

# Agilent 86122C Multi-Wavelength Meter Programming Guide



# Notices

© Agilent Technologies, Inc. 2002-2013  
This document contains proprietary information that is protected by copyright. All rights are reserved.

No part of this document may be reproduced in (including electronic storage and retrieval or translation into a foreign language) without prior agreement and written consent from Agilent Technologies GmbH as governed by United States and international copyright laws.

Agilent Technologies Deutschland GmbH  
Herrenberger Str. 130  
71034 Böblingen  
Germany

## Manual Part Number

86122-90C02

## Edition

First edition, April 2013

## 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 performed 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 indicated conditions are fully understood and met.**

---

## Agilent 86122C Programming

- [Programming Overview](#)
- [Limitations of LAN connections](#)
- [Summary for Experienced Programmers](#)
- [SCPI Syntax Rules](#)
- [Initializing the Interface and the Instrument](#)
- [Non-Sequential Commands](#)
- [Status Registers](#)
- [Queues](#)
- [SCPI Programming Commands](#)
- [Example Programs](#)
- [Returned Data Format](#)
- [Appendix - Messages and Errors](#)

## Programming Overview

To control the 86122C from an external computer the following steps are required:

1. [Connect the 86122C physically](#)
2. [Set up the software connection](#)
  - a) [Select the software standards](#)
  - b) [Configure your 86122C](#)
  - c) [Set up your controller \(e.g. personal computer\)](#)
3. [Control the 86122C programmatically](#)

1. Connect the 86122C physically

The 86122C provides two interface ports for controlling purposes. These are General Purpose Interface Bus (GPIB; IEEE 488), and Local Area Network (LAN) Ethernet.

- GPIB

The 86122C can be connected to GPIB controller interfaces using a GPIB cable. There is a variety of GPIB controller interfaces commercially available. The most common example is a GPIB card plugged into a PC (see example configuration block diagrams below). Another example is a GPIB controller interface that is connected via USB to a controlling computer.

- LAN

The 86122C can be connected via LAN cable to an existing LAN. In most cases the connection will be made to a LAN switch or LAN router. Any computer that is connected to the LAN can control the 86122C. A special cross-over LAN cable can also be used for a direct point-to-point connection between the computer and instrument.

Compared to GPIB connections there are some minor limitations to using a LAN connection. The vast majority of applications will not be affected. For details see [Limitations of LAN connections](#).

## 2. Set up the software connection

### a) Select the software standards

Independently of the physical connection, there is a variety of industry software standards to control measurement instruments like the 86122C. To fit into the specific environment, the 86122C supports many of them. Among others the following lists only the highest level standards that are relevant to the programmer:

- **VXIplug&play**

An easy way to program this instrument is by using the VXIplug&play universal instrument drivers. The VXIplug&play universal instrument drivers allow you to develop programs using the following applications: Agilent VEE, LabVIEW, LabWindows/CVI, C, C++, and Microsoft® Visual Basic and others.

The VXIplug&play driver provides many 86122C specific functions that you can call. Use e.g. use the function `ag8612x_StartSweep` to start a wavelength sweep of the 86122C. The set of specific functions allows you to programmatically control your 86122C for a many applications. Thus, normally you do not need to send SCPI commands if you use the VXIplug&play driver.

The VXIplug&play driver also provides pass-through functions (e.g. `ag8612x_QueryString`) to send SCPI commands too. This allows you to make use of all 86122C capabilities.

For more information about using the VXIplug&play driver see the example programs under `c:\VXIlnp\WinNT\ag8612x\Examples\` after installation of the driver.

- **Standard Commands for Programmable Instruments (SCPI)**

You can control the instrument remotely by using the SCPI commands. The commands allow you to control the test equipment as well as to make measurements quickly, accurately, and repeatably. You can also use SCPI Commands to perform other functions, such as printing.

If you have already controlled instruments using SCPI, [Summary for Experienced Programmers](#) has information you will find useful.

For more information on SCPI see also [SCPI Syntax Rules](#) and [SCPI Programming Commands](#).

- **VISA**

VISA is a software standard that enables programs to open, use and close data communication channels to instruments.

The program opens the communication channel using the function `viOpen(descriptor)`. Descriptor is a string that determines a communication path to a particular instrument via a particular physical connection. You can e.g. open a VISA communication channel to your 86122C via GPIB by

```
viOpen("GPIB0::30::INSTR")
```

(in this example the GPIB address of your 86122C is assumed to be set to 30). Alternatively you can open a VISA communication channel to your 86122C via LAN by

```
viOpen("TCPIP0::86122C-0321::inst0::INSTR")
```

(in this example the 86122C is assumed to be 86122C-0321; the trailing 4 digits are by convention the last 4 of the serial number of your 86122C).

Once a communication channel has been opened you can exchange commands and data by sending SCPI commands or reading replies of the 86122C. The underlying physical connection is handled by the VISA and need not be specified in the program.

VISA is a very common standard supported on many operating system controllers.

- **SICL**

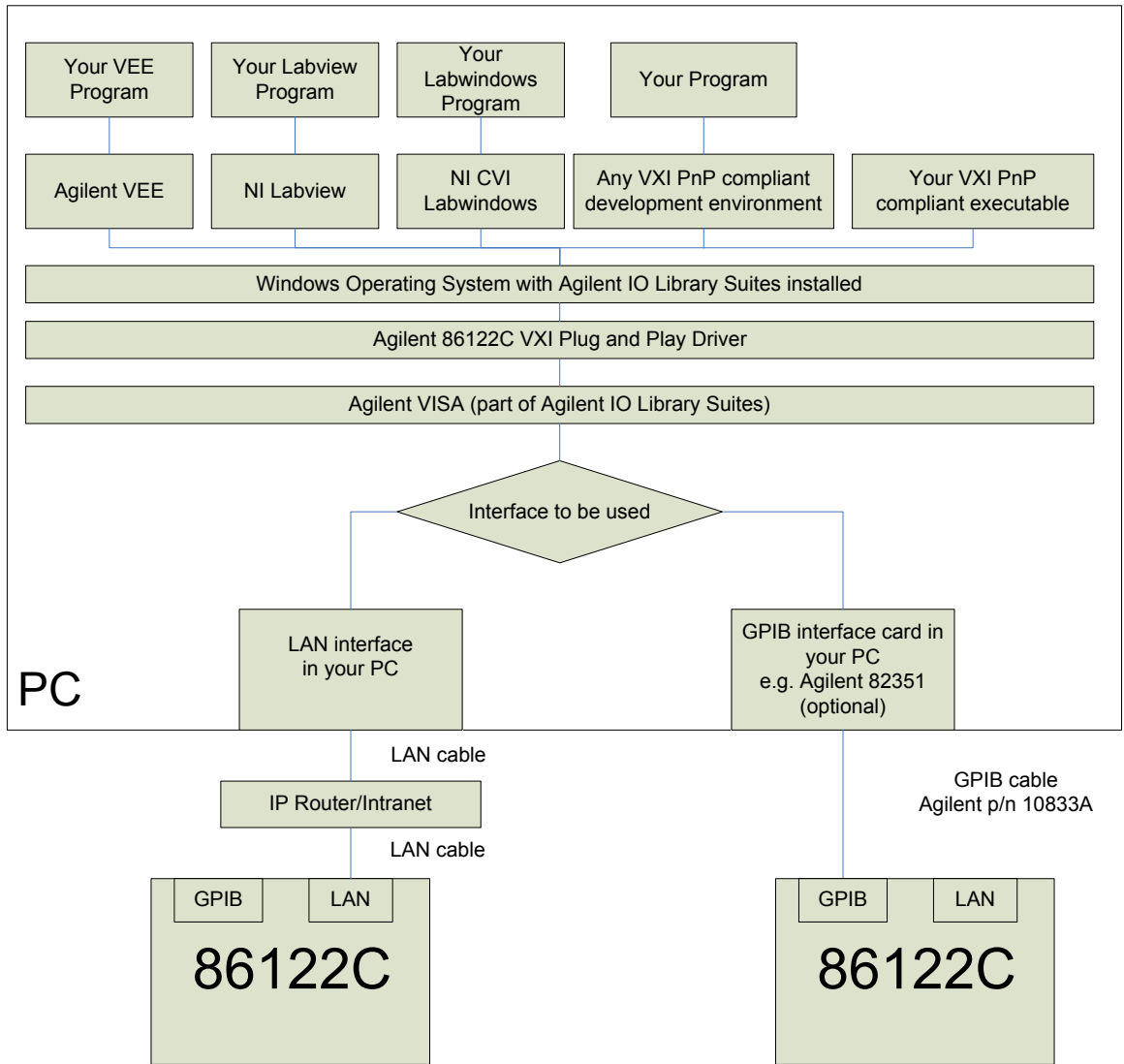
The SICL standard is similar to its successor VISA. For compatibility, the 86122C still supports SICL. Agilent recommends using VISA instead of SICL in new programs.

- **GPIB or IEEE 488.2**

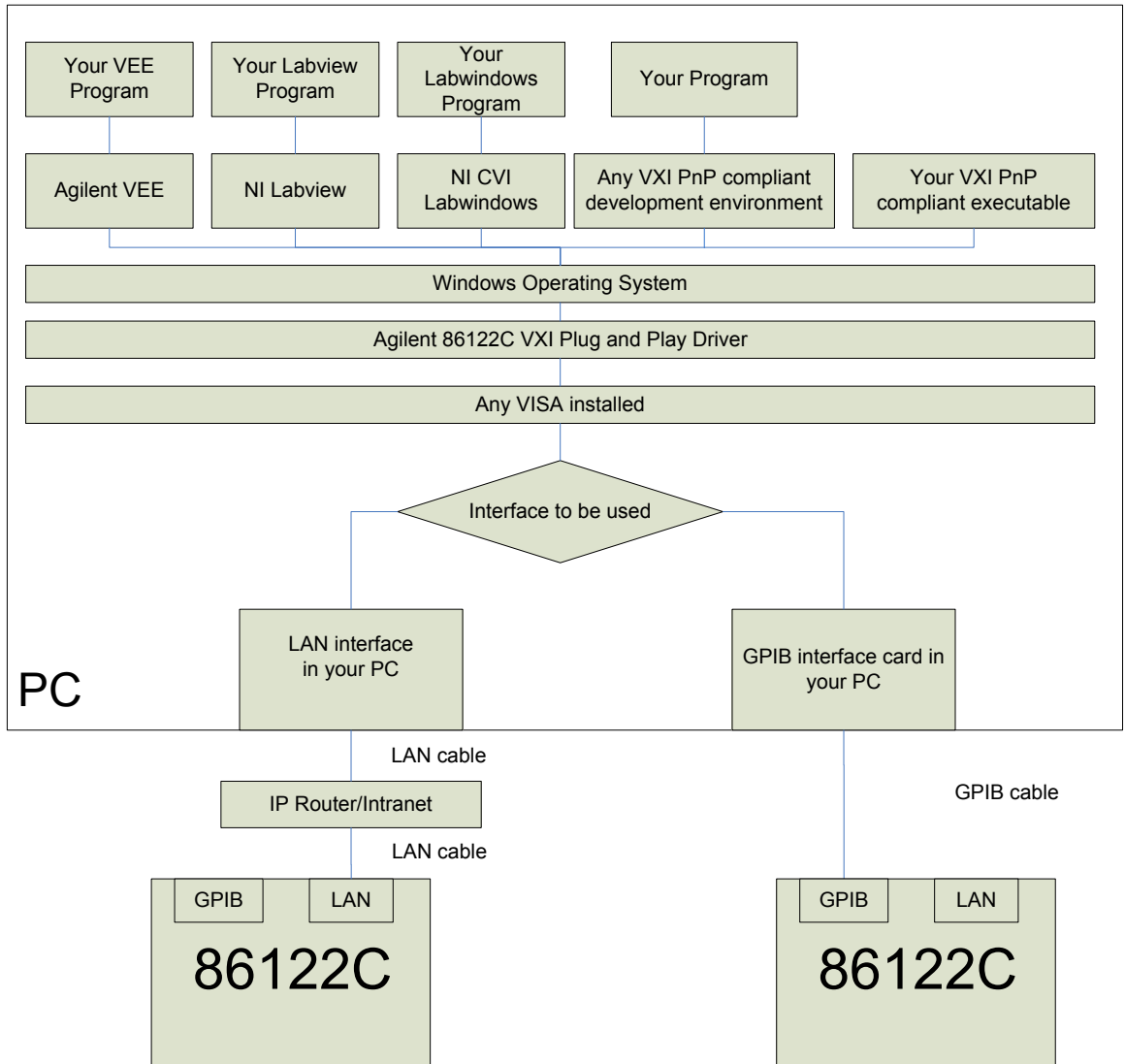
These standards provide communication to your 86122C over the GPIB bus only.

The following block diagrams show some example configurations. They also illustrate some hardware and software requirements to use the above standards. To control the 86122C programmatically, select first any valid combination that fits your environment best. Subsequent chapters show how configure your 86122C, and how to setup your controller (e.g. personal computer) to use the selected software standards.

## Controlling the 86122C using the 8612x VXI Plug and Play driver

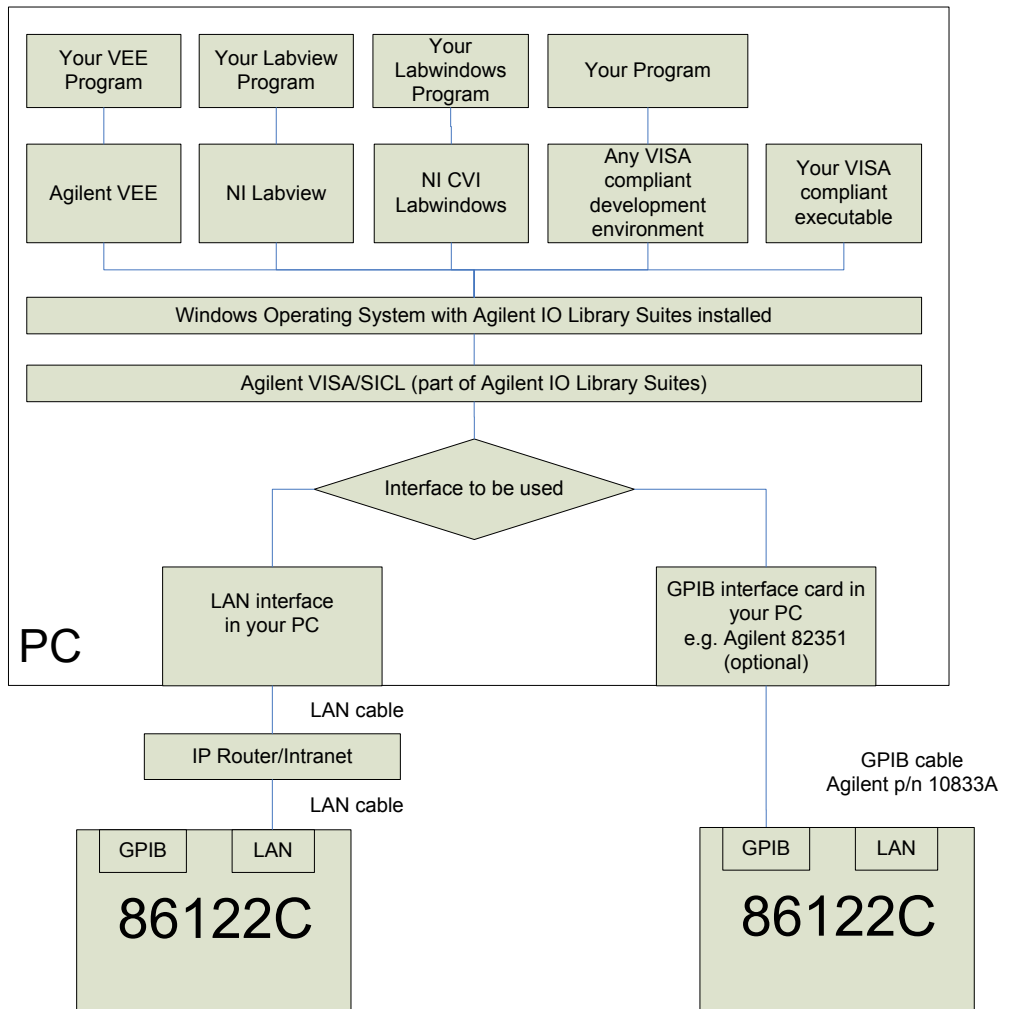


## Controlling the 86122C using the 8612x VXI Plug and Play driver

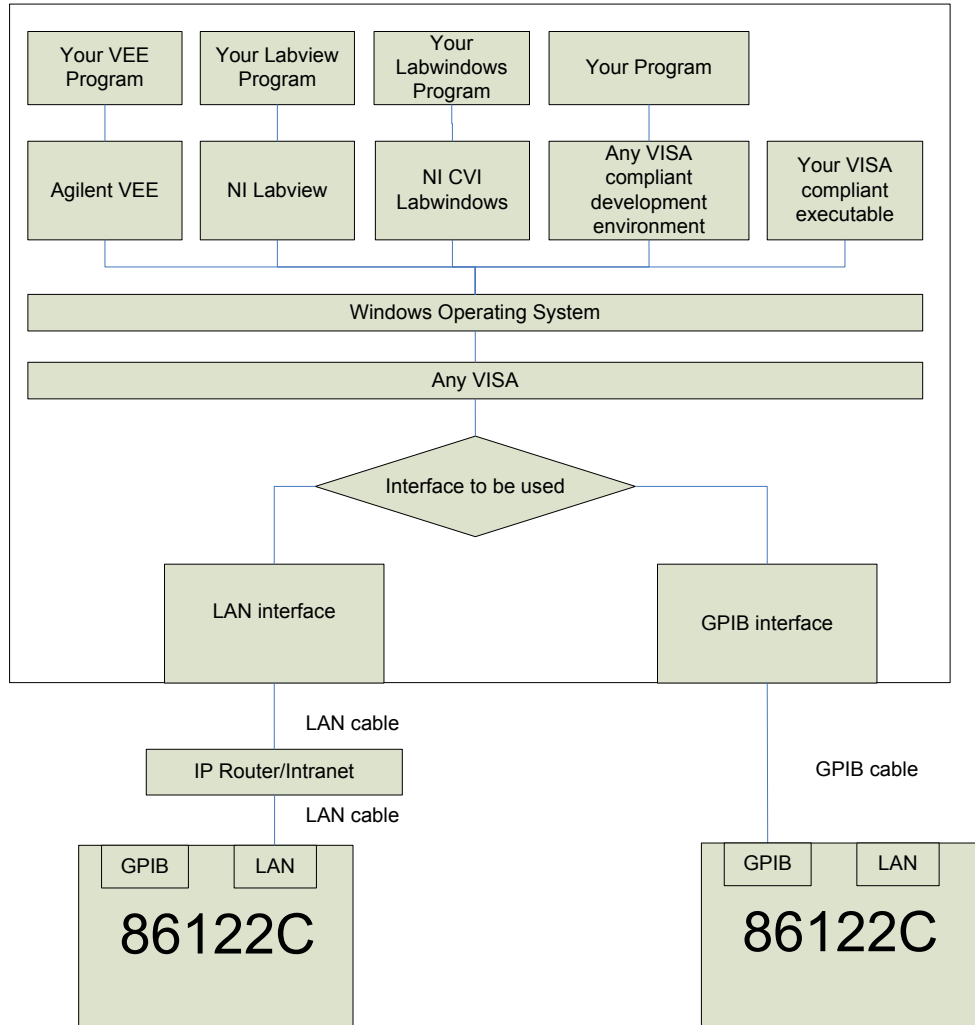




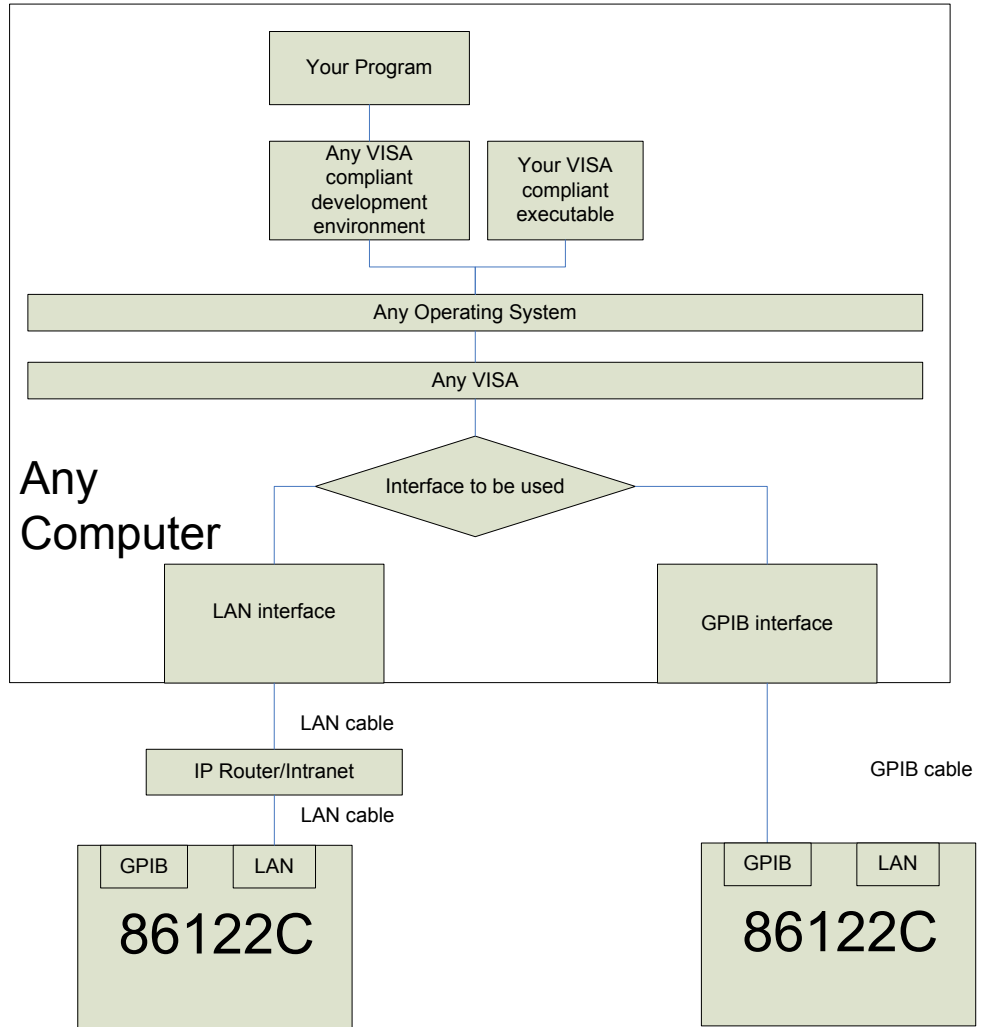
## Controlling the 86122C by SCPI language using the VISA or SICAL driver



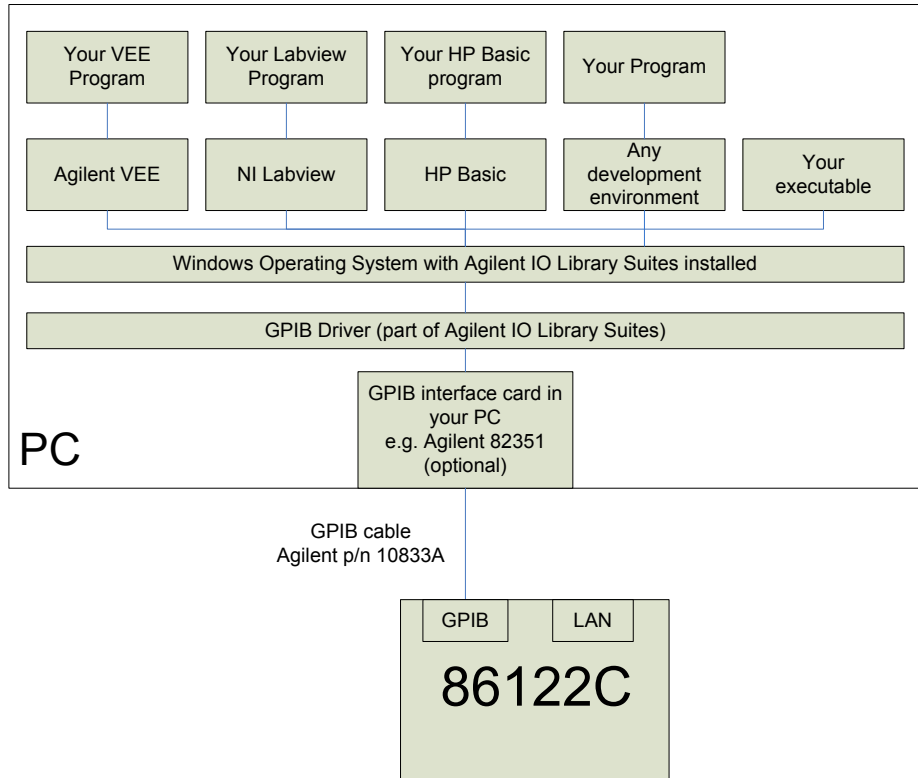
## Controlling the 86122C by SCPI language using VISA



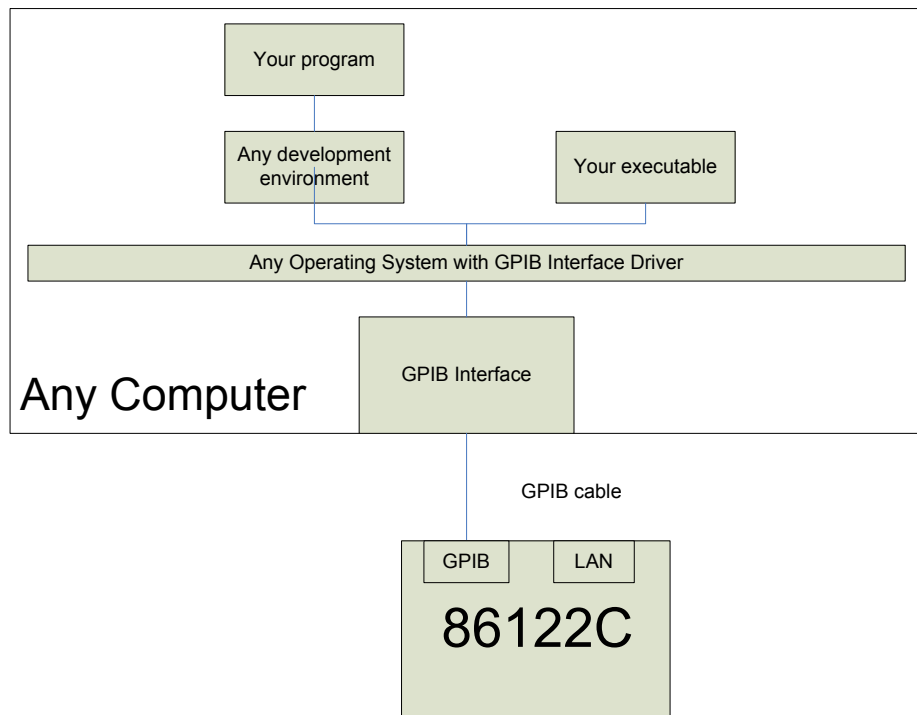
### Controlling the 86122C by SCPI language using VISA



### Controlling the 86122C by SCPI language using GPIB



### Controlling the 86122C by SCPI language using GPIB



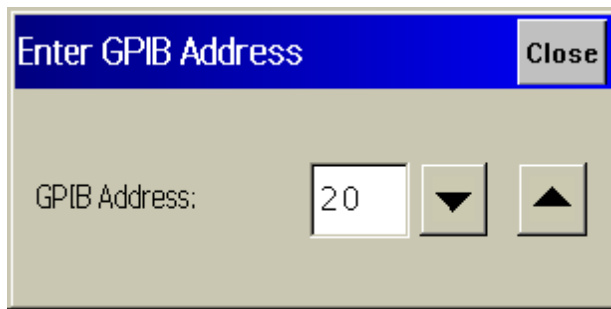
## b) Configure your 86122C

The 86122C has been prepared to support all the above mentioned software standards. Depending on the physical connection to be used, GPIB or LAN, only a few configuration settings are required.

**GPIB:** Set the GPIB Address of the 86122C using the menu **Setup | GPIB Setup**

This dialog box allows you to select the GPIB address for the multi-wavelength meter.

The default value is 20.



Although the default address is 20, addresses from 0 to 30 may be used if the default address is the same as another instrument's GPIB address. Address 21 is usually reserved for the computer interface Talk/Listen address and should not be used as the instrument address.

**LAN:** Once you integrate your 86122C into your existing LAN you can use all above mentioned standards. No software standard (i.e. VISA, SICL, etc.) specific settings are required.

- The Windows Computer Name has been factory set as 86122C-<last 4 digit of instrument serial number> (e.g. "86122C-2123").
- Ask your network administrator for support for integration of your 86122C into the LAN
- Use the menu File | Network | Getting Started... | Setting the Network Properties to get more help how to integrate the 86122C into your existing LAN
- Use the menu File | Network | Network Properties... to integrate the 86122C into your existing Network.

### c) Set up your controller

If your controller is a Windows based computer many required programs and drivers can be easily installed with the help of the Agilent N7700A Package Manager. The Package Manger is the file setup.exe in the root directory of the Photonic Application Suite CD that comes with the 86122C. You can get the N7700A Package Manager also from the internet under <http://www.agilent.com/find/N7700>. Click the link “Download Photonic Application Suite” and follow the instructions.

- **VXIplug&play**

On Windows based controllers start the Agilent Package Manager to install the latest version of the 8612x VXIplug&play driver.

**Note:** As shown in the above block diagrams, the 8612x VXIplug&play driver itself requires VISA to be installed (see below).

- **Standard Commands for Programmable Instruments (SCPI)**

There are no individual setup steps required on your controller to use SCPI. Of course you need to set up your controller for other software standards like VISA to enable it for sending SCPI commands to your 86122C.

- **VISA**

VISA is supported on many operating system platforms. See the documentation of the VISA provider.

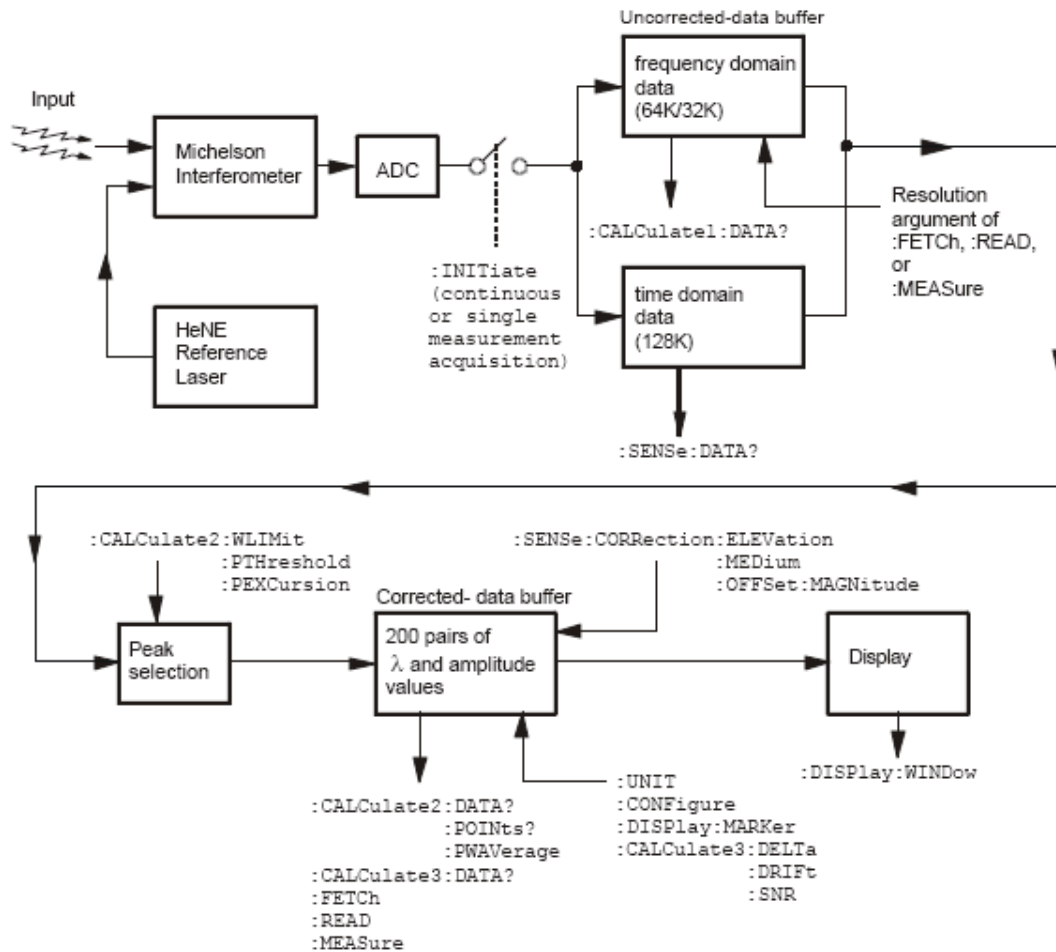
For use on Windows based computers you can use the Agilent Package Manager to install the Agilent IO Libraries Suite to install Agilent VISA.

- **SICL, GPIB IEEE 488.2**

For use on Windows based computers use the Agilent Package Manager to install the Agilent IO Libraries Suite to install the drivers for Agilent GPIB interfaces.

3. Control the 86122C programmatically

The following graphic shows a simplified programming model of the Agilent 86122C. This block diagram lists some of the available SCPI programming commands. Each command is placed next to the section of the instrument that it configures or queries data.



After collecting the uncorrected data, the Agilent 86122C searches the data for the first 1000 peak responses.



For WLIMit:OFF, searching starts at 1650 nm and progresses towards 1270 nm.  
 For WLIMit:ON, searching starts at WLIMit:START and progresses toward WLIMit:STOP.

These peak values are then placed into the corrected data buffer. Each peak value consists of an amplitude and wavelength measurement. Amplitude and wavelength correction factors are applied to this data.

Because there are so many different GPIB control platforms, the in-line code examples use a platform-independent pseudo code.

### **sendMessage xxxxx**

The “sendMessage xxxxx” indicates to send the text string xxxxx to the instrument. We recommend that you write your own function that adds error checking around the low level access routines.

- For VISA, the low level write routine is ivWrite().
- For NI 488, the low level routine is ibwrt(). For SICL, use iwrite().
- For HP Basic for windows, use OUTPUT.

### **receiveInteger**

The “receiveInteger” indicates to insert a call to a routine that reads an ASCII response from the instrument and converts it to an integer. The routine will return the integer value. Again, we recommend that you write your own function and do your own conversion from text to integer.

- For VISA, the low level read routine is ivRead().
- For NI 488, use ibrd(). For SICL use iread().
- For HP Basic for windows, use ENTER.

For more detailed information regarding GPIB, the IEEE 488.2 standard, or the SCPI standard, refer to the following books:

SCPI Consortium. *SCPI—Standard Commands for Programmable Instruments*, 1997.

International Institute of Electrical and Electronics Engineers. *IEEE Standard 488.1-1987, IEEE Standard Digital Interface for Programmable Instrumentation*. New York, NY, 1987.

International Institute of Electrical and Electronics Engineers. *IEEE Standard 488.2-1987, IEEE Standard Codes, Formats, Protocols and Common Commands For Use with ANSI/IEEE Std 488.1-1987*. New York, NY, 1987.



## Limitations of LAN connections

1. There may be a period of up to 5 seconds when you will be unable to open a LAN interface with the Agilent 86122C. 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 than 1-second."



## Initializing the Interface and the Instrument

You will notice that all programming examples in the online help send the commands to instrument address **720**. The programming examples assume the interface card is set to the interface select code “7” and the instrument GPIB address is set to “20” (default address). Set the output and input functions of your programming language to send the commands to this address. Alternatively, you can change the GPIB address in the [GPIB Setup dialog box](#), located in the Setup menu.

### Remote mode and front-panel lockout

When the instrument is in the remote operating mode, the remote indicator located on the lower right corner of the graphical display is lit.



Press  to return control of the instrument to the front panel.  and AC Power are the only front panel controls active when the instrument is in remote operation.



### Initializing the interface

To make sure the bus and all appropriate interfaces are in a known state, begin every program with an initialization statement. For example, a CLEAR command clears the interface buffer:

```
CLEAR 720 !
```



Programming examples are written in HP Basic.

When you use GPIB, CLEAR also resets the instrument's parser.

### Initializing the instrument

After the interface is cleared, you can initialize the instrument to a preset state with \*RST.

```
OUTPUT 720; "*RST" !
```

\*RST also sets the Agilent 86122C into the single measurement acquisition mode. The READ and MEASure data queries expect this measurement mode; therefore, the proper operation of these queries is ensured.

## Summary for Experienced Programmers

If you are familiar with GPIB instrument programming, terms like EOI, command synchronization, and SCPI should be familiar to you. If not, review [Programming Overview](#). This section is a summary to get your application operating quickly, focusing on ways in which the multi-wavelength meter differs from other instruments you may have encountered.

The following topics provide more information:

- [EOI Required](#)
- [Device Clear](#)
- [Single Sweep Mode](#)
- [Command Synchronization](#)
- [Maximum Query Rate](#)
- [Watching for Error Messages](#)

---

### 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 multi-wavelength meter sees the EOI, it closes the buffer and begins processing the commands. The multi-wavelength meter 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 HP Basic for Windows, you will need to enable EOI each time you open a session to the instrument. This is done using the ASSIGN @Mwm key words, and associating the END keyword with the session.

---

### Use device clear

The multi-wavelength meter 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 as needed while you debug your application.

The Selected Device Clear message is not a text message like \*CLS. It is an interface level command.

- ◆ For VISA, use viClear().
- ◆ For NI 488.2, use ibclr(). For HP SICL, use iclear().
- ◆ For HP Basic for windows, use CLEAR 720 or CLEAR @Mwm.

---

## 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 complicates the logic of the program. In general, unless an operator is tuning a device, it is best to turn continuous sweep off with:

```
INITiate:CONTInuous OFF
```

\*RST automatically turns continuous sweep off.

---

## Command synchronization

The multi-wavelength meter 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 :INITiate to take a single sweep, then wait for the sweep to finish before executing the next command.

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 HP Basic for Windows, use SPOLL 720 or SPOLL @Mwm.

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 do not need an explicit response from the instrument, \*WAI 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 HP 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.

For more information about error messages, see [Messages and Errors](#).

## SCPI Syntax Rules

Most of the commands used for controlling instruments on the GPIB are Standard Commands for Programmable Instruments (SCPI) commands.

The following topics will help you learn how to use SCPI commands in your programs.

- [Adding parameters to a command](#)
- [Querying data](#)
- [Combining commands from different subsystems](#)
- [Sending a command](#)
- [Combining commands in the same subsystem](#)
- [Sending common commands](#)
- [Notation conventions and definitions](#)
- [Using upper or lowercase letters](#)
- [Numbers](#)
- [Using short or long forms command names](#)
- [Program instruction terminator](#)
- [White space](#)

---

### Adding Command Parameters

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

```
OUTPUT 720;":INIT:CONT ON"
```

Notice the space between CONT and ON.

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

```
OUTPUT 720;":MEASure:SCALar:POWER:FREQuency? 1300, MAXimum"
```

---

### Combining Commands from Different Subsystems

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

```
OUTPUT 720;": CALCulate2:PEXCursion 12;:DISPlay:WINDow:GRAPh:STATus OFF"
```

---

### Combining Commands in the Same Subsystem

You can combine commands from the same subsystem as long as they are both on the same level in the subsystem hierarchy. Commands are separated with a semi-colon (;). For

example, the following two lines,

```
OUTPUT 720;":CALCulate2:PEXCursion 12"
OUTPUT 720;":CALCulate2:PTHReshold 20"
```

can be combined into one line.

```
OUTPUT 720;": CALCulate2:PEXCursion 12; PTHReshold 20"
```

Notice that the semicolon separates the two functions.

---

## Notation Conventions and Definitions

Review the following SCPI notation conventions.

Convention	Description
<>	Angle brackets indicate values entered by the programmer.
	Vertical bars, or "pipes", can be read as "or". They are used in syntax diagrams to separate alternative parameter options.
[]	Square brackets indicate that the enclosed items are optional. You can omit some keywords without changing the effect of the command. These optional, or default, keywords are used in many subsystems and are identified by brackets in syntax diagrams.
{ }	When several items are enclosed by braces, you can only select one of the items.
<integer>	An ASCII string representing an integer. This is defined by the IEEE 488.2 <NR1> format.
<real>	An ASCII string representing a real number. This is defined by the IEEE 488.2 <NR2> or <NRf> formats.

---

## Numbers

All numbers are expected to be strings of ASCII characters. Therefore, when you send 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 takes up 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
280E-1
28000m
```

0.028K  
28E-3K

If a measurement cannot be made, no response is given and an error is placed into the error queue. For example, the following string will timeout the controller and place a *Data stale or corrupt* error in the error queue because no power data has been received.

```
*RST
FETCh:POWer?
```

You can use the following multiplier suffixes:

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

---

### **Program Instruction Terminator**

The string of instructions sent to the instrument is executed after the instruction terminator is received. The terminator may be a new-line (NL) character, the End-Of-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.

---

### **Querying Data**

Data is requested from the instrument using a query. Queries can be used to find out how the instrument is currently configured or to get results of measurements made by the instrument, with the query actually activating the measurement. String responses are returned as upper-case 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 following query places the number of points in the data set in the output queue.

```
OUTPUT 720;":CALCulate2:POINts?"
```

In HP BASIC, the controller input statement

```
ENTER 720;Range
```

passes the value across the bus to the controller and places it in the variable Range. A newline character is appended to the response.



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 also 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, a semicolon separates each response.

## Sending a Command

It is easy to send a command to the instrument. Create a command string from the available commands, and place the string in your program language's output statement. For commands other than common commands, be sure to include a colon before the subsystem name. For example, the following string places the cursor on the peak laser line and returns the power level of this peak:

```
OUTPUT 720;":MEAS:SCAL:POW? MAX"
```

## Sending Common Commands

If a subsystem has been selected and the instrument receives a common command, the instrument remains in the selected subsystem. For example, if the following program message is sent to the instrument, the DISPLAY subsystem remains selected.

```
"DISPLAY:MARKer:MAXimum:LEFT;*CLS;DISPlay:MARKer:MAXimum:RIGHT"
```

If some other type of command is received within a program message, you must reenter the original subsystem after the command.

## Using Upper or Lowercase Letters

Program headers can be sent using any combination of uppercase or lowercase ASCII

characters. However, instrument responses are always returned in uppercase.

---

## Using Short or Long Forms Command Names

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

```
OUTPUT 720;":MEASure:SCALar:POWer? MAXimum"
```

This is the short form of the same command:

```
OUTPUT 720;":MEAS:SCAL:POW? MAX"
```

Programs written in long form are easily read and 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.

Use mnemonics to create a short form of a command from the long form. 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* applied if the length of the keyword is exactly four characters.

Examples of long and equivalent short forms:

Long Form	Equivalent Short Form
DISPlay	DISP
DATA	DATA
SYSTem	SYST
ERRor	ERR

---

## White Space

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



## Initializing the Interface and the Instrument

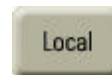
You will notice that all programming examples in the online help send the commands to instrument address **720**. The programming examples assume the interface card is set to the interface select code “7” and the instrument GPIB address is set to “20” (default address). Set the output and input functions of your programming language to send the commands to this address. Alternatively, you can change the GPIB address in the [GPIB Setup dialog box](#), located in the Setup menu.

### Remote mode and front-panel lockout

When the instrument is in the remote operating mode, the remote indicator located on the lower right corner of the graphical display is lit.



Press  to return control of the instrument to the front panel.  and AC Power are the only front panel controls active when the instrument is in remote operation.



### Initializing the interface

To make sure the bus and all appropriate interfaces are in a known state, begin every program with an initialization statement. For example, a CLEAR command clears the interface buffer:

```
CLEAR 720 !
```



Programming examples are written in HP Basic.

When you use GPIB, CLEAR also resets the instrument's [parser](#).

### Initializing the instrument

After the interface is cleared, you can initialize the instrument to a preset state with \*RST.

```
OUTPUT 720; "*RST" !
```

\*RST also sets the Agilent 86122C into the single measurement acquisition mode. The READ and MEASure data queries expect this measurement mode; therefore, the proper operation of these queries is ensured.

## Non-Sequential Commands

The Agilent 86122C normally processes its remote commands sequentially. The instrument waits until the actions specified by a particular command are completed before reading and executing the next command. However, there are a few non-sequential commands where this is not true. Non-sequential commands do not finish executing before the next command is interpreted.

The following is a list of non-sequential commands:

:CALCulate1:TRANSform:FREQuency:POINts

:CALCulate2:PEXCursion

:CALCulate2:PTHReshold

:CALCulate2:WLIMit:STARt:FREQuency

:CALCulate2:WLIMit:STARt:WAVelength

:CALCulate2:WLIMit:STARt:WNUMber

:CALCulate2:WLIMit:STOP:FREQuency

:CALCulate2:WLIMit:STOP:WAVelength

:CALCulate2:WLIMit:STOP:WNUMber

:CALCulate3:SNR:AUTO

:SENSe:CORRection:ELEVation

:INITiate:CONTInuous

:INITiate[:IMMEDIATE]

:HCOPY[:IMMEDIATE]

\*TRG



The command :SENSe:CORRection:ELEVation provides backwards compatibility with previous instruments and is not relevant to the 86122C.

The following additional commands are also non-sequential commands if CALCulate3:SNR:AUTO is set to OFF:

:CALCulate3:REFerence:FREQuency

:CALCulate3:REFerence:WAVelength

:CALCulate3:REFerence:WNUMber

Non-sequential commands can reduce overall execution time of programs, but can also be the source of annoying errors. Forcing the Agilent 86122C to wait for non-sequential commands to finish eliminates this problem.

Always use \*OPC or \*WAI with non-sequential commands to ensure your programs execute properly. For example, suppose that you wanted to set the elevation correction value and then

send an :INIT:IMM command. The following programming fragment results in an error “–213 *Init ignored*”. This occurs because the :ELEVation command causes the recalculation of the data which is like sending the :INIT:IMM command. When the actual :INIT:IMM is sent, the error occurs because the command is already in progress.

```
OUTPUT 720;”:INIT:IMM”
```

```
OUTPUT 720;”:SENSe:CORRection:ELEVation 1000”
```

```
OUTPUT 720;”:INIT:IMM”
```

Use an \*OPC? query to ensure that the :ELEVation command has completed as shown in the following lines:

```
OUTPUT 720;”:INIT:IMM”
```

```
OUTPUT 720;”:SENSe:CORRection:ELEVation 1000”
```

```
OUTPUT 720;”:*OPC?”
```

```
ENTER 720;Response$
```

```
OUTPUT 720;”:INIT:IMM”
```

Or, the \*WAI command could be used:

```
OUTPUT 720;”:INIT:IMM”
```

```
OUTPUT 720;”:SENSe:CORRection:ELEVation 1000”
```

```
OUTPUT 720;”:*WAI?”
```

```
OUTPUT 720;”:INIT:IMM”
```

## Status Registers

The Agilent 86122C provides four registers that you can query to monitor the instrument's condition. These registers allow you to determine the following:

- ◆ Status of an operation
- ◆ Availability of the measured data
  - Reliability of the measured data

The following topics provide more information about the status registers.

- [Status Byte](#)
- [Standard Event Enable](#)
- [OPERation Status](#)
- [QUEStionable Status](#)
- [Enabling Register Bits with Masks](#)

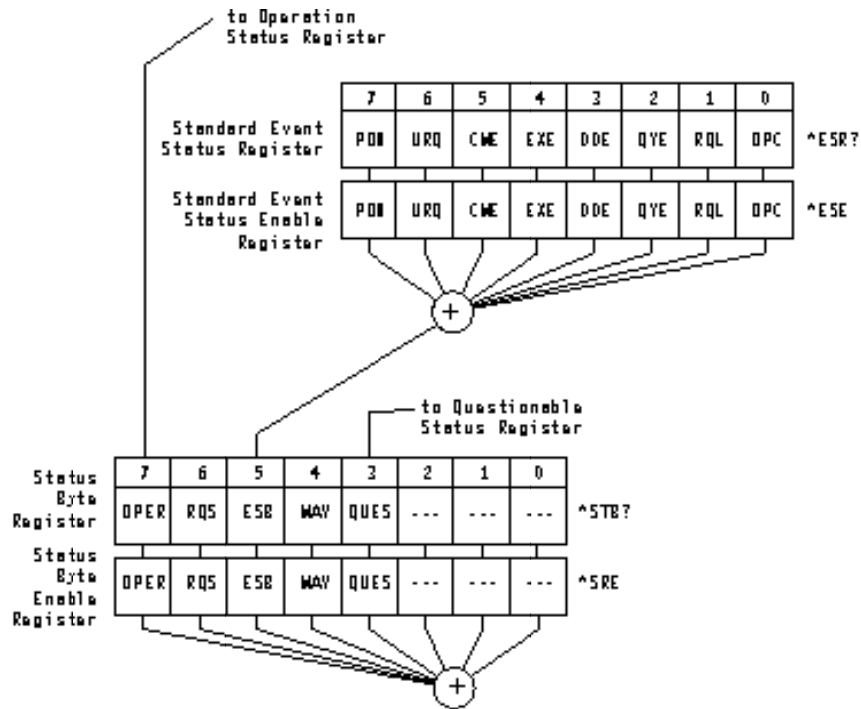
---

### Status Byte Register

The Status Byte Register monitors the status of the other three registers. This register contains summary bits that monitor activity in the other status registers and queues. The Status Byte Register's bits are set and cleared by the presence and absence of a summary bit from other registers or queues. Notice in the following figure that the bits in the Standard Event Status, OPERation Status, and QUEStionable Status registers are "or'd" to control a bit in the Status Byte Register.

If a bit in the Status Byte Register goes high, you can query the value of the source register to determine the cause.

The Status Byte Register can be read using either the [\\*STB? command](#) or the GPIB serial poll command. Both commands return the decimal-weighted sum of all set bits in the register. The difference between the two methods is that the serial poll command reads bit 6 as the Request Service (RQS) bit and clears the bit which clears the SRQ interrupt. \*STB? reads bit 6 as the Master Summary Status (MSS) and does not clear the bit or have any effect on the SRQ interrupt. The value returned is the total bit weights of all of the bits that are set at the present time.



The following table defines the bits in the Statue Byte Register.

Bit	Bit	Weight Condition
0	1	Not Used
1	2	Not Used
2	4	Error Queue Status
3	8	Questionable Status
4	16	Message Available (MAV)
5	32	Standard Event Status (ESB)
6	64	Master Summary Status (MSS)
7	128	Operation Status

---

## Standard Event Enable Status Register

The Standard Event Enable Status Register is the standard IEEE 488.2 register. This register monitors the following instrument status events:

- ◆ OPC - Operation Complete
- ◆ RQC - Request Control
- ◆ QYE - Query Error
- ◆ DDE - Device Dependent Error
- ◆ EXE - Execution Error
- ◆ CME - Command Error
- ◆ URQ - User Request
  - PON - Power On

When one of these events occur, the corresponding bit in the register is set. If the bits are enabled in the Standard Event Status Enable Register, the bits set in this register generate a summary bit to set bit 5 (ESB) in the Status Byte Register.

The contents of the Standard Event Status Register can be read and the register cleared by sending the \*ESR? query. The value returned is the total bit weights of all of the bits that are set at the present time.

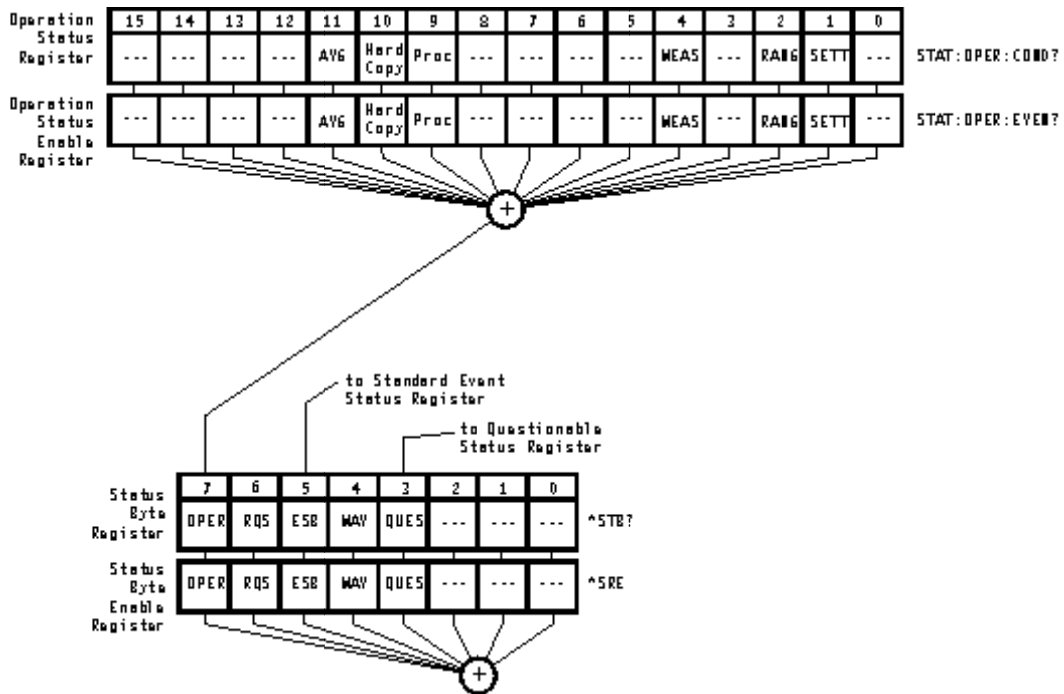
---

## OPERation Status Register

The OPERation Status Register contains the bits that report on the instrument's normal operation. You can query the value of the OPERation Status Register using commands in the STATus subsystem.

The STATus subsystem also has transition filter software that gives you the ability to select the





The following table defines the bits in the OPERAtion Status Register.

Bit	Definition
0	Not used.
1	<b>Setting</b> indicates that the instrument is waiting for the motor to reach the proper position before beginning data acquisition.
2	<b>Ranging</b> indicates that the instrument is currently gain ranging.
3	Not used.

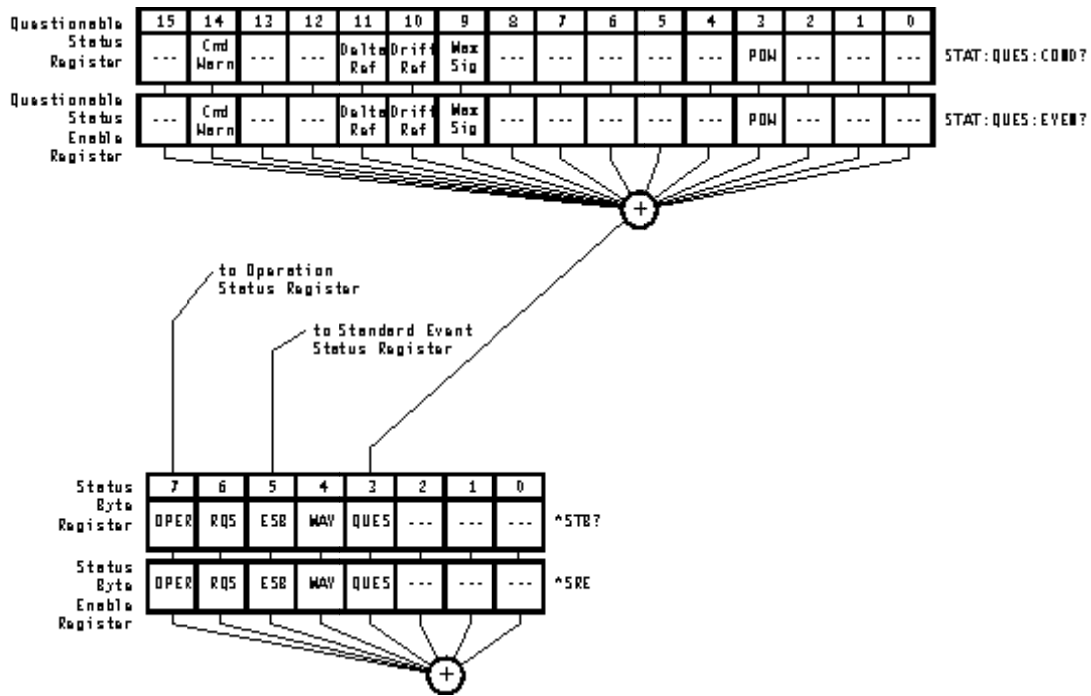
4	<b>Measuring</b> indicates that the instrument is currently making a measurement.
5 through 8	Not used.
9	<b>Processing</b> indicates that the instrument is currently processing the data acquired.
10	<b>Hardcopy</b> indicates that the instrument is currently printing the data to the parallel port.
11	<b>Averaging</b> indicates that the instrument is in the process of averaging the noise for the signal-to-noise ratio calculation.
12 through 15	Not used.

---

### Bits in Questionable Status Register

The Questionable Status Register contains the bits that report on the instrument's normal operation. You can query the value of the QUESTIONABLE Status Register using commands in the STATUS subsystem.

The STATUS subsystem also has transition filter software that gives you the ability to select the logic transitions that sets bits in the QUESTIONABLE Status Register. For example, you can define the POWER bit of the QUESTIONABLE Status register to report an event when the condition transitions from false to true. This is a positive transition. You can also specify a negative transition where the bit is set when the condition transitions from true to false.



The following table defines the bits in the OPERAtion Status Register.

Bit	Definition
0, 1, 2	Not used.
3	<b>Power</b> indicates that the instrument is measuring too high of a power.
4	<b>Over Temperature</b> indicates the HeNe laser is over temperature. Shut down instrument. If this condition persists, contact Agilent Service Center.
5 through 8	Not used.
9	<b>Maximum Signals</b> indicates that the instrument

	has found the maximum number of signals.
10	<b>Drift Reference</b> indicates that during one or more data acquisitions, the number of signals detected differed from the number of reference signals.
11	<b>Delta Reference</b> indicates that there is no delta reference signal.
12	<b>Not Stabilized</b> indicates that the HeNe laser is not stabilized. A measurement cannot be made until the laser is stabilized.
13	Not used
14	<b>Command Warning</b> indicates that the instrument has received some extra, unexpected parameters for one of the measurement functions.
15	Not used.

---

### Enabling Register Bits with Masks

Several masks are available which you can use to enable or disable individual bits in each register. For example, you can disable the Hardcopy bit in the OPERation Status Register so that even though it goes high, it can never set the summary bit in the status byte high.

Use \*SRE to set or query the mask for the Status Byte Register.

The masks for the OPERation Status and QUEStionable Status registers are set and queried using the STATus subsystem's ENABle commands.

Use \*ESE to set or query the mask for the Standard Event Status Register.

The \*CLS common command clears 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 example, suppose your application requires an interrupt whenever any type of error occurs. The error related bits in the Standard Event Status Register are bits 2 through 5. The sum of the decimal weights of these bits is 60. Therefore, you can enable any of these bits to generate the summary bit by sending \*ESE 60.

Whenever an error occurs, it sets one of these bits in the Standard Event Status Register. Because the bits are all enabled, a summary bit is generated to set bit 5 in the Status Byte Register.

If bit 5 (ESB) in the Status Byte Register is enabled (via the \*SRE command), an SRQ service request interrupt is sent to the external computer.

Standard Event Status Register bits that are not enabled still respond to their corresponding conditions (that is, they are set if the corresponding event occurs). However, because they are not enabled, they do not generate a summary bit to the Status Byte Register.

## Queues

There are two queues in the instrument: the output queue and the error queue. You can query the values in either queue.

### 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 Register. The method used to read the Output Queue depends upon the programming language and environment. For example, when using HP BASIC, you can read the output queue 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).

The error queue is read with SYSTEM:ERROR?. Executing this query reads and 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."

See [Error Messages](#) for more information on reading the error queue.

## SCPI Programming Commands

The programming commands in this instrument can be placed into one of three types: measurement instruction commands, common commands, and subsystem commands.

See the following command types for more information about the commands in the selected group.

- [Measurement Instructions](#)
- [Common Commands](#)
- [Subsystem Commands](#)

For an overall list of available commands, see the following:

- [List of Commands](#)

There are a variety of measurements that the Agilent 86122C can perform. The following table shows an example of these different measurements and their associated programming commands. There may be more than one method to obtain the desired data. Refer to the individual command information for the correct command syntax.

<b>Desired Measurement</b>	<b>Configuration Command</b>	<b>Data Query</b>
Wavelength (nm)	CONFigure, FETCh, READ, and MEASure	MEASure:ARRay:POWer:WAVelength?
Frequency (THz)	CONFigure, FETCh, READ, and MEASure	MEASure:ARRay:POWer:FREQuency?
Wave number	CONFigure, FETCh, READ, and MEASure	MEASure:ARRay:POWer:WNUMber?
Power (W, dBm)	CONFigure, FETCh, READ, and MEASure	MEASure:ARRay:POWer:?
Average Wavelength, Wave number, or Frequency	CALCulate2:PWAverage:STATe	CALCulate2:DATA?
Total Power (W, bDm)	CALCulate2: PWAverage:STATe	CALCulate2:DATA?
Fabry-Perot Laser	CALCulate3:FPERot	FPERot[::STATe]?
Laser-Line Separation	CALCulate3:DELTA:REFerence	CALCulate3:DATA?
Laser-Line Drift	CALCulate3:DRIFT: STATe	CALCulate3:DATA?
Signal-to-Noise Ratio	CALCulate3:SNR:STATe	CALCulate3:DATA?
Signal-to-Noise Ratio Average	CALCulate3:ASN:STATe	CALCulate3:DATA?
Time-Domain Data	CALCulate1:TRANsform :FREQuency:POINts	SENSe: DATA?

Uncorrected Frequency  
Domain Data

CALCulate1:TRANSform  
:FREQUENCY:POINTS

CALCulate1:DATA?



## Measurement Instructions

Use measurement instructions to perform measurements and return the desired results to your computer. The Agilent 86122C uses four basic measurement instructions: CONFigure, FETCh, READ, and MEASure. Measurement instructions have the following hierarchy:

```
{ :MEASure | :READ[?] | :FETCh[?] | :CONFigure[?] }
```

```
{ :ARRay | [:SCALar] }
```

```
:POWer[?]
```

```
:FREQuency[?] :WAVelength[?] :WNUMber[?]
```



MEASure and READ are identical to combining the following commands:

```
:MEASure :ABORT;:CONFigure;:READ
:READ :ABORT;:INITiate:IMMediate;:FETCh
```

To perform a measurement, add a POWer function to the measurement instruction. The POWer functions select power, frequency, wavelength, or wave number measurements.

You can specify whether FETCh, READ, or MEASure returns a single value (:SCALar) or multiple values (:ARRay). The following example specifies SCALar data:

```
:MEASure:SCALar:POWer:WAVelength? MAX
```

See the four basic measurement instructions for more information about the selected instruction group and its associated commands.

[CONFigure](#)

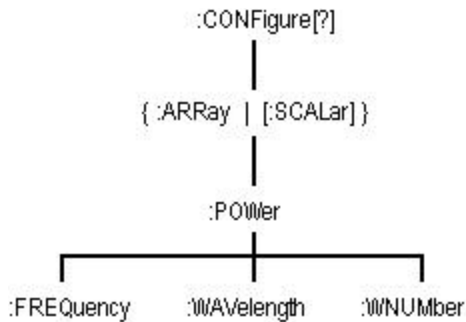
[MEASure](#)

[FETCh](#)

[READ](#)

## :CONFigure Command

Use CONFigure commands to change measurement settings without taking a measurement. The CONFigure commands place the instrument in the Multi-WL or Peak WL display mode. The CONFigure commands have the following hierarchy:



The CONFigure query returns the last configuration setup by CONFigure. The instrument returns a string that represents the current configuration of the instrument. The returned string is in the short command form, and returned values are in base units as in the following examples.

Command :CONFigure:SCALar:POWer:WAVelength 1300NM, MAX

Query "POW:WAV 1.300000e-6,0.01"

Response



The 1300NM and resolution (MAX) values track the actual instrument settings and input signals. Notice that the quotation marks are part of the returned string.

Command :CONFigure:SCALar:POWer -9 dBm

Query "POW 1.196978e-4,DEF"

Response



The power values are always returned in watts.

According to the SCPI command reference, :ARRay causes an instrument to take multiple measurements. The <size> parameter indicates the number of measurements to take. However, with the Agilent 86122C, the ARRArray command refers to the measurements performed for one measurement sweep; this will result in an array of measured signals. Therefore the <size> parameter will not apply to this command and the instrument will ignore any <size> parameter sent. A syntax error will not be generated if a <size> parameter is sent.

See the following CONFigure commands for more information about the selected command.



The command trees for each of the four basic [measurement instructions](#) are identical.

● [CONFigure {\[:ARRay\] | \[:SCALar\]} :POWer](#) ● [CONFigure {\[:ARRay\] | \[:SCALar\]} :POWer:WAVelength](#)

● CONFigure {:ARRay | [:SCALar]}  
:POWer:FREQuency

● CONFigure{:ARRay | [:SCALar]}  
:POWer:WNUMber

**:CONFigure{:ARRay | [:SCALar]} :POWER**

The command trees for each of the four basic [measurement instructions](#) are identical.

**Command**

:POWER [&lt;expected\_value&gt;[,&lt;resolution&gt;]]

Returns amplitude values.

Used With	<expected_value>	<resolution>
SCALar	optional	optional
ARRay	Ignored	optional

When used with :SCALar, a single value is returned. If the tabular display is on, the display is placed in the single-wavelength mode. The marker is then placed on the signal having a power level that is closest to the <expected\_value> parameter.

When used with :ARRay, an array of amplitudes is returned. If the tabular display is on, the display is placed in the list-by-power mode.

Returned values are in the current power units. Wavelength units are not affected.

**<expected\_value>  
Constants**

MAXimum  
The highest power signal

MINimum  
The lowest power signal

DEFault  
The signal at the current marker position

**<resolution>  
Constants**

MAXimum  
0.01 resolution

MINimum  
0.001 resolution

EXTended1  
0.0001 resolution

DEFault  
Current resolution

**Examples**

:CONF:ARR:POW

:CONF:SCAL:POW -10 dBm

The following line is an example of a returned string when :CONFigure:SCALar:POWER MAXimum is sent:

-5.88346500E+000

If six laser lines are located and :CONFigure:ARRay:POWer is sent, the following string could be returned. Notice that the first returned number indicates the number of laser-line values returned in the query. The measurement units are in dBm.

6,-1.37444400E+001,-1.10996100E+001,-9.62396600E+000,  
-7.94024500E+000,-7.01303200E+000,-1.04536200E+001

**:CONFigure{ :ARRay | [:SCALar]} :POWER:FREQuency**

The command trees for each of the four basic [measurement instructions](#) are identical.

**Command** :POWER:FREQuency [<expected\_value>[,<resolution>]]

Returns frequency values.

Used With	<expected_value>	<resolution>
SCALar	optional	optional
ARRay	Ignored	optional

When used with :SCALar, a single value is returned. If the tabular display is on, the display is placed in the single-wavelength mode. The marker is then placed on the signal having a frequency that is closest to the <expected\_value> parameter. Default units for the <expected\_value> parameter are Hz.

When used with :ARRay, an array of frequencies is returned. If the tabular display is on, the display is placed in the list-by-wavelength mode.

The <resolution> parameter sets the number of points for :CALCulate1:TRANSform:FREQuency:POINts. This parameter provides backwards compatibility with previous instruments. It does not affect the resolution or speed of :CONFigure. This parameter is a unitless number whose value will be limited to either 0.01, 0.001, or 0.0001.

Returned values are in Hz. Displayed units are shown in THz. Power units are not affected.



When using :ARRay, the <expected\_value> is ignored. However, this argument must be present if the resolution argument is specified. You can use DEF to specify the default value.

<expected_value>	MAXimum
Constants	The highest frequency signal
	MINimum
	The lowest frequency signal
	DEFault
	The signal at the current marker position
<resolution>	MAXimum
Constants	0.01 resolution
	MINimum

0.001 resolution

EXTended1

0.0001 resolution

DEFault

Current resolution

### Examples

```
:CONF:ARR:POW:FREQ DEF, MAX
```

```
:CONF:SCAL:POW:FREQ 230.8THZ, MAX
```

The following line is an example of a returned string when :CONFigure:SCALar:POWer:FREQuency MAXimum is sent:

```
+1.94055176E+014
```

If six laser lines are located and

:CONFigure:ARRay:POWer:FREQuency is sent, the following string is an example of the returned data. Notice that the first returned number indicates the number of laser-line values returned in the query.

```
6,+1.93050000E+014,+1.93250900E+014,+1.93452000E+014,  
+1.93653000E+014,+1.93854100E+014,+1.94055100E+014
```

**:CONFigure{ :ARRay | [:SCALar]} :POWer:WAVelength**

The command trees for each of the four basic [measurement instructions](#) are identical.

**Command** :POWer:WAVelength [<expected\_value>[,<resolution>]]

Returns wavelength values.

Used With	<expected_value>	<resolution>
SCALar	optional	optional
ARRay	Ignored	optional

When used with :SCALar, a single value is returned. If the tabular display is on, the display is placed in the single-wavelength mode. The marker is then placed on the signal having a wavelength that is closest to the <expected\_value> parameter. Default units for the <expected\_value> parameter are in meters.

When used with :ARRay, an array of wavelengths is returned. If the tabular display is on, the display is placed in the list-by-wavelength mode.

The <resolution> parameter sets the number of points for :CALCulate1:TRANSform:FREQuency:POINts. This parameter provides backwards compatibility with previous instruments. It does not affect the resolution or speed of :CONFigure. This parameter is a unitless number whose value will be limited to either 0.01, 0.001, or 0.0001.

Returned values are in meters. Displayed units are shown in nanometers. Power units are not affected.



When using :ARRay, the <expected\_value> is ignored. However, this argument must be present if the resolution argument is specified. You can use DEF to specify the default value.

<expected_value>	MAXimum
Constants	The highest wavelength signal
	MINimum
	The lowest wavelength signal
	DEFault
	The signal at the current marker position
<resolution>	MAXimum
Constants	0.01 resolution
	MINimum



0.001 resolution

EXTended1

0.0001 resolution

DEFault

Current resolution

### Examples

```
:CONF:ARR:POW:WAV DEF, MAX
```

```
:CONF:SCAL:POW:WAV 1300NM, MAX
```

The following line is an example of a returned string when :CONFigure:SCALar:POWer:WAVelength MAXimum is sent:

```
+1.5529258E-006
```

If six laser lines are located and

:CONFigure:ARRay:POWer:WAVelength is sent, the following string could be returned. Notice that the first returned number indicates the number of laser-line values returned in the query.

```
6,+1.54488100E-006,+1.54648400E-006,+1.54809000E-006,+1.54969900E-006,+1.55131100E-006,+1.55292600E-006
```

**:CONFigure{ :ARRay | [:SCALAr]} :POWer:WNUMber**

The command trees for each of the four basic [measurement instructions](#) are identical.

**Command** :POWer:WNUMber [<expected\_value>[,<resolution>]]

Returns a wave number value.

Used With	<expected_value>	<resolution>
SCALAr	optional	optional
ARRay	Ignored	optional

When used with :SCALAr, a single value is returned. If the tabular display is on, the display is placed in the single-wavelength mode. The marker is then placed on the signal having a wave number that is closest to the <expected\_value> parameter. Default units for the <expected\_value> parameter are in /m.

When used with :ARRay, an array of wave numbers is returned. If the tabular display is on, the display is placed in the list-by-wavelength mode.

The <resolution> parameter sets the number of points for :CALCulate1:TRANSform:FREQuency:POINts. This parameter provides backwards compatibility with previous instruments. It does not affect the resolution or speed of :CONFigure. This parameter is a unitless number whose value will be limited to either 0.01, 0.001, or 0.0001.

Returned values are in inverse meters. Displayed units are shown in inverse centimeters. Power units are not affected.



When using :ARRay, the <expected\_value> is ignored. However, this argument must be present if the resolution argument is specified. You can use DEF to specify the default value.

<expected_value>	MAXimum
Constants	The laser line having the largest wave number
	MINimum
	The laser line having the smallest wave number
	DEFault
	The signal at the current marker position
<resolution>	MAXimum
Constants	0.01 resolution

MINimum  
0.001 resolution

EXTended1  
0.0001 resolution

DEFault  
Current resolution

**Examples**

```
:CONF:ARR:POW:WNUM DEF, MAX
```

```
:CONF:SCAL:POW:WNUM 6451, MAX
```

If :CONFigure:SCALar:POWer:WNUMber 6451 is sent, and a 1550 nm laser line is present, the following response would be returned to the computer:

```
+6.45286262E+005
```

Notice that the returned units are /m. If six laser lines are measured and :CONFigure:ARRay:POWer:WNUMber is sent, the following response is returned. Notice that the first returned number indicates the number of laser-line values returned in the query.

```
6,+6.43945300E+005,+6.44615600E+005,+6.45286300E+005  
+6.45957000E+005,+6.46627900E+005,+6.47298400E+005
```

## :FETCh Command

Use the FETCh command to return data from previously performed measurements. However, this command does not initiate the collection of new data. This makes the FETCh command especially useful when characterizing transient data because new data is not collected. Because FETCh does not configure the instrument or acquire new input data, you can use FETCh repeatedly on the same set of acquired data. For example, use two FETCh commands to return wavelength values and then power values for the same measurement as shown in the following program fragment

```
OUTPUT 720;”:INIT:CONT OFF;”
OUTPUT 720;”:CONF:ARR:POW MAX”
OUTPUT 720;”:INIT:IMM”
OUTPUT 720;”:FETC:ARR:POW?”
ENTER 720:powers$
OUTPUT 720;”:FETC:ARR:POW:WAV?”
ENTER 720:wavelengths$
```

In the above example, the data in the power and wavelength arrays are returned in the same order so that powers can be matched to wavelengths. You can also send a MEASure command followed by a FETCh command.

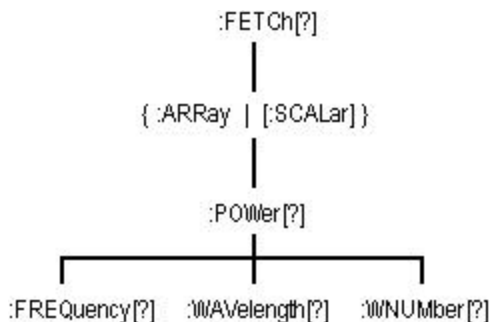


FETCh does not reconfigure the display. For example, if the display is in the Peak WL mode, sending :FETCh:ARRay does not configure the display to the Multi-WL mode even though an array of data is returned to the computer.



If no arguments are specified for :FETCh?, the previously specified arguments will be used. A common programming error occurs when :FETCh is used after an \*RST command. This generates error number – 230, “*Data corrupt or stale*”. In this instance, you must send :INITiate:IMMEDIATE after the \*RST command and before:FETCh to capture a new array of measurement data.

The FETCh commands have the following hierarchy:



See the following FETCh commands for more information about the selected command.



The command trees for each of the four basic measurement instructions are identical.

- FETCh{:ARRay | [:SCALar]}:POWer?
- FETCh{:ARRay | [:SCALar]}:POWer:WAVelength?
- FETCh{:ARRay | [:SCALar]}:POWer:FREQuency?
- FETCh{:ARRay | [:SCALar]}:POWer:WNUMber?

**:FETCh{:ARRay | [:SCALar]} :POWER?**

The command trees for each of the four basic [measurement instructions](#) are identical.

**Query**

:POWER? [<expected\_value>[,<resolution>]]

Returns amplitude values.

Used With	<expected_value>	<resolution>
SCALar	optional	optional
ARRay	Ignored	optional

When used with :SCALar, a single value is returned. If the tabular display is on, the display is placed in the single-wavelength mode. The marker is then placed on the signal having a power level that is closest to the <expected\_value> parameter.

When used with :ARRay, an array of amplitudes is returned. If the tabular display is on, the display is placed in the list-by-power mode.

Returned values are in the current power units. Wavelength units are not affected.

<expected\_value>  
**Constants**

MAXimum  
The highest power signal

MINimum  
The lowest power signal

DEFault  
The signal at the current marker position

<resolution>  
**Constants**

MAXimum  
0.01 resolution

MINimum  
0.001 resolution

EXTended1  
0.0001 resolution

DEFault  
Current resolution

**Examples**

:FETCh:ARR:POW?

:FETCh:SCAL:POW? MIN

**Query Response**

The following line is an example of a returned string when :FETCh:SCALar:POWER? MAXimum is sent:

-5.88346500E+000

If six laser lines are located and `:FETCh:ARRay:POWer?` is sent, the following string could be returned. Notice that the first returned number indicates the number of laser-line values returned in the query. The measurement units are in dBm.

6,-1.37444400E+001,-1.10996100E+001,-9.62396600E+000,  
-7.94024500E+000,-7.01303200E+000,-1.04536200E+001

**:FETCh{:ARRay | [:SCALar]} :POWer:FREQuency?**

The command trees for each of the four basic [measurement instructions](#) are identical.

**Query**

:POWer:FREQuency? [<expected\_value>[,<resolution>]]

Returns frequency values.

Used With	<expected_value>	<resolution>
SCALar	optional	optional
ARRay	Ignored	optional

When used with :SCALar, a single value is returned. If the tabular display is on, the display is placed in the single-wavelength mode. The marker is then placed on the signal having a frequency that is closest to the <expected\_value> parameter. Default units for the <expected\_value> parameter are Hz.

When used with :ARRay, an array of frequencies is returned. If the tabular display is on, the display is placed in the list-by-wavelength mode.

The <resolution> parameter sets the number of points for :CALCulate1:TRANSform:FREQuency:POINts. This parameter provides backwards compatibility with previous instruments. It does not affect the resolution or speed of :FETCh. This parameter is a unitless number whose value will be limited to either 0.01, 0.001, or 0.0001.

Returned values are in Hz. Displayed units are shown in THz. Power units are not affected.



When using :ARRay, the <expected\_value> is ignored. However, this argument must be present if the resolution argument is specified. You can use DEF to specify the default value.

<expected_value>	MAXimum
Constants	The highest frequency signal
	MINimum
	The lowest frequency signal
	DEFault
	The signal at the current marker position
<resolution>	MAXimum
Constants	0.01 resolution
	MINimum



	0.001 resolution
	EXTended1
	0.0001 resolution
	DEFault
	Current resolution
<b>Examples</b>	:FETCh:ARR:POW:FREQ? DEF, MIN
	:FETCh:SCAL:POW:FREQ? 230.8THz, MIN
<b>Query Response</b>	The following line is an example of a returned string when :FETCh:SCALar:POWer:FREQuency? MAXimum is sent:
	+1.94055176E+014
	If six laser lines are located and :FETCh:ARRay:POWer:FREQuency? is sent, the following string is an example of the returned data. Notice that the first returned number indicates the number of laser-line values returned in the query.
	6,+1.93050000E+014,+1.93250900E+014,+1.93452000E+014, +1.93653000E+014,+1.93854100E+014,+1.94055100E+014

**:FETCh{:ARRay | [:SCALar]} :POWer:WAVelength?**

The command trees for each of the four basic [measurement instructions](#) are identical.

**Query**

:POWer:WAVelength? [<expected\_value>[,<resolution>]]

Returns wavelength values.

Used With	<expected_value>	<resolution>
SCALar	optional	optional
ARRay	Ignored	optional

When used with :SCALar, a single value is returned. If the tabular display is on, the display is placed in the single-wavelength mode. The marker is then placed on the signal having a wavelength that is closest to the <expected\_value> parameter. Default units for the <expected\_value> parameter are in meters.

When used with :ARRay, an array of wavelengths is returned. If the tabular display is on, the display is placed in the list-by-wavelength mode.

The <resolution> parameter sets the number of points for :CALCulate1:TRANSform:FREQuency:POINts. This parameter provides backwards compatibility with previous instruments. It does not affect the resolution or speed of :FETCh. This parameter is a unitless number whose value will be limited to either 0.01, 0.001, or 0.0001.

Returned values are in meters. Displayed units are shown in nanometers. Power units are not affected.



When using :ARRay, the <expected\_value> is ignored. However, this argument must be present if the resolution argument is specified. You can use DEF to specify the default value.

<expected_value>	MAXimum
Constants	The highest wavelength signal
	MINimum
	The lowest wavelength signal
	DEFault
	The signal at the current marker position
<resolution>	MAXimum
Constants	0.01 resolution
	MINimum

	0.001 resolution
	EXTended1
	0.0001 resolution
	DEFault
	Current resolution
<b>Examples</b>	:FETCh:ARR:POW:WAV? DEF, MIN
	:FETCh:SCAL:POW:WAV? 1300NM, MIN
<b>Query Response</b>	The following line is an example of a returned string when :FETCh:SCALar:POWer:WAVelength? MAXimum is sent:
	+1.5529258E-006
	If six laser lines are located and :FETCh:ARRay:POWer:WAVelength? is sent, the following string could be returned. Notice that the first returned number indicates the number of laser-line values returned in the query.
	6,+1.54488100E-006,+1.54648400E-006,+1.54809000E-006,+1.54969900E-006,+1.55131100E-006,+1.55292600E-006

**:FETCh{:ARRay | [:SCALar]} :POWer:WNUMber?**

The command trees for each of the four basic [measurement instructions](#) are identical.

**Query**

:POWer:WNUMber? [<expected\_value>[,<resolution>]]

Returns a wave number value.

Used With	<expected_value>	<resolution>
SCALar	optional	optional
ARRay	Ignored	optional

When used with :SCALar, a single value is returned. If the tabular display is on, the display is placed in the single-wavelength mode. The marker is then placed on the signal having a wave number that is closest to the <expected\_value> parameter. Default units for the <expected\_value> parameter are in /m.

When used with :ARRay, an array of wave numbers is returned. If the tabular display is on, the display is placed in the list-by-wavelength mode.

The <resolution> parameter sets the number of points for :CALCulate1:TRANSform:FREQuency:POINts. This parameter provides backwards compatibility with previous instruments. It does not affect the resolution or speed of :FETCh. This parameter is a unitless number whose value will be limited to either 0.01, 0.001, or 0.0001.

Returned values are in inverse meters. Displayed units are shown in inverse centimeters. Power units are not affected.



When using :ARRay, the <expected\_value> is ignored. However, this argument must be present if the resolution argument is specified. You can use DEF to specify the default value.

<expected\_value>

MAXimum

Constants

The laser line having the largest wave number

MINimum

The laser line having the smallest wave number

DEFault

The signal at the current marker position

<resolution>

MAXimum

Constants

0.01 resolution

MINimum  
0.001 resolution

EXTended1  
0.0001 resolution

DEFault  
Current resolution

**Examples** :FETCh:ARR:POW:WNUM? DEF, MIN

:FETCh:SCAL:POW:WNUM? 6451, MIN

**Query Response** If :FETCh:SCALar:POWer:WNUMber? 6451 is sent, and a 1550 nm laser line is present, the following response would be returned to the computer:

+6.45286262E+005

Notice that the returned units are /m. If six laser lines are measured and :FETCh:ARRay:POWer:WNUMber? is sent, the following response is returned. Notice that the first returned number indicates the number of laser-line values returned in the query.

6,+6.43945300E+005,+6.44615600E+005,+6.45286300E+005  
+6.45957000E+005,+6.46627900E+005,+6.47298400E+005

## :MEASure Command

Use MEASure commands to configure the Agilent 86122C, capture new data, and return the data in one operation. For example, to measure the longest wavelength, send the following command:

```
:MEASure:SCALar:POWer:WAVelength? MAX
```



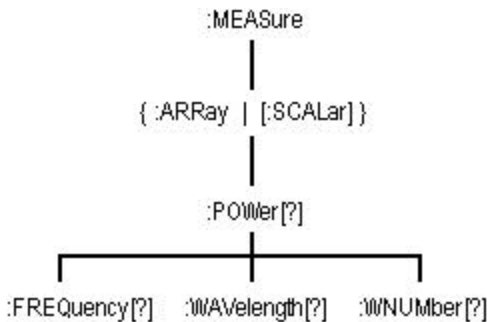
A common programming error is to send :MEASure when the instrument is in the continuous measurement acquisition mode. Because :MEASure contains an :INITiate:IMMediate command, which expects the single measurement acquisition mode, an error is generated, and :INITiate is ignored.



MEASure and READ are identical to combining the following commands:

```
:MEASure :ABORt;:CONFigure;:READ
:READ :ABORt;:INITiate:IMMediate;:FETCh
```

The MEASure commands have the following hierarchy:



See the following MEASure commands for more information about the selected command.



The command trees for each of the four basic [measurement instructions](#) are identical.

- [MEASure{:ARRay | \[:SCALar\]} :POWer?](#)
- [MEASure{:ARRay | \[:SCALar\]} :POWer:WAVelength?](#)
- [MEASure{:ARRay | \[:SCALar\]} :POWer:FREQuency?](#)
- [MEASure{:ARRay | \[:SCALar\]} :POWer:WNUMber?](#)

**:MEASure{:ARRAY | [:SCALar]} :POWer?**

The command trees for each of the four basic [measurement instructions](#) are identical.

<b>Query</b>	<p>:POWer? [&lt;expected_value&gt;[,&lt;resolution&gt;]]</p> <p>Returns amplitude values.</p> <table border="0" style="margin-left: 20px;"> <tr> <td style="padding-right: 20px;"><b>Used With</b></td> <td style="padding-right: 20px;">&lt;expected_value&gt;</td> <td>&lt;resolution&gt;</td> </tr> <tr> <td>SCALar</td> <td>optional</td> <td>optional</td> </tr> <tr> <td>ARRay</td> <td>Ignored</td> <td>optional</td> </tr> </table> <p>When used with :SCALar, a single value is returned. If the tabular display is on, the display is placed in the single-wavelength mode. The marker is then placed on the signal having a power level that is closest to the &lt;expected_value&gt; parameter.</p> <p>When used with :ARRay, an array of amplitudes is returned. If the tabular display is on, the display is placed in the list-by-power mode.</p> <p>Returned values are in the current power units. Wavelength units are not affected.</p>	<b>Used With</b>	<expected_value>	<resolution>	SCALar	optional	optional	ARRay	Ignored	optional
<b>Used With</b>	<expected_value>	<resolution>								
SCALar	optional	optional								
ARRay	Ignored	optional								
<b>&lt;expected_value&gt; Constants</b>	<p>MAXimum The highest power signal</p> <p>MINimum The lowest power signal</p> <p>DEFault The signal at the current marker position</p>									
<b>&lt;resolution&gt; Constants</b>	<p>MAXimum 0.01 resolution</p> <p>MINimum 0.001 resolution</p> <p>EXTended1 0.0001 resolution</p> <p>DEFault Current resolution</p>									
<b>Examples</b>	<p>:MEAS:ARR:POW?</p> <p>:MEAS:SCAL:POW? DEF</p>									
<b>Query Response</b>	<p>The following line is an example of a returned string when :MEASure:SCALar:POWer? MAXimum is sent:</p>									

-5.88346500E+000

If six laser lines are located and `:MEASure:ARRay:POWer?` is sent, the following string could be returned. Notice that the first returned number indicates the number of laser-line values returned in the query. The measurement units are in dBm.

6,-1.37444400E+001,-1.10996100E+001,-9.62396600E+000,  
-7.94024500E+000,-7.01303200E+000,-1.04536200E+001



**:MEASure{:ARRAY | [:SCALar]} :POWER:FREQuency?**

The command trees for each of the four basic [measurement instructions](#) are identical.

**Query**

:POWER:FREQuency? [<expected\_value>[,<resolution>]]

Returns frequency values.

Used With	<expected_value>	<resolution>
SCALar	optional	optional
ARRay	Ignored	optional

When used with :SCALar, a single value is returned. If the tabular display is on, the display is placed in the single-wavelength mode. The marker is then placed on the signal having a frequency that is closest to the <expected\_value> parameter. Default units for the <expected\_value> parameter are Hz.

When used with :ARRay, an array of frequencies is returned. If the tabular display is on, the display is placed in the list-by-wavelength mode.

The <resolution> parameter sets the number of points for :CALCulate1:TRANSform:FREQuency:POINts. This parameter provides backwards compatibility with previous instruments. It does not affect the resolution or speed of :MEASure. This parameter is a unitless number whose value will be limited to either 0.01, 0.001, or 0.0001.

Returned values are in Hz. Displayed units are shown in THz. Power units are not affected.



When using :ARRay, the <expected\_value> is ignored. However, this argument must be present if the resolution argument is specified. You can use DEF to specify the default value.

<expected_value>	MAXimum
Constants	The highest frequency signal
	MINimum
	The lowest frequency signal
	DEFault
	The signal at the current marker position
<resolution>	MAXimum
Constants	0.01 resolution
	MINimum

0.001 resolution  
 EXTended1  
 0.0001 resolution  
 DEFault  
 Current resolution

**Examples** :MEAS:ARR:POW:FREQ?

:MEAS:SCAL:POW:FREQ? 230.8THZ

**Query Response** The following line is an example of a returned string when :MEASure:SCALar:POWer:FREQuency? MAXimum is sent:

+1.94055176E+014

If six laser lines are located and :MEASure:ARRay:POWer:FREQuency? is sent, the following string is an example of the returned data. Notice that the first returned number indicates the number of laser-line values returned in the query.

6,+1.93050000E+014,+1.93250900E+014,+1.93452000E+014,  
 +1.93653000E+014,+1.93854100E+014,+1.94055100E+014

**:MEASure{:ARRAY | [:SCALar]} :POWER:WAVelength?**

The command trees for each of the four basic [measurement instructions](#) are identical.

**Query**

:POWER:WAVelength? [<expected\_value>[,<resolution>]]

Returns wavelength values.

Used With	<expected_value>	<resolution>
SCALar	optional	optional
ARRay	Ignored	optional

When used with :SCALar, a single value is returned. If the tabular display is on, the display is placed in the single-wavelength mode. The marker is then placed on the signal having a wavelength that is closest to the <expected\_value> parameter. Default units for the <expected\_value> parameter are in meters.

When used with :ARRay, an array of wavelengths is returned. If the tabular display is on, the display is placed in the list-by-wavelength mode.

The <resolution> parameter sets the number of points for :CALCulate1:TRANSform:FREQuency:POINts. This parameter provides backwards compatibility with previous instruments. It does not affect the resolution or speed of :MEASure. This parameter is a unitless number whose value will be limited to either 0.01, 0.001, or 0.0001.

Returned values are in meters. Displayed units are shown in nanometers. Power units are not affected.



When using :ARRay, the <expected\_value> is ignored. However, this argument must be present if the resolution argument is specified. You can use DEF to specify the default value.

<expected_value>	MAXimum
Constants	The highest wavelength signal
	MINimum
	The lowest wavelength signal
	DEFault
	The signal at the current marker position
<resolution>	MAXimum
Constants	0.01 resolution
	MINimum

	0.001 resolution
	EXTended1
	0.0001 resolution
	DEFault
	Current resolution
<b>Examples</b>	:MEAS:ARR:POW:WAV?
	:MEAS:SCAL:POW:WAV? 1300NM
<b>Query Response</b>	The following line is an example of a returned string when :MEASure:SCALar:POWer:WAVelength? MAXimum is sent:
	+1.5529258E-006
	If six laser lines are located and :MEASure:ARRay:POWer:WAVelength? is sent, the following string could be returned. Notice that the first returned number indicates the number of laser-line values returned in the query.
	6,+1.54488100E-006,+1.54648400E-006,+1.54809000E-006,+1.54969900E-006,+1.55131100E-006,+1.55292600E-006

**:MEASure{ :ARRAY | [:SCALar] } :POWER:WNUMber?**

The command trees for each of the four basic [measurement instructions](#) are identical.

**Query**

:POWER:WNUMber? [<expected\_value>[,<resolution>]]

Returns a wave number value.

Used With	<expected_value>	<resolution>
SCALar	optional	optional
ARRay	Ignored	optional

When used with :SCALar, a single value is returned. If the tabular display is on, the display is placed in the single-wavelength mode. The marker is then placed on the signal having a wave number that is closest to the <expected\_value> parameter. Default units for the <expected\_value> parameter are in /m.

When used with :ARRay, an array of wave numbers is returned. If the tabular display is on, the display is placed in the list-by-wavelength mode.

The <resolution> parameter sets the number of points for :CALCulate1:TRANSform:FREQuency:POINts. This parameter provides backwards compatibility with previous instruments. It does not affect the resolution or speed of :MEASure. This parameter is a unitless number whose value will be limited to either 0.01, 0.001, or 0.0001.

Returned values are in inverse meters. Displayed units are shown in inverse centimeters. Power units are not affected.



When using :ARRay, the <expected\_value> is ignored. However, this argument must be present if the resolution argument is specified. You can use DEF to specify the default value.

<expected_value>	MAXimum
Constants	The laser line having the largest wave number
	MINimum
	The laser line having the smallest wave number
	DEFault
	The signal at the current marker position
<resolution>	MAXimum
Constants	0.01 resolution

MINimum  
 0.001 resolution  
 EXTended1  
 0.0001 resolution  
 DEFault  
 Current resolution

**Examples**

:MEAS:ARR:POW:WNUM?

:MEAS:SCAL:POW:WNUM? 6451

**Query Response**

If :MEASure:SCALar:POWer:WNUMber? 6451 is sent, and a 1550 nm laser line is present, the following response would be returned to the computer:

+6.45286262E+005

Notice that the returned units are /m. If six laser lines are measured and :MEASure:ARRay:POWer:WNUMber? is sent, the following response is returned. Notice that the first returned number indicates the number of laser-line values returned in the query.

6,+6.43945300E+005,+6.44615600E+005,+6.45286300E+005  
 +6.45957000E+005,+6.46627900E+005,+6.47298400E+005

## :READ Command

The READ command is similar to the MEASure except that it does not configure the instrument's settings. Use CONFigure to configure the instrument for a particular measurement without returning any data.



MEASure and READ are identical to combining the following commands:

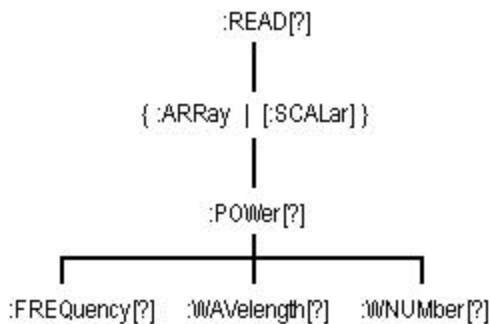
```
:MEASure :ABORt::CONFigure::READ
:READ :ABORt::INITiate:IMMEDIATE::FETCh
```

If no arguments are specified for :READ?, the previous READ or FETCh arguments will be used.



A common programming error is to send :READ when the instrument is in the continuous measurement acquisition mode. Because READ contains an :INITiate:IMMEDIATE command, which expects the single measurement acquisition mode, an error is generated, and :INITiate is ignored.

The READ commands have the following hierarchy:



See the following READ commands for more information about the selected command.



The command trees for each of the four basic [measurement instructions](#) are identical.

- [READ{:ARRay | \[:SCALar\]} :POWer?](#)
- [READ {:ARRay | \[:SCALar\]} :POWer:WAVelength?](#)
- [READ {:ARRay | \[:SCALar\]} :POWer:FREQuency?](#)
- [READ {:ARRay | \[:SCALar\]} :POWer:WNUMber?](#)

**:READ{:ARRay | [:SCALar]} :POWER?**

The command trees for each of the four basic [measurement instructions](#) are identical.

**Query**

:POWER? [<expected\_value>[,<resolution>]]

Returns amplitude values.

Used With	<expected_value>	<resolution>
SCALar	optional	optional
ARRay	Ignored	optional

When used with :SCALar, a single value is returned. If the tabular display is on, the display is placed in the single-wavelength mode. The marker is then placed on the signal having a power level that is closest to the <expected\_value> parameter.

When used with :ARRay, an array of amplitudes is returned. If the tabular display is on, the display is placed in the list-by-power mode.

Returned values are in the current power units. Wavelength units are not affected.

**<expected\_value>  
Constants**

MAXimum  
The highest power signal

MINimum  
The lowest power signal

DEFault  
The signal at the current marker position

**<resolution>  
Constants**

MAXimum  
0.01 resolution

MINimum  
0.001 resolution

EXTended1  
0.0001 resolution

DEFault  
Current resolution

**Examples**

:READ:ARR:POW?

:READ:SCAL:POW? MIN

**Query Response**

The following line is an example of a returned string when :READ:SCALar:POWER? MAXimum is sent:



-5.88346500E+000

If six laser lines are located and `:READ:ARRay:POWer?` is sent, the following string could be returned. Notice that the first returned number indicates the number of laser-line values returned in the query. The measurement units are in dBm.

6,-1.37444400E+001,-1.10996100E+001,-9.62396600E+000,  
-7.94024500E+000,-7.01303200E+000,-1.04536200E+001

**:READ{:ARRay | [:SCALar]} :POWer:FREQuency?**

The command trees for each of the four basic [measurement instructions](#) are identical.

**Query**

:POWer:FREQuency? [<expected\_value>[,<resolution>]]

Returns frequency values.

Used With	<expected_value>	<resolution>
SCALar	optional	optional
ARRay	Ignored	optional

When used with :SCALar, a single value is returned. If the tabular display is on, the display is placed in the single-wavelength mode. The marker is then placed on the signal having a frequency that is closest to the <expected\_value> parameter. Default units for the <expected\_value> parameter are Hz.

When used with :ARRay, an array of frequencies is returned. If the tabular display is on, the display is placed in the list-by-wavelength mode.

The <resolution> parameter sets the number of points for :CALCulate1:TRANSform:FREQuency:POINts. This parameter provides backwards compatibility with previous instruments. It does not affect the resolution or speed of :READ. This parameter is a unitless number whose value will be limited to either 0.01, 0.001, or 0.0001.

Returned values are in Hz. Displayed units are shown in THz. Power units are not affected.



When using :ARRay, the <expected\_value> is ignored. However, this argument must be present if the resolution argument is specified. You can use DEF to specify the default value.

<expected_value>	MAXimum
Constants	The highest frequency signal
	MINimum
	The lowest frequency signal
	DEFault
	The signal at the current marker position
<resolution>	MAXimum
Constants	0.01 resolution
	MINimum

	0.001 resolution
	EXTended1
	0.0001 resolution
	DEFault
	Current resolution
<b>Examples</b>	:READ:ARR:POW:FREQ?
	:READ:SCAL:POW:FREQ? 230.8THZ
<b>Query Response</b>	The following line is an example of a returned string when :READ:SCALar:POWer:FREQUency? MAXimum is sent: +1.94055176E+014  If six laser lines are located and :READ:ARRay:POWer:FREQUency? is sent, the following string is an example of the returned data. Notice that the first returned number indicates the number of laser-line values returned in the query.  6,+1.93050000E+014,+1.93250900E+014,+1.93452000E+014,+1.93653000E+014,+1.93854100E+014,+1.94055100E+014

**:READ{:ARRay | [:SCALar]} :POWer:WAVelength?**

The command trees for each of the four basic [measurement instructions](#) are identical.

**Query**

:POWer:WAVelength? [<expected\_value>[,<resolution>]]

Returns wavelength values.

Used With	<expected_value>	<resolution>
SCALar	optional	optional
ARRay	Ignored	optional

When used with :SCALar, a single value is returned. If the tabular display is on, the display is placed in the single-wavelength mode. The marker is then placed on the signal having a wavelength that is closest to the <expected\_value> parameter. Default units for the <expected\_value> parameter are in meters.

When used with :ARRay, an array of wavelengths is returned. If the tabular display is on, the display is placed in the list-by-wavelength mode.

The <resolution> parameter sets the number of points for :CALCulate1:TRANSform:FREQuency:POINts. This parameter provides backwards compatibility with previous instruments. It does not affect the resolution or speed of :READ. This parameter is a unitless number whose value will be limited to either 0.01, 0.001, or 0.0001.

Returned values are in meters. Displayed units are shown in nanometers. Power units are not affected.



When using :ARRay, the <expected\_value> is ignored. However, this argument must be present if the resolution argument is specified. You can use DEF to specify the default value.

<expected_value>	MAXimum
Constants	The highest wavelength signal
	MINimum
	The lowest wavelength signal
	DEFault
	The signal at the current marker position
<resolution>	MAXimum
Constants	0.01 resolution
	MINimum

	0.001 resolution
	EXTended1
	0.0001 resolution
	DEFault
	Current resolution
<b>Examples</b>	:READ:ARR:POW:WAV?
	:READ:SCAL:POW:WAV? 1300NM
<b>Query Response</b>	The following line is an example of a returned string when :READ:SCALar:POWer:WAVelength? MAXimum is sent: +1.5529258E-006  If six laser lines are located and :READ:ARRay:POWer:WAVelength? is sent, the following string could be returned. Notice that the first returned number indicates the number of laser-line values returned in the query.  6,+1.54488100E-006,+1.54648400E-006,+1.54809000E-006,+1.54969900E-006,+1.55131100E-006,+1.55292600E-006

**:READ{:ARRay | [:SCALar]} :POWer:WNUMber?**

The command trees for each of the four basic [measurement instructions](#) are identical.

**Query**

:POWer:WNUMber? [<expected\_value>[,<resolution>]]

Returns a wave number value.

Used With	<expected_value>	<resolution>
SCALar	optional	optional
ARRay	Ignored	optional

When used with :SCALar, a single value is returned. If the tabular display is on, the display is placed in the single-wavelength mode. The marker is then placed on the signal having a wave number that is closest to the <expected\_value> parameter. Default units for the <expected\_value> parameter are in /m.

When used with :ARRay, an array of wave numbers is returned. If the tabular display is on, the display is placed in the list-by-wavelength mode.

The <resolution> parameter sets the number of points for :CALCulate1:TRANSform:FREQuency:POINts. This parameter provides backwards compatibility with previous instruments. It does not affect the resolution or speed of :READ. This parameter is a unitless number whose value will be limited to either 0.01, 0.001, or 0.0001.

Returned values are in inverse meters. Displayed units are shown in inverse centimeters. Power units are not affected.



When using :ARRay, the <expected\_value> is ignored. However, this argument must be present if the resolution argument is specified. You can use DEF to specify the default value.

<expected_value>	MAXimum
Constants	The laser line having the largest wave number
	MINimum
	The laser line having the smallest wave number
	DEFault
	The signal at the current marker position
<resolution>	MAXimum
Constants	0.01 resolution

MINimum  
0.001 resolution

EXTended1  
0.0001 resolution

DEFault  
Current resolution

**Examples**

```
:READ:ARR:POW:WNUM?
```

```
:READ:SCAL:POW:WNUM? 6451
```

**Query Response**

If :READ:SCALar:POWer:WNUMber? 6451 is sent, and a 1550 nm laser line is present, the following response would be returned to the computer:

```
+6.45286262E+005
```

Notice that the returned units are /m. If six laser lines are measured and :READ:ARRay:POWer:WNUMber? is sent, the following response is returned. Notice that the first returned number indicates the number of laser-line values returned in the query.

```
6,+6.43945300E+005,+6.44615600E+005,+6.45286300E+005  
+6.45957000E+005,+6.46627900E+005,+6.47298400E+005
```

## Common Commands

Use common commands to control general device functions that are common among many different types of instruments. Common commands are defined by the IEEE 488.2 standard. They control functions that are common to all IEEE 488.2 instruments. Common commands can be received and processed by the instrument, whether they are sent over the GPIB as separate program messages or within other program messages.

These commands include

- [\\*CLS](#)                      ● [\\*RCL](#)                      ● [\\*STB?](#)
- [\\*ESE](#)                        ● [\\*RST](#)                      ● [\\*TRG](#)
- [\\*ESR?](#)                      ● [\\*SAV](#)                      ● [\\*TST?](#)
- [\\*IDN?](#)                      ● [\\*SRE](#)                      ● [\\*WAI](#)
- [\\*OPC](#)

---

### \*CLS - Clear Status

Command	*CLS
	Clears the instrument status byte by emptying the error queue and clearing all event registers. This command does not alter the instrument settings. After the *CLS command, the instrument is left in the idle state. This command also cancels any preceding *OPC command or query.
	*CLS cannot be issued as a query.

---

### \*ESE - Event Status Enable

Command	*ESE <integer>
	Sets the bits in the event status enable register.
<integer>	A mask value from 0 to 255.
Example	The following example sets ESB (event summary bit) bit 5 of the event status enable register. Therefore, when an incorrect programming command is received, the CME (command error bit) in the status byte register is set.  OUTPUT 720;”*ESE 32”



See [Status Enable Register](#) for more information about event status enable register bits.

---

### \*ESR? - Event Status Register Query

Query	*ESR?
	Returns the value of the event status register. When you read the standard event status register, the value returned is the total of the bit weights of all of the bits that are set to one (1) at the time you read the byte.
Example	<pre>OUTPUT 720;”*ESR?” ENTER 720;Event PRINT Event</pre> <p>See <a href="#">Status Enable Register</a> for more information about event status enable register bits.</p>

---

### \*IDN? - Identification Number Query

Query	*IDN?
	<p>Returns a string that uniquely identifies the instrument.</p> <p>The string is in the form of</p> <p>Agilent Technologies,86122C,&lt;serial number&gt;,&lt;software revision&gt;</p> <p>This query must be the last query in a program message. Any queries after *IDN? are ignored.</p> <p>The maximum length of the identification string is 50 bytes.</p>
Example	<pre>DIM Id\$[50] OUTPUT 720;”*IDN?” ENTER 720;Id\$ PRINT Id\$</pre>

---

### \*OPC - Operation Complete

Command	*OPC
	<p>Sets the operation complete bit in the event status register when all pending device operations have finished.</p> <p>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</p>

	Agilent 86122C has completed the operation.
Query	*OPC?  Places an ASCII “1” in the output queue when all pending device operations have finished.  Use *OPC? query to ensure all operations have completed before continuing the program. By following a command with *OPC? and an ENTER statement, the program will pause until the response (ASCII “1”) is returned by the instrument.  Be sure the computer’s timeout limit is at least two seconds, since some of the Agilent 86122C commands take approximately one second to complete.
Example	OUTPUT 720;”*OPC?” ENTER 720;Op\$

---

### \*RCL – Recall

Command	*RCL <integer>  Recalls a saved instrument state. This command is backward compatible with the 86120B/C.  For a more general save/recall function, see <a href="#">:MMEMory:LOAD</a> and <a href="#">:MMEMory:STORe</a> .
<integer>	A number from 1 to 4.

---

### \*RST - Reset

Command	*RST  Returns the Agilent 86122C to a known condition.  This command places the instrument in single measurement acquisition mode. Therefore, any current data is marked as invalid and a measurement query results in an error. For example, :FETCh? results in error number – 230, “Data corrupt or stale”. You must then initiate a new sweep with :INITiate:IMMEDIATE before you can use :FETCh.
---------	--

---

**\*SAV - Save**

Command	*SAV <integer> Stores the current state of the instrument in a save register.
<integer>	A number from 1 to 4.

---

**\*SRE - Service Request Enable**

Command	*SRE <integer> Sets the bits in the service request enable register.
<integer>	A mask value from 0 to 63 or from 128 to 191.  See <a href="#">Status Event Enable Register</a> for more information about service request enable register bits.
Query	*SRE?
Example	The following example enables ESB (event summary) bit 5 in the status byte register to generate a service request.  OUTPUT 720;”*SRE 32” ENTER 720;Op\$

---

**\*STB? - Status Byte**

Query	*STB? <integer> Returns the current value of the status byte register.  The master summary status (MSS) bit 6 indicates whether or not the device has at least one reason for requesting service. When you read the status byte register, the value returned is the total of the bit weights of all of the bits set to one at the time you read the byte. *STB? does not affect the contents of the status byte register.  See <a href="#">Status Byte Register</a> for more information about each bit in the status byte register.
<integer>	A number from 0 to 255.
Example	OUTPUT 720;”*STB?” ENTER 720;Value PRINT Value

---

## \*TRG - Trigger

Command      \*TRG

Acquires data according to the current settings. If a measurement is already in progress, a trigger is ignored, and an error is generated.

Example        The following example starts the data acquisition according to the current settings.

```
OUTPUT 720;”*TRG”
```



**Non-sequential command:** Always use an \*OPC? query or a \*WAI command to ensure that this command has the time to complete before sending any more commands to the instrument.

---

## \*TST? - Test

Query          \*TST?

Starts a self-test on the instrument. The result of the test is placed in the output queue. A zero indicates the test passed and a non-zero value indicates the test failed. The instrument will initiate a measurement and check for any hardware errors after the measurement is complete.

Example        OUTPUT 720;”\*TST?”  
ENTER 720;Result  
PRINT Result

---

## \*WAI - Wait

Command      \*WAI

Prevents the instrument from executing any further commands until the current command and all pending commands have completed.

## Subsystem Commands

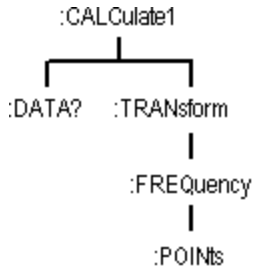
Use subsystem commands to control specific device functions.

See the following subsystems for more information about the selected subsystem and its associated commands.

- [CALCulate1](#)
- [CALCulate2](#)
- [CALCulate3](#)
- [DISPlay](#)
- [HCOPY](#)
- [MMEMory](#)
- [SENSe](#)
- [STATus](#)
- [SYSTem](#)
- [TRIGger](#)
- [UNIT](#)

## :CALCulate1 Subsystem

Use CALCulate1 commands to query *uncorrected* frequency-spectrum data. The measurement update returns 30,093 values. The commands in this subsystem have the following command hierarchy:



See the following CALCulate1 commands for more information about the selected command.

- [DATA?](#)
- [TRANSform:FREQuency:Points](#)

**:CALCulate1:DATA?**

Query :CALCulate1:DATA?

Queries *uncorrected* frequency-spectrum data of the input laser line.

Attribute Preset State: not affected  
 Summary SCPI Compliance: instrument specific  
 Query Only

Query Response The returned values are proportional to squared Watts (linear) units. No amplitude or frequency correction is applied to the values. To obtain the logarithmic (dB) result, normalize the returned values by the largest value, then take five times the logarithm of the normalized values.

Be prepared to process a large amount of data when this query is sent. The amount of data returned can be set using CALCulate1:TRANSform:FREQuency:POINts or the resolution argument of a configure measurement instrument function. The following floating point values can be returned:

Mode	Number of Points	Frequency Spacing	Start Frequency	Stop Frequency
MINimum	7525	7.226756134 GHz	181.6878 THz	236.061989 THz
MAXimum	15047	3.6133378067 GHz	181.691489 THz	236.058375 THz
EXTended1	30093	1.806689033 GHz	181.691489 THz	236.058375 THz

The following string is a typical example of the first few returned values:

```
+4.02646500E+001,+6.78125100E+001,+6.17986600E+001,+4.26768200E+001,+4.80245300E+001,+3.10491300E+001,+1.13409400E+001,+5.07832500E+001,+2.77746200E+001,+3.89150500E+001,+3.50217600E+001,+7.34649800E-001,+5.64983800E+000,
```

Notice that only measurement values are returned to the computer. There is no first value that indicates the number of values contained in the string as there is, for example, with the :FETCh, :READ, and :MEASure.

Use:CALCulate1:TRANSform:FREQuency:POINts? to query the number of points :CALC1:DATA? returns.



If your program is aborted or interrupted after sending this query, the Agilent 86122C continues to process the data but does not place it in the output buffer. Because of the amount of data processed, the instrument *will not respond* to any new commands in its input buffer for up to *20 seconds*.



This query will generate a “Settings Conflict” error if the instrument is in the signal-to-noise application, with averaging turned on.

### :CALCulate1:TRANSform:FREQuency:POINts

**Command** :CALCulate1:TRANSform:FREQuency:POINts{?} {<integer> | MINimum | MAXimum | EXTended1}}

Sets the size of the Fast Fourier transform (FFT) performed by the instrument.

**<integer>** Sets FFT size. Must be either 30,093, 15,047 or 7,525. Other values result in an error.

MINimum: 7,525  
 MAXimum: 15,047  
 EXTended1: 30,093

A MAXimum data points mode corresponds to an FFT size of 15,047. A MINimum data points mode corresponds to an FFT size of 7,525. The EXTended1 mode corresponds to an FFT size of 30,093. These values are a subset of the uncorrected-data buffer.

Changing the number of points causes the instrument to reprocess the current set of data.



These settings are for backward compatibility to the 86120 family and no longer affect the accuracy, resolution or update rate of the instrument.

**Attribute Summary** Non-sequential command  
 Preset State: array size set to 15,047  
 \*RST State: 15,047  
 SCPI Compliance: instrument specific

**Query Response** Returns the number of points in the data set. This is the number of measurement points that will be returned by CALCulate1:DATA?.

MINimum: +7,525  
 MAXimum: +15,047  
 EXTended1: +30,093

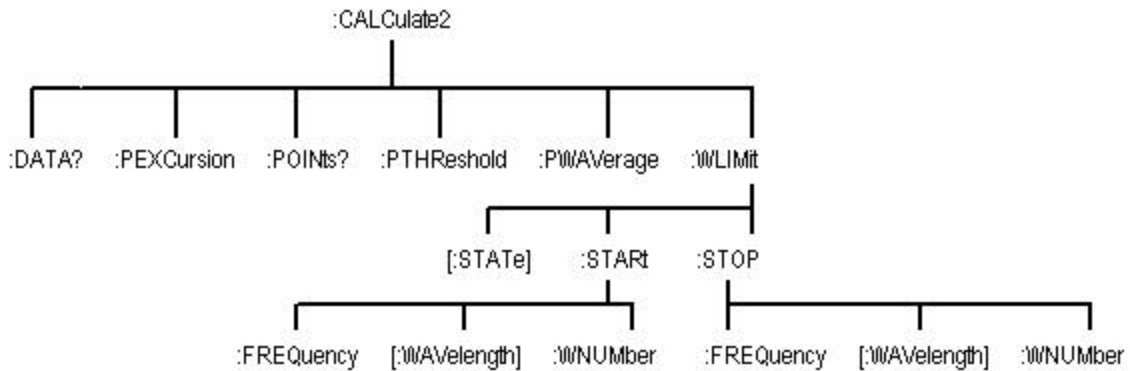


**Non-sequential command:** Always use an \*OPC? query or a \*WAI command to ensure that this command has the time to complete before sending any more commands to the instrument.



## :CALCulate2 Subsystem

Use CALCulate2 commands to query *corrected* values for frequency-spectrum data. The commands in this subsystem have the following command hierarchy:



See the following CALCulate2 commands for more information about the selected command.

- [DATA?](#)
- [PEXCursion](#)
- [POINTs?](#)
- [PTHReshold:ABSolute](#)
- [PTHReshold:MODe](#)
- [PTHReshold\[:RELative\]](#)
- [PWAVerage\[:STATe\]](#)
- [WLMit:START:FREQuency](#)
- [WLMit:START\[:WAVelength\]](#)
- [WLMit:START:WNUMber](#)
- [WLMit\[:STATe\]](#)
- [WLMit:STOP:FREQuency](#)
- [WLMit:STOP\[:WAVelength\]](#)
- [WLMit:STOP:WNUMber](#)

**:CALCulate2:DATA?**

**Query** :CALCulate2:DATA? {FREQuency | POWer | WAVelength | WNUMber}

Queries the corrected peak data of the input laser line.

**Constant** FREQuency

Queries the array of laser-line frequencies after the peak search is completed. If :CALCulate2:PWAVelength:STATe is on, the power-weighted average frequency is returned.

POWer

Queries the array of laser-line powers after the peak search is completed. If :CALCulate2:PWAVelength:STATe is on, the total input power is returned.

WAVelength

Queries the array of laser-line wavelengths after the peak search is completed. If :CALCulate2:PWAVelength:STATe is on, the power-weighted average wavelength is returned.

WNUMber

Queries the array of laser-line wave numbers after the peak search is completed. If :CALCulate2:PWAVelength:STATe is on, the power-weighted average wave number is returned.

**Attribute  
Summary**

Preset State: not affected

SCPI Compliance: standard

Query Only

Use [CALCulate2:POINts?](#) to determine the number of points CALCulate2:DATA? will return. The following string is a typical example of the first few returned values when WAVelength is specified:

```
+1.54488600E-006,+1.54649100E-006,+1.54808300E-006,+1.54969600E-006,+1.55131200E-006,+1.55293000E-006
```

This next string resulted by specifying the WNUMber argument:

```
+6.47296600E+005,+6.46625000E+005,+6.45959900E+005,+6.45287500E+005,+6.44615500E+005,+6.43943900E+005
```

Notice that only measurement values are returned to the computer. There is no first value that indicates the number of values contained in the string as there is, for example, with :FETCh, :READ, and :MEASure.

When there is no input signal, :POWer? returns –200 dBm; and :WAVelength? returns 100 nm.



The value for WAVelength? is corrected for temperature and pressure, and for air or vaccum.

**:CALCulate2:PEXCursion**

Command	:CALCulate2:PEXCursion{?  {<integer>   MINimum   MAXimum   DEFault}}
	Sets the peak excursion limit used by the instrument to determine valid laser line peaks. A laser line is identified as a valid peak if its amplitude is greater than the peak excursion plus the amplitudes of the closest local minima on either side of the peak. This command works in conjunction with the peak threshold setting. Changing the peak excursion limit causes the instrument to reprocess the current set of data.
<integer>	Represents logarithmic value in dB. Valid range is 1 - 30 dB.
Constant	MINimum: 1 dB MAXimum: 30 dB DEFault: 15 dB
Attribute Summary	Non-sequential command Preset State: 15 dB *RST State: 15 dB SCPI Compliance: instrument specific
Query Response	Returns the current value of the peak excursion limit. For example, if the current value is set to 15 dB, then <b>15</b> is returned.



**Non-sequential command:** Always use an \*OPC? query or a \*WAI command to ensure that this command has the time to complete before sending any more commands to the instrument.

**:CALCulate2:POINts?**

Query :CALCulate2:POINts?

Queries the number of points in the data set. This is the number of points that will be returned by [CALCulate2:DATA?](#).

Attribute  
Summary Preset State: not affected  
\*RST State: not affected  
SCPI Compliance: instrument specific  
Query Only

Query  
Response Returns the number of points in the data set. For example, if six laser lines are located, the value **+6** is returned.

### :CALCulate2:PTHReshold:ABSolute

**Command** :CALCulate2:PTHReshold:ABSolute {?} {<real> | <power units suffix> | MINimum | MAXimum | DEFault}}

Sets the peak threshold limit used by the instrument to determine valid laser line peaks. When in Absolute peak threshold mode, a laser line is identified as a valid peak if its amplitude is above the absolute peak threshold value in the units specified by the UNIT:POWer command. If a units suffix is included, the units suffix will take precedence over that specified by UNIT:POWer.

This setting works in conjunction with the peak excursion setting to determine which responses are valid peaks. Changing the peak threshold limit causes the instrument to reprocess the current set of data.

**<real>** Represents power units in dBm. Valid range is - 40 to 0.

**<power units suffix>** Value is in dBm or W, and can include a multiplier.

**Constant** MINimum: -40 dBm  
 MAXimum: 10 dBm  
 DEFault: -20 dBm

**Attribute Summary** Non-sequential command  
 Preset State: -20 dBm  
 \*RST State: -20 dBm  
 SCPI Compliance: instrument specific

**Query Response** Returns the current peak threshold value in units specified by UNIT:POWer, or by the power units suffix. For example, if the current value is set to -15 dBm and:UNIT:POWer is set to dBm, then **-15** is returned.



**Non-sequential command:** Always use an \*OPC? query or a \*WAI command to ensure that this command has the time to complete before sending any more commands to the instrument.

### :CALCulate2:PTHReshold:MODE

**Command** :CALCulate2:PTHReshold:MODE{?| {RELative | ABSolute}}

Sets the peak threshold mode of the instrument. If RELative is selected, then the peak threshold value is treated as a relative number in dB. If absolute is selected, then the peak threshold value is treated as an absolute power in dBm.

**Attribute Summary** Non-sequential command  
 Preset State: RELative  
 \*RST State: RELative  
 SCPI Compliance: instrument specific

**Query Response** Returns the current peak threshold mode.



**Non-sequential command:** Always use an \*OPC? query or a \*WAI command to ensure that this command has the time to complete before sending any more commands to the instrument.

**:CALCulate2:PTHReshold[:RELative]**

Command	:CALCulate2:PTHReshold[:RELative] {?} {<integer>   MINimum   MAXimum   DEFault}
	Sets the peak threshold limit used by the instrument to determine valid laser line peaks.
	When in Relative (default) peak threshold mode, a laser line is identified as a valid peak if its amplitude is above the maximum amplitude minus the peak threshold value. This subtraction is performed in dB.
	This setting works in conjunction with the peak excursion setting to determine which responses are located. Changing the peak threshold limit causes the instrument to reprocess the current set of data.
<integer>	Represents logarithmic value in dB. Valid range is 0 to 40.
Constant	MINimum: 0 dB MAXimum: 40 dB DEFault: 10 dB
Attribute Summary	Non-sequential command Preset State: 10 dB *RST State: 10 dB SCPI Compliance: instrument specific
Query Response	Returns the current peak threshold value in dB. For example, if the current value is set to 15 dB, then <b>15</b> is returned.



**Non-sequential command:** Always use an \*OPC? query or a \*WAI command to ensure that this command has the time to complete before sending any more commands to the instrument.

**:CALCulate2:PWAVerage[:STATe]**

**Command**           :CALCulate2:PWAVerage[:STATe]{?| {ON | OFF | 1 | 0}}

Places the instrument in the power-weighted average mode.

When the state is on, CALCulate2:DATA? POWER returns the total power and CALCulate2:DATA? WAVElength, FREQuency, or WNUMber returns the power-weighted average wavelength, frequency, or wave number values.

Turning power-weighted average mode on while making delta, Fabry-Perot, or signal-to-noise measurements results in a “-221 Settings Conflict” error.

**Attribute**           Preset State: off  
**Summary**            \*RST State: off



### :CALCulate2:WLIMit:START:FREQuency

**Command** :CALCulate2:WLIMit:START:FREQuency {?|<real> | MINimum | MAXimum }

Sets the start frequency for the wavelength limit range.

**<real>** Frequency value within the following limits:

MINimum: 181.6924 THz (1650 nm)

MAXimum: frequency limit stop value

The default units are Hz. The start frequency value must be less than or equal to the stop frequency value or the start frequency will be set to the stop frequency and a “Data out of range” error will be generated.

**Attribute Summary**  
 Non-sequential command  
 Preset State: 181.6924 THz  
 \*RST State: 181.6924 THz  
 SCPI Compliance: instrument specific



**Non-sequential command:** Always use an \*OPC? query or a \*WAI command to ensure that this command has the time to complete before sending any more commands to the instrument.

### :CALCulate2:WLIMit:START[:WAVelength]

**Command** :CALCulate2:WLIMit:START[:WAVelength] {?}{<real> | MINimum | MAXimum}}

Sets the start wavelength for the wavelength limit range.

**<real>** Wavelength value within the following limits:

MINimum: 1270 nm

MAXimum: wavelength limit stop value

The default units are meters. The start wavelength value must be less than or equal to the stop wavelength value or the start wavelength will be set to the stop wavelength and a “Data out of range” error will be generated. Setting the start wavelength is equivalent to setting the stop frequency/wave number because of the inverse relationship of frequency to wavelength.

**Attribute Summary** Non-sequential command  
 Preset State: 1270 nm  
 \*RST State: 1270 nm  
 SCPI Compliance: instrument specific



**Non-sequential command:** Always use an \*OPC? query or a \*WAI command to ensure that this command has the time to complete before sending any more commands to the instrument.

**:CALCulate2:WLIMit:START:WNUMber**

**Command** :CALCulate2:WLIMit:START:WNUMber {?}{<real> | MINimum | MAXimum}}

Sets the start wave number for the wavelength limit range.

**<real>** Wave number value within the following limits:

MINimum: 6060 /cm (1650 nm)

MAXimum: wave number limit stop value

The default units are inverse meters. The start wave number value must be less than or equal to the stop wave number value or the start wave number will be set to the stop wave number and a “Data out of range” error will be generated.

**Attribute** Non-sequential command  
**Summary** Preset State: 6.060606E5 /m  
 \*RST State: 6.060606E5 /m  
 SCPI Compliance: instrument specific



**Non-sequential command:** Always use an \*OPC? query or a \*WAI command to ensure that this command has the time to complete before sending any more commands to the instrument.

**:CALCulate2:WLIMit[:STATe]**

**Command** :CALCulate2:WLIMit[:STATe]{?| {ON | OFF | 1 | 0}}

Turns wavelength limiting on and off. When this function is on, the instrument has an input range from WLIMit START to WLIMit STOP values. When this function is off, the instrument displays peaks over the full wavelength range. The graphics display always shows the range between WLIMit:START and WLIMit:STOP, regardless of the state of this command.

When the instrument receives this command, it reprocesses the data and performs a new peak search.

**Attribute** Non-sequential command  
**Summary** Preset State: on  
 \*RST State: on  
 SCPI Compliance: instrument specific



**Non-sequential command:** Always use an \*OPC? query or a \*WAI command to ensure that this command has the time to complete before sending any more commands to the instrument.

### :CALCulate2:WLIMit:STOP:FREQuency

**Command** :CALCulate2:WLIMit:STOP:FREQuency {?}{<real> | MINimum | MAXimum }

Sets the stop frequency for the wavelength limit range.

**<real>** Frequency value within the following limits:

MINimum: start wavelength limit  
 MAXimum: 236.0571 THz (1270 nm)

The default units are Hz. The stop frequency value must be greater than or equal to the start frequency value or the stop frequency will be set to the start frequency and a “Data out of range” error will be generated.

**Attribute Summary** Non-sequential command  
 Preset State: 236.0571 THz  
 \*RST State: 236.0571 THz  
 SCPI Compliance: instrument specific



**Non-sequential command:** Always use an \*OPC? query or a \*WAI command to ensure that this command has the time to complete before sending any more commands to the instrument.

**:CALCulate2:WLIMit:STOP[:WAVelength]**

**Command** :CALCulate2:WLIMit:STOP[:WAVelength] {?}{<real> | MINimum | MAXimum }

Sets the stop wavelength for the wavelength limit range.

**<real>** Wavelength value within the following limits:

MINimum: start wavelength limit

MAXimum: 1650.0 nm

The default units are meters. The stop wavelength value must be greater than or equal to the start wavelength value or the stop wavelength will be set to the start wavelength and a “Data out of range” error will be generated. Setting the stop wavelength is equivalent to setting the start frequency/wave number because of the inverse relationship of frequency to wavelength.

**Attribute Summary** Non-sequential command  
 Preset State: 1650 nm  
 \*RST State: 1650nm  
 SCPI Compliance: instrument specific



**Non-sequential command:** Always use an \*OPC? query or a \*WAI command to ensure that this command has the time to complete before sending any more commands to the instrument.

**:CALCulate2:WLIMit:STOP:WNUMber**

**Command** :CALCulate2:WLIMit:STOP:WNUMber {?}{<real> | MINimum | MAXimum }

Sets the stop wave number for the wavelength limit range.

**<real>** Wave number value within the following limits:

MINimum: start wavelength limit  
MAXimum: 7874 /cm (1270 nm)

The default units are inverse meters. The stop wave number value must be less than or equal to the start wave number value or the stop wave number will be set to the start wave number and a “Data out of range” error will be generated.

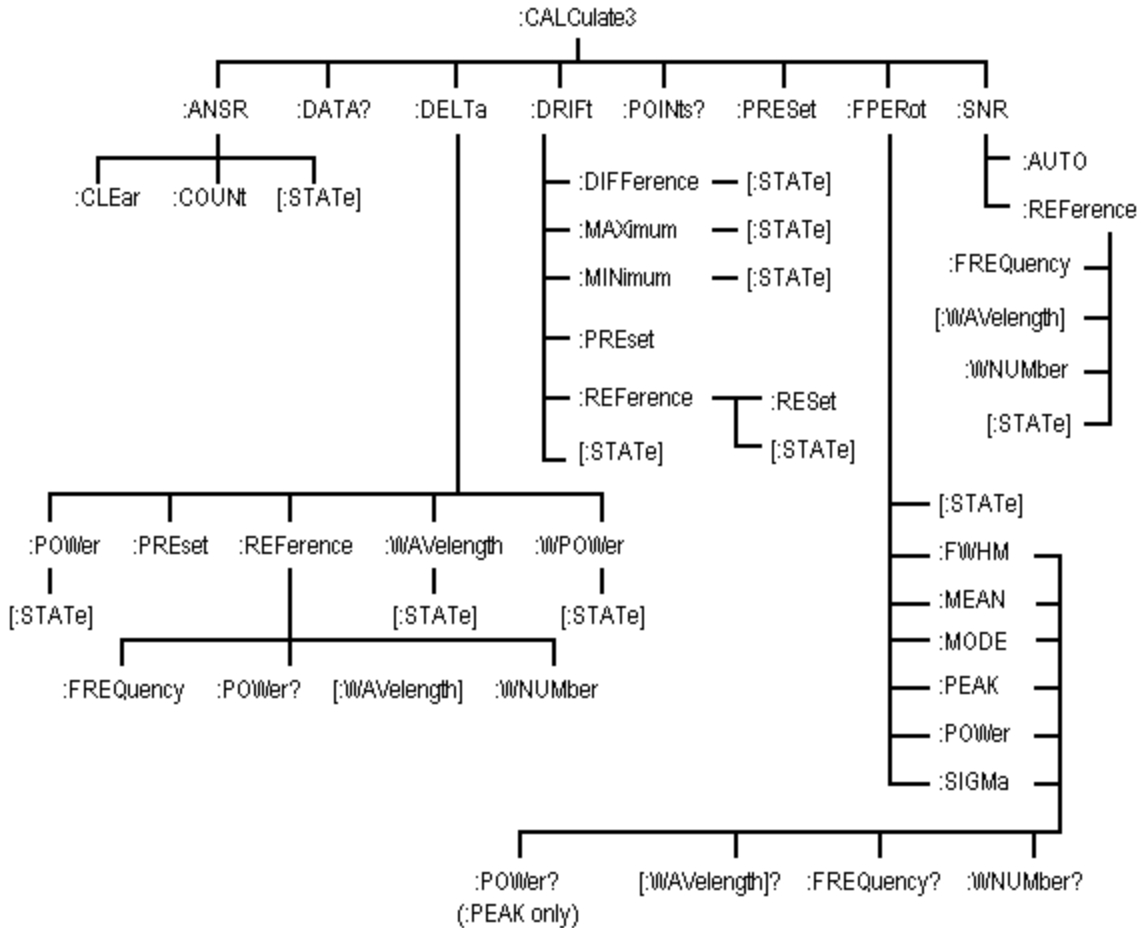
**Attribute** Non-sequential command  
**Summary** Preset State: 7.87401E5 /m  
\*RST State: 7.87401E5 /m  
SCPI Compliance: instrument specific



**Non-sequential command:** Always use an \*OPC? query or a \*WAI command to ensure that this command has the time to complete before sending any more commands to the instrument.

## :CALCulate3 Subsystem

Use the CALCulate3 commands to perform delta, drift, signal-to-noise, and Fabry-Perot measurements. The commands in this subsystem have the following command hierarchy:



See the following CALCulate3 commands for more information about the selected command.

- [ASNR:CLEar](#)
- [ASNR:COUNT](#)
- [ASNR\[:STATe\]](#)
- [DATA?](#)
- [DELTA:POWER\[:STATe\]](#)
- [DELTA:PRESet](#)
- [DELTA:REFERENCE:FREQUENCY](#)
- [DELTA:REFERENCE:POWER?](#)
- [DRIFT:REFERENCE:RESet](#)
- [DRIFT:REFERENCE\[:STATe\]](#)
- [FPERot\[:STATe\]](#)
- [FPERot:FWHM?](#)
- [FPERot:MEAN?](#)
- [FPERot:MODE:SPACing?](#)
- [FPERot:PEAK?](#)
- [FPERot:POWER?](#)



- [DELTA:REFERENCE:WAVelength](#)
- [DELTA:REFERENCE:WNUMber](#)
- [DELTA:WAVelength\[:STATe\]](#)
- [DELTA:WPOWER\[:STATe\]](#)
- [DRIFT:DIFFerence\[:STATe\]](#)
- [DRIFT:MAXimum\[:STATe\]](#)
- [DRIFT:MINimum\[:STATe\]](#)
- [DRIFT:PRESet](#)
- [DRIFT\[:STATe\]](#)
- [FPERot:SIGMa?](#)
- [POINts?](#)
- [PRESet](#)
- [SNR:AUTO](#)
- [SNR:REFERENCE:FREQuency](#)
- [SNR:REFERENCE:WAVelength](#)
- [SNR:REFERENCE:WNUMber](#)
- [SNR\[:STATe\]](#)

### **:CALCulate3:ASNR:CLEar**

**Command**           :CALCulate3:ASNR:CLEar

Clears the number of measurements used in the average signal-to-noise calculation. The current measurement is used as the new reference for the average signal-to-noise calculation.

**Attribute**           Preset State: not affected  
**Summary**            \*RST State: not affected  
                      SCPI Compliance: instrument specific

**:CALCulate3:ASNR:COUNT**

**Command**           :CALCulate3:ASNR:COUNT {?}{<integer> | MINimum | MAXimum }

Sets the number of measurements to be used for the average signal-to-noise calculation. If this count is changed while the average signal calculation is on, and the new count is less than the number of measurements already taken, the instrument will go into single measurement mode.

**<integer>**           Value within the following limits:

MINimum: 10

MAXimum: 900

**Attribute**           Preset State: 100

**Summary**            \*RST State: 100

SCPI Compliance: instrument specific

**:CALCulate3:ASNR[:STATe]**

**Command** :CALCulate3:ASNR[:STATe] {?}{ ON | OFF | 1 | 0 }

Turns the average signal-to-noise ratio on or off. Only one of the :CALCulate3 calculations (ASNR, DELT<sub>a</sub>, DRIFT, or SNR) can be turned on at a time. Turning on the ASNR calculation while another calculation is on will generate a “Settings Conflict” error.

When the calculation is first turned on, the lines measured in the current measurement will be used as the reference values for the signal-to-noise ratio. Subsequent measurements will average the noise values. The signal values are not updated until the number of measurements used to average the noise is greater than or equal to the COUNT value.



Only one STATE command can be turned on at any one time. Attempting to turn more than one state on at a time results in a “-221 Settings Conflict” error.

**Attribute** Preset State: off  
**Summary** \*RST State: off  
 SCPI Compliance: instrument specific

**:CALCulate3:DATA?**

Query	<p>:CALCulate3:DATA? {POWER   FREQuency   WAVelength   WNUMber   DROPPed}</p> <p>Queries the data resulting from delta, drift, data logging, and signal-to-noise measurements.</p>
Argument	<p><b>POWER</b> Queries the array of laser-line powers after the calculation is completed.</p> <p><b>FREQuency</b> Queries the array of laser-line frequencies after the calculation is completed.</p> <p><b>WAVelength</b> Queries the array of laser-line wavelengths after the calculation is completed.</p> <p><b>WNUMber</b> Queries the array of laser-line wave numbers after the calculation is completed.</p> <p><b>DROPPed</b> This applies only to the drift application. This query returns an array of values corresponding to the peaks in the reference sweep. The value for each element in the array will be</p> <ul style="list-style-type: none"> <li>◆ 0 if there have been no drops on the peak</li> <li>◆ 1 if the peak is currently below the peak threshold</li> <li>◆ 2 if the peak previously dropped and is currently above the peak threshold.</li> </ul>
Attribute Summary	<p>Preset State: not affected *RST State: not affected Query Only</p> <p>The data returned by the query depends upon which calculation state is on. An error is generated if no calculation state is on. The returned data is comma delimited.</p> <p>In the SNR or ASNR calculation, only the POWER argument is valid. The other arguments will generate a “Settings Conflict” error. Use CALCulate2:DATA? to retrieve the signal wavelengths and powers.</p>
Query Response	<p>The following string is a typical example of six values returned when POWER is specified from a delta power measurement:</p> <pre>-7.42833100E+000,-1.00087200E+000,- 2.52121400E+000, -3.41918900E+000,-3.80437200E+000,-</pre>

6.36282900E+000

Notice that only measurement values are returned to the computer. There is no first value that indicates the number of values contained in the string as there is, for example, with FETCh, READ, and MEASure commands. Use CALCulate3:POINts? to determine the number of points CALCulate3:DATA? returns.

**:CALCulate3:DELTA:POWer[:STATe]**

**Command** :CALCulate3:DELTA:POWer[:STATe]{?|{ON | OFF | 1 | 0}}

Turns the delta-power measurement mode on and off. When this state is on, the power value of the reference laser line is subtracted from the power values of all laser lines except the reference. The power of the reference laser line is returned as an absolute power (unnormalized). The frequency data returned is the array of absolute frequency values.

**Attribute** Preset State: off  
**Summary** \*RST State: off  
 SCPI Compliance: instrument specific



Only one STATE command can be turned on at any one time. Attempting to turn more than one state on at a time results in a “-221 Settings Conflict” error.

**:CALCulate3:DELTA:PRESet**

Command	:CALCulate3:DELTA:PRESet
	Turns off all delta measurement states.
Attribute	Preset State: not affected
Summary	*RST State: not affected
	SCPI Compliance: instrument specific
	Command Only



**:CALCulate3:DELTA:REFERENCE:FREQUENCY**

<b>Command</b>	:CALCulate3:DELTA:REFERENCE:FREQUENCY{?  {<real>   MINimum   MAXimum}}
	Selects the reference laser line for DELTA calculations. The reference will be the laser line at the frequency closest to the frequency entered. Subsequent measurements will use the frequency closest to the reference frequency used for the previous measurement.
<b>&lt;real&gt;</b>	Frequency value within the following limits: MINimum: 181.6924 THz MAXimum: 236.0571 THz
<b>Attribute Summary</b>	Preset State: 236.0571 THz (1270 nm) *RST State: 236.0571 THz (1270 nm) SCPI Compliance: instrument specific
<b>Query Response</b>	The query returns the reference laser line frequency. The default units are Hz.

**:CALCulate3:DELTA:REFerence:POWer?**

Query	:CALCulate3:DELTA:REFerence:POWer? Queries the reference laser line's power level.
Attribute	Preset State: not affected
Summary	*RST State: not affected SCPI Compliance: instrument specific Query Only

**:CALCulate3:DELTA:REFERENCE[:WAVElength]**

<b>Command</b>	:CALCulate3:DELTA:REFERENCE[:WAVElength]{?  {<real>   MINimum   MAXimum}}
	Selects the reference laser line for DELTA calculations. The reference will be the laser line at the wavelength closest to the wavelength entered. Subsequent measurements will use the wavelength closest to the reference wavelength used for the previous measurement.
<real>	Wavelength value within the following limits: MINimum: 1270 nm MAXimum: 1650 nm
<b>Attribute Summary</b>	Preset State: 1270 nm (236.0571 THz) *RST State: 1270 nm (236.0571 THz) laser line SCPI Compliance: instrument specific
<b>Query Response</b>	The query returns the current wavelength of the reference laser line. The default units are meters.

**:CALCulate3:DELTA:REFerence:WNUMber**

<b>Command</b>	:CALCulate3:DELTA:REFerence:WNUMber{?  {<real>   MINimum   MAXimum}}
	Selects the reference laser line for delta calculations. The reference will be the laser line at the wave number closest to the wave number entered. Subsequent measurements will use the wave number closest to the reference wave number used for the previous measurement.
<real>	Wave number value within the following limits: MINimum: 6,060 /cm (1650 nm) MAXimum: 7,824 /cm (1270 nm)
<b>Attribute Summary</b>	Preset State: 7,824 /cm (1270 nm) *RST State: 7,824 /cm (1270 nm) SCPI Compliance: instrument specific
<b>Query Response</b>	The query returns the current wave number of the reference laser line. The default units are /m.

**:CALCulate3:DELTA:WAVelength[:STATe]**

Command	:CALCulate3:DELTA:WAVelength[:STATe]{?  {ON   OFF   1   0}}
	Turns the delta wavelength measurement mode on and off. When turned on, the wavelength of the reference laser line is subtracted from the wavelength values of all laser lines except the reference.
Attribute	Preset State: off
Summary	*RST State: off SCPI Compliance: instrument specific
Query Response	When this command is set to on, the power data returned from CALCulate3:DATA? is the array of absolute powers measured for each laser line, the frequency data is the array of frequency values normalized to the frequency of the reference laser line, and the frequency of the reference laser line is returned as an absolute frequency (unnormalized).



Only one STATe command can be turned on at any one time. Attempting to turn more than one state on at a time results in a “-221 Settings Conflict” error.

**:CALCulate3:DELTA:WPOWer[:STATe]****Command**

Turns the delta wavelength and power measurement mode on and off. When turned on, the wavelength of the reference laser line is subtracted from the wavelength values of all laser lines except the reference. The power value of the reference is subtracted from the power values of all laser lines except the reference.

**Attribute  
Summary**

Preset State: off  
\*RST State: off  
SCPI Compliance: instrument specific

**Query  
Response**

When this command is set to on, the power data returned from CALCulate3:DATA? is the array of powers normalized to the power of the reference laser line, the power of the reference laser line is returned as an absolute power (unnormalized), the frequency data is the array of frequency values normalized to the frequency of the reference laser line, and the frequency of the reference laser line is returned as an absolute frequency (unnormalized).



Only one STATE command can be turned on at any one time. Attempting to turn more than one state on at a time results in a “-221 Settings Conflict” error.

**:CALCulate3:DRIFT:DIFFerence[:STATe]**

Command	:CALCulate3:DRIFT:DIFFerence[:STATe]{?  {ON   OFF   1   0}}
	Sets the drift calculation to subtract the minimum values measured from the maximum values measured.
	Use :CALCulate3:DRIFT:PRESet to turn off all the drift states before turning on this state.
Attribute	Preset State: off
Summary	*RST State: off SCPI Compliance: instrument specific
Query	When this command is set to on, CALCulate3:DATA? returns
Response	the maximum power, frequency, or wavelength minus the minimum power, frequency, or wavelength.



Only one STATe command can be turned on at any one time. Attempting to turn more than one state on at a time results in a “-221 Settings Conflict” error.

**:CALCulate3:DRIFT:MAXimum[:STATe]**

**Command** :CALCulate3:DRIFT:MAXimum[:STATe]{?} {ON | OFF | 1 | 0}

Sets the drift calculation to return the maximum power, frequency, and wavelength values measured.

Use :CALCulate3:DRIFT:PRESet to turn off all the drift states before turning on this state.

**Attribute** Preset State: off

**Summary** \*RST State: off

SCPI Compliance: instrument specific

**Query** When this command is set to on, CALCulate3:DATA? returns  
**Response** the maximum power, frequency, or wavelength.



Only one STATe command can be turned on at any one time. Attempting to turn more than one state on at a time results in a “-221 Settings Conflict” error.



**:CALCulate3:DRIFt:MINimum[:STATe]**

**Command** :CALCulate3:DRIFt:MINimum[:STATe][?]{ON | OFF | 1 | 0}

Sets the drift calculation to return the minimum power, frequency, and wavelength values measured.

Use :CALCulate3:DRIFt:PRESet to turn off all the drift states before turning on this state.

**Attribute** Preset State: off

**Summary** \*RST State: off  
SCPI Compliance: instrument specific

**Query** When this command is set to on, CALCulate3:DATA? returns  
**Response** the minimum power, frequency, or wavelength.



Only one STATE command can be turned on at any one time. Attempting to turn more than one state on at a time results in a “-221 Settings Conflict” error.

### **:CALCulate3:DRIFt:PRESet**

**Command** :CALCulate3:DRIFt:PRESet  
Turns off all the drift states for DIFFerence, MAXimum, MINimum, and REFerence.

**Attribute** Preset State: not affected  
**Summary** \*RST State: not affected  
SCPI Compliance: instrument specific  
Command Only



When this command is set to on, the query CALCulate3:DATA? returns the difference between the current measurement and the reference.

**:CALCulate3:DRIFt:REFerence:RESet**

**Command**           :CALCulate3:DRIFt:REFerence:RESet

Places the current list of laser lines into the reference list.

**Attribute**           Preset State: not affected

**Summary**            \*RST State: not affected

SCPI Compliance: instrument specific

Command Only

**:CALCulate3:DRIFT:REFerence[:STATe]**

**Command** :CALCulate3:DRIFT:REFerence[:STATe][?]{ON | OFF | 1 | 0}

Turns on and off the drift reference state.

Use :CALCulate3:DRIFT:PRESet to turn off all the drift states before turning on the drift reference state.

**Attribute** Preset State: off

**Summary** \*RST State: off

SCPI Compliance: instrument specific

**Query Response** When this command is set to on, :CALCulate3:DATA? returns the reference laser lines.



Only one STATE command can be turned on at any one time. Attempting to turn more than one state on at a time results in a “-221 Settings Conflict” error.

**:CALCulate3:DRIFT[:STATe]**

**Command** :CALCulate3:DRIFT[:STATe]{?| {ON | OFF | 1 | 0}}

Turns on and off the drift measurement calculation.

When the drift mode is first turned on, the current list of laser lines is placed into the reference. All subsequent measurements take the new data, subtract the reference data, and display the differences in wavelengths and powers.

**Attribute** Preset State: off

**Summary** \*RST State: off

SCPI Compliance: instrument specific

**Query Response** When this command is set to on, CALCulate3:DATA? returns the power and frequency of the current measurement minus the power and frequency of the reference.



Only one STATe command can be turned on at any one time. Attempting to turn more than one state on at a time results in a “-221 Settings Conflict” error.

**:CALCulate3:FPERot[:STATe]**

**Command** :CALCulate3:FPERot[:STATe] {? | {ON | OFF | 1 | 0}}

Turns on and off the Fabry-Perot measurement mode.

When the state is ON, the Agilent 86122C measures characteristics of the Fabry-Perot laser modes. The modes are defined by the peak excursion and peak threshold commands.

**Attribute** Preset State: off

**Summary** \*RST State: off

SCPI Compliance: instrument specific



Only one STATe command can be turned on at any one time. Attempting to turn more than one state on at a time results in a “-221 Settings Conflict” error.

**:CALCulate3:FPERot:FWMH?**

<b>Query</b>	:CALCulate3:FPERot:FWMH{[:WAVelength]   :FREQuency   :WNUMber}?
	Queries the full width half-maximum data of the selected modes.
<b>Argument</b>	<p>WAVelength Queries the array of laser-line wavelengths after the calculation is completed. Returned values are in meters (m).</p> <p>FREQuency Queries the array of laser-line frequencies after the calculation is completed. Returned values are in Hz.</p> <p>WNUMber Queries the array of laser-line wave numbers after the calculation is completed. Returned values are in inverse meters (/m).</p>
<b>Attribute Summary</b>	<p>Preset State: not affected</p> <p>*RST State: not affected</p> <p>SCPI Compliance: instrument specific</p> <p>Query Only</p>
<b>Example Query Response</b>	<p>WAVelength +5.47128800E-009</p> <p>FREQuency +6.93436400E+011</p> <p>WNUMber +2.31306200E+001</p>

**:CALCulate3:FPERot:MEAN?**

<b>Query</b>	:CALCulate3:FPERot:MEAN[:WAVelength]   :FREQuency   :WNUMber)?
	Queries the mean data of the selected modes.
<b>Argument</b>	WAVelength Returns the mean wavelength of the selected modes.
	FREQuency Returns the mean frequency of the selected modes.
	WNUMber Returns the mean wave number of the selected modes.
<b>Attribute Summary</b>	Preset State: not affected *RST State: not affected SCPI Compliance: instrument specific Query Only
<b>Example Query Response</b>	WAVelength +1.53878000E-006 FREQuency +1.94824800E+014 WNUMber +6.49865400E+003



**:CALCulate3:FPERot:MODE:SPACing?**

<b>Query</b>	:CALCulate3:FPERot:MODE:SPACing[:WAVelength]   :FREQuency   :WNUMber)?
	Queries the mode spacing data of the selected modes.
<b>Argument</b>	WAVelength Queries the mode spacing wavelength after the calculation is completed. Returned values are in meters (m).
	FREQuency Queries the mode spacing frequency after the calculation is completed. Returned values are in Hz.
	WNUMber Queries the mode spacing wave number after the calculation is completed. Returned values are in inverse meters (/m).
<b>Attribute Summary</b>	Preset State: not affected *RST State: not affected SCPI Compliance: instrument specific Query Only
<b>Example Query Response</b>	WAVelength +3.18277200E- 010 FREQuency +4.02787400E+011 WNUMber +1.34356200E+001

**:CALCulate3:FPERot:PEAK?**

**Query** :CALCulate3:FPERot:PEAK{[:WAVelength] | :FREQuency | :WNUMber | :POWer{[:DBM]][:WATTs}}?

Queries the peak data of the selected modes.

**Argument** WAVelength  
Returns the peak wavelength of the selected modes.

FREQuency  
Returns the peak frequency of the selected modes.

WNUMber  
Returns the peak wave number of the selected modes.

POWer  
Returns the peak amplitude of the selected modes in dBm or watts.

**Attribute Summary** Preset State: not affected  
\*RST State: not affected  
SCPI Compliance: instrument specific  
Query Only

**ExampleQuery Response** WAVelength  
+1.54073400E-006

FREQuency  
+1.94577600E+014

WNUMber  
++6.49041000E+003

POWer  
-9.09446600E+000 (dBm), +1.23183800E-004 (watts)

**:CALCulate3:FPERot:POWer?**

<b>Query</b>	:CALCulate3:FPERot:POWer{[:DBM]:WATTs}}?
	Queries the total power data of the selected modes.
<b>Argument</b>	DBM Returns the total power in dBm.
	WATTs Returns the total power in watts.
<b>Attribute</b>	Preset State: not affected
<b>Summary</b>	*RST State: not affected SCPI Compliance: instrument specific Query Only
<b>ExampleQuery</b>	dBm (DBM)
<b>Response</b>	-4.46895600E+000
	watts (WATTs) +3.57358800E- 004

**:CALCulate3:FPERot:SIGMa?**

<b>Query</b>	:CALCulate3:FPERot:SIGMa{[:WAVelength]   :FREQuency   :WNUMber}?
	Queries the sigma data of the selected modes.
<b>Argument</b>	WAVelength Returns the sigma wavelength of the selected modes.
	FREQuency Returns the sigma frequency of the selected modes.
	WNUMber Returns the sigma wave number of the selected modes.
<b>Attribute</b>	Preset State: not affected
<b>Summary</b>	*RST State: not affected SCPI Compliance: instrument specific Query Only
<b>ExampleQuery</b>	WAVelength
<b>Response</b>	+2.32784700E-009
	FREQuency
	+2.94452900E+011
	WNUMber
	+9.82124900E+000

### **:CALCulate3:POINts?**

Query	:CALCulate3:POINts? Queries the number of points in the data set.
Attribute Summary	Preset State: not affected *RST State: not affected SCPI Compliance: instrument specific Query Only
Query Response	Returns the number of points returned by the CALCulate3:DATA? query.

### **:CALCulate3:PRESet**

<b>Command</b>	:CALCulate3:PRESet
	Turns off any CALCulate3 calculation (delta, drift, Fabry-Perot, or signal-to-noise) that is on.
<b>Attribute</b>	Preset State: not affected
<b>Summary</b>	*RST State: not affected SCPI Compliance: instrument specific Command Only

**:CALCulate3:SNR:AUTO**

<b>Command</b>	<p>:CALCulate3:SNR:AUTO{?  {ON   OFF   1   0}}</p> <p>Selects the reference frequency value for measuring noise in the signal-to-noise calculation.</p> <p>The command argument allows you to select either an internally generated or a user-entered frequency reference for measuring the noise. To enter a value to use as the reference, use:SNR:REFerence:FREQuency, :SNR:REFerence:WAVelength, and :SNR:REFerence:WNUMber.</p>
<b>Constant</b>	<p>ON Selects internally generated reference frequency.</p> <p>OFF Selects user-entered reference frequency.</p>
<b>Attribute Summary</b>	<p>Preset State: on</p> <p>*RST State: on</p> <p>SCPI Compliance: instrument specific</p>

**:CALCulate3:SNR:REference:FREQuency**

**Command** :CALCulate3:SNR:REference:FREQuency{?} {<real> | MINimum | MAXimum}}

Enters a frequency that can be used for the noise measurement reference in signal-to-noise calculations.

**<real>** Frequency value within the following limits:

MINimum: 181.6924 THz

MAXimum: 236.0571 THz

After entering the frequency value, use SNR:AUTO to configure the instrument to use this value in subsequent signal-to-noise calculations. The default units are Hz.

**Attribute Summary** Preset State: not affected  
 \*RST State: 193.4145 THz (1550.0 nm in a vacuum)  
 SCPI Compliance: instrument specific



**:CALCulate3:SNR:REFerence[:WAVelength]**

**Command** :CALCulate3:SNR:REFerence[:WAVelength][?]{<real> | MINimum | MAXimum}}

Sets the wavelength used for the noise measurement reference in the signal-to-noise calculation.

**<real>** Wavelength value within the following limits:

MINimum: 1270 nm

MAXimum: 1650 nm

After entering the wavelength value, use SNR:AUTO to configure the instrument to use this value in subsequent signal-to-noise calculations. The number entered is converted internally to the corresponding frequency. The default units are meters.

**Attribute Summary** Preset State: not affected  
 \*RST State: 1550.0 nm in a vacuum  
 SCPI Compliance: instrument specific

**:CALCulate3:SNR:REference:WNUMber**

<b>Command</b>	:CALCulate3:SNR:REference:WNUMber{? {<real>   MINimum   MAXimum}}
	Sets the wave number used for the noise measurement reference in the signal-to-noise calculation.
<b>&lt;real&gt;</b>	Wave number value within the following limits: MINimum: 6060 /cm (1270 nm) MAXimum: 7874 /cm (1650 nm)  After entering the wavelength value, SNR:AUTO to configure the instrument to use this value in subsequent signal-to-noise calculations. The wave number entered is converted internally to the corresponding frequency. The default units are /m.
<b>Attribute Summary</b>	Preset State: not affected *RST State: 6451.614 cm <sup>-1</sup> (1550 nm) SCPI Compliance: instrument specific

### **:CALCulate3:SNR[:STATe]**

**Command** :CALCulate3:SNR[:STATe]{?| {ON | OFF | 1 | 0}}  
Turns the signal-to-noise calculation on and off.

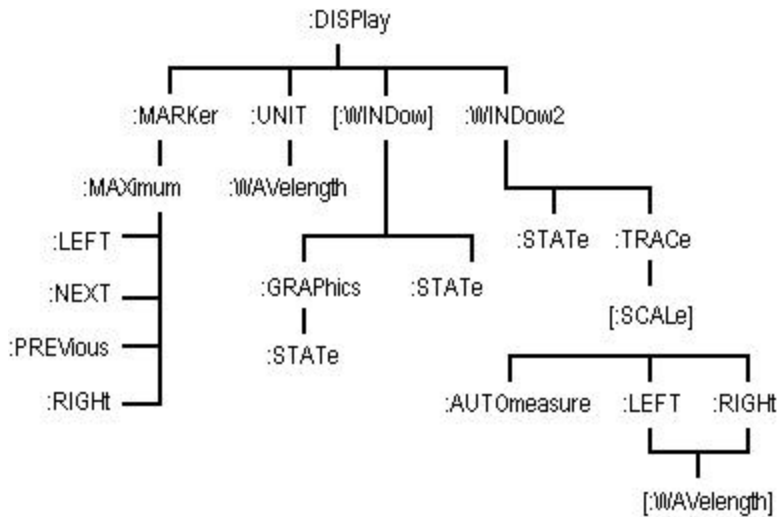
**Attribute** Preset State: off  
**Summary** \*RST State: off  
SCPI Compliance: instrument specific



Only one STATE command can be turned on at any one time. Attempting to turn more than one state on at a time results in a “-221 Settings Conflict” error.

## :DISPlay Subsystem

Use DISPlay commands to control the display of data on the instrument. This subsystem is divided into two windows: the default, tabular view (:WINDow) and the graphical display window (:WINDow2). The commands in this subsystem have the following command hierarchy:



See the following DISPlay commands for more information about the selected command.

- [MARKer:MAXimum](#)
- [MARKer:MAXimum:LEFT](#)
- [MARKer:MAXimum:NEXT](#)
- [MARKer:MAXimum:PREVIOUS](#)
- [MARKer:MAXimum:RIGHT](#)
- [UNIT:WAVelength](#)
- [WINDow:GRAPHics:STATe](#)
- [WINDow:STATe](#)
- [WINDow2:STATe](#)
- [WINDow2:TRACe\[:SCALE\]:AUTO](#)
- [WINDow2:TRACe\[:SCALE\]:LEFT\[:WAVelength\]](#)
- [WINDow2:TRACe\[:SCALE\]:RIGHT\[:WAVelength\]](#)

### **:DISPlay:MARKer:MAXimum**

Command	:DISPlay:MARKer:MAXimum
	Sets the marker to the laser line that has the maximum power.
Attribute	Preset State: marker set to maximum-power laser line
Summary	*RST State: marker set to maximum-power laser line SCPI Compliance: instrument specific Command Only

**:DISPlay:MARKer:MAXimum:LEFT**

**Command**           :DISPlay:MARKer:MAXimum:LEFT

Moves the marker from the current marker position to the next laser line having the following characteristics:

- ◆ shorter wavelength
- ◆ lower frequency
- ◆ lower wave number

If the display is in the List by Ampl mode, it will change to List by WL before the marker is moved.

**Attribute**           Preset State: marker set to maximum-power laser line  
**Summary**           \*RST State: marker set to maximum-power laser line  
SCPI Compliance: instrument specific  
Command Only

### **:DISPlay:MARKer:MAXimum:NEXT**

**Command**           :DISPlay:MARKer:MAXimum:NEXT

Moves the marker to the laser line with the next lower power level.

If the display is in the List by WL mode, it will change to List by Ampl before the marker is moved.

**Attribute**           Preset State: marker set to maximum-power laser line  
**Summary**            \*RST State: marker set to maximum-power laser line  
SCPI Compliance: instrument specific  
Command Only

**:DISPlay:MARKer:MAXimum:PREVious**

<b>Command</b>	:DISPlay:MARKer:MAXimum:PREVious
	Moves the marker to the laser line that has the next higher power level.
	If the display is in the List by WL mode, it will change to List by Ampl before the marker is moved.
<b>Attribute Summary</b>	Preset State: marker set to maximum-power laser line *RST State: marker set to maximum-power laser line SCPI Compliance: instrument specific Command Only



**:DISPlay:MARKer:MAXimum:RIGHT**

**Command**           :DISPlay:MARKer:MAXimum:RIGHT

Moves the marker from the current marker position to the next laser line having the following characteristics:

- ◆ longer wavelength
- ◆ higher frequency
  - higher wave number

If the display is in the List by Ampl mode, it will change to List by WL before the marker is moved.

**Attribute  
Summary**

Preset State: marker set to maximum-power laser line  
\*RST State: marker set to maximum-power laser line  
SCPI Compliance: instrument specific  
Command Only

**:DISPlay:UNIT:WAVelength**

**Command**           :DISPlay:UNIT:WAVelength { ? | THZ | NM | ICM }

Sets the wavelength units on the front panel display.

This command only affects the units displayed on the front panel and has no effect of the remote interface. It also takes precedence over other commands that would affect the units on the display.

**Attribute**           Preset State: NM

**Summary**            \*RST State: NM

SCPI Compliance: instrument specific

**:DISPlay[:WINDow]:GRAPhics:STATe**

Command	:DISPlay[:WINDow]:GRAPhics:STATe{?} {ON   OFF   1   0}
	Turns on and off the display of the power bars. When turned on, the power bars are displayed in all modes except the drift and signal-to-noise modes. When turned off, this command prevents the display of power bars for all instrument modes.
Attribute	Preset State: on
Summary	*RST State: on SCPI Compliance: instrument specific

**:DISPlay[:WINDow]:STATe**

<b>Command</b>	<code>:DISPlay[:WINDow]:STATe{?  {ON   OFF   1   0}}</code>  Turns the tabular display on and off. ON activates the tabular display. If the graphical display (:WINDow2) is on, the tabular display is shown below the graphical display. OFF turns off the tabular display and allows the graphical display to occupy the entire data display area.
<b>Attribute</b>	Preset State: on
<b>Summary</b>	*RST State: on SCPI Compliance: instrument specific

**:DISPlay:WINDow2:STATe**

Command	:DISPlay:WINDow2:STATe{?  {ON   OFF   1   0}}
	Turns the graphical display on and off. ON activates the graphical display. If the tabular display (:WINDow) is on, the graphical display is shown above the tabular display. OFF turns off the graphical display and allows the tabular display to occupy the entire data display area.
Attribute	Preset State: on
Summary	*RST State: on SCPI Compliance: instrument specific

**:DISPlay:WINDow2:TRACe[:SCALe]:AUTOmeasure**

**Command**           :DISPlay:WINDow2:TRACe[:SCALe]:AUTOmeasure

Initiates an AutoMeasure on the input signal.

The multi-wavelength meter will measure the input signal and format the display for optimal viewing. The split graph/table display will be selected. The graphical display limits will be adjusted to fit the entire input signal into the window. A tabular format will be selected based on the number of lines detected in the input signal.

**Attribute**           Preset State: none

**Summary**            \*RST State: none

SCPI Compliance: instrument specific

Command Only

**:DISPlay:WINDow2:TRACe[:SCALe]:LEFT[:WAVelength]**

<b>Command</b>	:DISPlay:WINDow2:TRACe[:SCALe]:LEFT[:WAVelength] {?}{<real>   MINimum   MAXimum}}
	Sets or queries the value represented by the left edge of the trace x-axis. If the left scale value given is greater than the right scale value, the right scale value will be set to the left scale value.
<b>&lt;real&gt;</b>	Wavelength value within the following limits: MINimum: 1270 nm MAXimum: :DISPlay:WINDow2:TRACe:SCALe:RIGHT value
<b>Attribute Summary</b>	Preset State: 1270 nm *RST State: 1270 nm SCPI Compliance: instrument specific

**:DISPlay:WINDow2:TRACe[:SCALe]:RIGHt[:WAVelength]**

**Command** :DISPlay:WINDow2:TRACe[:SCALe]:RIGHt[:WAVelength] {?|<real> | MINimum | MAXimum}}

Sets or queries the value represented by the right edge of the trace x-axis. If the right scale value given is less than the left scale value the left scale value will be set to the right scale value.

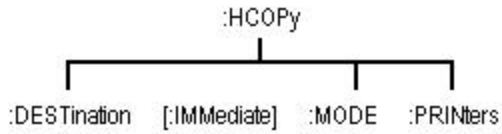
<real> Wavelength value within the following limits:  
 MINimum: :DISPlay:WINDow2:TRACe:SCALe:LEFT value  
 MAXimum: 1650 nm

**Attribute** Preset State: 1650 nm  
**Summary** \*RST State: 1650 nm  
 SCPI Compliance: instrument specific



## :HCOPy Subsystem

Use HCOPy commands to print the displayed measurement results to a printer. The commands in this subsystem have the following command hierachy:



See the following HCOPy commands for more information about the selected command.

- [DESTination](#)
- [\[:IMMediate\]](#)
- [MODE](#)
- [PRINters](#)

**:HCOPY:DESTination**

<b>Command</b>	:HCOPY:DESTination {<printer_number>   "<printer_string>"}
	Selects the default printer. Either number or string name can specify the default printer.
<printer_number>	Integer representing the attached printer. This number corresponds to the number returned with each printer name by HCOPY:PRINters?.
<printer_string>	Alphanumeric string representing the attached printer. The printer string must exactly match either the character strings in the File, Print Setup dialog boxes, or the strings returned by HCOPY:PRINters?.
<b>Attribute</b>	Preset State: none
<b>Summary</b>	*RST State: none SCPI Compliance: instrument specific

### **:HCOPY[:IMMEDIATE]**

Command :HCOPY[:IMMEDIATE]

Initiates a printout on the currently selected printer.

Printing to file is not supported. Do not use a printer driver that is configured to print to file.

Attribute Preset State: none  
Summary \*RST State: none  
SCPI Compliance: standard  
Command Only



**Non-sequential command:** Always use an \*OPC? query or a \*WAI command to ensure that this command has the time to complete before sending any more commands to the instrument.

**:HCOPY:MODE**

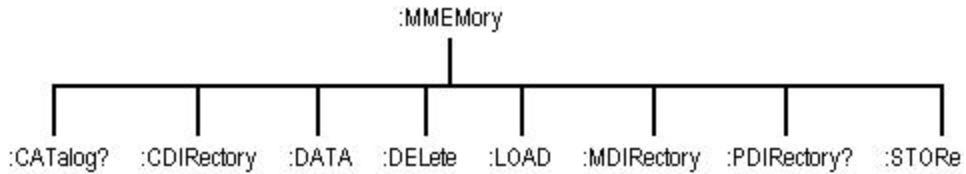
<b>Command</b>	:HCOPY:MODE {?   TABLE   SCREEN   ALL } Selects the output for all for hardcopy printouts.
<b>Constant</b>	TABLE Outputs all measured data in tabular form. SCREEN Outputs the graph and the portion of the tabular data that is on the screen. ALL Outputs both the tabular data and the screen.
<b>Attribute Summary</b>	Preset State: TABLE *RST State: TABLE SCPI Compliance: instrument specific

**:HCOPy:PRINters?**

<b>Query</b>	:HCOPy:PRINters?
	Queries the currently available printers.
<b>Attribute Summary</b>	Preset State: none *RST State: none SCPI Compliance: instrument specific Query Only
<b>Query Response</b>	Returns a list of the currently available printers.
<b>Returned Format</b>	<printer_count><NL><printer_data><NL>[,<printer_data><NL>]
<printer_count>	Number of printers currently installed.
<NL>	New line character.
<directory>	List of available printers, each line separated by a <NL>.
<b>Example</b>	3<NL>1,"HP LaserJet 8000 PS",DEF<NL>2,"HP LaserJet 4050 TN PS"<NL>3,"HP DeskJet 870Cxi"<NL><NL>

## :MMEMory Subsystem

Use MMEMory commands to manage the mass storage capabilities of this instrument. The commands in this subsystem have the following command hierarchy:



See the following MMEMory commands for more information about the selected command.

- [● CATalog?](#)
- [● LOAD](#)
- [● CDIRectory](#)
- [● MDIRectory](#)
- [● DATA](#)
- [● PWDirectory?](#)
- [● DElete](#)
- [● STORE](#)

**:MMEMory:CATalog?**

**Query** :MMEMory:CATalog? [“<directory>” | { ROOT | SIMages | DATA | SETups }]

Returns a listing of the contents of a directory.

The directory may be specified as a string, such as “D:\User Files\Data”, or as a parameter. If no parameter is used, a listing of the present working directory is returned.



The file times will be Greenwich Mean Time.

<b>Constant</b>	<p>ROOT Root of user directory structure on hard drive.</p> <p>SIMages Directory where screen images are stored.</p> <p>DATA Directory where all instrument and application generated data is stored.</p> <p>SETups Directory where instrument setup data is stored.</p>
<b>Attribute Summary</b>	<p>Preset State: none</p> <p>*RST State: none</p> <p>SCPI Compliance: instrument specific</p> <p>Query Only</p>
<b>Query Response</b>	<p>Returns the following:</p> <p>&lt;n&gt;&lt;NL&gt;&lt;directory&gt;</p>
<b>Example</b>	<pre>6 &lt;NL&gt; Directory of D:\User Files\Data&lt;NL&gt; &lt;NL&gt; 02/21/02 04:09 PM    0.45 KB LogData1.csv&lt;NL&gt; 03/05/02 02:13 PM    0.49 KB MeasData1.csv&lt;NL&gt; 03/05/02 02:13 PM    0.49 KB MeasData2.csv&lt;NL&gt; 03/05/02 02:13 PM    0.49 KB MeasData99.csv&lt;NL&gt;</pre>
<b>&lt;n&gt;</b>	Number of lines in the directory listing.
<b>&lt;NL&gt;</b>	New line character.
<b>&lt;directory&gt;</b>	List of directories and filenames, each line separated by an <NL>.

**:MMEMory: CDIRectory**

<b>Command</b>	[“<directory>”   { ROOT   SIMages   DATA   SETups } ] Changes the present working directory to the specified directory name. The desired directory can be specified either by name or by a parameter. “<directory>” is a character-quoted ASCII string, which can include the subdirectory designation. You must separate the directory name and any subdirectories with a backslash (\).
<b>&lt;filename&gt;</b>	ROOT Root of user directory structure on hard drive. SIMages Directory where screen images are stored. DATA Directory where all instrument and application generated data is stored. SETups Directory where instrument setup data is stored.
<b>Attribute Summary</b>	*RST State: none SCPI Compliance: instrument specific Command Only
<b>Query Response</b>	Returns an error if specified directory does not exist.



- To access external media like USB flash memory, USB hard disk drives or similar you can e.g. execute MMEM:CDIR “E:\”. Replace E: by the drive letter of your external media.
- If you attempt to execute :MMEM:CDIR “C:\”, the present working directory (PWD) is not changed.
- On the built-in drive D: the directory specified must be below “D:\User Files\”.



**:MMEMory:DATA**

**Command** :MMEMory:DATA {? "<filename>" | "<filename>",<data>

Loads the specified data into the file "<filename>". The destination directory is the working directory. <data> is in IEEE 4888.2 block data format.

This command operates only on files in the user file directory.

**<filename>** Name of the file. A maximum of 254 characters including the path name can be used. The filename assumes the present working directory if a path does not precede the file name.

An error will be generated if the filename includes subdirectories with the name of the file. An error will also be generated if a file exists with the same name as the working directory.

**<data>** The command can use either the IEEE 4888.2 data formats

- ◆ Definite Length Arbitrary Block Response Data
  - Indefinite Length Arbitrary Block Response Data

The :MMEMory:DATA query uses only the IEEE 4888.2 "Definite Length Arbitrary Block Response Data" format.

**Attribute Summary**  
 Preset State: none  
 \*RST State: none  
 SCPI Compliance: instrument specific  
 Command Only

### **:MMEMory:DELeTe**

<b>Command</b>	MMEMory:DELeTe "<filename>"  Deletes a specified file from the disk. An error is generated if the requested file does not exist.
<b>&lt;filename&gt;</b>	Name of the file. A maximum of 254 characters including the path name can be used. The filename assumes the present working directory if a path does not precede the file name.  The file name can include subdirectories with the name of the file.
<b>Attribute Summary</b>	Preset State: none *RST State: none SCPI Compliance: instrument specific Command Only



Prior to this command, you can change the present working directory by the ":MMEMory:CDIRectory" command.

**:MMEMory:LOAD**

**Command** :MMEMory:LOAD "<file\_name>"

Loads the specified instrument setup file from the disk.

**<file\_name>** Name of the file. A maximum of 254 characters including the path name can be used. The filename assumes the present working directory if a path does not precede the file name.

**Attribute** Preset State: none  
**Summary** \*RST State: none  
 SCPI Compliance: instrument specific  
 Command Only

**Query** Returns an error if the file does not exist.  
**Response**



Prior to this command, you can change the present working directory by the ":MMEMory:CDIRectory" command.

### **:MMEMory:MDIRectory**

**Command**           :MMEMory:MDIRectory "<directory>"

Creates a directory in the present working directory, with the designated directory name.

**Attribute**           Preset State: not affected

**Summary**            \*RST State: not affected

SCPI Compliance: instrument specific

Command Only



Prior to this command, you can change the present working directory by the ":MMEMory:CDIRectory" command.

**:MMEMory:PWDirectory?**

Query	:MMEMory:PWD?
	Returns the name of the present working directory, including the full path.
Attribute Summary	Preset State: none *RST State: none SCPI Compliance: instrument specific Query Only
Query Response	Returns the following: <present_working_directory><NL>
<NL>	New line character.

**:MMEMory:STORe**

**Command** :MMEMory:STORe <source>,"<file\_name>"

Stores an instrument setup or tabular results to the disk. The file name does not include a suffix. The instrument supplies the suffix.

<source> {TABLE | SETup}

<file\_name> Name of the file. A maximum of 254 characters including the path name can be used. The filename assumes the present working directory if a path does not precede the file name.

**Attribute** Preset State: none  
**Summary** \*RST State: none  
 SCPI Compliance: instrument specific  
 Command Only

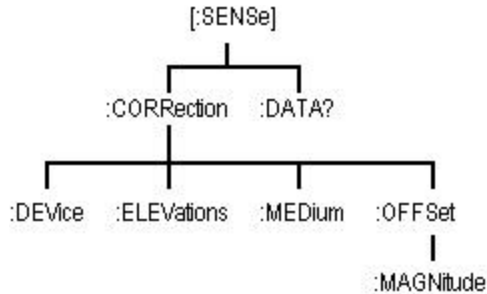
**Query** Returns an error if a file exists with the same name in the  
**Response** specified directory.



Prior to this command, you can change the present working directory by the ":MMEMory:CDIRectory" command.

## **[[:SENSe]] Subsystem**

Use SENSE commands to select between measurements in air or vacuum. You can also enter an amplitude offset. The commands in this subsystem have the following command hierarchy:



See the following SENSE commands for more information the selected command.

- [CORRection:DEvice](#)
- [CORRection:ELEVation](#)
- [CORRection:MEDium](#)
- [CORRection:OFFSet\[:MAGNitude\]](#)
- [DATA?](#)

## [:SENSe]:CORRection:DEVIce

Command	[:SENSe]:CORRection:DEVIce{?} {NARRow   BROad}
	Selects the wavelength measurement algorithm.
Constant	NARRow
	Selects wavelength measurements for narrowband devices such as DFB lasers and modes of FP lasers.
	BROad
	Selects wavelength measurements for broadband devices such as optical filters and LEDs.
	The narrow bandwidth algorithm, used for measuring lasers, determines the wavelength based upon the peak.
	The broad bandwidth algorithm, used for LEDs, filters, and chirped lasers, determines the wavelength based upon the center-of-mass of the power spectrum. The peak excursion function is used to determine the value of the integration limits. Care must be taken to ensure that the integration limits are above any noise. This is especially true when measuring devices with sloping noise floors, like an EDFA amplifier. For more information on peak excursion, refer to the <a href="#">PEXCursion</a> command.



Instrument specifications apply when the device is set to NARRow. Specifications do not apply in BROad mode.

Attribute	Non-sequential command
Summary	Preset State: NARRow *RST State: NARRow SCPI Compliance: instrument specific
Query Response	Returns the previously selected device. For example, NARRow



**Non-sequential command:** Always use an \*OPC? query or a \*WAI command to ensure that this command has the time to complete before sending any more commands to the instrument.



## [:SENSe]:CORRection:ELEVation



This command is available for backward compatibility to the 86120 family and no longer affects the accuracy, resolution or update rate of the instrument.

Command	[:SENSe]:CORRection:ELEVation{?} {<integer>   MINimum   MAXimum}}
	Sets the elevation value used by the instrument to compensate for air dispersion.
<integer>	Altitude in meters. MINimum: 0 m MAXimum: 5000 m
Attribute Summary	Non-sequential command Preset State: not affected *RST State: sets this value to the minimum SCPI Compliance: instrument specific
Query Response	Returns the current elevation setting as shown in the following example:  +1500



**Non-sequential command:** Always use an \*OPC? query or a \*WAI command to ensure that this command has the time to complete before sending any more commands to the instrument.

**[[:SENSe]:CORRection:MEDium**

<b>Command</b>	<pre>[[:SENSe]:CORRection:MEDium{?} {AIR   VACuum}}</pre> <p>Sets the Agilent 86122C to return wavelength readings in a vacuum or standard air.</p> <p>Standard air is defined to have the following characteristics:</p> <ul style="list-style-type: none"> <li>◆ barometric pressure - 760 torr</li> <li>◆ temperature - 15°C</li> <li>◆ relative humidity - 0%</li> </ul>
<b>Constant</b>	<p><b>AIR</b> Selects wavelength values in standard air.</p> <p><b>VACuum</b> Selects wavelength values in a vacuum.</p>
<b>Attribute Summary</b>	<p>Preset State: VAC *RST State: VAC SCPI Compliance: instrument specific</p>

**[[:SENSe]:CORRection:OFFSet[:MAGNitude]**

**Command**            [:SENSe]:CORRection:OFFSet:MAGNitude{?} {<real> | MINimum | MAXimum}}

Enters an offset for amplitude values.



With the Agilent 86122C, the power offset value is applied to measured power values. This includes data values displayed on the instrument front panel and those returned via remote commands (for example, CALC2:DATA? POW or FETC:SCAL:POW?). Please note that the Agilent 86120C applied this offset to front panel data and only to the CALC2:DATA? POW command when in the power weighted average mode (for example, CALC2:PWAV:STAT ON).

**<real>**                Logarithmic value in dB.  
                           MINimum: -40.0 dB  
                           MAXimum: 40.0 dB

**Attribute Summary**    Preset State: 0.0  
                           \*RST State: 0.0  
                           SCPI Compliance: instrument specific

**Query Response**       Returns the current offset setting as shown in the following example:  
                           +5.00000000E+000

**[[:SENSE]:DATA?**

## Query

[:SENSE]:DATA?

Queries the time domain samples of the input laser line.

Be prepared to process a large amount of data when this query is sent. The amount of data returned can be set using CALCulate1:TRANSform:FREQuency:POINts. The following number of values can be returned:

- ◆ MINimum:  $2^{16}$
- ◆ MAXimum:  $2^{17}$
- ◆ EXTended1:  $2^{18}$

The floating point values are scaled from 1.000 to 1.999756 ( $1 + 4095/4096$ ). Amplitude values are not calibrated.

The input laser line(s) generate an interference pattern on the photodetector as a function of the Michelson interferometer optical path delay. The time domain data is sampled at uniform optical path delay increments of the reference laser wavelength, or 0.632991 microns.

If your program is aborted or interrupted after sending this query, the Agilent 86122C continues to process the data but does not place it in the output buffer. Because of the amount of data processed, the instrument *will not respond* to any new commands in its input buffer *for 30 or 40 seconds*.

Attribute  
Summary

Preset State: none  
SCPI Compliance: instrument specific  
Query Only

Query  
Response

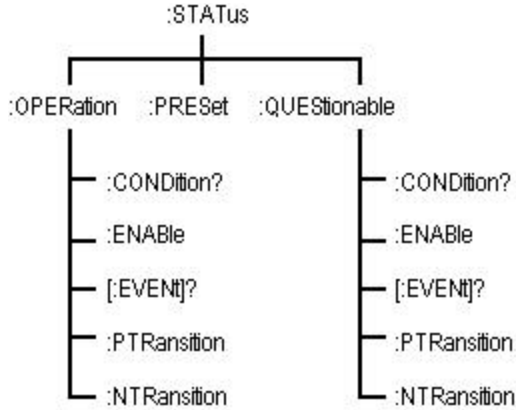
The following string shows an example of the first few measurements returned by this query.

```
+1.51367200E+000,+1.51855500E+000,+1.49902300E+000,+1.47949200E+000,+1.50488300E+000,+1.53320300E+000,+1.50097700E+000,+1.47265600E+000,+1.50293000E+000,+1.50781300E+000,+1.51171900E+000,+1.48242200E+000,+1.50097700E+000,+1.51855500E+000,+1.50683600E+000,+1.48632800E+000,+1.50488300E+000
```

Notice that only values are returned to the computer. There is no first value that indicates the number of values contained in the string as there is, for example, with FETCh, READ, and MEASure.

## :STATus Subsystem

Use STATus commands to control the Agilent 86122C's 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:



See the following STATus commands for more information about the selected command.

- [{OPERation | QUEStionable}:CONDition?](#)
- [{OPERation | QUEStionable}:NTRansition](#)
- [{OPERation | QUEStionable}:ENABle](#)
- [{OPERation | QUEStionable}:PTRansition](#)
- [{OPERation | QUEStionable}:EVENT\]](#)
- [PRESet](#)

**:STATus:{OPERation | QUEStionable}:CONDition?**

Query	:STATus:{OPERation   QUEStionable}:CONDition?  Queries the value of the questionable or operation condition register. Use this query to read the value of the <a href="#">OPERation Status</a> or <a href="#">QUEStionable Status registers</a> .
Attribute Summary	Preset State: none *RST State: none SCPI Compliance: standard Query Only
Query Response	0 to 32767
Example	OUTPUT 720;":STATUS:OPERATION:CONDITION?"

**:STATus:{OPERation | QUEStionable}:ENABle**

Command	:STATus:{OPERation   QUEStionable}:ENABle{?  <value>}
	Sets the enable mask for the questionable or operation event 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 or bit 7 for the operation status) in the status byte will be set.
<integer>	Number from 0 to 65535.
Attribute	Preset State: none
Summary	*RST State: none SCPI Compliance: standard
Query Response	The largest value that can be returned is 32767, because the most-significant register bit cannot be set true.
Example	OUTPUT 720;":STATus:QUEStionable:ENABle 1024"

**:STATus:{OPERation | QUEStionable}[:EVENT]**

<b>Command</b>	:STATus:{OPERation   QUEStionable}:EVENT?
	Queries the contents of the questionable or operation event registers.
<b>Attribute Summary</b>	Preset State: none *RST State: none SCPI Compliance: standard
<b>Query Response</b>	Returns a number from 0 to 32767 indicating which bits are set. Reading the register clears the register.
<b>Example</b>	OUTPUT 720;”:STATUS:OPERATION:EVENT?”



**:STATus:{OPERation | QUEStionable}:NTRansition**

<b>Command</b>	:STATus:OPERation:NTRansition{?} <integer>
	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 cause the associated OPERation Status or QUEStionable Status register bits 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
<integer>	Number from 0 to 65535.
<b>Attribute Summary</b>	Preset State: none *RST State: none SCPI Compliance: standard
<b>Query Response</b>	When queried, the largest value that can be returned is 32767 because the most significant register bit cannot be set true.
<b>Example</b>	OUTPUT 720;":STATus:OPER:NTRansition 16"

**:STATus:{OPERation | QUEStionable}:PTRansition**

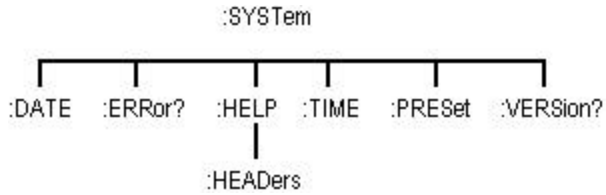
<b>Command</b>	:STATus:OPERation:PTRansition{?  <integer>}  Selects bits in the event register which can be set by positive transitions of the corresponding bits in the condition register.  Changes in the state of a condition register bit cause the associated OPERation Status or QUEStionable Status event register bits 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.
<integer>	Number from 0 to 65535.
<b>Attribute Summary</b>	Preset State: none *RST State: none SCPI Compliance: standard
<b>Query Response</b>	When queried, the largest value that can be returned is 32767 because the most significant register bit cannot be set true.
<b>Example</b>	OUTPUT 720;":STATus:OPER:PTRansition 16"

**:STATus:PRESet**

Command	:STATus:PRESet
	Presets the enable registers and the PTRansition and NTRansition filters.
	This Preset command is defined by SCPI to affect the enable register. If you want to clear all event registers and queues, use <a href="#">*CLS</a> .
Attribute Summary	Preset State: none *RST State: none SCPI Compliance: standard Command Only
Example	OUTPUT 720;”:STATUS:PRESET”

## :SYSTem Subsystem

Use SYSTem commands to preset the Agilent 86122C, set instrument utilities, and query error messages. The commands in this subsystem have the following command hierarchy:



See the following SYSTem commands for more information the selected command.

- [DATE](#)
- [PRESet](#)
- [ERRor?](#)
- [TIME](#)
- [HELP:HEADers?](#)
- [VERsion](#)

**:SYSTem:DATE**

Command	:SYSTem:DATE {?   <year>,<month>,<day>}
	Sets or queries the system date. This command is not affected by <u>*RST</u> .
Attribute Summary	Preset State: none *RST State: none SCPI Compliance: standard
Query Response	<year><month><day><NL>
<year>	Specifies the year in the format <yyyy>.
<month>	Specifies the month in the format <1,2,...12>   <JAN, FEB, MAR,...>.
<day>	Specifies the day in the format, for example, <1..31>.

**ERRor?**

Query :SYSTem:ERRor?

Queries an error from the error queue.

The Agilent 86122C has a 30 entry error queue. The queue is a first-in, first-out buffer. If you repeatedly send the SYSTEM:ERROR?, the instrument returns the error numbers and descriptions in the order in which they occur until the queue is empty. Any further queries return +0, "No errors" until another error occurs.

Attribute Preset State: none  
 Summary \*RST State: none  
 SCPI Compliance: standard  
 Query Only

Query Response <value>, <string>

The following is an example of a response:

-113,"Undefined header"

<value> Integer value that represents the error message number.

<string> Quoted text of the error message.

Example DIM Error\$[250]  
 OUTPUT 720;":SYSTEM:ERROR?"  
 ENTER 720;Error\$  
 PRINT Error\$

See [Error Messages](#) to view a complete list of error messages.

**:SYSTem:HELP:HEADers?**

Query :SYSTem:HELP:HEADers?

Queries a listing of all the remote programming commands available for the Agilent 86122C.

Attribute Preset State: none  
 Summary \*RST State: none  
 SCPI Compliance: standard  
 Query Only

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.

Each command in the listing is separated by a line feed character.



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.

```
#44387
:ABORt/nquery/
:CALCulate:DATA?/qonly/
:CALCulate:TRANSform:FREQuency:POINts
:CALCulate1:DATA?/qonly/
:CALCulate1:TRANSform:FREQuency:POINts
:CALCulate2:DATA?/qonly/
.
.
.
*IDN?/qonly/
*OPC
*RCL/nquery/
*RST/nquery/
*SAV/nquery/
*SRE
*STB?/qonly/
*TRG/nquery/
*TST?/qonly/
*WAI/nquery/
```

## **:SYSTem:PRESet**

**Command**           :SYSTem:PRESet  
Returns the Agilent 86122C to a known condition.

**Attribute**           Preset State: none  
**Summary**            \*RST State: none  
SCPI Compliance: standard  
Command Only



**:SYSTem:TIME**

Command	:SYSTem:TIME {?   <hour>,<minute>,<second>}
	Sets or queries the system time. This command is not affected by <a href="#">*RST</a> .
Attribute Summary	Preset State: none *RST State: none SCPI Compliance: standard
Query Response	<hour><minute><second><NL>
<hour>	Specifies the hour in the format <0..23>.
<minute>	Specifies the minute in the format <0..59>.
<second>	Specifies the second in the format <0..59>.

## **:SYSTem:VERSion?**

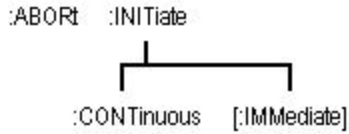
Query           :SYSTem:VERSion

Queries the version of SCPI that the instrument complies with. The SCPI version used in the Agilent 86122C is 1999.0.

Attribute       Preset State: none  
Summary        \*RST State: none  
                  SCPI Compliance: standard  
                  Query Only

## [:TRIGger] Subsystem

Use TRIGger commands to stop the current measurement and allow the instrument to acquire new measurement data. You can also use the TRIGger commands to select single or continuous acquisition of measurement data. The commands in this subsystem have the following command hierarchy:



The SCPI definition defines the :TRIGger subsystem to include :ABORt, :ARM, :INITiate, and :TRIGger commands. The Agilent 86122C uses no :ARM or :TRIGger commands.

See the following TRIGger commands for more information about the selected command.

- [ABORt](#)
- [INITiate:CONTInuous](#)
- [INITiate\[:IMMediate\]](#)

## **:ABORt**

### **Command**

:ABORt

Halts the current measurement sequence and places the instrument in the idle state.

If the instrument is configured for continuous measurements, a new measurement sequence will begin. Otherwise, the instrument stays in the idle state until a new measurement is initiated.

### **Attribute**

Preset State: not affected

### **Summary**

SCPI Compliance: standard

Command Only

### **:INITiate:CONTinuous**

**Command**           :INITiate:CONTinuous{?| {ON | OFF | 1 | 0}}

Selects single or continuous measurement acquisition mode. When ON, the instrument continuously measures the input spectrum.

**Attribute**           Non-sequential command  
**Summary**            Preset State: on  
                      \*RST State: off  
                      SCPI Compliance: standard



**Non-sequential command:** Always use an \*OPC? query or a \*WAI command to ensure that this command has the time to complete before sending any more commands to the instrument.

## **:INITiate[:IMMediate]**

Command :INITiate:IMMediate

Initiates a new measurement sequence.

Attribute Non-sequential command

Summary Preset State: none  
SCPI Compliance: standard  
Command Only



**Non-sequential command:** Always use an \*OPC? query or a \*WAI command to ensure that this command has the time to complete before sending any more commands to the instrument.

## :UNIT Subsystem

Use UNIT commands to set the amplitude units to watts or dBm. The commands in this subsystem have the following command hierarchy:

```
:UNIT
|
[:POWer]
```



Use the [:DISPlay:UNIT:WAVelength](#) command, to set the wavelength units.

---

### [:POWer]

Command	:UNIT[:POWer]{?  {W   DBM}}
	Sets the power units to watts (linear) or dBm (logarithmic).
Attribute	Preset State: dBm
Summary	*RST State: dBm
	SCPI Compliance: standard

## List of Commands



**Codes:** “S” indicates a standard SCPI command. ”I” indicates an instrument specific command.

Common Command	Description	Code
*CLS	Clears all event registers and the error queue.	
*ESE	Sets the bits in the standard-event status enable register.	
*ESR?	Queries value standard-event status register.	
*IDN?	Queries instrument model number and firmware version.	
*OPC	Sets operation complete bit of the standard-event status register.	
*RCL	Recalls a saved instrument state.	
*RST	Resets instrument.	
*SAV	Saves an instrument state.	
*SRE	Sets bits in service-request enable register.	
*STB	Queries value of status byte.	
*TRG	Triggers acquisition of measurement data.	
*TST?	Performs an instrument self test.	
*WAI	Causes instrument to finish processing current command before continuing.	
Measurement Instructions	Description	Code
:CONFigure	Configures instrument for wavelength, wavenumber, frequency, power, and measurements.	I
:FETCh	Queries wavelength, wavenumber, frequency, power, and measurements that have already been captured.	I
:MEASure	Configures, measures, and queries wavelength, wavenumber, frequency, power, and measurements.	I
:READ	Measures and queries wavelength, wavenumber, frequency, power, and	I



		measurements.	
<b>CALCulate1 Subsystem</b>	<b>Description</b>		<b>Code</b>
:CALCulate1:DATA?	Queries the uncorrected frequency-spectrum data of the input signal.		S
:CALCulate1:TRANSform:FREQuency:POINts?	Sets and queries the number of points in the data set.		S
<b>CALCulate2 Subsystem</b>	<b>Description</b>		<b>Code</b>
:CALCulate2:DATA?	Queries the corrected frequency-spectrum data of the input signal.		S
:CALCulate2:PEXCursion	Sets the peak excursion limit.		I
:CALCulate2:POINts?	Queries the number of points in the data set.		I
:CALCulate2:PTHReshold	Sets the peak threshold limit.		I
:CALCulate2:PWAVerage[:STATe]?	Places the instrument in the average-wavelength mode. Data queries return the power-weighted average frequency, wavelength, or wavenumber or total power.		I
:CALCulate2:WLIMit[:STATe]	Turns wavelength limiting on and off.		I
:CALCulate2:WLIMit:START:FREQuency	Sets the starting frequency for the wavelength limit range.		I
:CALCulate2:WLIMit:START[:WAVelength]	Sets the starting wavelength for the wavelength limit range.		I
:CALCulate2:WLIMit:START:WNUMber	Sets the starting wavenumber for the wavelength limit range.		I
:CALCulate2:WLIMit:STOP:FREQuency	Sets the stopping frequency for the wavelength limit range.		I
:CALCulate2:WLIMit:STOP[:WAVelength]	Sets the stopping wavelength for the wavelength limit range.		I
:CALCulate2:WLIMit:STOP:WNUMber	Sets the stopping wavenumber for the wavelength limit range.		I
<b>CALCulate3 Subsystem</b>	<b>Description</b>		<b>Code</b>
:CALCulate3:ASNR:CLEAr	Resets and restarts the signal-to-noise ratio averaging.		I
:CALCulate3:ASNR:COUNT	Sets the number of measurements to		I

:CALCulate3:ASNR:COUNT	average the signal-to-noise ratio.	
:CALCulate3:ASNR:[:STATe]	Turns signal-to-noise ratio averaging mode on and off.	I
:CALCulate3:DATA?	Queries the data resulting from delta, drift, and signal-to-noise measurements.	S
:CALCulate3:DELTA:POWer [:STATe]	Turns the delta power measurement mode on and off.	I
:CALCulate3:DELTA:REFerence: FREQUency	Selects the signal to be used as the reference for the DELTA calculations.	I
:CALCulate3:DELTA:REFerence: POWer?	Queries the power level of the reference signal.	I
:CALCulate3:DELTA:REFerence [:WAVelength]	Selects the signal to be used as the reference for the DELTA calculations.	I
:CALCulate3:DELTA:REFerence: WNUMber	Selects the signal to be used as the reference for the DELTA calculations.	I
:CALCulate3:DELTA:WAVelength [:STATe]	Turns the delta wavelength measurement mode on and off.	I
:CALCulate3:DELTA:WPOWer [:STATe]	Turns the delta wavelength and power measurement mode on and off.	I
:CALCulate3:DRIFT:DIFFerence [:STATe]	Sets the drift calculation to subtract the minimum values measured from the maximum values measured.	I
:CALCulate3:DRIFT:MAXimum [:STATe]	Sets the drift calculation to return the maximum power (frequency) values measured.	I
:CALCulate3:DRIFT:MINimum [:STATe]	Sets the drift calculation to return the minimum power (frequency) values measured.	I
:CALCulate3:DRIFT:PRESet	Turns off all the drift states for DIFFerence, MAXimum, MINimum, and REFerence.	I
:CALCulate3:DRIFT:REFerence: RESet	Places the current list of signals into the reference list.	I
:CALCulate3:DRIFT:REFerence [:STATe]	Turns the drift state on and off so that CALC3:DATA? will return the reference signal list.	I
:CALCulate3:DRIFT[:STATe]	Turns the drift measurement calculation on and off.	I

:CALCulate3:FPERot[:STATe]	Turns the Fabry-Perot measurement mode on and off.	I
:CALCulate3:FPERot:MEAN[:WAVelength]?	Queries the mean wavelength of the selected modes.	I
:CALCulate3:FPERot:MEAN:FREQuency?	Queries the mean frequency of the selected modes.	I
:CALCulate3:FPERot:MEAN:WNUMber?	Queries the mean wavenumber of the selected modes.	I
:CALCulate3:FPERot:PEAK[:WAVelength]?	Queries the peak wavelength of the selected modes.	I
:CALCulate3:FPERot:PEAK:FREQuency?	Queries the peak frequency of the selected modes.	I
:CALCulate3:FPERot:PEAK:WNUMber?	Queries the peak wavenumber of the selected modes.	I
:CALCulate3:FPERot:PEAK:POWer?	Queries the peak power of the selected modes.	I
:CALCulate3:FPERot:FWMH[:WAVelength]?	Queries full width half-max wavelength of selected modes.	I
:CALCulate3:FPERot:FWMH:FREQuency?	Queries full width half-max frequency of selected modes.	I
:CALCulate3:FPERot:FWMH:WNUMber?	Queries full width half-max wavenumber of selected modes.	I
:CALCulate3:FPERot:MODE:SPACing[:WAVelength]?	Queries the mode spacing wavelength of the selected modes.	I
:CALCulate3:FPERot:MODE:SPACing:FREQuency?	Queries the mode spacing frequency of the selected modes.	I
:CALCulate3:FPERot:MODE:SPACing:WNUMber?	Queries the mode spacing wavenumber of the selected modes.	I
:CALCulate3:FPERot:POWer[:WAVelength]?	Queries the power wavelength of the selected modes.	I
:CALCulate3:FPERot:POWer:FREQuency?	Queries the power frequency of the selected modes.	I
:CALCulate3:FPERot:POWer:WNUMber?	Queries the power wavenumber of the selected modes.	I
:CALCulate3:FPERot:SIGMa	Queries the sigma wavelength of the selected modes.	I

[:WAVelength]?	selected modes.	
:CALCulate3:FPERot:SIGMa:FREQUency?	Queries the sigma frequency of the selected modes.	I
:CALCulate3:FPERot:SIGMa:WNUMber?	Queries the sigma wavenumber of the selected modes.	I
:CALCulate3:POINts?	Queries the number of points in the data set.	I
:CALCulate3:PRESet	Turns off any CALCulate3 calculation that is on.	I
:CALCulate3:SNR:AUTO	Selects the internal or externally entered frequency value for the noise measurement reference in the SNR calculation.	I
:CALCulate3:SNR:REFerence:FREQUency	Sets the frequency used for the noise measurement reference in the SNR calculation.	I
:CALCulate3:SNR:REFerence[:WAVelength]	Sets the wavelength used for the noise measurement reference in the SNR calculation.	I
:CALCulate3:SNR:REFerence:WNUMber	Sets the wave number used for the noise measurement reference in the SNR calculation.	I
:CALCulate3:SNR[:STATe]	Turns the SNR calculation on and off.	I
<b>DISPlay Subsystem</b>	<b>Description</b>	<b>Code</b>
:DISPlay:MARKer:MAXimum	Sets the marker to the signal with the largest power.	I
:DISPlay:MARKer:MAXimum:LEFT	Moves marker to signal with the next lower wavelength or frequency.	I
:DISPlay:MARKer:MAXimum:NEXT	Moves the marker to the signal with the closest power level just below the power level of the signal at the current marker position.	I
:DISPlay:MARKer:MAXimum:PREVIOUS	Moves the marker to the signal with the closest power level just above the power level of the signal at the current marker position.	I
:DISPlay:MARKer:MAXimum:RIGHT	Moves marker to the next higher wavelength or frequency.	I
:DISPlay[:WINDow]:GRAPhics:STATe	Turns the instrument display of the power bars on and off.	S

<b>HCOPy Subsystem</b>	<b>Description</b>	<b>Code</b>
:HCOPy:DESTination	Selects the default printer.	S
:HCOPy[:IMMEDIATE]	Starts a printout.	S
:HCOPy:MODE	Selects the output for all for hardcopy printouts.	S
:HCOPy:PRINters?	Queries the currently available printers.	S
<b>SENSe Subsystem</b>	<b>Description</b>	<b>Code</b>
:SENSe:CORRection:DEVice	Configures wavelength measurements for narrowband or wideband devices.	I
:SENSe:CORRection:ELEVation	Sets the elevation value used by the instrument to compensate for air dispersion.	I
:SENSe:CORRection:OFFSet: MAGNitude	Sets the power offset value used by the instrument.	S
:SENSe:CORRection:MEDIum	Sets the instrument to return the wavelength reading in a vacuum when the parameter is on. Parameters are VAC and AIR.	I
:SENSe:DATA?	Queries the time domain samples of the input signal.	I
<b>STATus Subsystem</b>	<b>Description</b>	<b>Code</b>
:STATus:{OPERation   QUESTionable}:CONDition?	Returns the value for the condition register for the node.	S
:STATus:{OPERation   QUESTionable}:EVENT?	Returns the value of the event register for the node.	S
:STATus:{OPERation   QUESTionable}:ENABle	Sets the enable register.	S
:STATus:{OPERation   QUESTionable}:PTRansition	Sets the positive transition filter register.	S
:STATus:{OPERation   QUESTionable}:NTRansition	Sets the negative transition filter register.	S
:STATus:PRESet	Presets the enable registers for all status nodes.	S
<b>SYSTem Subsystem</b>	<b>Description</b>	<b>Code</b>
:SYSTem:ERRor?	Queries an error from the error queue.	S
:SYSTem:HELP:HEADers?	Queries an ASCII listing of all Agilent 86120C remote commands.	I

:SYSTem:PRESet	Performs the equivalent of a front-panel PRESET key press.	S
:SYSTem:VERSion	Queries the version of SCPI with which this instrument is compliant.	S
<b>TRIGger Subsystem</b>	<b>Description</b>	<b>Code</b>
:ABORt	Stops the current measurement sequence.	S
:INITiate:IMMEDIATE	Places the instrument into the initiated state and initiates a new measurement sequence.	S
:INITiate:CONTinuous	Sets the instrument for single or continuous measurement.	S
<b>UNIT Subsystem</b>	<b>Description</b>	<b>Code</b>
:UNIT[:POWer]	Sets the power units to watts (linear) or dBm (logarithmic).	S

## Example Programs

The following programs provide examples of remote programming commands used in typical situations. These programs are not meant to teach general programming techniques or provide ready-to-use solutions. They provide an understanding of how measurements are performed and how to return data to the computer.



All of the following examples are written in the HP BASIC programming language.

- ◆ [Measuring a DFB laser](#)
- ◆ [Measuring WDM channels](#)
- ◆ [Measuring WDM channel drift](#)
- ◆ [Measuring WDM channel separation](#)
- ◆ [Measuring signal-to-noise ratio of each WDM channel](#)
  - [Increasing a source's wavelength accuracy](#)

---

## \*TRG - Trigger

Command      \*TRG

Acquires data according to the current settings. If a measurement is already in progress, a trigger is ignored, and an error is generated.

Example        The following example starts the data acquisition according to the current settings.

```
OUTPUT 720;”*TRG”
```



**Non-sequential command:** Always use an \*OPC? query or a \*WAI command to ensure that this command has the time to complete before sending any more commands to the instrument.

---

## \*TST? - Test

Query          \*TST?

Starts a self-test on the instrument. The result of the test is placed in the output queue. A zero indicates the test passed and a non-zero value indicates the test failed. The instrument will initiate a measurement and check for any hardware errors after the measurement is complete.

Example        OUTPUT 720;”\*TST?”  
ENTER 720;Result  
PRINT Result

---

## \*WAI - Wait

Command      \*WAI

Prevents the instrument from executing any further commands until the current command and all pending commands have completed.



```
ENTER @Mwm;Identity$  
RETURN Identity$  
FNEND
```

## Measuring WDM Channels

The following example program measures the multiple laser lines of a WDM system. It measures both the power and wavelengths of each line. First, the program sets the Agilent 86122C in the single acquisition measurement mode. It then triggers the Agilent 86122C with the MEASure command to capture measurement data of the input spectrum. Because the data is stored in the instrument's memory, it can be queried as needed.

This example uses the [Set\\_ese](#), [Error\\_msg](#), [Err\\_mngmt](#), and [FNIdentity function](#) subroutines.

```

COM /Instrument/ @Mwm
ASSIGN @Mwm TO 720
ON ERROR GOTO Error_msg,
Set_ese
PRINT USING "37A,33A";"Multi-Wavelength Meter Identity is : ";FNIdentity$

OUTPUT @Mwm;":INIT:CONT OFF"
ON TIMEOUT 7,5 CALL Err_mngmt
OUTPUT @Mwm;":MEAS:ARR:POW:WAV?"
ENTER @Mwm USING "#,K";Nb_wl
ALLOCATE Current_wl(1:Nb_wl)
ENTER @Mwm USING "#,K";Current_wl(*)
OUTPUT @Mwm;":FETC:ARR:POW?"
ENTER @Mwm USING "#,K";Nb_wl
ALLOCATE Current_pwr(1:Nb_wl)
ENTER @Mwm USING "#,K";Current_pwr(*)
FOR I=1 TO Nb_wl
    PRINT USING "22A,2D,6A,4D.2DE,4A,S2D.2D,3A";"The wavelength number ";Current_wl(I);" at ";Current_pwr(I);"dBm"
NEXT I
OFF TIMEOUT
STOP

Error_msg:
    PRINT "the prgm is aborted due to : ";ERRM$
END

Err_mngmt:SUB Err_mngmt
    COM /Instrument/ @Mwm
    DIM Err_msg$[255]
    INTEGER Cme
    CLEAR 7
    REPEAT
        OUTPUT @Mwm; "**ESR?"
        ENTER @Mwm;Cme
        OUTPUT @Mwm; ":SYST:ERR?"
        ENTER @Mwm;Err_msg$
        PRINT Err_msg$
        UNTIL NOT BIT(Cme,2) AND NOT BIT(Cme,4) AND NOT BIT(Cme,5) AND Err$,"+0")
    Subend:SUBEND

```

```
Set_ese:SUB Set_ese
  COM /Instrument/ @Mwm
  OUTPUT @Mwm; "*"ESE";IVAL("00110100",2)
SUBEND
```

```
Identity:DEF FNIdentity$;
  COM /Instrument/ @Mwm
  DIM Identity$[50]
  Identity$=""
  OUTPUT @Mwm;"*RST"
  OUTPUT @Mwm;"*OPC?"
  ENTER @Mwm;Opc_done
  OUTPUT @Mwm;"*IDN?"
  ENTER @Mwm;Identity$
  RETURN Identity$
FNEND
```

## Measuring WDM Channel Drift

The following example program measures the drift of channels in a WDM system. It measures drift in both power and wavelength of each line. First, the program sets the Agilent 86122C in the continuous-acquisition measurement mode. It then measures drift using commands from the [CALCulate3 subsystem](#).

This example uses the [Error\\_msg](#), [Set\\_ese](#), [Err\\_mngmt](#), [Cmd\\_opc](#), [Tempo](#), and [FNIdentity function](#) subroutines.

You will notice the use of the Tempo subroutine to pause the program for 10 seconds while the instrument measures the drift on the system.



The use of the Err\_mngmt subroutine is optional. Err\_mngmt was modified for the programming example to allow this subroutine to indicate the last command that was sent to the instrument before an error occurred.

```
COM /Instrument/ @Mwm
ASSIGN @Mwm TO 720
DIM Key$[1]
ON ERROR GOTO Error_msg
Set_ese
PRINT USING "37A,33A";"Multi-Wavelength Meter Identity is: " ;FNIdentity$
ON TIMEOUT 7,5 CALL Err_mngmt
Cmd_opc("**RST")
Cmd_opc("INIT:IMM")
Cmd_opc("**OPC")
Cmd_opc("CONF:ARR:POW:WAV")
```

*! Turn on the drift calculation*

```
Cmd_opc("CALC3:DRIF:STAT ON")
Err_mngmt("CALC3:DRIF:STAT ON")
```

*! Turn off all drift states*

```
Cmd_opc("CALC3:DRIF:PRES")
Err_mngmt("CALC3:DRIF:PRES")
```

*! Turn on drift reference state*

```
Cmd_opc("CALC3:DRIF:REF:STAT ON")
Err_mngmt("CALC3:DRIF:REF:STAT ON")
```

*! Query the number of data points*

```
OUTPUT @Mwm;"CALC3:POIN?"
ENTER @Mwm USING "#,K";Nb_pt
ALLOCATE Current_ref_wl(1:Nb_pt)
```

```
ALLOCATE Current_ref_pwr(1:Nb_pt)
```

```
! Query reference wavelengths and powers
```

```
OUTPUT @Mwm;":CALC3:DATA? WAV"
ENTER @Mwm USING "#,K";Current_ref_wl(*)
OUTPUT @Mwm;":CALC3:DATA? POW"
ENTER @Mwm USING "#,K";Current_ref_pwr(*)
```

```
! Turn off drift reference state
```

```
Cmd_opc("CALC3:DRIF:REF:STAT OFF")
Err_mngmt("CALC3:DRIF:REF:STAT OFF")
```

```
! Turn on drift max min calculation
```

```
Cmd_opc("CALC3:DRIF:DIFF:STAT ON")
Err_mngmt("CALC3:DRIF:DIFF:STAT ON")
```

```
Tempo(10)
```

```
ALLOCATE Current_diff_wl(1:Nb_pt)
ALLOCATE Current_diff_pw(1:Nb_pt)
```

```
! Query drift wavelengths and powers
```

```
OUTPUT @Mwm;":CALC3:DATA? WAV"
ENTER @Mwm USING "#,K";Current_diff_wl(*)
OUTPUT @Mwm;":CALC3:DATA? POW"
ENTER @Mwm USING "#,K";Current_diff_pw(*)
OFF TIMEOUT
FOR I=1 TO Nb_pt
    PRINT USING "18A,2D,6A,M4D.2DE,3A,21A,MDD.3DE,3A";"Wavelength number "
    ;I;" is : ";Current_ref_wl(I);" m";" with a drift from : ";Current_diff_wl(I);" m"
    PRINT USING "28A,SDD.2DE,4A,20A,MDD.3DE,3A,/" ;"it has a power level of
    : ";Current_ref_pwr(I);" dBm";" with a drift from : ";Current_diff_pw(I);" dB";
NEXT I
STOP
```

```
Error_msg: !
    PRINT "The program is aborted due to : ";ERRM$
END
```

```
Err_mngmt:SUB Err_mngmt(OPTIONAL Cmd_msg$)
    COM /Instrument/ @Mwmt
    DIM Err_msg$(255)
    INTEGER Cme
    CLEAR @Mwm
    REPEAT
        OUTPUT @Mwm;"*ESR?"
        ENTER @Mwm;Cme
        OUTPUT @Mwm;":SYST:ERR?"
        ENTER @Mwm;Err_msg$
```

```

        IF NPAR>0 AND NOT POS(Err_msg$,"+0") THEN PRINT "This command ";Cmd_msg$;" makes the following error :"  

        IF NOT POS(Err_msg$,"+0") THEN PRINT Err_msg$  

        UNTIL NOT BIT(Cme,2) AND NOT BIT(Cme,4) AND NOT BIT(Cme,5) AND POS(Err_msg$,"+0")  

Subend:SUBEND

```

```

Set_ese:SUB Set_ese  

    COM /Instrument/ @Mwm  

    OUTPUT @Mwm;"*ESE ";IVAL("00110100",2)  

SUBEND

```

```

Identity:DEF FNIdentity$;  

    COM /Instrument/ @Mwm  

    DIM Identity$[50]  

    Identity$=""  

    OUTPUT @Mwm;"*RST"  

    OUTPUT @Mwm;"*OPC?"  

    ENTER @Mwm;Opc_done  

    OUTPUT @Mwm;"*IDN?"  

    ENTER @Mwm;Identity$  

    RETURN Identity$  

FNEND

```

```

Cmd_opc:SUB Cmd_opc(Set_cmd$)  

    COM /Instrument/ @Mwm  

    OUTPUT @Mwm;Set_cmd$  

    OUTPUT @Mwm;"*OPC?"  

    ENTER @Mwm;Opc_done$  

SUBEND

```

```

Tempo:SUB Tempo(Temp)  

    FOR I=Temp TO 0 STEP -1  

        DISP "Waiting for ";VAL$(I);" sec . . . "  

        WAIT 1  

    NEXT I  

    DISP ""  

SUBEND

```

## Measuring WDM Channel Separation

The following example program measures the line separations on a WDM system. It measures separation (delta) between power and wavelength of each line using commands from the [CALCulate3 subsystem](#).

This example uses the [Error\\_msg](#), [Set\\_ese](#), [Err\\_mngmt](#), [Cmd\\_opc](#), and [FNIdentity](#) function subroutines.



The use of the Err\_mngmt subroutine is optional. Err\_mngmt was modified for the programming example to allow this subroutine to indicate the last command that was sent to the instrument before an error occurred.

```
COM /Instrument/ @Mwm
ASSIGN @Mwm TO 720
DIM Key$[1]
ON ERROR GOTO Error_msg
Set_ese
PRINT USING "37A,33A";"Multi-Wavelength Meter Identity is : ";FNIdentity$
ON TIMEOUT 7,5 CALL Err_mngmt
Cmd_opc("*RST")
```

*! Change to list-by-wavelength display*

```
Cmd_opc(":CONF:ARR:POW:WAV")
```

*! Trigger and wait for one measurement*

```
Cmd_opc(":INIT")
Cmd_opc("*WAI")
```

*! Turn on delta mode*

```
Cmd_opc(":CALC3:DELT:WPOW:STAT ON")
```

*! Set first wavelength as reference*

```
Cmd_opc(":CALC3:DELT:REF:WAV MIN")
```

*! Query number of data points*

```
OUTPUT @Mwm;":CALC3:POIN?"
ENTER @Mwm USING "#,K";Nb_pt
ALLOCATE Delta_wl(1:Nb_pt)
ALLOCATE Delta_pwr(1:Nb_pt)
```

*! Query wavelengths and powers*

```
OUTPUT @Mwm;":CALC3:DATA? WAV"
```

```

ENTER @Mwm;Delta_wl(*)
OUTPUT @Mwm;":CALC3:DATA? POW"
ENTER @Mwm;Delta_pwr(*)
OFF TIMEOUT
FOR I=1 TO Nb_pt-1
  PRINT USING "6A,2D,17A,M4D.3D,31A,S2D.2D,4A";"Line :";I;" wavelength is : ";(Delta_wl(I)+((NOT
  I=1)*Delta_wl(1)))/1.0E-9;" nm. Absolute line level is : ";Delta_pwr(I)+(NOT I=1)*Delta_pwr(1);" dBm"
  PRINT USING "17A,2D,6A,M4D.3D,23A,2D,6A,S2D.2D,3A";"Delta Wl to line ",I+1," is : ";(Delta_wl(I+1)-(NOT
  I=1)*Delta_wl(I))/1.E-9;" nm, Delta Pwr to line ",I+1," is : ";(I=1)*(Delta_pwr(I+1))+(NOT I=1)*(Delta_pwr(I+1)-Delta_pwr(I));"
  dB"
NEXT I
  PRINT USING "6A,2D,17A,M4D.3D,31A,S2D.2D,4A";"Line :";I;" wavelength is : ";(Delta_wl(1)+Delta_wl(Nb_pt))/1.0E-9;"
  nm. Absolute line level is : ";Delta_pwr(1)+Delta_pwr(Nb_pt);" dBm"
STOP
Error_msg: !
  PRINT "The program is aborted due to : ";ERRM$
END

Err_mngmt:SUB Err_mngmt(OPTIONAL Cmd_msg$)
  COM /Instrument/ @Mwmt
  DIM Err_msg$[255]
  INTEGER Cme
  CLEAR @Mwm
  REPEAT
    OUTPUT @Mwm;"*ESR?"
    ENTER @Mwm;Cme
    OUTPUT @Mwm;":SYST:ERR?"
    ENTER @Mwm;Err_msg$
    IF NPAR>0 AND NOT POS(Err_msg$,"+0") THEN PRINT "This command ";Cmd_msg$;" makes the following error : "
    IF NOT POS(Err_msg$,"+0") THEN PRINT Err_msg$
  UNTIL NOT BIT(Cme,2) AND NOT BIT(Cme,4) AND NOT BIT(Cme,5) AND POS(Err_msg$,"+0")
Subend:SUBEND

Set_ese:SUB Set_ese
  COM /Instrument/ @Mwm
  OUTPUT @Mwm;"*ESE ";IVAL("00110100",2)
SUBEND

Identity:DEF FNIdentity$;
  COM /Instrument/ @Mwm
  DIM Identity$[33]
  Identity$=""
  OUTPUT @Mwm;"*RST"
  OUTPUT @Mwm;"*OPC?"
  ENTER @Mwm;Opc_done
  OUTPUT @Mwm;"*IDN?"
  ENTER @Mwm;Identity$
  RETURN Identity$
FNEND

Cmd_opc:SUB Cmd_opc(Set_cmd$)
  COM /Instrument/ @Mwm
  OUTPUT @Mwm;Set_cmd$

```



```
OUTPUT @Mwm;"*OPC?"  
ENTER @Mwm;Opc_done$  
SUBEND
```

## Measure Signal-to-Noise Ratio of Each WDM Channel

The following example program measures signal-to-noise ratio on a WDM system. It measures the ratio for each line using commands from the [CALCulate3 subsystem](#).

This example uses the [Error\\_msg](#), [Set\\_ese](#), [Err\\_mngmt](#), [Cmd\\_opc](#), and [FNIdentity](#) function subroutines.



The use of the Err\_mngmt subroutine is optional. Err\_mngmt was modified for the programming example to allow this subroutine to indicate the last command that was sent to the instrument before an error occurred.

```
COM /Instrument/ @Mwm
ASSIGN @Mwm TO 720
DIM Key$[1]
ON ERROR GOTO Error_msg,
Set_ese
PRINT USING "37A,33A";"Multi-Wavelength Meter Identity is : ";FNIdentity$
ON TIMEOUT 7,5 CALL Err_mngmt
Cmd_opc("*RST")
OUTPUT @Mwm;":MEAS:ARR:POW:WAV?"
ENTER @Mwm USING "#,K";Nb_pt
ALLOCATE Current_wl(1:Nb_pt)
ENTER @Mwm USING "#,K";Current_wl(*)
OUTPUT @Mwm;":FETC:ARR:POW?"
ENTER @Mwm USING "#,K";Nb_pt
ALLOCATE Current_pwr(1:Nb_pt)
ENTER @Mwm USING "#,K";Current_pwr(*)
```

*! Turn signal-to-noise ratio on*

```
Cmd_opc("CALC3:SNR:STAT ON")
Err_mngmt("CALC3:SNR:STAT ON")
```

*! Set first wavelength as noise reference*

```
Cmd_opc("CALC3:SNR:REF:WAV MIN")
Err_mngmt("CALC3:SNR:REF:WAV MIN")
```

*! Query number of data points*

```
OUTPUT @Mwm;":CALC3:POIN?"
ENTER @Mwm USING "#,K";Nb_pt
ALLOCATE Snr_pwr(1:Nb_pt)
```

*! Query signal-to-noise values*

```
OUTPUT @Mwm;":CALC3:DATA? POW"
ENTER @Mwm;Snr_pwr(*)
```

```

OFF TIMEOUT
FOR I=1 TO Nb_pt
  PRINT USING "7A,2D,17A,M4D.3D,25A,S2D.2D,22A,2D.2D,3A";"Line : ";I;" wavelength is : ";Current_wl(I)/1.0E-9;" nm,
  absolute level is : ";Current_pwr(I);" dBm, with a SNR of : ";Snr_pwr(I);" dB"
NEXT I
STOP

Error_msg: !
  PRINT "The program is aborted due to : ";ERRM$
END

Err_mngmt:SUB Err_mngmt(OPTIONAL Cmd_msg$)
  COM /Instrument/ @Mwmt
  DIM Err_msg$(255)
  INTEGER Cme
  CLEAR @Mwm
  REPEAT !
    OUTPUT @Mwm;"*ESR?"
    ENTER @Mwm;Cme
    OUTPUT @Mwm;".SYST:ERR?"
    ENTER @Mwm;Err_msg$
    IF NPAR>0 AND NOT POS(Err_msg$,"+0") THEN PRINT "This command ";Cmd_msg$;" makes the following error : "
    IF NOT POS(Err_msg$,"+0") THEN PRINT Err_msg$
  UNTIL NOT BIT(Cme,2) AND NOT BIT(Cme,4) AND NOT BIT(Cme,5) AND POS(Err_msg$,"+0")
Subend:SUBEND

Set_ese:SUB Set_ese
  COM /Instrument/ @Mwm
  OUTPUT @Mwm;"*ESE ";IVAL("00110100",2)
SUBEND

Identity:DEF FNIdentity$;
  COM /Instrument/ @Mwm
  DIM Identity$(50)
  Identity$=""
  OUTPUT @Mwm;"*RST"
  OUTPUT @Mwm;"*OPC?"
  ENTER @Mwm;Opc_done
  OUTPUT @Mwm;"*IDN?"
  ENTER @Mwm;Identity$
  RETURN Identity$
FNEND

Cmd_opc:SUB Cmd_opc(Set_cmd$)
  COM /Instrument/ @Mwmd
  OUTPUT @Mwm;Set_cmd$
  OUTPUT @Mwm;"*OPC?"
  ENTER @Mwm;Opc_done$
SUBEND

```

## Increasing a Source's Wavelength Accuracy

The following example program uses the Agilent 86122C to increase the absolute wavelength accuracy of Agilent 8167A, 8168B, and 8168C tunable laser sources. Essentially, the Agilent 86122C transfers its accuracy to the tunable laser source.

For more information about the absolute accuracy of this instrument, refer to Specifications.



In order to run this program, the firmware of the tunable laser source must support the automatic alignment command, WAVEACT.

The example program uses the following measurement algorithm:

- Identify and initialize the Agilent 86122C and tunable laser source
- Ask user for desired wavelength
- Set wavelength of tunable laser source
- Turn tunable laser source's output on
- Enter loop
  - Measure wavelength
  - Compare wavelength to desired wavelength
  - Realign tunable laser source's wavelength
  - Check if wavelength changed from last pass
- Repeat until (delta wavelength < 0.0015 nm or wavelength is stable)

```
COM Current_wl,Diff_wl,Target_wl,Previous_diff,Diff_diff
Current_wl=0
Diff_wl=0
Target_wl=0
Previous_diff=0
Diff_diff=0
```

```
ASSIGN @Tls TO 724
ASSIGN @Mwm TO 720
```

*! Initialize instrument*

```
DIM Identity$(50)
Identity$=""
OUTPUT @Tls;"*CLS"
OUTPUT @Tls;"*IDN?"
ENTER @Tls;identity$
PRINT "TLS IS A ";identity$
OUTPUT @Mwm;"*RST"
OUTPUT @Mwm;"*CLS"
OUTPUT @Mwm;"*IDN?"
ENTER @Mwm;identity$
PRINT "MWM IS A ";identity$
```

*! Ask user for desired wavelength*

```

INPUT "What wavelength (nm)do you wish to have",Target_wl
Target_wl=Target_wl*1.0E-9
PRINT "the target wavelength is : ";Target_wl

```

*! Set wavelength of tunable laser source*

```

OUTPUT @TIs; ":WAVE ";VAL$(Target_wl)
OUTPUT @TIs; ":OUTP ON"

```

*! Enter realignment loop*

```

REPEAT
  OUTPUT @Mwm;":MEAS:SCAL:POW:WAV?"
  ENTER @Mwm;Current_wl
  PRINT "The current wavelength is : ";VAL$(Current_wl)
  Diff_wl=PROUND(ABS(Target_wl-Current_wl),-16)
  PRINT "Diff between target & Current is (+ or -) : ";VAL$(Diff_wl)
  OUTPUT @TIs;":WAVEACT ";VAL$(Current_wl)
  Diff_diff=PROUND(ABS(Diff_wl-Previous_diff),-16)
  PRINT "differential difference between two turn : ";VAL$(Diff_diff)
  Previous_diff=Diff_wl
  UNTIL (Diff_wl<1.5*1.0E-12) OR (Diff_diff=0)
END

```

## Returned Data Format

Review the following information to understand the returned data format from a query.

### Measurements are returned as strings

All measurement values are returned from the Agilent 86122C as ASCII strings. When an array is returned, the comma character separates the individual values.

### Determining the number of data points

When FETCh, READ, or MEASure is used (with ARRAy specified), the first returned value indicates the total number of measurement values returned in the query.

If you use :CALCulate1:DATA?, :CALCulate2:DATA?, or :CALCulate3:DATA? to query data, first send :POINTs? to determine the number of values returned in the string. The string does not contain a first value, which specifies the string length. This is shown in the following example:

```
OUTPUT 720;";:CALCulate1:POINTs?"
ENTER 720;Length
OUTPUT 720;";:CALCulate1:DATA?"
ENTER 720;Result$
```

### Correcting for elevation and vacuum

The Agilent 86122C contains an internal weather station that provides temperature and barometric pressure values. This allows the instrument to automatically correct for differences in elevation and temperature. Use:SENSe:CORRection:MEDium to select readings in air or vacuum.

### Setting amplitude units

The default amplitude units are dBm. If you need measurements in watts, use :UNIT:POWer. When the Agilent 86122C is turned on, the amplitude units are automatically set to the units used before the instrument was last turned off.

## Messages and Errors Overview

Instrument messages are displayed in the text box at the bottom of the display screen.






These messages provide information about the status of the instrument or an action that has been performed. Messages also appear to inform you of errors that may have occurred. These errors may appear as a result of

- ◆ Invalid or null data
- ◆ Internal software errors
  - Hardware errors.



Error messages remain visible for 3 seconds before reverting back to the standard display.

There are two categories of error messages: instrument specific and general SCPI. Both types of messages use symbols that represent the severity of the error condition.

- ◆  indicates current instrument status. No action is required.
- ◆  allows you to take a recommended action. This type of message indicates if an operation did not complete or was cancelled.
  -  indicates instrument hardware or software failures. This type of message will remain on the display until the error condition is corrected.

Click/touch one of the following topics to view the lists of messages that may occur.

- [Instrument Specific Messages](#)
- [General SCPI Messages](#)

## Instrument Specific Messages

Click/touch any of the following error messages to view more information about the selected message.



To view a list of remote programming error messages, click/touch [General SCPI Error Messages](#).

Error Number	Error Message
12	<a href="#">Power level too high</a>
13	<a href="#">Data download problem</a>
14	<a href="#">Data acquisition problem</a>
15	<a href="#">Max number of signals found</a>
48	<a href="#">No reference signal</a>
49	<a href="#">Gain ranging error</a>
103	<a href="#">Path not allowed</a>
115	<a href="#">Network path not found</a>
117	<a href="#">Reference HeNe not stabilized</a>
118	<a href="#">Reference HeNe Over Temperature</a>
147	<a href="#">Printer error</a>

---

### Power Level Too High

<b>Number:</b>	12
<b>Explanation:</b>	The ADC over-range bit is detected and the current gain is already 0.
<b>Solution:</b>	If problem persists, please contact the Agilent service center.

---

### Data Download Problem

<b>Number:</b>	13
<b>Explanation:</b>	The data acquisition bit was not true after initiating data acquisition.
<b>Solution:</b>	If problem persists, please contact the Agilent service center.

---

### Data Acquisition Problem

<b>Number:</b>	14
<b>Explanation:</b>	The motion complete bit was not true after initiating motor movement.
<b>Solution:</b>	If problem persists, please contact the Agilent service center.



---

### Max Number of Signals Found

**Number:** 15

**Explanation:** 1000 signals were detected.

---

### No Reference Signal

**Number:** 48

**Explanation:** The drift application expected, but did not find any signals in its reference sweep.

---

### Gain Ranging Error

**Number:** 49

**Explanation:** Could not find proper gain level after 6 attempts.

---

### Path Not Allowed

**Number:** 103

**Explanation:** The remote command used does not allow file name with path.

---

### Network Path Not Found

**Number:** 115

**Explanation:** The network path may be unavailable or unmapped. For example, if you attempt to load or save a file to an unmapped or non-existent network path.

**Solution:** Contact your network administrator to verify the network path and the network configuration of your instrument. If the problem persists, please contact the Agilent service center.

---

### Reference HeNe Not Stabilized

**Number:** 117

**Explanation:** During sweep, the HeNe laser is detected as not stabilized.

**Solution:** When the reference laser is not stable the instrument cannot take measurements. This is normal operation for the reference laser during warmup with the high accuracy option.

With the exceptions of the following commands, an

error is generated if a command is given to the instrument. The following commands are operational so that instrument status information can be obtained:

```
*CLS
*ESE[?]
*ESR?
*IDN?
*SRE[?]
*STB?
:SYST:ERR?
:STAT:OPER:COND?
:STAT:OPER:ENAB[?]
:STAT:OPER:EVEN?
:STAT:OPER:NTR[?]
:STAT:OPER:PTR[?]
:STAT:PRESET
:STAT:QUES:COND?
:STAT:QUES:ENAB[?]
:STAT:QUES:EVEN?
:STAT:QUES:NTR[?]
:STAT:QUES:PTR[?]
```

If problem persists, please contact the Agilent service center.

---

## Reference HeNe Over Temperature

**Number:** 118

**Explanation:** During sweep, the HeNe laser is detected in an over temperature condition.

**Solution:** When the reference laser is not stable the instrument cannot take measurements. This is normal operation for the reference laser during warmup with the high accuracy option.

With the exceptions of the following commands, an error is generated if a command is given to the instrument. The following commands are operational so that instrument status information can be obtained:

```
*CLS
*ESE[?]
*ESR?
*IDN?
*SRE[?]
*STB?
:SYST:ERR?
:STAT:OPER:COND?
:STAT:OPER:ENAB[?]
:STAT:OPER:EVEN?
:STAT:OPER:NTR[?]
:STAT:OPER:PTR[?]
```

```
:STAT:PRESET  
:STAT:QUES:COND?  
:STAT:QUES:ENAB[?]  
:STAT:QUES:EVEN?  
:STAT:QUES:NTR[?]  
:STAT:QUES:PTR[?]
```

If problem persists, please contact the Agilent service center.

---

## Printer Error

**Number:** 147

**Explanation:** A printer error occurred. Additional string information is added to describe specific printer error conditions. For example,

Printer Error: Install and select a default printer

The instrument was unable to locate the default printer.

**Solution:** The solution for this example is to install a printer and set up that printer as the default printer. Please refer to the Agilent 86122A Quick Start Guide for a list of printers this instrument supports.

## General SCPI Messages

Click/touch any of the following error messages to view more information about the selected message.



To view a list of instrument specific error messages, click/touch [Instrument Error Messages](#).

<b>Error Number</b>	<b>Error Message</b>
-100	<a href="#">Command error (unknown command)</a>
-101	<a href="#">Invalid character</a>
-102	<a href="#">Syntax error</a>
-103	<a href="#">Invalid Separator</a>
-104	<a href="#">Data Type Error</a>
-105	<a href="#">GET Not Allowed</a>
-108	<a href="#">Parameter Not Allowed</a>
-109	<a href="#">Missing parameter</a>
-112	<a href="#">Program mnemonic too long</a>
-113	<a href="#">Undefined header</a>
-120	<a href="#">Numeric data error</a>
-123	<a href="#">Exponent too large</a>
-124	<a href="#">Too many digits</a>
-128	<a href="#">Numeric data not allowed</a>
-131	<a href="#">Invalid suffix</a>
-134	<a href="#">Suffix too long</a>
-138	<a href="#">Suffix not allowed</a>
-141	<a href="#">Invalid character data</a>
-148	<a href="#">Character data not allowed</a>
-150	<a href="#">String data error</a>
-151	<a href="#">Invalid string data</a>
-158	<a href="#">String data not allowed</a>
-161	<a href="#">Invalid block data</a>
-168	<a href="#">Block data not allowed</a>
-170	<a href="#">Expression error</a>
-171	<a href="#">Invalid expression</a>
-178	<a href="#">Expression data not allowed</a>
-200	<a href="#">Execution error</a>
-211	<a href="#">Trigger ignored</a>
-213	<a href="#">Init ignored</a>
-221	<a href="#">Settings Conflict</a>

-222	<a href="#"><u>Data out of range</u></a>
-223	<a href="#"><u>Too much data</u></a>
-224	<a href="#"><u>Illegal parameter value</u></a>
-230	<a href="#"><u>Data corrupt or stale</u></a>
-232	<a href="#"><u>Data questionable</u></a>
-250	<a href="#"><u>Mass storage error</u></a>
-251	<a href="#"><u>Missing mass storage</u></a>
-252	<a href="#"><u>Missing media</u></a>
-253	<a href="#"><u>Corrupt media</u></a>
-254	<a href="#"><u>Media full</u></a>
-255	<a href="#"><u>Directory full</u></a>
-256	<a href="#"><u>File name not found</u></a>
-257	<a href="#"><u>File name error</u></a>
-258	<a href="#"><u>Media protected</u></a>
-273	<a href="#"><u>Illegal macro label</u></a>
-440	<a href="#"><u>Query UNTERMINATED after indef response</u></a>

---

### Command Error (Unknown Command)

**Number:** -100

**Explanation:** This is a generic syntax error for devices that cannot detect more specific errors. The code indicates only that a command error as defined in IEEE 488.2, 11.5.1.1.4 has occurred.

---

### Invalid Character

**Number:** -101

**Explanation:** A syntactic command contains a character that is invalid for that type. For example, a header containing an ampersand, SETUP&. This error might be used in place of error numbers -114, -121, -141 and some others.

---

### Syntax Error

**Number:** -102

**Explanation:** An unrecognized command or data type was encountered. For example, a string was received when the device does not accept strings.

---

### Invalid Separator

**Number:** –103

**Explanation:** The parser was expecting a separator and encountered an illegal character. For example, the semicolon was omitted after a program message unit.

---

### Data Type Error

**Number:** –104

**Explanation:** The parser recognized a data element that is not allowed. For example, numeric or string data was expected, but block data was encountered.

---

### GET Not Allowed

**Number:** –105

**Explanation:** A Group Execute Trigger was received within a program message (see IEEE 488.2, 7.7). Correct the GPIB controller program so that the GET does not occur within a line of GPIB program code.

---

### Parameter Not Allowed

**Number:** –108

**Explanation:** More parameters were received than expected for the header. For example, the \*ESE common command only accepts one parameter, so receiving \*ESE 0,1 is not allowed.

---

### Missing Parameter

**Number:** –109

**Explanation:** Fewer parameters were received than required for the header. For example, the \*ESE common command requires one parameter, so receiving \*ESE is not allowed.

---

### Program Mnemonic Too Long

**Number:** –112

**Explanation:** The header contains more than twelve characters (see IEEE 488.2, 7.6.1.4.1).

---

### Undefined Header

**Number:** –113

**Explanation:** An error was detected in the header. This message is used when the device cannot detect the more specific errors described for error –112.

---

### Numeric Data Error

**Number:** –120

**Explanation:** This error, as well as errors -121 through -129, are generated when parsing a data element which appears to be numeric, including non-decimal numeric types. This particular error message is used if the device cannot detect a more specific error.

---

### Exponent Too Large

**Number:** –123

**Explanation:** The magnitude of an exponent was greater than 32000 (see IEEE 488.2, 7.7.2.4.1).

---

### Too Many Digits

**Number:** –124

**Explanation:** The mantissa of a decimal-numeric data element contained more than 255 digits excluding leading zeros (see IEEE 488.2, 7.7.2.4.1).

---

### Numeric Data Not Allowed

**Number:** –128

**Explanation:** A legal numeric data element was received, but the device does not accept one in this position for the header.

---

### Invalid Suffix

**Number:** –131

**Explanation:** The suffix does not follow the syntax described in IEEE 488.2, 7.7.3.2, or the suffix is inappropriate for this device.

---

### Suffix Too Long

**Number:** –134

**Explanation:** The suffix contained more than twelve characters (see IEEE 488.2, 7.7.3.4).

---

### Suffix Not Allowed

**Number:** –138

**Explanation:** A suffix was encountered after a numeric element which does not allow suffixes.

---

### Invalid Character Data

**Number:** –141

**Explanation:** Either the character data element contains an invalid character or the particular element received is not valid for the header.

---

### Character Data Not Allowed

**Number:** –148

**Explanation:** A legal character data element was encountered where prohibited by the device.

---

### String Data Error

**Number:** –150

**Explanation:** This error is generated when parsing a string data element. This particular error message is used if the device cannot detect a more specific error.

---

### Invalid String Data

**Number:** –151

**Explanation:** A string data element was expected, but was invalid (see IEEE 488.2, 7.7.5.2). For example, an END message was received before the terminal quote character.

---

### String Data Not Allowed

**Number:** –158

**Explanation:** A string data element was encountered, but not allowed by the device at this point in the parsing.

---

### Invalid Block Data



**Number:** -161

**Explanation:** A block data element was expected, but was invalid (see IEEE 488.2, 7.7.6.2). For example, an END message was received before the end length was satisfied.

---

### Block Data Not Allowed

**Number:** -168

**Explanation:** A legal block data element was encountered, but not allowed by the device at this point in the parsing.

---

### Expression Error

**Number:** -170

**Explanation:** This error, as well as errors -171 through -179, are generated when parsing an expression data element. This particular error message is used if the device cannot detect a more specific error.

---

### Invalid Expression

**Number:** -171

**Explanation:** The expression data element was invalid (see IEEE 488.2, 7.7.7.2). For example, unmatched parentheses or an illegal character.

---

### Expression Data Not Allowed

**Number:** -178

**Explanation:** A legal expression data was encountered, but was not allowed by the device at this point in parsing.

---

### Execution Error

**Number:** -200

**Explanation:** For devices that cannot detect more specific errors, this code indicates only that an execution error as defined in IEEE 488.2, 11.5.1.1.5 has occurred.

---

### Trigger Ignored

**Number:** -211

**Explanation:** Caused by sending the \*TRG command while a measurement is already in progress or when the

instrument is in continuous measurement mode.

---

### Init Ignored

**Number:** -213

**Explanation:** Caused by sending an INITiate:IMMEDIATE, READ, or MEASure command while a measurement is already in progress or while the instrument is in continuous measurement mode.

---

### Settings Conflict

**Number:** -221

**Explanation:** Caused by trying to set the instrument to a state that is not allowed. For example, turning on drift maximum and drift minimum state simultaneously, or turning on OSNR state while drift or delta state is on.

---

### Data Out of Range

**Number:** -222

**Explanation:** A legal program data element was parsed but could not be executed because the interpreted value was outside the legal range defined by the device (see IEEE 488.2 11.5.1.1.5).

---

### Too Much Data

**Number:** -223

**Explanation:** A legal program data element of block, expression or string type was received that contained more data than the device could handle due to memory or related device-specific requirements.

---

### Illegal Parameter Value

**Number:** -224

**Explanation:** Used where an exact value, from a list of possibilities, was expected.

---

### Data Corrupt or Stale

**Number:** -230

**Explanation:** Caused by trying to query measurement data  
\*RST

immediately after a \*RST command. For example, sending \*RST;FETCh, or sending \*RST;:CALCulate2:DATA? POWer.

---

### Data Questionable

**Number:** -232

**Explanation:** Caused by sending a resolution value in one of the measurement functions that is outside the instrument's range.

---

### Mass Storage Error

**Number:** -250

**Explanation:** Indicates that a mass storage error occurred.

---

### Missing Mass Storage

**Number:** -251

**Explanation:** Indicates that a legal program command or query could not be executed because of missing mass storage. For example, an option that was not installed.

---

### Missing Media

**Number:** -252

**Explanation:** Indicates that a legal program command or query could not be executed because of missing media; for example, no disk.

---

### Corrupt Media

**Number:** -253

**Explanation:** Indicates that a legal program command or query could not be executed because of corrupt media; for example, bad disk or wrong format.

---

### Media Full

**Number:** -254

**Explanation:** Indicates that a legal program command or query could not be executed because the media was full; for example, there was no room on the disk.

---

### Directory Full

**Number:** –255

**Explanation:** Indicates that a legal program command or query could not be executed because the media directory was full.

---

### File Name Not Found

**Number:** –256

**Explanation:** Indicates that a legal program command or query could not be executed because the file name on the device media was not found; for example, an attempt was made to read or copy a nonexistent file.

---

### File Name Error

**Number:** –257

**Explanation:** Indicates that a legal program command or query could not be executed because the file name on the device media was in error; for example, an attempt was made to copy to a duplicate file name.

---

### Media Protected

**Number:** –258

**Explanation:** Indicates that a legal program command or query could not be executed because the media was protected; for example, the write-protect tab on a disk was present.

---

### Illegal Macro Label

**Number:** –273

**Explanation:** The macro label defined in the \*DMC command was a legal string syntax, but could not be accepted by the device (see IEEE 488.2, 10.7.3 and 10.7.6.2). For example, the label was too long, the same as a common command header, or contained invalid header syntax.

---

### Query UNTERMINATED After Indef Response

**Number:** –440

**Explanation:** Query was unterminated after an indefinite response.

[www.agilent.com](http://www.agilent.com)

© Copyright Agilent Technologies 2002 - 2013  
First edition, April 2013



86122-90C02

