS|US|MS|S]
```

CALCulate[1|2|3|4|5|6]:MEAN:RANGe:UPPer[:WAVelength]

Sets the upper X-axis limit for the trace mean range calculation.

Setting this value when CALCulate[1|2|3|4|5|6]:MEAN:RANGe:STATe is off will automatically turn CALCulate[1|2|3|4|5|6]:MEAN:RANGe:STATe on. The range used for the trace mean range is the same range used for the total power calculation, marker search range, and wavelength sweep range. Changing the range with this command will change all four ranges. Default units for the parameter are meters.

Sending the command when the instrument is in a zero span will generate a "Settings conflict" error.

Syntax

```
CALC:MEAN:RANG:UPP[:WAV] <numeric_value>
[M|UM|NM|PM|A]
```

Example

OUTPUT @Osa;"calc1:mean:state on"	! turn on trace mean function for trace A
OUTPUT @Osa;"calc1:mean:rang:upp:wav 1550nm"	! set high range for trace mean function !using Wavelength units

CALCulate[1|2|3|4|5|6]:MEAN:STATe

Turns the mean power calculation for a trace on or off.

Only one mean power calculation can be turned on at a time. For example, if a mean power calculation is being performed on trace A, turning a mean power calculation for trace B on will turn the calculation for trace A off.

Syntax

```
CALC[1|2|3|4|5|6]:MEAN:STAT OFF|ON|0|1

CALC[1|2|3|4|5|6]:MEAN:STAT?
```

Example

OUTPUT @Osa;"calc1:mean:state on"	! turn on trace mean function for trace A
OUTPUT @Osa;"calc1:mean:rang:stat on"	! turn on range for trace a mean function
OUTPUT @Osa;"calc1:mean:rang:low:freq 196 thz"	! set low range for mean trace function using !frequency units
OUTPUT @Osa;"calc1:mean:rang:upp:wav 1550nm"	! set high range for trace mean function !using Wavelength units
OUTPUT @Osa;"calc1:mean:data?"	! query for mean trace data, trace a
ENTER @Osa;Mean	
PRINT Mean	

CALCulate[1|2|3|4|5|6]:MINimum:CLEar

Clears the current minimum hold values for the trace and allows a new minimum hold to occur.

The trace will be initialized to the current value of the trace. If the specified trace is not in the minimum hold state, sending this command will have no effect.

Syntax

CALC:MIN:CLE

Example

OUTPUT @Osa;"calc1:min:state on"	! turn on trace mean function for trace A
OUTPUT @Osa;"calc1:min:clear"	! clear max data for trace A
OUTPUT @Osa;"init:cont on"data for trace A	! begin sweeping to acquire max hold

Related Key

RESET MIN/MAX HOLD

CALCulate[1|2|3|4|5|6]:MINimum[:STATe]

Turns minimum hold for a trace on or off.

The minimum hold operation compares the current amplitude value of each point on a trace in the current sweep to the corresponding point detected during the previous sweep, then stores the minimum value. CACLulate[1|2|3|4|5|6] refers to traces A, B, C, D, E, and F respectively.

Each trace can be set to one of four mutually exclusive states: normal, average (CALCulate[1|2|3|4|5|6]:AVERage[:STATe]), maximum hold (CALCulate[1|2|3|4|5|6]:MAXimum[:STATe]), or minimum hold (CALCulate[1|2|3|4|5|6]:MINimum[:STATe]). Enabling one state will disable any previous setting. For example, turning on trace averaging for trace B (CALCulate2:AVERage ON) would turn off a minimum hold state (CALCulate2:MINimum OFF).

Trace averaging, maximum, and minimum holds can be used in conjunction with trace math. If the trace that you wish to place a minimum hold on is the result of a math expression (C=log A–B), the minimum result of the math expression will be held. TRC will be the minimum difference of TRA and TRB.

Syntax

```
CALC[1|2|3|4|5|6]:MIN[:STAT] OFF|ON|0|1

CALC[1|2|3|4|5|6]:MIN[:STAT]?
```

Example

OUTPUT @Osa;"calc1:min:state on"	! turn on trace mean function for trace A
OUTPUT @Osa;"calc1:min:clear"	! clear max data for trace A
OUTPUT @Osa;"init:cont on"data for trace A	! begin sweeping to acquire max hold

Related Key

HOLD A NONE|MAX|MIN

CALCulate[1|2|3|4|5|6]:OFFset <param>

Sets the trace offset for a trace.

The Trace Math menu includes a “per trace” offset for each trace. The offset for each trace can be set independently. The units are determined by the amplitude units setting. When the amplitude units are set to “auto”, the units will be in dB in log display mode and “X” (unitless ratio) in linear display mode.

Syntax

```
CALC:OFF <param>
```

Example

```
OUTPUT @Osa;"calc1:offs -5db"! set trace A offset to -5 db
```

Related Key

TRACE A OFFSET

CALCulate[1|2|3|4|5|6]:POWer[:DATA]?

For 86141B, 86144B, 86146B Power Meter Mode only.

Reads the total input power and offset measured at the photodetector input.

Syntax

```
CALC:POW[:DATA]?
```

Example

OUTPUT @Osa;"calc1:pow?"	! query for total power
ENTER @Osa;Power	
PRINT Power	

CALCulate[1|2|3|4|5|6]:SIGMa[:DATA]?

Returns the sigma calculation results in meters.

Sigma is the rms value of the spectral width of the trace points based on a Gaussian distribution. The power and wavelength of each spectral component is used to calculate mean wavelength.

$$sigma = \sqrt{\sum_{i=1}^{n} \frac{P_i}{P_o} \left(\frac{trace\ point\ spacing}{resolution\ bandwidth} \right) (\lambda_i - \bar{\lambda})^2}$$

where:

$\bar{\lambda}$ is mean wavelength (Center of Mass) as defined in CALCulate[1|2|3|4|5|6]:CENTermass

λ_i is the wavelength of a single trace point

P_i is the power of a single trace point

P_o is total power as defined in CALCulate[1|2|3|4|5|6]:CENTermass

CALCulate[1|2|3|4|5|6]:FWHM:STATe must be on for this calculation. Trace A corresponds to CALCulate 1 and so on. Corrections to all calculations are made for the slope and variation of the resolution bandwidth filter over the wavelength range of the trace. When CALCulate[1|2|3|4|5|6]:TPOWer:IRANge is on, the calculation is performed over the upper and lower range limits. All five common calculation ranges (SOURce, CENTermass, FWHM, SIGMa, and TPOWer) share the same limits. Sending a CALCulate[1|2|3|4|5|6]:SIGMa? query when the
CALCulate[1|2|3|4|5|6]:FWHM:STATe is off will generate an error.

Syntax

```
CALC:SIGM[:DATA]?
```

Example

OUTPUT @OSA;"calc1:fwhm:state on"	! turn full width half max on for trace a
OUTPUT @OSA;"calc1:sigm:data?"	! query for sigma calculation results

CALCulate[1|2|3|4|5|6]:THReshold

Sets the value for the marker search threshold.

Syntax

```
CALC:THR <param>[W|MW|UW|DBM]
CALC:THR?
```

Example

OUTPUT @Osa;"calc:mark1:stat on"	! turn marker one on
OUTPUT @Osa;"calc:thr:stat on"	! turn marker search threshold on
OUTPUT @Osa;"calc:thr -60dbm"	! sets marker search threshold to 60db

Related Key

(MARKER SETUP) THRESHOLD VALUE

CALCulate[1|2|3|4|5|6]:THReshold:STATe

Turns on the marker search threshold function.

When this threshold function is on, marker peak searches will ignore peaks below the threshold value.

Syntax

```
CALC:THR:STAT ON|OFF|1|0

CALC:THR:STAT?
```

Example

OUTPUT @Osa;"calc:mark1:stat on"	! turn marker one on
OUTPUT @Osa;"calc:thr:stat on"	! turn marker search threshold on

Related Key

(MARKER SETUP) USE MARKER SEARCH THRESHOLD

CALCulate[1|2|3|4|5|6]:TPOWer[:DATA]?

Returns the total power of the specified trace. Trace A corresponds to CALCulate1, trace B to CALCulate2, and so on. Corrections to the total power are made for the slope and variation of the resolution bandwidth filter over the wavelength range of the trace. When CALCulate[1|2|3|4|5|6]:TPOWer:IRANge is on, the total power is calculated over the upper and lower range limits; otherwise, the total power is calculated over the entire trace. Sending this query when the CALCulate[1|2|3|4|5|6]:TPOWer:STATe is off will generate a "Settings conflict" error.

Syntax

```
CALC:TPOW[:DATA]?
```

Example

OUTPUT @Osa;"calc1:tpow:stat on"	! turn total power calculation on for trace a
OUTPUT @Osa;"calc1:tpow:iran:stat on"	! turn on range limits for total power calculation
OUTPUT @Osa;"calc1:tpow:iran:upp 1555nm"	! set upper range for total power to 1555nm
OUTPUT @Osa;"calc1:tpow:iran:low 1530nm"	! set lower range for total power calculation to !1530nm
OUTPUT @Osa;"calc1:Tpow:data?"	! query for total power data
ENTER @Osa;Tpow	
PRINT Tpow	

CALCulate[1|2|3|4|5|6]:TPOWer:IRANge:LOWer

Sets the lower X-axis limit range for the TPOWer, SOURce, CENTermass, FWHM, and SIGMa calculations for all traces. Setting this value when CALCulate[1|2|3|4|5|6]:TPOWer:IRANge[:STATe] is off will automatically turn CALCulate[1|2|3|4|5|6]:TPOWer:IRANge[:STATe] on. The range used for the total power integration is the same range used for the marker search range, trace mean range, and wavelength range. Changing the range with this command will change all four ranges.

Default units for the parameter are meters. Sending the command when the instrument is in a zero span will generate a "Settings conflict" error.

Syntax

```
CALC[1|2|3|4|5|6]:TPOW:IRAN:LOW
<numeric_value>[M|UM|NM|A|HZ|KHZ|MHZ|GHZ|THZ]

CALC[1|2|3|4|5|6]:TPOW:IRAN:LOW?
```

Example

OUTPUT @Osa;"calc1:tpow:stat on"	! turn total power calculation on for trace a
OUTPUT @Osa;"calc1:tpow:iran:stat on"	! turn on range limits for total power calculation
OUTPUT @Osa;"calc1:tpow:iran:upp 1555nm"	! set upper range for total power to 1555nm
OUTPUT @Osa;"calc1:tpow:iran:low 1530nm"	! set lower range for total power calculation to !1530nm
OUTPUT @Osa;"calc1:Tpow:data?"	! query for total power data
ENTER @Osa;Tpow	
PRINT Tpow	

CALCulate[1|2|3|4|5|6]:TPOWer:IRANge[:STATe]

Turns the total power calculation range for all traces on or off. Setting IRANge:STATe to on will set the corresponding TPOWer:STATe to on. Although there is a single range controlling the total power integration, trace mean calculation, marker search range, and wavelength sweep range, there are four independent state settings for limiting them to the range. If all four states are off, setting the CALCulate[1|2|3|4|5|6]:TPOWer:STATe to on, will initialize the lower limit to

$$\text{start} + \frac{\text{span}}{3}$$

and the upper limit to

$$\text{start} + \left(2 \times \frac{\text{span}}{3}\right) \quad .$$

Sending the command when the instrument is in a zero span will generate a "Settings conflict" error.

Syntax

```
CALC[1|2|3|4|5|6]:TPOW:IRAN[STAT] OFF|ON|0|1

CALC[1|2|3|4|5|6]:TPOW:IRAN[STAT]?
```

Example

OUTPUT @Osa;"calc1:tpow:stat on"	! turn total power calculation on for trace a
OUTPUT @Osa;"calc1:tpow:iran:stat on"	! turn on range limits for total power calculation
OUTPUT @Osa;"calc1:tpow:iran:upp 1555nm"	! set upper range for total power to 1555nm
OUTPUT @Osa;"calc1:tpow:iran:low 1530nm"	! set lower range for total power calculation to !1530nm
OUTPUT @Osa;"calc1:Tpow:data?"	! query for total power data
ENTER @Osa;Tpow	
PRINT Tpow	

CALCulate[1|2|3|4|5|6]:TPOWer:IRANge:UPPer

Sets the upper X-axis limit range for the TPOWer, SOURce, CENTermass, FWHM, and SIGMa calculations for all traces. Setting this value when the CALCulate[1|2|3|4|5|6]:TPOWer:IRANge[:STATe] is off will automatically turn the CALCulate[1|2|3|4|5|6]:TPOWer:IRANge[:STATe] on. The range used for the total power calculation is the same range used for the marker search range, trace mean range and wavelength range. Changing the range with this command will change all four ranges.

Default units for the parameter are meters. Sending the command when the instrument is in a zero span will generate a "Settings conflict" error.

Syntax

CALC[1|2|3|4|5|6]:TPOW:IRAN:UPP
<numeric_value>[M|UM|NM|A|HZ|KHZ|MHZ|GHZ|THZ]

CALC[1|2|3|4|5|6]:TPOW:IRAN:UPP?

Example

OUTPUT @Osa;"calc1:tpow:stat on"	! turn total power calculation on for trace a
OUTPUT @Osa;"calc1:tpow:iran:stat on"	! turn on range limits for tpow calculation
OUTPUT @Osa;"calc1:tpow:iran:upp 1555nm"	! set upper range for tpow to 1555nm
OUTPUT @Osa;"calc1:tpow:iran:low 1530nm"	! set lower range for tpow calculation to !1530nm
OUTPUT @Osa;"calc1:Tpow:data?"	! query for total power data
ENTER @Osa;Tpow	
PRINT Tpow	

CALCulate[1|2|3|4|5|6]:TPOWer:STATe

Turns the total power calculation for a trace on or off. Only one total power calculation can be turned on at a time. For example, if a total power calculation is being performed on trace A, turning a total power calculation for trace B on will turn the calculation for trace A off. Turning this function on in zero span generates a "Settings conflict" error.

Syntax

```
CALC[1|2|3|4|5|6]:TPOW:STAT OFF|ON|0|1

CALC[1|2|3|4|5|6]:TPOW:STAT?
```

Example

OUTPUT @Osa;"calc1:tpow:stat on"	! turn total power calculation on for trace a
OUTPUT @Osa;"calc1:tpow:iran:stat on"	! turn on range limits for total power calculation
OUTPUT @Osa;"calc1:tpow:iran:upp 1555nm"	! set upper range for total power to 1555nm
OUTPUT @Osa;"calc1:tpow:iran:low 1530nm"	! set lower range for total power calculation to !1530nm
OUTPUT @Osa;"calc1:Tpow:data?"	! query for total power data
ENTER @Osa;Tpow	
PRINT Tpow	

CALibration Subsystem Commands

This subsystem has the function of performing system calibration. The commands in this subsystem have the following command hierarchy:

CALibration

```
:ALIGn
    [:AUTO]:MARKer
    EXTernal
    :FILTER
    :INTernal
    :MARKer
    :PRESet
    :TADD
:DATE?
:POWer
    :DATE?
    :PATH OFF|ON|0|1
    :EXTernal
    :STATe ON|OFF|1|0
    :VALue <param>
    :WAVelength <numeric_value>[M|UM|NM|A|HZ|KHZ|MHZ|GHZ]
:PRESet
:STATe OFF|ON|0|1
:WAVelength
    :DATE?
    :EWC
        :FUNCtion OFF|ON|0|1
        :RANGe FULL|TELEcom
    [:EXTernal]
        :MULTipoint
            :DATA
            :DELete
            :MARKer
        [:NORMal]
            :MARKer
        :VALue
    :INTernal
        :MULTipoint
        [:NORMal]
    :MODE NORMal|MULTipoint
    :PATH OFF|ON|0|1
    :STATe OFF|ON|0|1
    :USER:DATA <string>
:ZERO[:AUTO] OFF|ON|0|1|ONCE
```

CALibration:ALIGnment

Performs an automatic alignment of the instrument at the wavelength of the largest signal found in full span. This aligns the monochromator output with the photodetector for improved amplitude accuracy. Sending this command with a marker on screen will generate a "Settings conflict" error.

Syntax

```
CAL:ALIG
```

Related Key

AUTO ALIGN

CALibration:ALIGnment:EXTernal

Performs an alignment of the instrument using an external broadband source. After you set the desired start and stop wavelengths, the instrument performs an alignment at each of several wavelengths and stores the values in a wavelength alignment table. This results in improved amplitude accuracy.

NOTE

Error 5056, Trajectory align cannot find input signal, will occur if there is not enough power in the start and stop wavelength region.
Error 5060, Trajectory align failed, will occur if the align procedure failed.

Syntax

```
CAL:ALIG:EXT
```

Related Commands

```
CALibration:ALIGnment:EXTernal
```

CALibration:ALIGnment:FILTer

Available for the 86144B/86146B while in Filter Mode only.

Performs an alignment of the instrument at the filter marker. This aligns the monochromator output with photodetector input for improved amplitude accuracy.

Syntax

```
CAL:ALIG:FILT
```

Related Key

Auto Align

CALibration:ALIGnment:INTernal

Adjusts the mechanical position of the instrument's internal optical components ensuring amplitude accuracy of your measurements. Before initiating the alignment, connect the internal calibrator to the front-panel input connector.

The instrument automatically sets the start wavelength at 1490 nm, stop wavelength at 1590 nm, span, and reference level, and then performs a fully automatic, internal auto align. The input signal is aligned at equally spaced alignments (minimum 50 nm spacing between points) for the internal, multi-mode fiber or (minimum 5 nm spacing between points) for the external, single-mode fiber (Agilent 86144B/86146B only).

NOTE

Error 5056, Trajectory align cannot find input signal, will occur if a broadband light source is not connected to the front-panel input connector.
Error 5060, Trajectory align failed, will occur if the align procedure failed.

Syntax

```
CAL:ALIG:INT
```

Related Key

Calibrator Multi-Pt Align

CALibration:ALIGnment:MARKer[1|2|3|4]

Performs an automatic alignment of the instrument at the wavelength of the specified marker. This aligns the monochromator output with the photodetector for improved amplitude accuracy. Sending this command without the specified marker on will generate a "Settings conflict" error.

Syntax

```
CAL:ALIG:MARK1|2|3|4]
```

Related Key

Auto Align

CALibration:ALIGnment:PRESet

Sets the alignment of the instrument to the preset factory-calibrated values. After sending this command, it is recommended that an auto align be performed. (See CALibrate:ALIGnment and CALibration:ALIGnment:MARKer).

Syntax

```
CAL:ALIG:PRES
```

Related Commands

CALibrate:ALIGnment
CALibration:ALIGnment:MARKer

Related Key

Factory Preset (IP)

CALibration:ALIGnment:TADD

Allows alignment at several wavelengths, ensuring amplitude accuracy of your measurements. Before initiating the alignment, connect a broadband light source to the front-panel input connector. (You may add and/or update the existing points in the trajectory table.) Enter CALibration:ALIGnment:PRESet to clear the table.

Turn on an active marker (using the CALCulate:MARKer [1|2|3|4]:X:WAVelength command) at the desired wavelength. If an active marker is not turned on, the instrument will automatically turn on an active marker and auto align at the largest input signal. The minimum recommended spacing between points is 5 nm for the external, single mode fiber (Agilent 86144B/86146B only) and 50 nm for the internal, multi-mode fiber. The span must be >3 nm for the single mode fiber and >25 nm for the multi-mode fiber.

NOTE

Error 5001, Auto align cannot find input signal, will occur if a broadband light source is not connected to the front-panel input connector.
Error 5070, Trajectory Add Failed, will occur if the trajectory table is full or the computed trajectory table is invalid.

Syntax

```
CAL:ALIG:TADD
```

Related Key

Auto Align & Add To Trajectory

CALibration:DATE?

Returns the date of the most recent factory calibration.

Syntax

```
CAL:DATE?
```

Related Key

WAVELENGTH CAL SETUP

CALibration:POWer

Performs a power calibration. The calibration is aborted if the power measured on the input signal is more than 3 dB higher or 10 dB lower than the value specified in the CALibration:POWer:VALue command.

Syntax

```
CAL:POW
```

Related Commands

CALibraton:POWer:VALue

Related Key

POWER CAL SETUP, PERFORM CALIBRATION

CALibration:POWer:DATE?

Returns the date of the most recent user power calibration.

Syntax

```
CAL:POW:DATE?
```

Related Key

POWER CAL SETUP

CALibration:POWer:PATH

Available for the 86144B/86146B only.

Enables power calibration for the external fiber path (path 2). Calibration for Path 1 is always enabled.

For CALibration:POWer:PATH2 ON, a power calibration (for example, CALibration:POWer:INTernal) will calibrate both the external and internal paths. A 9 μm patchcord must be connected between the Monochromator Output and the Photodetector Input before performing a power calibration on the external path.

Syntax

```
CAL:POW:PATH[1|2] <on|off|0|1>

CAL:POW:PATH[1|2]?
```

Related Commands

CALibration:WAVelength:PATH

Related Keys

(Power Cal Setup) Power Cal External Path

CALibration:POWer:STATe

Specifies whether or not the calibration power data is applied. Amplitude accuracy is only specified with power calibration on.

Syntax

```
CAL:POW:STAT OFF|ON|0|1

CAL:POW:STAT?
```

CALibration:POWer:VALue

Specifies the power to be used for calibration. Default units are set by the UNIT:POWer command.

Syntax

```
CAL:POW:VAL <param>

CAL:POW:VAL?
```

Related Commands

CALibration:POWer:VALue

Related Key

(POWER CAL SETUP) SET CALIBRATION POWER

CALibration:POWer:WAVelength

Specifies the wavelength of the signal used for the amplitude calibration.

Syntax

```
CAL:POW:WAV<numeric_value>
[M|UM|NM|A|HZ|KHZ|MHZ|GHZ]

CAL:POW:WAV?
```

Related Key

(WAVELENGTH CAL SETUP) SET CALIBRATION WAVELENGTH

CALibration:PRESet

Presets the calibration of the instrument to factory-calibrated values. This cancels the effect of any previous CALibration:POWer or CALibration:WAVelength.

Syntax

```
CAL:PRES
```

Related Key

Factory Preset (IP)

CALibration:STATe

Specifies if the user calibration data is applied or not. Amplitude accuracy and wavelength accuracy are only specified when calibration is on. The response value is the logical *and* of CALibration:POWer:STATe? and CALibration
:WAVelength:STATe?.

Syntax

```
CAL:STAT OFF|ON|0|1

CAL:STAT?
```

CALibration:WAVelength:DATE?

Returns the date of the most recent user wavelength calibration.

Syntax

```
CAL:WAV:DATE?
```

CALibration:WAVelength:EWC:FUNCtion

Enables or disables the enhanced wavelength calibration for subsequent calibrations. EWC must be enabled for wavelength accuracy specifications to apply in the range selected.
0 = disables EWC
1 = enables EWC (default on factory preset)

Syntax

```
CAL:WAV:EWC:FUNC ON|OFF|0|1

CAL:WAV:EWC:FUNC?
```

Related Key

ENHANCED WVL CAL SETUP

CALibration:WAVelength:EWC:RANGe

Sets the range over which the enhanced wavelength calibration (EWC) is performed. The two ranges for the EWC are FULL and TELEcom. FULL covers the range from 605 nm to 1670 nm. TELEcom covers the smaller span more relevant to telecommunications: 1270 to 1670 nm. Factory preset is TELEcom.

When enabled, the EWC is applied during internal calibrations. EWC must be enabled for wavelength accuracy specifications to apply in the range selected. Setting the range to FULL will require a longer calibration time for an internal calibration, but will provide enhanced wavelength accuracy over the full range.

Syntax

```
CAL:WAV:EWC:RANG FULL|TEL

CAL:WAV:EWC:RANG?
```

Related Key

ENHANCED WVL CAL SETUP

CALibration:WAVelength[:EXTernal]:Multipoint

Performs a single point enhanced wavelength calibration using an external source. The multipoint data is adjusted at the wavelength selected by the CALibration:WAVelength:EXTernal:VALue command. If the wavelength measured on the input signal differs more than ±2.5 nm from the value specified, the calibration is aborted.

NOTE For this command to function properly, it must be used in the correct sequence with the following commands:
CALibration:WAVelength:EXTernal:VALue <param>
CALibration:WAVelength[:EXTernal]:MULTipoint

Syntax

```
CAL:WAV[:EXT]:Multipoint
```

Related Key

(Wavelength Calibration Setup) Calibration Data will be: Offset

CALibration:WAVelength[:EXTernal]:Multipoint:DATA

Enters user measured multipoint wavelength calibration data. The command takes the data in <string> format and writes it to the wavelength calibration tables.

Xn are wavelengths in vacuum in meters of the wavelength standard (not the value indicated by the OSA). The Xn minimum spacing is 2 pm, and must be in increasing order. There is a maximum of 10000 pairs. Linear interpolation is used between the data points when calculating the wavelength corrections.

Yn are wavelength errors in vacuum (indicated wavelength - actual wavelength) in meters. Yn magnitude must be less than 200 pm.

The spacing between data points must be larger than the magnitude of the change in error between data points. Specifically, the magnitude of the slope must be less than 1. Where slope = (Y(n+1) - Y(n))/(X(n+1)-X(n)). For example if Xn are 10 pm apart, Yn must change by less than 10 pm.

The query returns any external multipoint wavelength calibration data in string format. For example:

+1.45011471E-006,+0.00000000E+000,+1.50011168E-006,+9.20199449E-13, +1.56010779E-006,-1.12468277E-012,+1.61010432E-006,+0.00000000E+000

Previous multipoint wavelength data are cleared each time the command is used. Therefore, to modify the multipoint wavelength calibration data, use the query to obtain the existing table of data, then make changes to the table and reenter it using this command.

When measuring new external multipoint calibration data, use “CALibration:WAVelength:MODE:NORMal” to disable previous wavelength calibration data.

Syntax

```
CAL:WAV[:EXT]:MULT:DATA X1,Y1,X2,Y3,....,Xn,Yn

CAL:WAV[:EXT]:MULT:DATA?
```

CALibration:WAVelength[:EXTernal]:MULTipoint:DELete

Deletes calibration data entered by CALibration[:EXTernal]:WAVelength:MULTipoint:DATA.

Syntax

```
CAL:WAV[:EXT]:MULT:DEL
```

Related Key

Factory Preset (IP)

CALibration:WAVelength[:EXTernal]:MULTipoint:MARKer[1|2|3|4|]

Performs a single point enhanced wavelength calibration using the signal nearest the marker. The marker location must be selected before this command can be run. The multipoint data is adjusted at the wavelength selected by the CALibration:WAVelength:EXTernal:VALue command. If the wavelength measured on the input signal differs more than ±2.5 nm from the value specified, the calibration is aborted. If no multipoint data exists, the calibration is aborted and a settings conflict error is generated.

This command is necessary if a signal with two or more peaks is input to the optical spectrum analyzer during the calibration. If a source has more than one peak, the marker is used to determine which peak will be calibrated.

NOTE

For this command to function properly, it must be used in the correct sequence with the following commands:
CALibration:WAVelength:EXTernal:VALue <param>
CALCulate:MARKer[1|2|3|4]:X:WAVelength <param>
CALibration:WAVelength[:EXTernal]:MULTipoint:MARKer[1|2|3|4]

Syntax

```
CAL:WAV[:EXT:MULT:MARK[1|2|3|4|]
```

CALibration:WAVelength[:EXTernal][:NORMal]

Performs a single point enhanced wavelength calibration using an external source. All multipoint wavelength calibration offsets are disabled. The multipoint data can also be disabled with CALibration:WAVelength:MODE:NORMal.

If the wavelength measured on the input signal differs more than ±2.5 nm from the value specified, the calibration is aborted.

NOTE For this command to function properly, it must be used in the correct sequence with the following commands:
CALibration:WAVelength:EXTernal:VALue <param>
CALibration:WAVelength[:EXTernal] [:NORMal]

Syntax

```
CAL:WAV[:EXT][:NORM]
```

Related Key

(Wavelength Calibration Setup) Calibration values will be: Replaced

CALibration:WAVelength[:EXTernal][:NORMal]:MARKer[1|2|3|4|]

Performs a single point enhanced wavelength calibration using the signal nearest the marker. The marker location must be selected before this command can be run. All multipoint wavelength calibration offsets are disabled. The multipoint data can also be disabled with CALibration:WAVelength :MODE:NORMal. If the wavelength measured on the input signal differs more than ±2.5 nm from the value specified in the CALibration:WAVelength:VALue command, the calibration is aborted.

This command is necessary if a signal with two or more peaks is input to the optical spectrum analyzer during the calibration. If a source has more than one peak, the marker is used to determine which peak will be calibrated.

NOTE For this command to function properly, it must be used in the correct sequence with the following commands:
CALibration:WAVelength:EXTernal:VALue <param>
CALCulate:MARKer[1|2|3|4]:X:WAVelength <param>
CALibration:WAVelength[:EXTernal][:NORMal]:MARKer[1|2|3|4]

Syntax

```
CAL:WAV[:EXT[:NORM]:MARK[1|2|3|4|]
```

CALibration:WAVelength[:EXTernal]:VALue

Specifies the wavelength for a single point calibration. Default units for the parameter are meters.

Syntax

```
CAL:WAV[:EXT]:VAL <param>[M|UM|NM|A]

CAL:WAV[:EXT]:VAL?
```

Related Key

(Wavelength Calibration Setup) Set Calibration Wavelength

CALibration:WAVelength:INTernal:MULTipoint

Performs an enhanced wavelength calibration using the internal calibrator. Any existing multipoint wavelength calibration data is adjusted relative to this calibration. If no multipoint data exists, the calibration is aborted and a settings conflict error is generated.

NOTE The internal calibrator must be connected to the input before sending this command.

Syntax

```
CAL:WAV:INT:MULT
```

Related Key

(Wavelength Calibration Setup) Signal Source: Calibration

(Wavelength Calibration Setup) Calibration Data will be: Offset

CALibration:WAVelength:INTernal[:NORMal]

Performs the enhanced wavelength calibration using the internal calibrator. Any existing multipoint wavelength calibration data is cleared.

NOTE The internal calibrator must be connected to the input before sending this command.

Syntax

```
CAL:WAV:INT[:NORM]
```

Related Key

(Wavelength Calibration Setup) Signal Source: Calibration

(Wavelength Calibration Setup) Calibration Data will be: Replaced

CALibration:WAVelength:MODE

Enables or disables the multipoint wavelength calibration data. NORMal disables the multipoint wavelength calibration data. MULTipoint enables the data from the last multipoint wavelength calibration per CALibration:WAVelength:MULT:DATA. This data must be entered before MULTipoint mode can be selected.

The following commands change the setting of CALibration:WAVelength:MODE to NORMal:

CALibration:WAVelength[:EXTernal]:NORMal
CALibration:WAVelength[:EXTernal]:NORMal:MARKer [1|2|3|4]

Once multipoint data is entered, the following commands will enable the multipoint data. Refer to the specific commands for further information.

CALibration:WAVelength[:EXTernal]:MULTipoint
CALibration:WAVelength[:EXTernal]:MULTipoint:MARKer [1|2|3|4]
CALibration:WAVelength:INTernal:MULTipoint
CALibration:WAVelength:MULTipoint:DATA

Syntax

```
CAL:WAV:MODE NORM|MULT

CAL:WAV:MODE?
```

Related Key

(Wavelength Calibration Setup) when calibration completes, user multi-point wavelength calibration will be :offset replaced.

CALibration:WAVelength:STATe

Specifies whether or not the calibration wavelength data is applied. Wavelength accuracy is only specified with wavelength calibration on.

Syntax

```
CAL:WAV:STAT OFF|ON|0|1

CAL:WAV:STAT?
```

CALibration:WAVelength:PATH

For 86144B/86146B only.

Turns wavelength calibration of the external fiber path on or off. When on, wavelength calibration of the external fiber path (path 2) is enabled. Calibration for Path 1 is always enabled.

For CALibration:WAVelength:PATH2 ON, a wavelength calibration (for example, CALibration:WAVelength:INTernal) will calibrate both the external and internal paths. A 9 µm patchcord must be connected between the Monochromator Output and the Photodiode Input before preforming a wavelength calibration on the external path.

Syntax

```
CAL:WAV:PATH[1|2] <on|off|1|0>
CAL:WAV:PATH?
```

Related Commands

CALibration:POWer:PATH

Related Keys

(Wavelength Cal Setup) Wavelength Cal External Path

CALibration:WAVelength:USER:DATA

Although this command is available, some OSA firmware versions do not support it. In place of this command, it is recommended that you use: CALibration:WAVelength[:EXTernal]:MULTipoint:DATA.

All information given for CALibration:WAVelength:MULTipoint:DATA will apply to this command.

Syntax

```
CAL:WAV:USER:DATA <string>
CAL:WAV:USER:DATA?
```

CALibration:ZERO[:AUTO]

Specifies whether or not auto zeroing is enabled. Auto zeroing measures and compensates for the dark current of the photodetector for improved amplitude accuracy. The ONCE parameter causes the dark current to be measured one time, and then the resulting correction is applied to *all* subsequent measurements. Auto Zero on causes the dark current to be measured every 10 seconds, and then the resulting correction is applied to the next sweep.

Syntax

```
CAL:ZERO[:AUTO] OFF|ON|0|1|ONCE

CAL:ZERO[:AUTO]?
```

Related Key

(AMPLITUDE SETUP) AUTO ZERO
ZERO NOW

DISPlay Subsystem Commands

The DISPlay controls the selection and presentation of textual, graphical, and TRACe information.

The commands in this subsystem have the following command hierarchy:

DISPlay

```
[:WINDow]
[:WINDow[1]]
    :ANNotation[:ALL] OFF|ON|0|1
    :POPup[1|2|3|4][:ALL] OFF|ON|0|1
    :TEXT
        :CLEar
        :DATA <string> | <data_block>
    :TRACe
        :ALL [:SCALe][:AUTO]
            :MARKer OFF|ON|0|1
            :OPTimize OFF|ON|0|1
        :GRATicule:GRID[:STATe] OFF|ON|0|1
        [:STATe] <trace_name>, OFF|ON|0|1
        :X[:SCALe]:AUTO:SPAN <numeric_value> [M|NM|UM]
            ;AUTO OFF|ON|0|1
        :Y[SCALe]
            :AUTO:PDIVision <numeric_value>[DB]
            :SPACing LINear|LOGarithmic
        :Y[1|2][SCALe]
            :AUTO:PDIVision:AUTO OFF|ON|0|1
            :LINear OFF|ON|0|1
            :PDIVision <numeric_value>[DB]
            :RLEVel <numeric_value>[DBM|W|MW|UM|NW|DB]
            :RPOSition <numeric_value>
```

DISPlay[:WINDow[1]]

Turns the display of the interface on or off.

Disabling the display speeds up the remote commands because the graphics no longer need to be updated. This is useful for manufacturing applications that use a PC to display test results.

The display is enabled by sending the DISPlay:WINDow:ON command. Turning the display on or off does not affect the instrument state. Note that there is not a front panel function for turning off the display.

Syntax

```
DISP[:WIND[1]] OFF|ON|0|1
```

Example

```
OUTPUT @Osa;"disp:wind off"
OUTPUT @Osa;"disp:wind on"
```

DISPlay[:WINDow[1]]:ANNotation[:ALL]

Turns the screen annotation on or off. This command affects only the X-axis and Y-axis labeling and labeling within the graticule.

Syntax

```
DISP[:WIND[1]]:ANN[:ALL] ON|OFF|0|1
DISP[:WIND[1]]:ANN[:ALL]?
```

DISPlay[:WINDow[1]]:POPup[1|2|3|4][:ALL]

Turns the power meter area on the optical spectrum analyzer screen on or off.

Syntax

```
DISP[:WIND[1]]:POP[1|2|3|4][:ALL] OFF|ON|0|1
```

DISPlay[:WINDow[1]]:TEXT:CLEar

Clears the title on the display resulting from previous use of the DISPlay[:WINDow[1]]:TEXT:DATA command.

Syntax

```
DISP[:WIND[1]]:TEXT:CLE
```

Example

OUTPUT @Osa;"disp:wind:ann:all off"	! example 17 display string
OUTPUT @Osa;"disp:wind:text:data 'display string'"	! place string onto display
OUTPUT @Osa;"disp:wind:text:clear"	! remove string from display

DISPlay[:WINDow[1]]:TEXT:DATA

Writes text on the display in the Title area. Uses the <data_block> parameter to send extended ASCII characters such as control codes and symbols.

Syntax

```
DISP[:WIND[1]]:TEXT:DATA <string>|<data_block>

DISP[:WIND[1]]:TEXT:DATA?
```

Example

OUTPUT @Osa;"disp:wind:ann:all off"	! example 17 display string
OUTPUT @Osa;"disp:wind:text:data 'display string'"	! place string onto display
OUTPUT @Osa;"disp:wind:text:clear"	! remove string from display

Related Key

SET TITLE

DISPlay[:WINDow[1]]:TRACe:ALL[:SCALe][:AUTO]

Finds the largest input signal using trace A and sets the span and vertical scale to display that signal. This command performs the same function as the front-panel AUTO-MEAS key.

The following defines the instrument state settings altered by Auto Measure. State settings that are not listed are not altered.

Because many instrument state setting are altered, it is recommended you use this command only to find unknown signals. It is not recommend this command be used in the middle of a measurement routine.

Center Wavelength	According to signal wavelength and bandwidth
Span	Set according to automeasure setup panel. In some cases, this may also be a function of signal characteristics.
Grating Order	Auto
Sensitivity	Set according to automeasure setup panel. In some cases, this may also be a function of signal characteristics.
dB/div	Set according to automeasure setup panel. In some cases, this may also be a function of signal characteristics.
Video Bandwidth	Auto
Auto Range Enable	On
Trans-Z Lock	Off
Repetitive Sweep	On (front panel), Off (remote control)
Sweep Time	Auto
Auto Chop Mode	On
Gated Sweep Enable	Off
Sweep Trigger Mode	Internal
Trace Length	1001
Wavelength Limit	On
Reference Level	According to signal amplitude
Linear Display Mode	Off
Resolution Bandwidth	According to signal characteristics
Res-BW to Span Ratio	0.01
Peak Search on EOS	Off
Line Markers	Off
Trace Integration Limit	Off
Search Limit	Off
Trace Integration	Off
Trace Mean Calculation	Off

For each trace, except trace A:

Trace Math	Off
Update	Off
View	Off
Hold Mode	None
Averaging	Off

Trace A is identical, except:

Update	On
View	On

For each marker, except marker 1:

Visibility	Off
Marker BW	Off
Delta Mode	Off
Marker Trace	Trace A
Noise Marker	Off

Marker 1 is identical, except when the final span is non-zero as follows:

Visibility	On
Wavelength	Highest point on selected signal

Syntax

```
DISP[:WIND[1]]:TRAC:ALL[:SCAL][:AUTO]
```

Example

OUTPUT @Osa;"disp:trac:all:mark on"	! set automeasure to find signal closest to marker
OUTPUT @Osa;"disp:trac:all:opt on"	! set automeasure to optimize sensitivity
OUTPUT @Osa;"disp:trac:x:auto:span:auto off"	! turn automeasure auto span off
OUTPUT @Osa;"disp:trac:x:auto:span 5nm"	! set automeasure final span value

OUTPUT @Osa;"disp:trac:Y1:auto:pdiv:auto off"	! turn automeasure auto vertical scaling off
OUTPUT @Osa;"disp:trac:Y1:auto:pdiv 5"	! set vertical power scale to 5/division
OUTPUT @Osa;"disp:trac:all:auto"	! perform automeasure

Related Key

Auto Measure

DISPlay[:WINDow[1]]:TRACe:ALL[:SCALe][:AUTO]:MARKer

Changes the DISPlay:WINDow:TRACe:ALL:SCALe:AUTO (Auto Measure) command to find the input signal closest to the marker and set the span and vertical scale to view that signal. This command also sets single sweep mode.

Syntax

```
DISP[:WIND[1]]:TRAC:ALL[:SCAL][:AUTO]:MARK OFF|ON|0|1

DISP[:WIND[1]]:TRAC:ALL[:SCAL][:AUTO]:MARK?
```

Example

OUTPUT @Osa;"disp:trac:all:mark on"	! set automeasure to find signal closest to marker

Related Key

(Auto Measure Setup) AutoMeas at Marker

DISPlay[:WINDow[1]]:TRACe:ALL[:SCALe][:AUTO]:OPTimize

Changes the DISPlay:WINDow:TRACe:ALL:SCALe:AUTO (Auto Measure) command to optimize sensitivity after finding the input signal. This command also sets single sweep mode.

Syntax

```
DISP[:WIND[1]]:TRAC:ALL[:SCAL][:AUTO]:OPT OFF|ON|0|1

DISP:WIND[1]]:TRAC:ALL[:SCAL][:AUTO]:OPT?
```

Example

OUTPUT @Osa;"disp:trac:all:opt on"	! set automeasure to optimize sensitivity

Related Key

(Auto Measure Setup) Optimize Sensitivity

DISPlay[:WINDow[1]]:TRACe:GRATicule:GRID[:STATe]

Turns the graticule on or off.

Syntax

```
DISP[:WIND[1]]:TRAC:GRAT:GRID[:STAT]OFF|ON|0|1

DISP[:WIND[1]]:TRAC:GRAT:GRID[:STAT]?
```

Example

```
OUTPUT @Osa;"disp:trac:grat:grid off"      ! turn grid off
```

DISPlay[:WINDow[1]]:TRACe[:STATe]:

Turns the trace display on or off. Specifying any trace other than the ones listed will generate an “Illegal parameter value” error.

Syntax

```
DISP[:WIND[1]]:TRAC[:STAT]TRA|TRB|TRC|TRD|TRE|TRF,
OFF|ON|0|1

DISP[:WIND[1]]:TRAC[:STAT]?TRA|TRB|TRC|TRD|TRE|TRF
```

Example

```
OUTPUT @Osa;"disp:trac trb,on"      ! display trace B
```

DISPlay[:WINDow[1]]:TRACe:X[:SCALe]:AUTO:SPAN

Specifies the final span after a DISPlay:WINDow:TRACe:ALL:SCALe:AUTO (Auto Measure) command.

Syntax

```
DISP[:WIND[1]]:TRAC:X[:SCAL]:AUTO:SPAN
<numeric_value>[M|UM|NM]

DISP[:WIND[1]]:TRAC:X[:SCAL]:AUTO:SPAN?
```

Example

OUTPUT @Osa;"disp:trac:x:auto:span 5nm"	! set automeasure final span value

Related Key

(Auto Measure Setup) Span

DISPlay[:WINDow[1]]:TRACe:X[:SCALe]:AUTO:SPAN:AUTO

Specifies whether the final span after a DISPlay:WINDow:TRACe:ALL:SCALe :AUTO (Auto Measure) command should be set automatically, based on properties of the measured signal.

Syntax

```
DISP[:WIND[1]]:TRAC:X[:SCAL]:AUTO:SPAN:AUTO OFF|ON|0|1

DISP[:WIND[1]]:TRAC:X[:SCAL]:AUTO:SPAN:AUTO?
```

Example

OUTPUT @Osa;"disp:trac:x:auto:span:auto off"	! turn automeasure auto span off

Related Key

(Auto Measure Setup) Span

DISPlay[:WINDow[1]]:TRACe:Y[:SCALe]:AUTO:PDIVision

Specifies the final vertical scale after performing a DISPlay:WINDow:TRACe :ALL:SCALe:AUTO (Auto Measure) command.

Syntax

```
DISP[:WIND[1]]:TRAC:Y[:SCAL]:AUTO:PDIV
<numeric_value>[DB]

DISP[:WIND[1]]:TRAC:Y[:SCAL]:AUTO:PDIV?
```

Example

OUTPUT @Osa;"disp:trac:Y1:auto:pdiv 5"	! set vertical power scale to 5/division

Related Key

(Auto Measure Setup) Scale/div

DISPlay[:WINDow[1]]:TRACe:Y[:SCALe]:SPACing

Specifies the scaling of the vertical axis as logarithmic or linear. In LOG scale, the scale in dB per division is specified by the DISPlay[:WINDow[1]]:TRACe :Y[:SCALe]:PDIVision command.

Syntax

```
DISP[:WIND[1]]:TRAC:Y[:SCAL]:SPAC LIN|LOG

DISP[:WIND[1]]:TRAC:Y[:SCAL]:SPAC?
```

Example

```
OUTPUT @Osa;"disp:trac:y:scal:Lin on"       ! set y scale to linear
```

Related Key

Display Mode Log Linear

DISPlay[:WINDow[1]]:TRACe:Y[1|2][:SCALe]:AUTO:PDIVision:AUTO

Specifies whether the final vertical scale after a DISPlay:WINDow:TRACe :ALL:SCALe:AUTO (Auto Measure) command should be adjusted automatically, based on signal properties. Y1 refers to the left (power) scale, and Y2 refers to the right (ratio) scale.

Syntax

```
DISP[:WIND[1]]:TRAC:Y[1|2][:SCAL]:AUTO:PDIV:AUTO
OFF|ON|0|1

DISP[:WIND[1]]:TRAC:Y[1|2][:SCAL]:AUTO:PDIV:AUTO?
```

Example

OUTPUT @Osa;"disp:trac:Y1:auto:pdiv:auto off"	! turn automeasure auto vertical scaling off

Related Key

(Auto Measure Setup) Scale/div

DISPlay[:WINDow[1]]:TRACe:Y:SCALe:LINear

Specifies whether the vertical scale is in linear units or in log units. Y1 refers to the left (power) scale, and Y2 refers to the right (ratio) scale.

Syntax

```
DISP[:WIND[1]]:TRAC:Y:SCAL:LIN OFF|ON|0|1

DISP[:WIND[1]]:TRAC:Y:SCAL:LIN?
```

Example

```
OUTPUT @Osa;"disp:trac:y:scal:Lin on"      ! set y scale to linear
```

Related Key

(Amplitude Setup) Amp Display Mode Log | Lin

DISPlay[:WINDow[1]]:TRACe:Y[1 | 2][:SCALe]:PDIVision

Specifies the dB per division of the vertical scale. Y1 refers to the left (power) scale, and Y2 refers to the right (ratio) scale.

NOTE The maximum value for the vertical scale is 20 dB per division for the power scale or the ratio scale. The minimum value is 0.01 dB per division. The Preset value is 10 dB per division.

Syntax

```
DISP[:WIND[1]]:TRAC:Y[1 | 2][:SCAL]:PDIV<numeric_value>
[DB]

DISP[:WIND[1]]:TRAC:Y[1 | 2][:SCAL]:PDIV?
```

Example

OUTPUT @Osa;"disp:trac:Y1:auto:pdiv 5"	! set vertical power scale to 5 dB/division

Related Key

SCALE/DIV

DISPlay[:WINDow[1]]:TRACe:Y[1 | 2][:SCALe]:RLEVel

Specifies the power value of the reference level. Default units are set by the UNIT:POWer command for Y1 and the UNIT:RATio command for Y2. Y1 refers to the dBm (power) scale, and Y2 refers to the dB (ratio) scale.

NOTE The units sent must match the specified Y axis, for example, dBm for Y1 and dB for Y2.

NOTE

The maximum value for the power reference level is the equivalent of +300 dBm. The minimum value is –300 dBm. The Preset value for the power reference level is
instrument dependent.

The maximum value for the ratio reference level is 270 dB. The minimum value is –330 dB. The Preset value for the ratio reference level is 0 dB.

Syntax

```
DISP[:WIND[1]]:TRAC:Y[1|2][:SCAL]:RLEV<numeric_value>
[DB]

DISP[:WIND[1]]:TRAC:Y[1|2][:SCAL]:RLEV?
```

Example

```
OUTPUT @Osa;"disp:trac:y1:rpos 9"     ! set reference level to display 9th graticule line
```

Related Key

REFERENCE LEVEL

DISPlay[:WINDow[1]]:TRACe:Y[1|2][:SCALe]:RPOSition

Selects the position at which the reference level is displayed. The top and bottom graticule lines correspond to 10 and 0, respectively. The default is 9. Y1 refers to the dBm (power) scale, and Y2 refers to the dB scale.

Syntax

```
DISP[:WIND[1]]:TRAC:Y[1|2][:SCAL]:RPOS<numeric value>

DISP[:WIND[1]]:TRAC:Y[1|2][:SCAL]:RPOS?
```

Example

```
OUTPUT @Osa;"disp:trac:y1:rpos 9"     ! set reference level to display 9th graticule line
```

Related Key

(AMPLITUDE SETUP) REFERENCE LEVEL POSITION

FORMat Subsystem Commands

The FORMat subsystem sets a data format for transferring numeric and array information

The commands in this subsystem have the following command hierarchy:

FORMat

[:DATA] REAL[32|64]|ASCii

FORMat[:DATA]

Specifies the trace data format used during data transfer via GPIB.

This command affects data transfers for the :TRACe[:DATA] subsystem. The ASCII format is a comma-separated list of numbers. The REAL format is a definite-length block of 32- or 64-bit floating-point binary numbers. The definite-length block is defined by IEEE 488.2: a "#" character, followed by one digit (in ASCII) specifying the number of length bytes to follow, followed by the length (in ASCII), followed by length bytes of binary data. The binary data is a sequence of 8-byte (32- or 64-bit) floating point numbers.

Syntax

```
FORM[:DATA] REAL[32|64]|ASC

FORM[:DATA]?
```

HCOPy Subsystem Commands

The HCOPy subsystem controls the setup of printing to an external device.

The commands in this subsystem have the following command hierarchy:

HCOPy

:DATA?

:DESTination
"SYSTem:COMMunicate:CENTronics" | "SYSTem:COMMunicate:FSHare[1|2|3|4]" |
"SYSTem:COMMunicate:INTernal"

:DEVice

:LANGuage <PCL|CGM>

:PSHare[1|2|3|4]

:ADDRess

[:PATH]<param>

:IMMediate

HCOPy:DATA?

Returns the currently defined printer output as an indefinite length block. After removing the #0 prefix and new line suffix, this block can be saved by the controller and sent directly to a suitable printer.

Syntax

```
HCOP:DATA?
```

HCOPy:DESTination "SYSTem:COMMunicate:INTernal" |"SYSTem :COMMunicate:CENTronics" | "SYSTem:COMMunicate:FSHare[1|2|3|4]"

Selects the I/O port for hard copy output. This affects subsequent presses of the PRINT key and the HCOPy[:IMMediate] command.

Syntax

```
HCOP:DEST "SYST:COMM:INT"|"SYST:COMM:CENT"|"SYST

:COMM:FSH[1|2|3|4]"

HCOP:DEST?
```

Example

OUTPUT @Osa;"hcop:dest 'syst:comm"	! set internal printer for next hcop:imm
OUTPUT @Osa;"hcop:dev:mode ALL"	! set printing to both table and graph
OUTPUT @Osa;"hcop:imm"	! print

Related Key

(PRINTER SETUP) PRINTER LOCATION

HCOPy:DEVice:LANGuage

Sets the plot print format to either PCL (Printer Command Language) or CGM (Computer Graphics Metafile), or GIF (Graphics Interchange Format) mode.

The query identifies whether the plot print format is in PCL or CGM mode.

Syntax

HCOP:DEV:LANG <PCL|CGM|GIF>

HCOP:DEV:LANG?

Related Key

(SAVE MENU) GRAPHIC FORMAT

HCOPy:DEVice:PSHare[1|2|3|4]:ADDRess <param>

Sets the network printer's IP address. The IP is specified in dot IP notation (for example, 192.162.1.1). This is an optional parameter. If an IP address is specified, no name to IP lookup is done.

NOTE If the networking is not configured, the command will generate a "Settings conflict" error.

Syntax

HCOP:DEV:PSH[1|2|3|4]:ADDR <param>

HCOP:DEV:PSH[1|2|3|4]:ADDR?

Example

OUTPUT @Osa;"hcop:dev:psh1:addr '111.111.111.111'"	! set printer IP number
OUTPUT @Osa;"hcop:dev:psh1:path'\\systemName\shareName'"	! set printer share path
OUTPUT @Osa;"hcop:dest 'syst:comm:psh1"	! set printer share one current
OUTPUT @Osa;"hcop:imm"	! initiate print to printer share one

Related Commands

HCOPy:DEVice:PSHare[1|2|3|4]:PATH

Related Key

(Printer Shares) Opt. IP Address

HCOPy:DEVice:PSHare[1|2|3|4][:PATH]

Sets the network printer's path. The path is specified as "\\printserver\printer_name".

NOTE If the networking is not configured, the command will generate a "Settings conflict" error.

Syntax

HCOP:DEV:PSH[1|2|3|4][:PATH] <param>

HCOP:DEV:PSH[1|2|3|4][:PATH]?

Example

OUTPUT @Osa;"hcop:dev:psh1:addr '111.111.111.111'"	! set printer IP number
OUTPUT @Osa;"hcop:dev:psh1:path'\\systemName\shareName'"	! set printer share path
OUTPUT @Osa;"hcop:dest 'syst:comm:psh1"	! set printer share one current
OUTPUT @Osa;"hcop:imm"	! initiate print to printer share one

Related Commands

HCOPy:DEVice:PSHare[1|2|3|4]:ADDREss <param>

Related Key

(Printer Shares) Share Path

HCOPy:IMMediate

Prints out the test results to the port defined by the HCOPy:DESTination command. The data printed is affected by the HCOPY:DEVice:MODE command.

Syntax

HCOP:IMM

Related Commands

HCOPy:DEVice:MODE

Related Key

Print

INITiate Subsystem Commands

The commands in this subsystem have the following command hierarchy:

INITiate

:CONTinuous OFF|ON|0|1

:IMMediate]

INITiate:CONTinuous

Specifies repeat or single sweep.

The query returns the state of continuous sweep setting.

Syntax

```
INIT:CONT OFF|ON|0|1

INIT:CONT?
```

Example

OUTPUT @Osa;"init:cont off"	! turn repeat sweep mode off
OUTPUT @Osa;"init:imm"	! initiate a single sweep

Related Key

REPEAT SWEEP ON|OFF

INITiate[:IMMediate]

Disables the monochromator output and photodetector input, takes a sweep with the current optical spectrum analyzer settings and displays a trace, and then enables the monochromator output and photodetector input.

Syntax

```
INIT[:IMM]
```

Example

OUTPUT @Osa;"init:cont off"	! turn repeat sweep mode off
OUTPUT @Osa;"init:imm"	! initiate a single sweep

Related Commands

INITiate:CONTinuous OFF|ON|0|1

Related Key

SINGLE SWEEP

INPut Subsystem Commands

For Agilent 86144B/86146B filter mode only

The commands in this subsystem are to set the filter mode to the specified peak or pit. The command hierarchy is as follows:

INPut

```
:FILTer
    :MAXimum
        :LEFT
        :NEXT
            :RIGHt
    :MINimum
        :LEFT
        :NEXT
            :RIGHt
:   :SCENT
    :SRLevel
    :X
    :Y?
```

INPut:FILTer:MAXimum

For Agilent 86144B/86146B filter mode only

Sets the filter marker to the specified peak.

NOTE If the filter mode is not selected, the command will generate a "Settings Conflict" error.

Syntax

```
INP:FILT:MAX
```

Example

OUTPUT @Osa;"inst:nsel 1"	! turn on filter mode
OUTPUT @Osa;"init:imm;*opc?"	! sweep once
ENTER @Osa;Temp"	
OUTPUT @Osa;"inp:filt:max"	! set filter marker highest peak

Related Commands

INPut:FILTer:MAXimum:LEFT
INPut:FILTer:MAXimum:NEXT
INPut:FILTer:MAXimum:RIGHt

INPut:FILTer:MAXimum:LEFT

For Agilent 86144B/86146B filter mode only

Sets the filter marker to the left of the specified peak.

NOTE If the filter mode is not selected, the command will generate a "Settings Conflict" error.

Syntax

INP:FILT:MAX:LEFT

Example

OUTPUT @Osa;"inst:nsel 1"	! turn on filter mode
OUTPUT @Osa;"init:imm;*opc?"	! sweep once
ENTER @Osa;Temp"	
OUTPUT @Osa;"inp:filt:max"	! set filter marker highest peak
OUTPUT @Osa;"inp:filt:max:left"	! set filter marker to next left peak

Related Commands

INPut:FILTer:MAXimum
INPut:FILTer:MAXimum:NEXT
INPut:FILTer:MAXimum:RIGHt

INPut:FILTer:MAXimum:NEXT

For Agilent 86144B/86146B filter mode only

Sets the filter marker to the next peak to the specified peak.

NOTE If the filter mode is not selected, the command will generate a "Settings Conflict" error.

Syntax

```
INP:FILT:MAX:NEXT
```

Example

OUTPUT @Osa;"inst:nsel 1"	! turn on filter mode
OUTPUT @Osa;"init:imm;*opc?"	! sweep once
ENTER @Osa;Temp"	
OUTPUT @Osa;"inp:filt:max"	! set filter marker highest peak
OUTPUT @Osa;"inp:filt:max:next"	! set filter marker to next highest peak

Related Commands

INPut:FILTer:MAXimum
INPut:FILTer:MAXimum:LEFT
INPut:FILTer:MAXimum:RIGHt

INPut:FILTer:MAXimum:RIGHt

For Agilent 86144B/86146B filter mode only

Sets the filter marker to the right of the specified peak.

NOTE If the filter mode is not selected, the command will generate a "Settings Conflict" error.

Syntax

```
INP:FILT:MAX:LEFT
```

Example

OUTPUT @Osa;"inst:nsel 1"	! turn on filter mode
OUTPUT @Osa;"init:imm;*opc?"	! sweep once
ENTER @Osa;Temp"	
OUTPUT @Osa;"inp:filt:max"	! set filter marker highest peak
OUTPUT @Osa;"inp:filt:max:right"	! set filter marker to next right peak

Related Commands

INPut:FILTer:MAXimum
INPut:FILTer:MAXimum:RIGHt
INPut:FILTer:MAXimum:NEXT

INPut:FILTer:MINimum

For Agilent 86144B/86146B filter mode only

Sets the filter marker to the lowest pit.

NOTE If the filter mode is not selected, the command will generate a "Settings Conflict" error.

Syntax

```
INP:FILT:MIN
```

Example

OUTPUT @Osa;"inst:nsel 1"	! turn on filter mode
OUTPUT @Osa;"init:imm;*opc?"	! sweep once
ENTER @Osa;Temp"	
OUTPUT @Osa;"inp:filt:min"	! set filter marker to lowest pit

Related Commands

INPut:FILTer:MINimum:LEFT
INPut:FILTer:MINimum:NEXT
INPut:FILTer:MINimum:RIGHt

INPut:FILTer:MINimum:LEFT

For Agilent 86144B/86146B filter mode only

Sets the filter marker to the next pit to the left.

NOTE If the filter mode is not selected, the command will generate a "Settings Conflict" error.

Syntax

```
INP:FILT:MIN:LEFT
```

Example

OUTPUT @Osa;"inst:nsel 1"	! turn on filter mode
OUTPUT @Osa;"init:imm;*opc?"	! sweep once
ENTER @Osa;Temp"	
OUTPUT @Osa;"inp:filt:min"	! set filter marker to lowest pit
OUTPUT @Osa;"inp:filt:min:left"	! set filter marker to next left pit

Related Commands

INPut:FILTer:MINimum
INPut:FILTer:MINimum:NEXT
INPut:FILTer:MINimum:RIGHt

INPut:FILTer:MINimum:NEXT

For Agilent 86144B/86146B filter mode only

Sets the filter marker to the next lowest pit.

NOTE If the filter mode is not selected, the command will generate a "Settings Conflict" error.

Syntax

```
INP:FILT:MIN:NEXT
```

Example

OUTPUT @Osa;"inst:nsel 1"	! turn on filter mode
OUTPUT @Osa;"init:imm;*opc?"	! sweep once
ENTER @Osa;Temp"	
OUTPUT @Osa;"inp:filt:min"	! set filter marker to lowest pit
OUTPUT @Osa;"inp:filt:min:next"	! set filter marker to next lowest pit

Related Commands

INPut:FILTer:MINimum
INPut:FILTer:MINimum:LEFT
INPut:FILTer:MINimum:RIGHt

INPut:FILTer:MINimum:RiGHt

For Agilent 86144B/86146B filter mode only

Sets the filter marker to the next pit to the right.

NOTE If the filter mode is not selected, the command will generate a "Settings Conflict" error.

Syntax

INP:FILT:MIN:RIGHT

Example

OUTPUT @Osa;"inst:nsel 1"	! turn on filter mode
OUTPUT @Osa;"init:imm;*opc?"	! sweep once
ENTER @Osa;Temp"	
OUTPUT @Osa;"inp:filt:min"	! set filter marker to lowest pit
OUTPUT @Osa;"inp:filt:min:right"	! set filter marker to next right pit

Related Commands

INPut:FILTer:MINimum
INPut:FILTer:MINimum:LEFT
INPut:FILTer:MINimum:NEXT

INPut:FILTer:SCENter

For Agilent 86144B/86146B filter mode only

Sets the center wavelength to the filter marker's wavelength.

NOTE If the filter mode is not selected, the command will generate a "Settings Conflict" error.

Syntax

```
INP:FILT:SCEN
```

Example

OUTPUT @Osa;"inst:nsel 1"	! turn on filter mode
OUTPUT @Osa;"init:imm;*opc?"	! sweep once
ENTER @Osa;Temp"	
OUTPUT @Osa;"inp:filt:max"	! set filter marker highest peak
OUTPUT @Osa;"inp:filt:scen"	! set center to filter marker wavelength

Related Commands

INPut:FILTer:SRLevel
INPut:FILTer:X
INPut:FILTer:Y?

INPut:FILTer:SRLevel

For Agilent 86144B/86146B filter mode only

Sets the reference level to the amplitude of the filter marker.

NOTE If the filter mode is not selected, the command will generate a "Settings Conflict" error.

Syntax

```
INP:FILT:SRL
```

```
INP:FILT:SRL?
```

Example

OUTPUT @Osa;"inst:nsel 1"	! turn on filter mode
OUTPUT @Osa;"init:imm;*opc?"	! sweep once
ENTER @Osa;Temp"	
OUTPUT @Osa;"inp:filt:max"	! set filter marker highest peak
OUTPUT @Osa;"inp:filt:srl"	! set reference level to amplitude of filter marker

Related Commands

INPut:FILTer:SCENter
INPut:FILTer:X
INPut:FILTer:Y?

INPut:FILTer:X

For Agilent 86144B/86146B filter mode only

Sets the x-axis value of the filter marker. The x value of the filter marker corresponds to the grating position.

NOTE If the filter mode is not selected, the command will generate a "Settings Conflict" error.

Syntax

INP:FILT:X<numeric_value>M|UM|NM|PM|A]

INP:FILT:X?

Example

OUTPUT @Osa;"inst:nsel 1"	! turn on filter mode
OUTPUT @Osa;"init:imm;*opc?"	! sweep once
ENTER @Osa;Temp"	
OUTPUT @Osa;"inp:filt:x 1315nm"	! set filter marker to 1315 nm

Related Commands

INPut:FILTer:SCENter
INPut:FILTer:SRLevel
INPut:FILTer:X
INPut:FILTer:Y?

INPut:FILTer:Y?

For Agilent 86144B/86146B filter mode only

Queries the y-axis value of the filter marker.

NOTE If the filter mode is not selected, the command will generate a "Settings Conflict" error.

Syntax

```
INP:FILT:Y?
```

Example

OUTPUT @Osa;"inst:nsel 1"	! turn on filter mode
OUTPUT @Osa;"init:imm;*opc?"	! sweep once
ENTER @Osa;Temp"	
OUTPUT @Osa;"inp:filt:max"	! set filter marker highest peak
OUTPUT @Osa;"inp:filt:y?"	! get amplitude at filter marker
ENTER @Osa;Maxamp	
PRINT Maxamp	

Related Commands

INPut:FILTer:SCENter
INPut:FILTer:SRLevel
INPut:FILTer:X

INSTrument Subsystem Commands

The commands in this subsystem have the following command hierarchy:

INSTrument

:CATalog?

:FULL?

:NSELect <numeric value>

:SELect

INSTrument:CATalog?

Returns a comma-separated list of strings denoting the applications and instrument modes supported by the instrument.

Syntax

```
INST:CAT?
{OSA,Filter,PowerMeter,PassiveComponent,
WDM_Autoscan,Amp_ISS_Test}
```

Example

DIM C$[128]	
OUTPUT @Osa;"inst:cat?"	! query instrument for installed applications and measurement modes
ENTER @Osa;C$	! store comma separated string
PRINT C$	! print

INSTrument:CATalog:FULL?

Returns a comma-separated list of string-numeric pairs denoting the instrument modes and applications supported by the instrument.

Syntax

```
INST:CAT:FULL?
{OSA,0,Filter,1,PowerMeter,2,PassiveComponent,3,
WDM_Autoscan,4,Amp_ISS_Test,5}
```

Example

```
DIM C$[128]
OUTPUT @Osa;"inst:cat:full?"
ENTER @Osa;C$
PRINT C$
```

INSTrument:NSELect

Selects the instrument mode or application using a numeric value. When a new instrument mode is selected, the behavior of the base firmware may change. When an application is selected, the application is loaded.

The numeric value for each application can be listed using the INSTrument :CATalog:FULL? command.

Syntax

```
INST:NSEL <numeric_value>
```

Example

```
OUTPUT @Osa;"inst:nsel 5"   ! switch to application, amplifier iss in this case
```

INSTrument:SELect

Selects the instrument mode or application. When a new instrument mode is selected, the behavior of the base firmware may change. When an application is selected, the application is loaded.

The identifier for each application can be listed using the INSTrument :CATalog or INSTrument:CATalog:FULL? commands.

NOTE A "Settings Conflict" error will occur is the instrument is in zero span.

Syntax

```
INST:SEL <identifier>
```

Example

```
OUTPUT @Osa;"inst:sel 'PassiveComponent'"    ! start passive component application
```

MEMory Subsystem Commands

The purpose of the MEMory subsystem is to manage instrument memory and specifically excludes memory used for mass storage, which is defined in the MMEMory subsystem.

The commands in this subsystem have the following command hierarchy:

MEMory

:STATe[:EXTended]?

MEMory:STATe[:EXTended]?

Returns the extended state information as an indefinite length block.

Syntax

```
MEM:STAT[:EXT]?
```

MMEMory Subsystem Commands

The MMEMory subsystem provides mass storage capabilities for the instrument.

The commands in this subsystem have the following command hierarchy:

MMEMory

:CATalog? FSHare[1|2|3|4]|INTernal|FLOPpy

:DATA <file_name>, <data_block>

:DELete <file_name> FSHare[1|2|3|4]|INTernal|FLOPpy

:FSHAre[1|2|3|4]

:ADDRess

[:PATH]

:INITialize [:INTernal|FLOPpy]

:LOAD:TRACe <trace_name>, <file_name> FSHare[1|2|3|4]|INTernal|FLOPpy

:STORe:TRACe <trace_name>, <file_name> FSHare[1|2|3|4]|INTernal|FLOPpy

MMEMory:CATalog?

This query returns information on the current contents and state of the mass storage media.

Lists all files in the current directory. The return data will be formatted as:

<mem_used>,<mem_free> {,<file listing>}

Each <file listing> indicates the name, type, and size of one file in the directory list:

<file_name>,<file_type>,<file_size>

Syntax

```
MMEM:CAT? [INT|FLOP]
```

Example

DIM S$[128]	! make buffer large enough to hold all the data
OUTPUT @Osa;"mmem:cat? int"	! query files stored on internal hard disk
ENTER @Osa;S$	
PRINT S$	

MMEMory:DATA

Stores <data_block> in the memory location <file_name>.

The query response is the <data_block> stored in <file_name>, where <data_block> is an indefinite block.

Syntax

```
MMEM:DATA <file_name>,<data_block>

MMEM:DATA? <file_name>
```

Example

MMEMory:DATA 'UPPER.LIMit', #0 <limit file contents>

MMEMory:DELete

Deletes the specified file. The file name extension must be specified as shown in the examples.

The DELete command removes a file from the specified mass storage device. The <file_name> parameter specifies the file_name to be removed.

Syntax

```
MMEM:DEL <file_name>, INT|FLOP
```

Example

For Traces - MMEMory:DELete "CBG.csv", FLOPpy
For State (*SAVe) - MMEMory:DELete "CBG.dat", FLOPpy

Related Key

DELETE MENU

MMEMory:FSHAre[1|2|3|4]:ADDRess

Sets the network file share's IP address for the specified file share. The IP is specified in dot IP notation (for example, 192.162.1.1). This is an optional parameter. If an IP address is specified, no name to IP lookup is done.

NOTE If networking is not configured, the command will generate a "Settings conflict" error.

Syntax

MMEM:FSH[1|2|3|4][:ADDR] <param>

Related Commands

MMEMory:FSHAre [1|2|3|4][:PATH] <param>

MMEMory:FSHAre [1|2|3|4][:PATH]

Set the network file share's path. The path is specified as "\\fileserver\directory". The directory can only be one level deep (for example, \\server\stateFiles).

NOTE If networking is not configured then the command will generate a "settings conflict" error.

Syntax

MMEM:FSHA [1|2|3|4][:PATH] <param>

MMEM:FSHA [1|2|3|4][:PATH]?

Related Commands

MMEMory:FSHAre [1|2|3|4][:ADDRess] <param>

MMEMory:INITialize

Formats a disk in the instrument's 3.5 inch disk drive.

Syntax

MMEM:INIT [FLOP]

MMEMory:LOAD:TRACe

Loads the specified trace from mass storage. Refer to *RCL for measurement recall.

Syntax

MMEM:LOAD:TRA TRA|TRB|TRC|TRD|TRE|TRF,
<file_name>[INT|FLOP|FSH1|FSH2|FSH3|FSH4]

Example

OUTPUT @Osa;"mmem:stor:trac TRA,'traceA',INT"	! store trace A as a file called 'TraceA in !INTernal memory
OUTPUT @Osa;"mmem:load:trac TRB,'traceA',INT"	! recall 'traceA; into Trace B from INTernal !memory

Related Commands

MMEMory:DELete? <file name> [INTernal|FLOPpy]
MMEMory:STORe <file name>[INTernal|FLOPpy]
:*RCL

MMEMory:STORe:TRACe

Stores a specified trace to mass storage. Refer to *SAV to save measurement data.

Syntax

MMEM:STOR:TRAC TRA|TRB|TRC|TRD|TRE|TRF,
<filename> [INT|FLOP|FSH1|FSH2|FSH3|FSH4]

Example

OUTPUT @Osa;"mmem:stor:trac TRA,'traceA',INT"	! store trace A as a file called 'TraceA in !INTernal memory
OUTPUT @Osa;"mmem:load:trac TRB,'traceA',INT"	! recall 'traceA; into Trace B from INTernal !memory

Related Commands

MMEMory:LOAD:TRACe TRA|TRB|TRC|TRD|TRE|TRF,
<file_name> [INTernal|FLOPpy]
:*SAV

ROUTe Subsystem Commands

The Route subsystem switches either the 9 μm external path or the 50 μm internal path.

ROUTe

:PATH <INTernal | EXTernal>

ROUTe:PATH <INTernal | EXTernal>

For Agilent 86144B/86146B only

Selects between the 50 μm internal path or the 9 μm external path.

Syntax

ROUT:PATH <INT | EXT>

Example

```
OUTPUT @OSA;"ROUT:PATH EXT"    ! SWITCH TO EXTERNAL PATH
```

SENSe Subsystem Commands

The SENSe subsystem deals with controls that directly affect device-specific settings and not those related to the signal-oriented characteristics. The commands in this subsystem have the following command hierarchy:

SENSe

```
:BANDwidth|BWIDth
    [:RESolution]
        :AUTO OFF|ON|0|1
        :RATio <numeric_value>
    :VIDeo <numeric_value> [HZ|KHZ|MHZ|GHZ]
        :AUTO OFF|ON|0|1
:CHOP[:STATe] OFF|ON|0|1
:CORRection
    :CSET[:SELect] 1|2|3|4
    :DATA wvl,ratio{wvl,ratio...}
    :RVELocity:MEDium
    :STATe OFF|ON|0|1
    :X:SPACing LOG|LIN
:GORDer[:AUTO] OFF|ON|0|1
:POWer[:DC]:RANGe
    :AUTO OFF|ON|0|1
    :LOCK OFF|ON|0|1
    :LOWer <numeric_value> [DBM|W|MW|UW|NW]
        :AUTO OFF|ON|0|1
:SWEep
    :POINts <numeric_value>
    :TIME
        :AUTO OFF|ON|0|1
[:WAVelength]
    :CENTer <numeric_value> [M|UM|NM|PM|A|HZ|KHZ|MHZ|GHZ|THZ]
        :STEP
            :AUTO OFF|ON|0|1
            [:INCRement] <numeric_value> [M|UM|NM|A]
    :LIMit OFF|ON|0|1
    :OFFSet <numeric_value> [M|UM|NM|A]
    :SPAN <numeric_value> [M|UM|NM|A]
        :FULL
    :SRANge
        :LOWer <numeric_value>
[M|UM|NM|PM|A|HZ|KHZ|MHZ|GHZ|THZ]
        [:STATe] OFF|ON|0|1
        :UPPer <numeric_value>
[M|UM|NM|PM|A|HZ|KHZ|MHZ|GHZ|THZ]
    :STARt <numeric_value> [M|UM|NM|PM|A|HZ|KHZ|MHZ|GHZ|THZ]
    :STOP <numeric_value> [M|UM|NM|PM|A|HZ|KHZ|MHZ|GHZ|THZ]
```

[SENSe]:BANDwidth | BWIDth[:RESolution]

Sets the resolution bandwidth.

Syntax

```
[SENS]:BAND | BWID[:RES] <param>[M | UM | NM | PM | A]
```

Example

```
OUTPUT @Osa;"band 2nm" ! set resolution bandwidth to 2 nm
```

Related Key

RES BW MAN

[SENSe]:BANDwidth | BWIDth[:RESolution]:AUTO

Couples the resolution bandwidth to the wavelength span, setting the it to *span resolution bandwidth ratio*.

Syntax

```
[SENS]:BAND | BWID[:RES]:AUTO OFF | ON | 0 | 1

[SENS]:BAND | BWID[:RES]:AUTO?
```

Example

OUTPUT @Osa;"band:auto on"	! switch to auto mode: RBW = (SPAN) X (RBW ratio)
OUTPUT @Osa;"band:rat .02"	! set the ratio used in Auto RWB calculation

Related Key

RES BW AUTO

[SENSe]:BANDwidth | BWIDth[:RESolution]:RATio

Specifies the ratio of the resolution bandwidth to the span. This parameter is multiplied by the span width to determine the automatic setting of the resolution bandwidth. The default ratio is .01.

Syntax

```
[SENS]:BAND|BWID[:RES]:RAT <numeric_value>

[SENS]:BAND|BWID[:RES]:RAT?
```

Example

OUTPUT @Osa;"band:auto on"	! switch to auto mode: RBW = (SPAN) X (RBW ratio)
OUTPUT @Osa;"band:rat .02"	! set the ratio used in Auto RWB calculation

[SENSe]:BANDwidth | BWIDth:VIDeo

Specifies the bandwidth of the post-detection video filter. The minimum value for the video bandwidth is 0.1 Hz. The maximum value is the lesser between 3 kHz and the bandwidth of the currently selected transimpedance amplifier.

Syntax

```
[SENS]:BAND|BWID:VID <numeric_value> [HZ|KHZ|MHZ|GHZ]

[SENS]:BAND|BWID:VID?
```

Example

```
OUTPUT @Osa;"band:vid 100Hz"      ! set video bandwidth to 1 Hertz
```

Related Commands

VIDEO BW MAN

[SENSe]:BANDwidth | BWIDth:VIDeo:AUTO

Enables and disables automatic coupling of the video bandwidth.

Video bandwidth filtering occurs after detection of the light. In the auto coupled mode, the video bandwidth has an extremely wide range. This allows the optical spectrum analyzer to avoid unnecessary filtering that would reduce the sweep speed more than required.

Normally, the video bandwidth is coupled to the sensitivity. Manually entering a video bandwidth breaks this coupling. The video bandwidth can be manually set from 100 MHz to 3 kHz, or the bandwidth of the currently selected transimpedance amplifier, whichever is less.

The following functions affect video bandwidth:

- Sensitivity
- Reference level
- Autoranging

The range of video bandwidth available in *Auto* mode is much greater than can be set manually from the front panel. A lower value of video bandwidth requires a longer sweep time. Because of the interdependence between the video bandwidth and sensitivity, it is recommended that either the sensitivity *or* the video bandwidth be changed, whichever is the most important to the measurement task being performed.

Because of the interdependence of sensitivity and video bandwidth, these parameters cannot be set individually. If one of the parameters is set manually, the other is forced into Auto coupled mode and set by the instrument. Set either the desired sensitivity or the desired video bandwidth, depending on which parameter is most important to the current measurement task.

NOTE The Preset state of the video bandwidth coupling is AUTO.

Syntax

```
[SENS]:BAND|BWID:VID:AUTO OFF|ON|0|1

[SENS]:BAND|BWID:VID:AUTO?
```

Example

```
OUTPUT @Osa;"band:vid:auto on"      ! set video bandwidth auto mode
```

Related Key

VIDEO BW AUTO

[SENSe]:CHOP[:STATe]

All models except Agilent 86144B/86146B.

Turns the spectrum analyzer chop mode on or off. Chop mode increases dynamic range for long sweep times by subtracting ambient light.

Syntax

```
[SENS]:CHOP[:STAT] OFF|ON|0|1

[SENS]:CHOP[:STAT]?
```

Example

```
OUTPUT @Osa;"chop on"      ! turn chop mode on
```

[SENSe]:CORRection:CSET[:SELect]

Selects the amplitude correction set table. Up to four correction sets can be created using the [SENSe]:CORRection:DATA command. Only one set may be selected at a time.

The query returns the number of the currently selected correction set.

Syntax

```
[SENS]:CORR:CSET[:SEL] 1|2|3|4

[SENS]:CORR:CSET[:SEL]?
```

Example

OUTPUT @Osa;"corr:cset 1"	! select amplitude correction set [1-4]
OUTPUT @Osa;"corr:data 1.31E-6,-2"	! make correction entry for 1310nm, -2db
OUTPUT @Osa;"corr:x:spac log"	! set spacing to logarithmic
OUTPUT @Osa;"corr:stat on"	! turn amplitude correction on

Related Commands

[SENSe]:CORRection:DATA

[SENSe]:CORRection:DATA

Allows you to create an amplitude correction set. Enter correction factors as comma-separated wavelength and amplitude pairs. Enter wavelength values in equal or increasing order to prevent an error condition. Up to 10001 wavelength and amplitude pairs can be entered. Wavelength values must be entered in meters (M) and amplitude values entered in decibels (dB).

Correction values are applied to the currently selected correction set. New values will replace the old values and there is no way to retrieve the old, replaced values. You should check that the desired correction set is selected by using [SENSe]:CORRection:CSET? Entered correction values will be written to memory and will take operation time to process.

The query returns the data of the currently selected correction set.

Syntax

[SENS]:CORR:DATA wvl,ratio{wvl,ratio...}

[SENS]:CORR:DATA?

Example

OUTPUT @Osa;"corr:cset 1"	! select amplitude correction set [1-4]
OUTPUT @Osa;"corr:data 1.31E-6,-2"	! make correction entry for 1310nm, -2db
OUTPUT @Osa;"corr:x:spac log"	! set spacing to logarithmic
OUTPUT @Osa;"corr:stat on"	! turn amplitude correction on

Related Commands

[SENSe]:CORRection:CSET?

[SENSe]:CORRection:RVELocity:MEDium

Wavelengths referenced in air or vacuum.

Syntax

[SENS]:CORR:RVEL:MED<AIR|VAC>

Example

```
OUTPUT @Osa;"corr:rvel:med VAC"       ! set wavelength values to vacuum
```

Related Key

(Wavelength Setup) Wavelengths Referenced In

[SENSe]:CORRection:STATe

Turns AMPCOR (amplitude correction) on or off.

When turned on, the correction set specified by the [SENSe]:CORRection:CSET command is enabled. When turned off, all the correction sets are disabled.

When AMPCOR is turned on, the correction points are applied across the active measurement range and added to all measurement results. Between points, the correction values are interpolated linearly or logarithmically. When measuring at wavelengths outside the first and last correction points, the first or last value (as appropriate) is used as the correction value.

Whenever AMPCOR is active, the currently selected correction set is displayed in the lower left corner of the screen. For example, if correction set number one is selected, "A1" is displayed. Refer to "Amplitude Correction Remote Commands" on page 32 for an overview of the amplitude correction remote commands.

The query returns the status of AMPCOR, 0 (zero) for off or 1 for on.

Syntax

[SENS]:CORR:STAT ON|OFF|0|1

[:SENS]:COR:STAT?

Example

OUTPUT @Osa;"corr:cset 1"	! select amplitude correction set [1-4]
OUTPUT @Osa;"corr:data 1.31E-6,-2"	! make correction entry for 1310nm, -2db
OUTPUT @Osa;"corr:x:spac log"	! set spacing to logarithmic
OUTPUT @Osa;"corr:stat on"	! turn amplitude correction on

Related Key

AMPLITUDE SETUP

[SENSe]:CORRection:X:SPACing

Specifies the interpolation method for amplitude corrections being applied to current data: logarithmic or linear.

Syntax

[SENS]:CORR:X:SPAC LOG|LIN

[SENS]:CORR:X:SPAC?

Example

OUTPUT @Osa;"corr:cset 1"	! select amplitude correction set [1-4]
OUTPUT @Osa;"corr:data 1.31E-6,-2"	! make correction entry for 1310nm, -2db
OUTPUT @Osa;"corr:x:spac log"	! set spacing to Linear
OUTPUT @Osa;"corr:stat on"	! turn amplitude correction on

[SENSe]:GORDer[:AUTO]

Specifies the optical spectrum analyzer grating order mode. When on, allows the instrument to select the best reflection order for the wavelength range. When off causes the instrument to use the first-order reflection, regardless of the wavelength.

Syntax

```
[SENS]:GORD[:AUTO] OFF|ON|0|1

[SENS]:GORD[:AUTO]?
```

Example

```
OUTPUT @Osa;"gord on"       ! turn grating order on
```

Related Key

GRATING ORDER AUTO

[SENSe]:POWer[:DC]:RANGe:AUTO

Turns the automatic ranging feature on or off and locks the transimpedance amplifier to the currently selected range. For improved dynamic range, automatic ranging changes the input range during the sweep

NOTE It is recommended this function be in AUTO mode.

Syntax

```
[SENS]:POW[:DC]:RANG:AUTO ON|OFF|0|

[SENS]:POW[:DC]:RANG?
```

Example

OUTPUT @Osa;"pow:rang:auto off"	! turn autoranging off
OUTPUT @Osa;"pow:rang:lock on"	! lock trans-z

Related Key

(AMPLITUDE SETUP) AUTO RANGING

[SENSe]:POWer[:DC]:RANGe:LOCK

Locks the transimpedance amplifiers to the lowest gain stages, for the fastest measurement speed.

Syntax

[SENS]:POW[:DC]:RANG:LOCK OFF|ON|0|1

[SENS]:POW[:DC]:RANG:LOCK?

Example

OUTPUT @Osa;"pow:rang:auto off"	! turn autoranging off
OUTPUT @Osa;"pow:rang:lock on"	! lock trans-z

Related Key

TRANSZ LOCK 2-3 ON OFF

[SENSe]:POWer[:DC]:RANGe:LOWer

Specifies the desired value for sensitivity. Default units are set by the UNIT:POWer command.

NOTE

The maximum value for Sensitivity is +300 dBm. The minimum value is the value that causes the sweep time to become 1000 seconds, and is an attribute of each individual optical spectrum analyzer. The minimum value will always be less than the values for sensitivity shown in the Specifications section of the User's Guide.

Syntax

```
[SENS]:POW[:DC]:RANG:LOW <param>

[SENS]:POW[:DC]:RANG:LOW?
```

Example

OUTPUT @Osa;"pow:rang:low:auto off"	! sensitivity set to manual model
OUTPUT @Osa;"pow:rang:low -33dbm"	! set sensitivity to -33 dbm

Related Key

SENSITIVITY AUTO MAN

[SENSe]:POWer[:DC]:RANGe:LOWer:AUTO

Turns the automatic setting of sensitivity on or off. Specifying a value for sensitivity with the [SENSe]:POWer[:DC]:RANGe:LOWer command will turn Auto off.

The sensitivity setting requests the lowest amplitude signal that can be measured relative to the power at "top of screen". It is defined as the signal that is six times the RMS noise. The range of settings is between –94 dB and 300 dB. An error will be reported for values outside of this range and the sensitivity will round to the nearest valid sensitivity.

Auto sensitivity uses one of six transimpedance amplifier stages. The stage is selected based on the amplitude value at the top of the display. At least 40 dB of dynamic range can be seen with one stage.

Manual sensitivity allows you to specify the lowest signal you want to measure.

```
[SENS]:POW[:DC]:RANe:LOW:AUTO ON|OFF|0|1

[SENS]:POW[:DC]:RANG:LOW?
```

Example

OUTPUT @Osa;"pow:rang:low:auto off"	! sensitivity set to manual model
OUTPUT @Osa;"pow:rang:low -33dbm"	! set sensitivity to -33 dbm

Related Key

SENSITIVITY AUTO MAN

[SENSe]:SWEep:POINts

Sets the number of the data points acquired during a sweep. This command is used in conjunction with the TRACe:POINts command when downloading a trace into the OSA. The minimum number of data points is three and the maximum is 10001.

Syntax

```
[SENS]:SWE:POIN <numeric_value>

[SENS]:SWE:POIN?
```

Example

OUTPUT @Osa;"swe:poin 500"	! set sweep points to 500

Related Key

SWEEP POINTS

[SENSe]:SWEep:TIME

Specifies the amount of time required for the instrument to sweep the current measurement range. The instrument automatically selects sweep times based on coupling of the following instrument settings:

- wavelength span
- resolution bandwidth
- video bandwidth
- sensitivity
- trace length
- power level

Coupling of these parameters yields optimum amplitude accuracy. When Sweep Time is set to Auto, the instrument always uses the fastest sweep possible while still maintaining the specified accuracy. Coupled, sweep times range from 56.3 mn to a maximum value that depends on the number of trace points used to draw the trace. This relationship is shown in the following equation:

$$56{,}3ms \leq \ sweep\ time\ \leq (1min)(\ trace\ points\ )$$

The default number of trace points is 1001, so the maximum sweep time is normally 100 seconds. When Sweep Time is in manual mode, the sweep time can be set from 56.3 ms to a maximum of 1000 seconds. If you change the number of trace points, the maximum sweep time changes as well.

NOTE

If the sweep time is set too fast, an over sweep message appears indicating the display is no longer calibrated and that trace data may not meet specifications. Increase the sweep time until the over sweep message disappears. If the sweep time is set too slow, measurement times may be excessively long.

Syntax

```
[SENS]:SWE:TIME <numeric_value>[US|MS|S]

[SENS]:SWE:TIME?
```

Example

OUTPUT @Osa;"swe:time:auto off"	! set sweep time to manual mode
OUTPUT @Osa;"swe:time 500ms"	! set sweep time to 500 ms

[SENSe]:SWEep:TIME:AUTO

When this function is on, the sweep time is determined by the trace length, span, and the sensitivity.

Syntax

```
[SENS]:SWE:TIME:AUTO OFF|ON|0|1

[SENS]:SWE:TIME:AUTO?
```

Example

OUTPUT @Osa;"swe:time:auto off"	! set sweep time to manual mode
OUTPUT @Osa;"swe:time 500ms"	! set sweep time to 500 ms

Related Key

SWEEP TIME AUTO MAN

[SENSe][:WAVelength]:CENTer

Specifies the center wavelength.

The start and stop wavelength and, if necessary, the span are adjusted so that:

$$Center = Start + \left(\frac{Span}{2}\right) \quad \text{and} \quad Center = Stop - \left(\frac{Span}{2}\right)$$

NOTE

With Wavelength Limit Off, the minimum value for the Center Wavelength is nominally 350.1 nm. The maximum value is 1999.9 nm. These limits are valid for wavelengths referenced in air or vacuum.

With Wavelength Limit On, the minimum value for the Center Wavelength is nominally 600.1 nm. The maximum value is 1699.9 nm. These limits are valid for wavelengths referenced in air or vacuum.

The Preset value for Wavelength Limit is On. The Preset value for Center Wavelength is 1150 nm.

Syntax

```
[SENS][:WAV]:CENT <numeric_value>
[M|UM|NM|PM|A|HZ|KHZ|MHZ|GHZ|THZ]

[SENS][:WAV:CENT?
```

Example

OUTPUT @Osa;"cent:step:auto off"	! turn step size to manual mode
OUTPUT @Osa;"centerwl up"	! set step to 10nm

Related Key

CENTER WL

[SENSe][:WAVelength]:CENTer:STEP:AUTO

When on, the step size is automatic. When off, the step size is fixed. The value is set by the [SENSe][:WAVelength]:CENTer:STEP[:INCRement] command.

Syntax

```
[SENS][:WAV]:CENT:STEP:AUTO OFF|ON|0|1

[SENS][:WAV]:CENT:STEP:AUTO?
```

[SENSe][:WAVelength]:CENTer:STEP[:INCRement]

Specifies the center wavelength step size.

Syntax

```
[SENS][:WAV]:CENT:STEP[:INCR]<numeric_value>
[M|UM|NM|A]
[SENS][:WAV]:CENT:STEP[:INCR]?
```

Related Key

(WAVELENGTH SETUP) WAVELENGTH STEP SIZE

[SENSe][:WAVelength]:LIMit

Specifies whether the span is limited to the specified range of 600 to 1700 nm. When off, the start wavelength can be set as low as 350 nm, and the stop wavelength as high as 2000 nm.

The performance of the optical spectrum analyzer is not specified and the amplitude is not calibrated outside the 600 to 1700 nm range.

Syntax

```
[SENS][:WAV]:LIM OFF|ON|0|1

[SENS][:WAV]:LIM?
```

Example

```
OUTPUT @Osa;"sens:wav:lim off"        ! remove start, stop limits
```

Related Key

WAVELENGTH LIMIT

[SENSe][:WAVelength]:OFFSet

Specifies the wavelength offset. This is the offset between the measured wavelength and the displayed wavelength. Also offsets wavelength values returned via remote commands.

Syntax

```
[SENS][:WAV]:OFFS <numeric_value>[M|UM|NM|A]

[SENS][:WAV]:OFFS?
```

Example

```
OUTPUT @Osa;"sens:wav:offs 1nm" ! set wavelength offset to 1 nm
```

Related Key

(WAVELENGTH SETUP) WAVELENGTH OFFSET

[SENSe][:WAVelength]:SPAN

Specifies the wavelength span.

The start and stop wavelength and, if necessary, the center wavelength are adjusted so that:

$Span = 2(Center - Start)$ and $Span = 2(Stop - Center)$

NOTE

The minimum value for Wavelength Span is 0.2 nm.

With Wavelength Limit Off, the maximum value for Wavelength Span is 1650 nm.

With Wavelength Limit On, the maximum value for Wavelength Span is 1100 nm.

The Preset value for Wavelength Limit is On. The Preset value for Wavelength Span is 1100 nm.

Syntax

```
[SENS][:WAV]:SPAN <numeric_value>[M|UM|NM|A]

[SENS][:WAV]:SPAN?
```

Example

OUTPUT @Osa;"span:full"	! set span to max
OUTPUT @Osa;"span 10nm"	! set span to 10 nm

Related Key

SPAN

[SENSe][:WAVelength]:SPAN:FULL

Sets the wavelength span of the spectrum analyzer to full span.

Syntax

```
[SENS][:WAV]:SPAN:FULL
```

Example

OUTPUT @Osa;"span:full"	! set span to max
OUTPUT @Osa;"span 10nm"	! set span to 10 nm

[SENSe][:WAVelength]:SRANge:LOWer

Sets the lower limit for the wavelength sweep range. Setting this value when [SENSe]:WAVelength:SRANge:STATe is off will automatically turn [SENSe]:WAVelength:SRANge:STATe on. The range used for the wavelength sweep range is the same range used for the total power integration, trace mean range, and marker search range. Changing the range with this command will change all four ranges.

Sending the command when the instrument is in a zero span will generate a "Settings conflict" error.

Default units for the parameter are meters. Frequency units are allowed.

Syntax

```
[SENS][:WAV]:SRAN:LOW <numeric_value>
[M|UM|NM|A|HZ|KHZ|MHZ|GHZ]

[SENS][:WAV]:SRAN:LOW?
```

Example

OUTPUT @Osa;"sran:low 1250nm"	! set lower range for sweep limit
OUTPUT @Osa;"sran:upp 1350nm	! set upper range for sweep limit
OUTPUT @Osa;"sran on"	! activate sweep range
OUTPUT @Osa;"init:cont on"	! sweep continuously over the specified range

[SENSe][:WAVelength]:SRANge[:STATe]

Turns the wavelength sweep range on or off. When the sweep range is on, the instrument will only sweep between the upper and lower sweep range limits. There is a single range controlling the total power integration, trace mean calculation, marker search range, and wavelength sweep range, but there are four independent state settings for limiting them to the range. If all four states are off, setting [SENSe]:WAVelength:SRANge:STATe to on will initialize
the lower limit to: $Start + \frac{Span}{3}$ and
the upper limit to: $Start + \left(2 \times \frac{Span}{3}\right)$

Sending the command when the instrument is in a zero span will generate a "Settings conflict" error.

Syntax

```
[SENS][:WAV]:SRAN[:STAT] OFF|ON|0|1

[SENS][:WAV]:SRAN[:STAT]?
```

Example

OUTPUT @Osa;"sran:low 1250nm"	! set lower range for sweep limit
OUTPUT @Osa;"sran:upp 1350nm	! set upper range for sweep limit
OUTPUT @Osa;"sran on"	! activate sweep range
OUTPUT @Osa;"init:cont on"	! sweep continuous over the specified range

[SENSe][:WAVelength]:SRANge:UPPer

Sets the upper limit for the wavelength sweep range. Setting this value when [SENSe]:WAVelength:SRANge:STATe is off will automatically turn [SENSe]:WAVelength:SRANge:STATe on. The range used for the wavelength sweep range is the same range used for the total power integration, trace mean range and marker search range. Changing the range with this command will change all four ranges.

Sending the command when the instrument is in a zero span will generate a “Settings conflict” error. Default units for the parameter are meters. Frequency units are allowed.

Syntax

```
[SENS][:WAV]:SRAN:UPP<numeric_value>
[M|UM|NM|A|HZ|KHZ|MHZ|GHZ]

[SENS][:WAV]:SRAN:UPP?
```

Example

OUTPUT @Osa;"sran:low 1250nm"	! set lower range for sweep limit
OUTPUT @Osa;"sran:upp 1350nm	! set upper range for sweep limit
OUTPUT @Osa;"sran on"	! activate sweep range
OUTPUT @Osa;"init:cont on"	! sweep continuous over the specified range

[SENSe][:WAVelength]:STARt

Specifies the start wavelength.

The center wavelength and span are adjusted so that:

$$Start = Center - \left(\frac{Span}{2}\right) \quad \text{and} \quad Stop = Center + \left(\frac{Span}{2}\right)$$

If the instrument is in zero span, this command sets the center wavelength to the value specified.

NOTE

With Wavelength Limit Off, the minimum value for the Start Wavelength is nominally 350 nm. The maximum value is 1999.8 nm. These limits are valid for wavelengths referenced in air or vacuum.

With Wavelength Limit On, the minimum value for the Start Wavelength is nominally 600 nm. The maximum value is 1699.8 nm. These limits are valid for wavelengths referenced in air or vacuum.

The Preset value for Wavelength Limit is On. The Preset value for Start Wavelength is 600 nm.

Syntax

```
[SENS]:WAV]:STAR <numeric_value>
[M|UM|NM|PM|A|HZ|KHZ|MHZ|GHZ|THZ]

[SENS][:WAV]:STAR?
```

Example

```
OUTPUT @Osa;"star 1300nm"
OUTPUT @Osa;"stop 1320nm"
```

Related Key

START WL

[SENSe][:WAVelength]:STOP

Specifies the stop wavelength.

The center wavelength and span are adjusted so that:

$$Start = Center - \left(\frac{Span}{2}\right) \quad \text{and} \quad Stop = Center + \left(\frac{Span}{2}\right)$$

If the instrument is in zero span, this command sets the center wavelength to the value specified.

NOTE

With Wavelength Limit Off, the minimum value for the Stop Wavelength is nominally 350.2 nm. The maximum value is 2000 nm. These limits are valid for wavelengths referenced in air or vacuum.

With Wavelength Limit On, the minimum value for the Stop Wavelength is nominally 600.2 nm. The maximum value is 1700 nm. These limits are valid for wavelengths referenced in air or vacuum.

The Preset value for Wavelength Limit is On. The Preset value for Stop Wavelength is 1700 nm.

Syntax

```
[SENS][:WAV]:STOP <numeric_value>
  [M|PM|NM|UM|A|HZ|KHZ|MHZ|GHZ|THZ]

[SENS][:WAV]:STOP?
```

Example

```
OUTPUT @Osa;"star 1300nm"
OUTPUT @Osa;"stop 1320nm"
```

Related Key

STOP WL

SOURce[n] Subsystem Commands

The commands in this subsystem have the following command hierarchy:

SOURce[n]

```
:CATalog?
:CURRent
    [:LEVel][:IMMediate][:AMPLitude] <param>
    :LIMit[:AMPLitude] <param>
    :PULSe:STATe OFF|ON|0|1
:PULSe
    :DCYCle <param>
    :WIDTh <param>
:STATe <identifier> OFF|ON|0|1
```

SOURce[n]:CATalog?

Returns a comma-separated list of sources in this instrument, such as: CALI, EELED 1550, EELED 1300, CURR, WHTLT.

An instrument can have five possible sources:

a rear panel current source (CURR), option 001

a broadband white light source (WHTLT), option 002,

a 1310 nm EELED (EELED 1300), included with a 1550 nm EELED in option 004

a 1550 nm EELED (EELED 1550), included with a 1310 nm EELED in option 004, or by itself in option 005,

and finally, an internal wavelength calibration source (CAL1), option 006

Syntax

```
SOUR[n]:CAT?
```

Example

```
DIM A$[128]
OUTPUT @Osa;"sour:cat?"
ENTER @Osa;A$
PRINT A$
```

SOURce[n]:CURRent[:LEVel][:IMMediate][:AMPLitude]

Specifies the source output level :IMMediate indicates that the subsequent specification of the new signal amplitude is to be processed by the device without waiting for further commands.

Syntax

```
SOUR[n]:CURR[:LEV][:IMM][:AMPL] <param>

SOUR[n]:CURR[:LEV][:IMM][:AMPL]?
```

SOURce[n]:CURRent:LIMit[:AMPLitude]

Sets the limit current of the current source output.

Syntax

```
SOUR[n]:CURR:LIM[:AMPL] <param>

SOUR[n]:CURR:LIM[:AMPL]?
```

SOURce[n]:CURRent:PULSe:STATe

Enables or disables the pulse mode for the current source.

Syntax

```
SOUR[n]:CURR:PULS:STAT OFF|ON|0|1

SOUR[n]:CURR:PULS:STAT?
```

SOURce[n]:PULSe:DCYCle

Sets the duty cycle of the current source output and the sync output.

Syntax

```
SOUR[n]:PULS:DCYC <param>

SOUR[n]:PULS:DCYCI?
```

SOURce[n]:PULSe:WIDTh

Sets the pulse width of the current source output and the sync output.

Syntax

```
SOUR[n]:PULS:WIDT <param>

SOUR[n]:PULS:WIDT?
```

SOURce[n]:STATe

Turns a specified source on or off.

This command is used in conjunction with the SOURce:CATalog? query to turn light sources on and off. For example, SOURce:STATe “WHTLT”, ON

Syntax

```
SOUR[n]:STAT <identifier>, OFF|ON|0|1

SOUR[n]:STAT? <identifier>
```

Example

OUTPUT @Osa;"sour:stat 'EELED1550', off"	! set lower range for sweep limit
OUTPUT @Osa;"init:imm;*opc?"	
ENTER @Osa;Temp	

STATus Subsystem Commands

This subsystem controls the SCPI-defined status-reporting structures. These structures provide registers that you can use to determine if certain events have occurred.

The commands in this subsystem have the following command hierarchy:

STATus

```
:OPERation
    :CONDition?
    ENABle
    [:EVENt]?
    :NTRansition <int_value>
    :PTRansition <int_value>
:PRESet
:QUEStionable
    :CONDition?
    :ENABle <int_value>
    [:EVENt]?
```

STATus:OPERation:CONDition?

Queries the contents of the operation condition register.

The query returns a value from 0 to 32767.

Syntax

```
STAT:OPER:COND?
```

Example

```
OUTPUT @OSA;"STAT:OPER:COND?"
```

STATus:OPERation:ENABle

Sets the contents of the operation enable register.

The enable mask selects which conditions in the event register cause the summary bit in the status byte to be set. If a bit in the enable mask is set true and the corresponding event occurs, the summary bit (bit 7 for the operation status) in the status byte will be set.

When queried, the largest value that can be returned is 65535. This is because the most-significant register bit cannot be set true.

Syntax

```
STAT:OPER:ENAB <int_value>

STAT:OPER:ENAB?
```

Example

```
OUTPUT @OSA;"STAT:OPER:ENAB 1024"
```

STATus:OPERation[:EVENt]?

Queries the contents of the operation event register. This query reads the contents of the register and then clears it. The response will be a number from 0 to 32767 indicating which bits are set. Reading the register clears the register.

Syntax

```
STAT:OPER[:EVEN]?
```

Example

```
OUTPUT @OSA;"STAT:OPER:EVEN?"
```

STATus:OPERation:NTRansition

Selects bits in the event register which can be set by negative transitions of the corresponding bits in the condition register.

Changes in the state of a condition register bit causes the associated Operation Status or Questionable Status register bit to be set. This command allows you to select a negative bit transition to trigger an event to be recognized. A negative transition is defined to occur whenever the selected bit changes states from a 1 to a 0. You can enter any value from 0 to 65535.

When queried, the largest value that can be returned is 32767. This is because the most-significant register bit cannot be set true.

Syntax

```
STAT:OPER:NTR <int_value>

STAT:OPER:NTR?
```

Example

```
OUTPUT @OSA;"STAT:OPER:NTR"
```

STATus:OPERation:PTRansition

Selects bits in the event register which can be set by positive transitions of the corresponding bits in the condition register.

Changes in the state of a condition register bit causes the associated Operation Status or Questionable Status event register bit to be set. This command allows you to select a positive bit transition to trigger an event to be recognized. A positive transition is defined to occur whenever the selected bit changes states from a 0 to a 1. You can enter any value from 0 to 65535.

When queried, the largest value that can be returned is 32767. This is because the most-significant register bit cannot be set true.

Syntax

```
STAT:OPER:PTR <int_value>

STAT:OPER:PTR?
```

Example

```
OUTPUT @OSA;"STAT:OPER:PTR"
```

STATus:PRESet

Clears the event registers and sets all bits in the enable registers.

The STATus:PRESet command is defined by SCPI to affect the enable register. If you want to clear all event registers and queues, use the *CLS command.

Table 10 Preset Values

Status Node	Preset Value
Operation enable register	0
Questionable enable register	0
PTRansition filters	32767
NTRansition filters	0

Syntax

```
STAT:PRES
```

Example

```
OUTPUT @OSA;"STAT:PRES"
```

Related Commands

*CLS

STATus:QUEStionable:CONDition?

Queries the contents of the questionable condition register.

Use this command to read the value of the Questionable Status register.

The query returns a value from 0 to 32767.

Syntax

```
STAT:QUES:COND?
```

Example

```
OUTPUT @OSA;"STAT:QUES:COND?"
```

STATus:QUEStionable:ENABle

Sets or queries the contents of the questionable enable register.

The enable mask selects which conditions in the event register cause the summary bit in the status byte to be set. If a bit in the enable mask is set true and the corresponding event occurs, the summary bit (bit 3 for the questionable status) in the status byte will be set.

When queried, the largest value that can be returned is 65535. This is because the most-significant register bit cannot be set true.

Syntax

```
STAT:QUES:ENAB <int_value>

STAT:QUES:ENAB?
```

Example

```
OUTPUT @OSA;"STAT:QUES:ENAB"
```

STATus:QUEStionable[:EVENt]?

Queries the contents of the questionable event register and then clears it.

The response will be a number from 0 to 32767 indicating which bits are set. Reading the register clears the register.

Syntax

```
STAT:QUES:EVEN?
```

Example

```
OUTPUT @OSA;"STAT:QUES:EVEN?"
```

SYSTem Subsystem Commands

The SYStem subsystem collects the functions that are not related to instrument performance. Examples include functions for performing general housekeeping and functions related to setting global configurations, such as TIME or DATE.

The commands in this subsystem have the following command hierarchy:

SYSTem

```
:COMMunicate
    :GPIB[:SELF]:BUFFer OFF|ON|0|1
    :NETWork
        :PASSword
        :USERname
        :WORKgroup <oaran>
:DATE?
:ERRor:[:NEXT]?
:HELP:HEADers
:PON[:TYPE] PRESet|LAST
:PRESet
:TIME?
:TZONe:NAME?
:VERSion?
```

SYSTem:COMMunicate:GPIB:BUFFer

Controls the internal GPIB input buffer. The buffer should be left on for normal operation. Default setting is off.

Syntax

```
SYST:COMM:GPIB[:SELF]:BUFF OFF|ON|0|1

SYST:COMM:GPIB[:SELF]:BUFF?
```

SYSTem:COMMunicate:NETWork:PASSword

Sets the password to authenticate network file or print actions.

NOTE If networking is not configured, the command will generate a “Settings conflict” error. There is no query version of this command.

Syntax

```
SYST:COMM:NETW:PASS <param>
```

Example

OUTPUT @Osa;"syst:comm:netw:user 'My_username'"	! set samba username
OUTPUT @Osa;"syst:comm:netw:work 'My_workgroup'"	! set samba workgroup
OUTPUT @Osa;"syst:comm:netw:pass 'My_password'"	! set samba password (see security !warning)[a]

[a] Security warning: Setting the samba share password over GPIB is a potential security risk. For maximum security, this parameter should be set from the front panel.

Related Commands

```
SYSTem:COMMunicate:NETWork:USERname
SYSTem:COMMunicate:NETWork:WORKgroup
```

SYSTem:COMMunicate:NETWork:USERname

Sets the username to authenticate network file or print actions.

NOTE If networking is not configured, the command will generate a “Settings conflict” error.

Syntax

```
SYST:COMM:NETW:USER <param>

SYST:COMM:NETW:USER?
```

Example

OUTPUT @Osa;"syst:comm:netw:user 'My_username'"	! set samba username
OUTPUT @Osa;"syst:comm:netw:work 'My_workgroup'"	! set samba workgroup
OUTPUT @Osa;"syst:comm:netw:pass 'My_password'"	! set samba password (see security !warning)[a]

[a] Security warning: Setting the samba share password over GPIB is a potential security risk. For maximum security, this parameter should be set from the front panel.

Related Commands

SYSTem:COMMunicate:NETWork:PASSword
SYSTem:COMMunicate:NETWork:WORKgroup

SYSTem:COMMunicate:NETWork:WORKgroup

Sets the domain to authenticate network file or print actions.

NOTE If networking is not configured, the command will generate a "Settings conflict" error.

Syntax

SYST:COMM:NETW:WORK

SYST:COMM:NETW:WORK?

Example

OUTPUT @Osa;"syst:comm:netw:user 'My_username'"	! set samba username
OUTPUT @Osa;"syst:comm:netw:work 'My_workgroup'"	! set samba workgroup
OUTPUT @Osa;"syst:comm:netw:pass 'My_password'"	! set samba password (see security !warning)[a]

[a] Security warning: Setting the samba share password over GPIB is a potential security risk. For maximum security, this parameter should be set from the front panel.

Related Commands

```
SYSTem:COMMunicate:NETWork:PASSword
SYSTem:COMMunicate:NETWork:USERname
```

SYSTem:DATE?

Queries the date of the real-time clock of the optical spectrum analyzer.

Syntax

```
SYST:DATE?
```

Example

```
DIM A$[48]
OUTPUT @Osa;"syst:date?"
ENTER @Osa;A$
PRINT A$
```

SYSTem:ERRor[:NEXT]?

Queries the earliest entry in the error queue, then deletes it. The *CLS command clears the entire error queue.

The Agilent 86140B series has a 30 entry error queue. The queue is a first-in, first-out buffer. Repeatedly sending the query SYSTem:ERRor? returns the error numbers and descriptions in the order in which they occur until the queue is empty. Any further queries returns +0,"No errors" until another error occurs.

For a complete list of error messages, refer to "SCPI Defined Errors" in the Agilent 86140B User's Guide.

Syntax

```
SYST:ERR[:NEXT]?
```

Example

```
DIM A$[48]
OUTPUT @Osa;"syst:error?"
ENTER @Osa;A$
PRINT A$
```

Related Key

SHOW CRITICAL ERRORS
SHOW HW ERRORS
SHOW WARNINGS
SHOW NOTICES

SYSTem:HELP:HEADers?

Returns a list of all commands and queries implemented by the instrument.

The returned ASCII string of commands is in the IEEE 488.2 arbitrary-block data format. The first line indicates the total number of bytes returned to the computer. That is, the # character is followed by one digit which indicates how many of the following digits convey the byte count. The next digits give the actual byte count. For example, in the listing below, 4387 bytes are indicated in the file.

Each command in the listing is separated by a line-feed character.

Syntax

```
SYST:HELP:HEAD?
```

Example

The following is an example of the first few lines and last few lines returned in the string. The term *nquery* indicates that a command cannot be sent as a query.

The term *qonly* indicates that a command can only be sent as a query.

```
#44387
:ABORt/nquery/
.
.
.
*IDN?/qonly/
*OPC
*RCL/nquery/
*RST/nquery/
*SAV/nquery/
*SRE
*STB?/qonly/
*TRG/nquery/
*TST?/qonly/
*WAI/nquery/
```

SYSTem:PON[:TYPE]

Selects the state, IP or Last, of the instrument when it is turned on. The default state is IP.

If IP is selected, the instrument will turn on in a known, preset state. With the settings as they would be after pressing the front-panel Preset key, or sending a SYSTem:PRESet command. For a list of parameter settings, Refer to "SYSTem:PRESet" on page 244.

If Last state is selected, the instrument will turn on with the settings as they were when the instrument was turned off. This is equivalent to recalling a saved instrument state or measurement file.

Syntax

```
SYST:PON[:TYPE] PRES|LAST

SYST:PON[:TYPE]?
```

Example

OUTPUT @Osa;"syst:pon LAST"	! set osa to return to last state after power up

Related Key

POWER ON STATE IP
PRESET

SYSTem:PRESet

Performs the equivalent of pressing the front-panel PRESET key.

The instrument state is set according to the settings shown in the following table:

Syntax

```
SYST:PRES
```

Example

OUTPUT @Osa;"syst:pres"	! preset instrument

Related Key

PRESET

SYSTem:TIME?

Queries the time of the real-time clock of the optical spectrum analyzer.

Syntax

```
SYST:TIME?
```

SYSTem:TZONe:NAME?

Returns the time zone used by the real-time clock of the spectrum analyzer. The time zone must be one of the following:

HST10	Hawaii Standard	GMT0bst	Greenwich Mean/British Summer
AST10adt	Aleutian Standard/Daylight	WAT1	Algeria, West Central Africa
YST9ydt	Yukon Standard/Daylight	MET1metdst	Middle European/Daylight
PST8PDT	Pacific Standard/Daylight	EET2	Turkey, Finland, Romania, Greece
MST7	Mountain Standard only	CAT2	Egypt, Sudan, Zaire, Central Africa
MST7MDT	Mountain Standard/Daylight	SAST2sadt	Republic of South Africa
CST6CDT	Central Standard/Daylight	WST3	Western Russia (Moscow)
EST5	Eastern Standard only	EAT3	Eastern Africa, Kenya, Ethiopia
EST5EDT	Eastern Standard/Daylight	WAT3	Moscow, Saudi Arabia, Syria
AST4ADT	Atlantic Standard/Daylight	PST5	Pakistan
NST330NDT	Newfoundland Standard/Daylight	IST530	India
SAT5	Peru, Ecuador, Columbia	TST7	Thailand
SAT430	Venezuela, Guyana, Surinam	SST7	Singapore
SAT4	Western Brazil, Bolivia, Chile	EAT8	Philippines, Hong Kong, China, Taiwan
SAT3	Argentina, Eastern Brazil	WST8	Western Australia
IST1	Iceland	JST9	Japan, Korea
WAT0	NW Africa, Morocco, Mauritania	CST930cdt	Central Australia Standard/Daylight
WET0WETDST	Western Europe/Daylight	EST10edt	Eastern Australia Standard/Daylight
PWT0pst	Portuguese Winter/Summer	NZST12nzdt	New Zealand Standard/Daylight

Syntax

```
SYST:TZON:NAME?
```

Example

```
DIM A$[50]
OUTPUT @Osa;"syst:tzone:name?"  ! set osa to return to last state after power up
ENTER @Osa;A$
PRINT A$
```

SYSTem:VERSion?

Queries the SCPI version. The SCPI version used in the Agilent Variable Optical Attenuator modules and Agilent Variable Optical Attenuator modules with Power Control is 1997.0.

Syntax

```
SYST:VERS?
```

Example

DIM A$[50]	
OUTPUT @Osa;"syst:version?"	! set osa to return to last state after power up
ENTER @Osa;A$	
PRINT A$	

TRACe Subsystem Commands

A TRACe or a DATA area is a named entity stored in instrument memory.

The commands in this subsystem have the following command hierarchy:

TRACe

```
[:DATA]
    :X
        :STARt? <trace_name>
        :STOP? <trace_name>
        :TIMe:SSTop <trace_name>,<numeric_value>,<numeric_value>
        :TYPE? <trace_name>
        [:WAVelength]:SSTop <trace_name>,<numeric_value>,<numeric_value>
    [:Y]? <trace_name>
        :[:POWER] <trace_name>,<data_block>|<numeric_value>{,<numeric_value>}
        :RATio <trace_name>,<data_block>|<numeric_value>{,<numeric_value>}
:EXCHange <trace_1> | <trace_2>
:FEED:CONTrol <trace_name>, ALWays | NEVer
:POINts <trace_name>,<numeric_value>
```

TRACe[:DATA]:X:STARt?

Returns the start value for the X-axis data for the trace.

The X-axis data will be evenly spaced points from STARt to STOP. The number of points is determined by the TRACe:POINts setting.

Syntax

```
TRAC[:DATA]:X:STAR? TRA|TRB|TRC|TRD|TRE|TRF
```

Example

OUTPUT @Osa;"trac:x:star? tra"	! query start wavelength for trace a
ENTER @Osa;Trastart	
OUTPUT @Osa;"trac:y:stop? tra"	! query stop wavelength for trace a
ENTER @Osa;Trastop	

TRACe[:DATA]:X:STOP?

Returns the stop value for the X-axis data for the trace.

The X-axis data will be evenly spaced points from STARt to STOP. The number of points is determined by the TRACe:POINts setting.

Syntax

```
TRAC[:DATA]:X:STOP? TRA|TRB|TRC|TRD|TRE|TRF
```

Example

OUTPUT @Osa;"trac:x:star? tra"	! query start wavelength for trace a
ENTER @Osa;Trastart	
OUTPUT @Osa;"trac:y:stop? tra"	! query stop wavelength for trace a
ENTER @Osa;Trastop	

TRACe[:DATA]:X:TIME:SSTop

This command sets the start and stop values for the X-axis data for the trace and must be in zero span.

The first <numeric_value> corresponds to the start, and the second corresponds to the stop. If the stop value is greater than the start value, a "Data out of range" error will be generated. The X-axis data will be evenly spaced points from start to stop. The number of points is determined by the TRACe:POINts setting. If the trace has an expression defined, this expression will be cleared when changing the X-axis start/stop.

Changing the X-axis data in a trace used in an expression (CALCulate[1|2|3|4|5|6]:MATH:EXPRession) by another trace may cause an error in the expression if the X-axis data in the operands of the expression no longer match.

Syntax

```
TRAC[:DATA]:X:TIME:SST TRA|TRB|TRC|TRD|TRE|TRF
  <numeric_value>,<numeric_value>[S|MS|US]
```

Example

OUTPUT @Osa;"span 0"	
OUTPUT @Osa;"trac:x:time:sstop tra,0ms,350ms"	! set start, stop for trace a in zero span

Related Key

Zero Span

TRACe[:DATA]:X:TYPE?

This query reads the X-axis type for the trace.

The X-axis will be WAVelength for a trace acquired in a normal span, or TIME for a trace acquired in zero span.

The trace names defined for the instrument are: TRA, TRB, TRC, TRD, TRE, and TRF. Specifying any other will generate an "Illegal parameter value" error.

Syntax

```
TRAC[:DATA]:X:TYPE? TRA|TRB|TRC|TRD|TRE|TRF
```

Example

DIM A$[10]	
OUTPUT @Osa;"trac:x:type? TRA"	! will return TIME in zero span, WAV otherwise
ENTER @Osa;A$	
PRINT A$	

TRACe[:DATA]:X[:WAVelength]:SSTop

Sets the start and stop values for the X-axis data for the trace.

The first <numeric_value> corresponds to the start, and the second corresponds to the stop. If the stop value is a shorter wavelength than the start value, a "Data out of range" error will be generated. The X-axis data will be evenly spaced points from start to stop. The number of points is

determined by the TRACe:POINts setting. If the trace has an expression defined, this expression will be cleared when the X-axis start/stop is changed.

Changing the X-axis data in a trace used in an expression (CALCulate[1|2|3|4|5|6]:MATH:EXPRession) by another trace may cause an error in the expression if the X-axis data in the operands of the expression no longer match.

Syntax

```
TRAC[:DATA]:X[:WAV]:SST TRA|TRB|TRC|TRD|TRE|TRF,
  <numeric_value>[M|UM|NM|PM|A|HZ|KHZ|MHZ|GHZ|THZ]
```

TRACe[:DATA][:Y]?

Returns the Y-axis data points for the trace.

The units are determined by the definition of the trace. The trace data format used in the command is determined by the FORMat subsystem.

Syntax

```
TRAC[:DATA][:Y]? TRA|TRB|TRC|TRD|TRE|TRF
```

Example

DIM Tdata(1:100)	
OUTPUT @Osa;"sens:swe:poin 100"	! set number of points to 100
OUTPUT @Osa;"form ascii"	! set to return data in ascii
OUTPUT @Osa;"init:imm;*opc?"	! acquire data for trace a
ENTER @Osa;Temp	
OUTPUT @Osa;"trac? tra"	! query for trace a data
ENTER @Osa;Tdata(*)	! store trace data in array
FOR I=1 TO 100	! print trace a data
PRINT Tdata(I)	
NEXT I	

TRACe[:DATA][:Y][:POWer]

Sets the Y-axis data point values for the trace. The number of Y-axis data points is determined by the TRACe:POINts setting. If a single numeric value is given, all of the Y-axis data points will be set to that value. If more than one value is sent, the trace length will be set to the number of values sent.

This command should be used where trace data represents power. The trace data format to be used with this command is determined by the FORMat subsystem.

Syntax

```
TRAC[:DATA][:Y][:POWer] TRA|TRB|TRC|TRD|TRE|TRF,
<data_block>|<numeric_value>{<numeric_value>}
```

TRACe[:DATA][:Y]:RATio

Sets the Y-axis data points for the trace.

The number of Y-axis data points is determined by the TRACe:POINts setting. If a single numeric value is given, all the Y-axis data points will be set to that value. If more than one value is sent, the trace length will be set to the number of values sent.

This command should be used when the trace data represents a power ratio (unitless number). The trace data format to be used with this command is determined by the FORMat subsystem.

Syntax

```
TRAC[:DATA][:Y]:RAT TRA|TRB|TRC|TRD|TRE|TRF,
  <data_block>|<numeric_value> {,<numeric_value>}
```

TRACe:EXCHange

Exchanges both the X-axis and Y-axis data of the two traces.

The only trace pairs that can be exchanged are TRA with any trace, and TRB with TRC. Specifying any other trace will generate an “Illegal parameter value” error.

The TRACe:FEED:CONTrol of the two traces is set to NEVer (trace updating is turned off) before the data is exchanged. Both X-axis and Y-axis data will be exchanged. After the data is exchanged, the TRACe:FEED:CONTrol of the two traces is left at NEVer.

Changing the X-axis data in a trace used in an expression (CALCulate[1|2|3|4|5|6]:MATH:EXPRession) may cause an error in the expression if the
X-axis data in the operands of the expression no longer match.

Syntax

```
TRAC:EXCH TRA, TRB|TRC|TRD|TRE|TRF
TRAC:EXCH TRB, TRC
```

Example

OUTPUT @Osa;"trac:exch tra,trb"	! exchange trace a with trace b

Related Key

EXCHANGE MENU

TRACe:FEED:CONTrol

Controls how often the specified trace accepts new data.

Setting the TRACe:FEED:CONTrol command to ALWays will allow the trace to always accept new data whenever data is available from the FEED. This is equivalent to turning on the trace update from the front panel.

Setting the TRACe:FEED:CONTrol command to NEVer will cause no new data to be fed into the trace. This is equivalent to turning off the trace update from the front panel.

When switching from NEVer to ALWays, all the valid data from the data source is immediately copied into the trace. If, for example:

- the instrument is in single sweep mode.
- TRA has SENSe:DATA as the FEED and NEVer as the FEED:CONTrol.
- SENSe:DATA contains valid measurement data

Setting the TRACe:FEED:CONTrol command from NEVer to ALWays for TRA will immediately copy the SENSe:DATA into trace A. If the instrument was in continuous sweep mode, and a sweep was in progress, setting the CONTrol command from NEVer to ALWays would immediately copy all the valid SENSe:DATA for the partial sweep.

Syntax

TRAC:FEED:CON TRA|TRB|TRC|TRD|TRE|TRF, ALW|NEV

TRAC:FEED:CONT? TRA|TRB|TRC|TRD|TRE|TRF

Example

OUTPUT @Osa;"trac:feed:cont tra,NEV"	! turn trace update off for trace

Related Key

UPDATE A...F ON|OFF

TRACe:POINts

Sets the number of data points to be used in the trace. Use this command only for downloading data with TRACe subsystem commands. Use SENSe :SWEep:POINts for changing measurement trace length. To download the trace, this number must match the number in the SENSe:SWEep:POINts command.

Syntax

TRAC:POIN TRA|TRB|TRC|TRD|TRE|TRF,<numeric_value>

TRAC:POIN?

Related Commands

SENSe:SWEep:POINts

TRIGger Subsystem Commands

The commands in this subsystem have the following command hierarchy:

TRIGger

```
[:SEQuence]
    :DELay <numeric_value> [S|MS|US|NS]
    :OUTPut OFF|ON|0|1
        :PULSe
            :DCYCle <numeric_value>
            :STATe OFF|ON|0|1
            :WIDTh <numeric_value> [S|MS|US|NS]
    :SLOPe POSitive|NEGative|EITHer
    :SOURce IMMEdiate|EXTernal|INTernal
```

TRIGger[:SEQuence]:DELay

Specifies the trigger delay used to start a measurement.

Syntax

```
TRIG[:SEQ]:DEL <numeric_value>[S|MS|US|NS]

TRIG[:SEQ]:DEL?
```

Example

OUTPUT @Osa;"trig:del 2ms"	! set trigger delay to 2ms

Related Key

TRIGGER DELAY

TRIGger[:SEQuence]:OUTPut

Controls the ADC trigger output. When off the signal will be a TTL low. When on the signal will be a TTL high. When AUTO is specified, the signal will go high before the sampling interval of the detector and go low after the sampling interval of the detector.

Syntax

```
TRIG[:SEQ]:OUTP OFF|ON|AUTO
TRIG[:SEQ]:OUTP?
```

Example

OUTPUT @Osa;"trig:output auto"	! set ADC trigger output to auto

Related Key

TRIGGER MODE

TRIGger[:SEQuence]:OUTPut:PULSe:DCYCle

Sets the duty cycle of the sync output. This is equivalent to the SOURce[n] :PULSe:DCYCle command.

Syntax

```
TRIG[:SEQ]:OUTP:PULS:DCYC <numeric_value>
TRIG[:SEQ]:OUTP:PULS:DCYC?
```

Example

OUTPUT @Osa;"trig:outp:puls:dcyc 1"	! set ADC duty cycle

Related Key

ADC SYNC OUT DUTY CYCLE

TRIGger[:SEQuence]:OUTPut:PULSe:STATe

Enables the ADC Sync Output on the rear panel of the optical spectrum analyzer.

Syntax

```
TRIG[:SEQ]:OUTP:PULS:STAT ON|OFF|0|1
```

Related Key

ADC Sync Out Pulse Width

TRIGger[:SEQuence]:OUTPut:PULSe:WIDTh

Sets the pulse width of the sync output. This is equivalent to the SOURce[n]
:PULSe:WIDTh command.

Syntax

```
TRIG[:SEQ]:OUTP:PULS:WIDT <numeric_value>[S|MS|US|NS]

TRIG[:SEQ]:OUTP:PULS:WIDT?
```

Example

OUTPUT @Osa;"trig:outp:puls:widt 2nm"	! set pulse width of sync output

Related Key

ADC Sync Out Pulse Width

TRIGger[:SEQuence]:SLOPe

Specifies the polarity of triggering used to start a measurement. Specifying the slope sets the trigger source to INTernal.

Syntax

```
TRIG[:SEQ]:SLOP POS|NEG|EITH

TRIG[:SEQ]:SLOP?
```

Example

```
OUTPUT @Osa;"trig:slop POS"   ! set trigger polarity to positive
```

Related Key

TRIGGER MODE

TRIGger[:SEQuence]:SOURce

Specifies the source, or type, of triggering used to start a measurement. Setting the source to anything other than INTernal sets the slope to EITHer.

Syntax

```
TRIG[:SEQ]:SOUR IMM|EXT|INT

TRIG[:SEQ]:SOUR?
```

Example

```
OUTPUT @Osa;"trig:sour INT" ! set trigger source to internal
```

Related Key

TRIGGER MODE

UNIT Subsystem Commands

Default values are defined, where applicable, for each SCPI command. The UNIT subsystem provides a mechanism to change the default values. The units selected apply to the designated command parameters for both command and response.

The commands in this subsystem have the following command hierarchy:

UNIT

:POWer DBM|W

:RATio DB|LINear|AUTO

UNIT:POWer

Selects the units in dBm or mW for the measured value.

Syntax

UNIT:POW DBM|W

Example

OUTPUT @Osa;"unit:pow DBM"	! set power units to dBm
OUTPUT @Osa;"unit:rat DB"	! set ratio units to decibels

Related Key

(AMPLITUDE SETUP) AMPLITUDE UNITS

UNIT:RATio

Specifies units for the input and output of values that represent power ratios. These commands are:

CALCulate:MARKer[1|2|3|4]:FUNCtion:BWIDth|BANDwidth:NDB
DISPlay:WINDow:TRACe:Y:SCALe: AUTO:PDIVision
DISPlay:WINDow:TRACe:Y:SCALe:PDIVision
TRACe:DATA:Y?

This command also sets UNIT:POWer to the corresponding setting.

Syntax

```
UNIT:RAT DB|LIN|AUTO

UNIT:RAT?
```

Example

OUTPUT @Osa;"unit:pow DBM"	! set power units to dBm
OUTPUT @Osa;"unit:rat DB"	! set ratio units to decibels

Agilent 71450 Series Commands to Agilent 86140B Series Equivalents

The following table provides a list of the Agilent 71450 series commands and the SCPI equivalent commands for the Agilent 86140B series analyzers.

NOTE: For the 86140B series OSA, any space(s) or characters between a token (_) and the query (?) symbol are not supported. For example:

FP_? will not work on the 86140B series OSA, but will work for the 71450 series OSA.

FP_? will work on the 86140B series OSA.

Table 11 Agilent 71450 Series Commands to Agilent 86140B Series Commands (Sheet 1 of 13)

Agilent 71450 Series Command	Equivalent Agilent 86140B Series Command
ABORT	
ABS	CALCulate1:MATH:EXPRession[:DEFine] <expression> CALCulate1:MATH:STATe ON
ACTDEF	
ACTPARM	
ADAPBTL	CALCulate1:MATH:EXPRession[:DEFine] <expression> CALCulate1:MATH:STATe ON
ADAPBPCTL	CALCulate1:MATH:EXPRession[:DEFine] <expression> CALCulate1:MATH:STATe ON
ADBTL	CALCulate1:MATH:EXPRession[:DEFine] <expression> CALCulate1:MATH:STATe ON
ADCTL	CALCulate1:MATH:EXPRession[:DEFine] <expression> CALCulate1:MATH:STATe ON
ADCTRG	TRIGger[:SEQuence]:SLOPe POSitive | NEGative | EITHer TRIGger[:SEQuence]:SOURce IMMediate | EXTernal | INTernal
ADCTRGDLY	TRIGger[:SEQuence]:DELay <numeric_value>[<unit>]
ADCTRGSYN	TRIGger[:SEQuence]:OUTPut OFF | ON | 0 | 1 | AUTO
ADD	CALCulate1:MATH:EXPRession[:DEFine] <expression> CALCulate1:MATH:STATe ON
ALIGN	CALibration:ALIGn:EXTernal
ALIGNPRST	CALibration:ALIGn:PRESet

Table 11 Agilent 71450 Series Commands to Agilent 86140B Series Commands (Sheet 2 of 13)

Agilent 71450 Series Command	Equivalent Agilent 86140B Series Command
AMB	CALCulate1:MATH:EXPRession[:DEFine] <expression> CALCulate1:MATH:STATe ON
AMBMC	CALCulate1:MATH:EXPRession[:DEFine] <expression> CALCulate1:MATH:STATe ON
AMBMCPL	CALCulate1:MATH:EXPRession[:DEFine] <expression> CALCulate1:MATH:STATe ON
AMBPL	CALCulate1:MATH:EXPRession[:DEFine] <expression> CALCulate1:MATH:STATe ON
AMC	CALCulate1:MATH:EXPRession[:DEFine] <expression> CALCulate1:MATH:STATe ON
AMCPL	CALCulate1:MATH:EXPRession[:DEFine] <expression> CALCulate1:MATH:STATe ON
AMETER	
AMPCOR	CALCulate1:OFFSet
AMPMKR	CALCulate:MARKer[1|2|3|4]:FUNCtion:BWIDth|BANDwidth:NDB <param>
AMPU	
ANNOFF	
ANNOT	DISPlay[:WINDow[1]]:ANNotation[:ALL] OFF|ON|0|1
APB	CALCulate1:MATH:EXPRession[:DEFine] <expression> CALCulate1:MATH:STATe ON
APBDCTL	CALCulate1:MATH:EXPRession[:DEFine] <expression> CALCulate1:MATH:STATe ON
AUNITS	UNIT:POWer DBM|W|AUTO
AUTOALIGN	CALibration:ALIGn[:AUTO] CALibration:ALIGn:MARKer[1|2|3|4]
AUTOMDB	DISPlay[:WINDow[1]]:TRACe:Y:SCALe:AUTO:PDIVision <param> DISPlay[:WINDow[1]]:TRACe:Y:SCALe:AUTO:PDIVision:AUTO:PDIVision :AUTO OFF|ON|0|1
AUTOMEAS	DISPlay[:WINDow[1]]:TRACe:ALL[:SCALe][:AUTO]
AUTOMMKR	DISPlay[:WINDow[1]]:TRACe:ALL[:SCALe][:AUTO]:MARKer OFF|ON|0|1
AUTOMOPT	DISPlay[:WINDow[1]]:TRACe:ALL[:SCALe][:AUTO]:OPTimize OFF|ON|0|1
AUTOMSP	DISPlay[:WINDow[1]]:TRACe:X:SCALe:AUTO:SPAN <param> DISPlay[:WINDow[1]]:TRACe:X:SCALe:AUTO:SPAN:AUTO OFF|ON|0|1
AUTORNG	[SENSe]:POWer[:DC]:RANGe:AUTO OFF|ON|0|1

Table 11 Agilent 71450 Series Commands to Agilent 86140B Series Commands (Sheet 3 of 13)

Agilent 71450 Series Command	Equivalent Agilent 86140B Series Command
AVG	CALCulate1:MATH:EXPRession[:DEFine] <expression> CALCulate1:MATH:STATe ON
AXB	TRACe:EXCHange TRA, TRB
AXC	TRACe:EXCHange TRA, TRC
BIT	
BLANK	DISPlay[:WINDow[1]]:TRACe[:STATe] <trace>,OFF|ON|0|1
BML	CALCulate1:MATH:EXPRession[:DEFine] <expression> CALCulate1:MATH:STATe ON
BP	
BTC	
BXC	TRACe:EXCHange TRB, TRC
CAL	CALibration:PRESet CALibration:POWer CALibration:WAVelength CALibration:WAVelength:MARKer
CALCOR	CALibration:STATe CALibration:POWer:STATe CALibration:WAVelength:STATe
CALDATA	
CALPWR	CALibration:POWer:VALue
CALWL	CALibration:WAVelength:VALue
CATALOG	MMEMory:CATalog?
CENTERWL	[SENSe:]WAVelength:CENTer <param>
CHEIGHT	
CHOP	[SENSe]:CHOP[:STATe] OFF|ON|0|1
CLRDSP	
CLRW	TRACe:FEED:CONTrol <trace>, ALWays
CLS	*CLS
COMPRESS	
CONCAT	
CONFIG	
CONTS	INITiate:CONTinuous ON|1
CORSEL	
CORTOLIM	
CWIDTH	
DE	

Table 11 Agilent 71450 Series Commands to Agilent 86140B Series Commands (Sheet 4 of 13)

Agilent 71450 Series Command	Equivalent Agilent 86140B Series Command
DEBUG	
DELETE	
DFB_	
DISPOSE	
DISPU	
DIV	CALCulate1:MATH:EXPRession[:DEFine] <expression> CALCulate1:MATH:STATe ON
DL	
DONE	*OPC?
DSPLY	
DSPMODE	
DSPTEXT	CALCulate1:MATH:EXPRession[:DEFine] <expression> CALCulate1:MATH:STATe ON
DWINDOW	
ENTER	
ERASE	
ERR	SYSTem:ERRor?
EXP	CALCulate1:MATH:EXPRession[:DEFine] <expression> CALCulate1:MATH:STATe ON
FETCH	
FFT	
FFTKNL	
FORMAT	MMEMory:INITialize [FLOPpy]
FP_	
FP_MKBW	
FP_TH	
FS	[SENSe][:WAVelength]:SPAN:FULL
FUNCDEF	
GATESWP	
GRAPH	
GRAT	DISPlay[:WINDow[1]]:TRACe:GRATicule:GRID[:STATe] OFF | ON | 0 | 1
GRATORDER	[SENSe]:GORDer[:AUTO] OFF | ON | 0 | 1
GRATSCRL	
GRID	

Table 11 Agilent 71450 Series Commands to Agilent 86140B Series Commands (Sheet 5 of 13)

Agilent 71450 Series Command	Equivalent Agilent 86140B Series Command
HD	
ID	*IDN?
IF/THEN	
IGEN	
IGENDTYCY	SOURce[n]:PULSe:DCYCle <numeric_value>
IGENLIMIT	
IGENPW	SOURce[n]:PULSe:WIDTh <numeric_value>
INSTMODE	
INT	CALCulate1:MATH:EXPRession[:DEFine] <expression> CALCulate1:MATH:STATe ON
IP	SYSTem:PRESet
IT	
IWINDOW	
KEYCLR	
KEYDEF	
KEYPST	
LED_	
LG	DISPlay[:WINDow[1]]:TRACe:Y[:SCALe]:SPACing LOGarithmic DISPlay[:WINDow[1]]:TRACe:Y[:SCALe]:PDIVision <numeric_value>[<unit>]
LIGHT	
LIMIAMP	
LIMIBEEP	
LIMIBOT	
LIMIDEL	
LIMIDONE	
LIMIEDIT	
LIMIFAIL	
LIMIHALF	
LIMILINE	
LIMINEXT	
LIMIRCL	
LIMIREL	
LIMISAV	

Table 11 Agilent 71450 Series Commands to Agilent 86140B Series Commands (Sheet 6 of 13)

Agilent 71450 Series Command	Equivalent Agilent 86140B Series Command
LIMISCRL	
LIMISDEL	
LIMISEG	
LIMITEST	
LIMITYPE	
LIMIWL	
LIMTOCOR	
LINES	
LINET	
LN	DISPlay[:WINDow[1]]:TRACe:Y[:SCALe]:SPACing LINear
LOAD	
LOG	
MDS	FORMat[:DATA] <param>
MEAN	CALCulate[1|2|3|4|5|6]:MEAN:STATe ON CALCulate[1|2|3|4|5|6]:MEAN[:DATA]?
MEASU	
MEASURE	
MEM	
MIN	CALCulate1:MATH:EXPRession[:DEFine] <expression> CALCulate1:MATH:STATe ON
MINH	CALCulate[1|2|3|4|5|6]:MINimum[:STATe] ON
MK	
MKA	CALCulate:MARKer[1|2|3|4]:Y?
MKACT	CALCulate:MARKer[1|2|3|4][:STATe] ON|1
MKAL	CALCulate:MARKer[1|2|3|4]:FUNCtion:BWIDth|BANDwidth:X:LEFT?
MKAR	CALCulate:MARKer[1|2|3|4]:FUNCtion:BWIDth|BANDwidth:X:RIGHt?
MKBW	CALCulate:MARKer[1|2|3|4]:FUNCtion:BWIDth|BANDwidth:RESult? CALCulate:MARKer[1|2|3|4]:FUNCtion:BWIDth|BANDwidth[:STATe]
MKBWA	CALCulate:MARKer[1|2|3|4]:FUNCtion:BWIDth|BANDwidth:NDB CALCulate:MARKer[1|2|3|4]:FUNCtion:BWIDth|BANDwidth:RESult?
MKCONT	
MKCWL	CALCulate:MARKer[1|2|3|4]:SCENter
MKD	CALCulate:MARKer[1|2|3|4]:FUNCtion:BWIDth|BANDwidth:NDB CALCulate:MARKer[1|2|3|4]:FUNCtion:BWIDth|BANDwidth:RESult?
MKDACT?	

Table 11 Agilent 71450 Series Commands to Agilent 86140B Series Commands (Sheet 7 of 13)

Agilent 71450 Series Command	Equivalent Agilent 86140B Series Command
MKDREFA	
MKDREFF	
MKMIN	CALCulate:MARKer[1 | 2 | 3 | 4]:MINimum
MKN	CALCulate:MARKer[1 | 2 | 3 | 4]:FUNCtion:PRESet CALCulate:MARKer[1 | 2 | 3 | 4]:X <param>
MKNOISE	CALCulate:MARKer[1 | 2 | 3 | 4]:FUNCtion:NOISe[:STATe] OFF | ON | 0 | 1 CALCulate:MARKer[1 | 2 | 3 | 4]:FUNCtion:NOISe:RESult?
MKOFF	CALCulate:MARKer:AOFF CALCulate:MARKer[1 | 2 | 3 | 4][:STATe] OFF | 0
MKP	
MKPABS	
MKPAUSE	
MKPITX	CALCulate:MARKer[1 | 2 | 3 | 4]:PEXCursion:PIT <param>
MKPK	CALCulate:MARKer[1 | 2 | 3 | 4]:MAXimum CALCulate:MARKer[1 | 2 | 3 | 4]:MAXimum:NEXT CALCulate:MARKer[1 | 2 | 3 | 4]:MAXimum:LEFT CALCulate:MARKer[1 | 2 | 3 | 4]:MAXimum:RIGHt CALCulate:MARKer[1 | 2 | 3 | 4]:MINimum:NEXT CALCulate:MARKer[1 | 2 | 3 | 4]:MINimum:LEFT CALCulate:MARKer[1 | 2 | 3 | 4]:MINimum:RIGHt
MKPX	CALCulate:MARKer[1 | 2 | 3 | 4]:PEXCursion[:PEAK] <param>
MKREAD	CALCulate:MARKer[1 | 2 | 3 | 4]:FUNCtion:BWIDth | BANDwidth:READout FREQuency | WAVelength | TIME CALCulate:MARKer[1 | 2 | 3 | 4]:FUNCtion:DELTa:X:READout FREQuency | WAVelength | TIME
MKRL	CALCulate:MARKer[1 | 2 | 3 | 4]:SRLevel
MKSP	
MKSS	
MKSTOP	
MKT	
MKTRACE	CALCulate:MARKer[1 | 2 | 3 | 4]:TRACe <source_trace>
MKTRACK	
MKTUNE	
MKTV	

Table 11 Agilent 71450 Series Commands to Agilent 86140B Series Commands (Sheet 8 of 13)

Agilent 71450 Series Command	Equivalent Agilent 86140B Series Command
MKTYPE	CALCulate:MARKer[1|2|3|4]:FUNCtion:NOISe[:STATe] OFF|ON|0|1 CALCulate:MARKer[1|2|3|4]:FUNCtion:DELTa[:STATe] OFF|ON|0|1 CALCulate:MARKer[1|2|3|4]:FUNCtion:BWIDth|BANDwidth[:STATe] OFF|ON|0|1 CALCulate:MARKer[1|2|3|4]:FUNCtion:PRESet
MKWL	CALCulate:MARKer[1|2|3|4]:X <param>
MOD	CALCulate1:MATH:EXPRession[:DEFine] <expression> CALCulate1:MATH:STATe ON
MODADD	
MODID	
MOV	
MPY	CALCulate1:MATH:EXPRession[:DEFine] <expression> CALCulate1:MATH:STATe ON
MSG	
MSI	
MXM	CALCulate1:MATH:EXPRession[:DEFine] <expression> CALCulate1:MATH:STATe ON
MXMH	CALCulate[1|2|3|4|5|6]:MAXimum[:STATe] ON
NORM	
NSTATE	
ONEOS	
ONMENU	
ONMKR	
ONUSER	
ONWINDOW	
OP	
OPTSW	
OR	
OUTPUT	
OVRW	
PA	
PAUSE	
PD	
PDA	
PDL_	
PDLCALC	

Table 11 Agilent 71450 Series Commands to Agilent 86140B Series Commands (Sheet 9 of 13)

Agilent 71450 Series Command	Equivalent Agilent 86140B Series Command
PDLDEV	
PDL_DEV?	
PDLEXIT	
PDLINIT	
PDLREV	
PDL_REV?	
PDLSCALE	
PDLSRC	
PDL_SRC?	
PDMEAS	
PDWL	
PEAKS	CALCulate:MARKer[1|2|3|4]:MAXimum CALCulate:MARKer[1|2|3|4]:X? CALCulate:MARKer[1|2|3|4]:MAXimum:NEXT CALCulate:MARKer[1|2|3|4]:X?
PEN	
PERASE	
PERSIST	
PLOT	HCOPy[:IMMediate]
POSU	
POWERON	SYSTem:PON[:TYPE] PRESet|LAST
PR	
PREFX	
PROTECT	
PSTATE	
PU	
PURGE	MMEMory:DELete <file_name>[INTernal|FLOPpy]
PWRBW	
RB	[SENSe]:BANDwidth|BWIDth[:RESolution] <param> [SENSe]:BANDwidth|BWIDth[:RESolution]:AUTO OFF|ON|0|1
RBR	[SENSe]:BANDwidth|BWIDth[:RESolution]:RATio <param>
RCLD	
RCLS	*RCL <numeric_value>|<filename>[INTernal|FLOPpy]
RCLT	MMEMory:LOAD:TRACe <trace>, <file_name>
RCLU	

Table 11 Agilent 71450 Series Commands to Agilent 86140B Series Commands (Sheet 10 of 13)

Agilent 71450 Series Command	Equivalent Agilent 86140B Series Command
READMENU	
RELHPIB	
REPEAT/ UNTIL	
RETURN	
REV	*IDN?
RL	DISPlay[:WINDow[1]]:TRACe:Y[:SCALe]:RLEVel <numeric_value>[<unit>]
RLPOS	DISPlay[:WINDow[1]]:TRACe:Y[:SCALe]:RPOSition <numeric_value>
RMS	
ROFFSET	CALCulate [1|2|3|4|5|5]:OFFSet
RQS	*SRE
SAVED	
SAVES	*SAV <numeric_value>|<filename>[INTernal|FLOPpy]
SAVET	MMEMory:STORe:TRACe <trace>, <file_name>
SAVEU	
SCALE	
SENS	[SENSe]:POWer:AC:RANGe:LOWer <numeric_value>|<step> [SENSe]:POWer:AC:RANGe:AUTO OFF|ON|0|1
SER	*IDN?
SMOOTH	
SNGLS	INITiate:CONTinuous OFF|0
SP	[SENSe]:WAVelength:SPAN <param>
SPANWL	[SENSe]:WAVelength:SPAN <param>
SQR	CALCulate1:MATH:EXPRession[:DEFine] <expression> CALCulate1:MATH:STATe ON
SRINPUT	
SRQ	
SS	[SENSe]:WAVelength:CENTer:STEP[:INCRement] <param> [SENSe]:WAVelength:CENTer:STEP:AUTO OFF|ON|0|1
ST	[SENSe]:SWEep:TIME <param> [SENSe]:SWEep:TIME:AUTO OFF|ON|0|1
STARTUP	
STARTWL	[SENSe]:WAVelength:STARt <param>
STATE	
STB	*STB?

Table 11 Agilent 71450 Series Commands to Agilent 86140B Series Commands (Sheet 11 of 13)

Agilent 71450 Series Command	Equivalent Agilent 86140B Series Command
STDEV	
STOPWL	[SENSe]:WAVelength:STOP <param>
STOR	
STORREF	
SUB	CALCulate1:MATH:EXPRession[:DEFine] <expression> CALCulate1:MATH:STATe ON
SUM	CALCulate[1|2|3|4|5|6]:TPOWer:STATe ON CALCulate[1|2|3|4|5|6]:TPOWer[:DATA]?
SUMSQR	
SWEEP	INITiate:CONTinuous OFF|ON|0|1
SWPMODE?	INITiate:CONTinuous?
TDF	FORMat[:DATA] <param>
TEST	*TST?
TEXT	DISPlay[:WINDow[1]]:TEXT:DATA <string>|<block>
TH	CALCulate[1|2|3|4|5|6]:THReshold <param>
THREED	
THREEDH	
THREEDV	
TIME	
TITLE	DISPlay[:WINDow[1]]:TEXT:DATA <string>|<block>
TM	TRIGger[:SEQuence]:SOURce IMMediate|EXTernal|INTernal
TP	
TRA/TRB/TRC	TRACe:DATA[:Y]? <trace_name> TRACe[:DATA][:Y][:POWer] <trace_name>,<data_block>| <numeric_value>{<numeric_value>} TRACe:DATA[:Y]:RATio <trace_name>,<block>| <numeric_value>{<numeric_value>}
TRCOND	
TRDEF	SENSe:SWEep:POINts <numeric value>
TRDSP	DISPlay[:WINDow[1]]:TRACe[:STATe] <trace>,OFF|ON|0|1
TRNSZLOCK	[SENSe]:POWer:RANGe:LOCK

Table 11 Agilent 71450 Series Commands to Agilent 86140B Series Commands (Sheet 12 of 13)

Agilent 71450 Series Command	Equivalent Agilent 86140B Series Command
TRPST	DISPlay[:WINDow[1]]:TRACe[:STATe] TRA,ON DISPlay[:WINDow[1]]:TRACe[:STATe] TRB,OFF DISPlay[:WINDow[1]]:TRACe[:STATe] TRC,OFF CALCulate1:MATH:STATe OFF CALCulate1:AVERage[:STATe] OFF CALCulate2:AVERage[:STATe] OFF CALCulate3:AVERage[:STATe] OFF TRACe:POINts TRA,800 TRACe:POINts TRB,800 TRACe:POINts TRC,800
TRSTAT	DISPlay[:WINDow[1]]:TRACe[:STATe]? <trace>
TS	INITiate[:IMMediate]
TWNDOW	
USERERR	
USERKEY	
USERLOCK	
USERMSG	
USERWARN	
USTATE	
VARDEF	
VARIANCE	
VAVG	CALCulate[1|2|3|4|5|6]:AVERage:COUNt <numeric_value> CALCulate[1|2|3|4|5|6]:AVERage[:STATe] OFF|ON|0|1
VB	[SENSe]:BANDwidth|BWIDth:VIDeo <param> [SENSe]:BANDwidth|BWIDth:VIDeo:AUTO OFF|ON|0|1
VIEW	DISPlay[:WINDow[1]]:TRACe[:STATe] <trace>,OFF|ON|0|1
VTDL	
VTH	
VTL	
VW	
WAIT	
WARN?	
WARNCTRL	
WLLIMIT	[SENSe]:WAVelength]:LIMit OFF|ON|0|1
WLMKRL	CALCulate[1|2|3|4|5|6]:TPOWer:IRANge:LOWer <param>
WLMKRR	CALCulate[1|2|3|4|5|6]:TPOWer:IRANge:UPPer <param>
WLOFFSET	[SENSe]:WAVelength]:OFFSet <numeric_value>

Table 11 Agilent 71450 Series Commands to Agilent 86140B Series Commands (Sheet 13 of 13)

Agilent 71450 Series Command	Equivalent Agilent 86140B Series Command
WLUNITS	
XAMPSW	
XCH	
XERR	SYSTem:ERRor?
XWARN	
ZERO	CALibration:ZERO[:AUTO] OFF | ON | 0 | 1 | ONCE
ZOOMRB	CALCulate:MARKer[1 | 2 | 3 | 4]:SCENter [SENSe]:BANDwidth | BWIDth[:RESolution] <param> [SENSe]:WAVelength]:SPAN 0

Agilent 71450 series commands available on the Agilent 86140B series

The following table provides a list of the Agilent 71450 series commands directly available on the Agilent 86140B series analyzers.

Table 12 Legacy Commands

Agilent 71450 Series Command	Agilent 71450 Series Command	Agilent 71450 Series Command
AMB ON/OFF	MKBW OFF	MKWL
AUNITS	MKBW?	MXMH
AUTO ALIGN	MKBWA	RB
AUTO MEAS	MKCF	RL
AUTO MSP	MKCWL	RLPOS
AUTOMMKR ON	MKD	ROS
AUTORNG	MKMIN	SENS
BLANK	MKN	SNGLS
CAL PRESET	MKNOISE	SP
CALCOR PWR	MKOFF	ST
CALCOR WL	MKOFF ALL	STARTWL
CALPWR	MKP	STB?
CALWL	MKPITX	STOPWL
CENTERWL	MKPK	SWEEP ON/OFF
CLRDSP	MKPK CP	TM
CLS	MKPK CPIT	TRA...F?
CONTS	MKPK HI	TRDEF
DONE?	MKPK HIP	TRDSP
ERR?	MKPK MI	TS
ID?	MKPK MIPIT	VAVG
IP	MKPK NH	VB
LG	MKPK NL	VIEW
LN	MKPK NLPIT	WAIT
MDS W	MKPK NM	WLOFFSET
MINH	MKPK NR	WLUNITS
MKA?	MKPK NRPIT	XCH TRA,TRB
MKACT	MKPX	ZERO
MKAL	MKRL	
MKAR	MKSP	
MKBW ON	MKTRACETRN	

NOTE

General Differences in Formats and Syntax

All queries returning floating point values return them in scientific notation <sign><decimal number>E<sign><exponent>. For example, 12.34 is returned as +1.234E+001

Predefined variables, functions and user-defined variables and functions are not supported by the 8614X.

Integers returned from a query have a sign.

Commands requiring a floating point number may be expressed with a sign, exponent, and exponent sign, all optional. Use “E” (case insensitive) to denote the exponent. For example: "CENTERWL 800nm", "CENTERWL +800.0E-009M" or "CENTERWL 8e-7M" all do the same thing.

Queries are only allowed in the SCPI format. That is, the QUERY token MUST be followed immediately by a "?", and then the arguments, comma separated.

<command>? [,<argument>...]

Wavelength units of "ANG" are "A".

"X" units are not supported.

ANG units not supported, use A.

User defined traces not supported.

Index

T

U

W

www.agilent.com

Printed in Germany January 2005
Second edition, January 2005

86140-90069

Agilent Technologies