Agilent 86082A
Wavelength Domain
Component Analyzer
User’s Guide

Agilent Technologies

Agilent Part No. 86082-90001
Printed in USA
August 2001

Agilent Technologies
Lightwave Division
3910 Brickway Boulevard,
Santa Rosa, CA 95403, USA

Notice.
The information contained in this document is subject to change without notice. Companies, names, and data used in examples herein are fictitious unless otherwise noted. Agilent Technologies makes no warranty of any kind with regard to this material, including but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Agilent Technologies shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Restricted Rights Legend.
Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 for DOD agencies, and subparagraphs (c) (1) and (c) (2) of the Commercial Computer Software Restricted Rights clause at FAR 52.227-19 for other agencies.

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 Technologies will, at its option, either repair or replace products which prove to be defective. For warranty service or repair, this product must be returned to a service facility designated by Agilent Technologies. Buyer shall prepay shipping charges to Agilent Technologies and Agilent Technologies 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 Technologies from another country.

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

Safety Symbols.
CAUTION

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

WARNING

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

⚠ The instruction manual symbol. The product is marked with this warning symbol when it is necessary for the user to refer to the instructions in the manual.

The laser radiation symbol. This warning symbol is marked on products which have a laser output.

∿ The AC symbol is used to indicate the required nature of the line module input power.

| The ON symbols are used to mark the positions of the instrument power line switch.

❍ The OFF symbols are used to mark the positions of the instrument power line switch.

The CE mark is a registered trademark of the European Community.

The CSA mark is a registered trademark of the Canadian Standards Association.

The C-Tick mark is a registered trademark of the Australian Spectrum Management Agency.

ISM1-A This text denotes the instrument is an Industrial Scientific and Medical Group 1 Class A product.

Typographical Conventions.
The following conventions are used in this book:

Key type for keys or text located on the keyboard or instrument.

Softkey type for key names that are displayed on the instrument's screen.

`Display type` for words or characters displayed on the computer's screen or instrument's display.

User type for words or characters that you type or enter.

Emphasis type for words or characters that emphasize some point or that are used as place holders for text that you type.

General Safety Considerations

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

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

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

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

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

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

CAUTION Fiber-optic connectors are easily damaged when connected to dirty or damaged cables and accessories. The Agilent 86082A's front-panel input connector is no exception. When you use improper cleaning and handling techniques, you risk expensive instrument repairs, damaged cables, and compromised measurements. Before you connect any fiber-optic cable to the Agilent 86082A, refer to "Cleaning Connections for Accurate Measurements" on page 8-11.

CAUTION

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

CAUTION

Do not use too much liquid in cleaning the optical spectrum analyzer. Water can enter the front-panel keyboard, damaging sensitive electronic components.

CAUTION

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

CAUTION

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

CAUTION

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

CAUTION

Do not connect ac power until you have verified the line voltage is correct as described in “Line Power Requirements” on page 1-9. Damage to the equipment could result.

CAUTION

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

CAUTION

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

1

Getting Started

Product Overview

The Agilent Technologies 86082A wavelength domain component analyzer (WDCA) is designed for the manufacture of narrow-wavelength filters.

As bandwidth requirements accelerate, demand is increasing for higher bit rates and more channels. As more channels are added, the filter width and the spacing between channels are decreasing to keep all the channels within the critical range of optical amplifiers. The optical filter is a crucial component in determining channel separation and location.

The optical filter's absolute wavelength position, location relative to the ITU grid, width, and shape are all important specifications to the designers, manufacturers, and users of these filters. As filter widths move from 200 GHz to 25 GHz and lower, these specifications become increasingly difficult to measure. The WDCA offers both the high precision and fast measurement speeds necessary for these demanding measurements.

Instrument features

The WDCA combines the best of Agilent Technologies optical spectrum analyzer (OSA) and tunable laser source (TLS). The receiver in the 86082A has the dynamic range, digital signal processing, large display, feature set, and graphical user interface of Agilent's OSA. This receiver is integrated with Agilent's TLS to provide the speed, wavelength range, and narrow 1 pm resolution features which the TLS offers. Together, they offer the speed and ease of use needed for demanding narrow-wavelength filter measurements.

The WDCA's dual input channels allow you to test two devices at the same time or make a reflection and insertion loss measurements without making a connector change.

The WDCA includes a dedicated built-in calibration reference cell. This provides the instrument with better than 10 pm absolute wavelength accuracy and a dynamic range of –65 dB as close as 2.5 pm from the signal peak.

The WDCA allows you to select from three performance modes to optimize the measurements for your specific needs.

- *Normal* mode provides excellent accuracy and the speed of two sweeps per second over a 2 nm span.
- *Precision* mode offers the increased accuracy needed for final filter test applications.
- *Fast* mode allows wider spans with faster sweep speed. In this mode you can measure filters with multiple channels or see if the rejection band holds over wide ranges.

To help with real-time adjustments on larger signals, the WDCA provides a range-lock feature. This feature locks the displayed information to the top 30 dB of the measurement range for the fastest trace update rate. Faster trace updates allow easier tuning and adjusting of the filter’s insertion loss and passband location. Range-lock is available with any of the performance modes.

With the WDCA, you can make stimulus-response measurements without the use of external programs, controllers, or programming. Because it is an integrated system, there is no concern with linking and synchronizing the source and receiver. Manual users will find the features, interface, and large display easy to learn and use. The marker features and built-in measurement aids allow you to focus on the device under test and not worry about the instrumentation or controller. Experienced OSA users will find many features, markers, and operational functions to be familiar.

For remote programming and automatic testing, the WDCA builds on the extensive programming command set of Agilent Technologies’ OSA family. You can leverage your current OSA programs for fast program implementation.

Standard Product, Options, and Accessories

Table 1-1. Standard Product Accessories

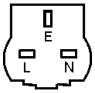
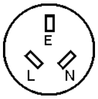
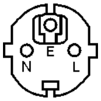
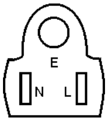
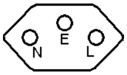
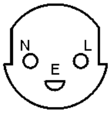
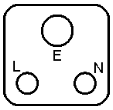
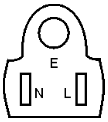
Description	Product Number	Item Quantity
BNC Cable (24 inches)	8120-1839	1
GPIB Cable	8120-3444	1
TLS Cover	86082-40002	1
FC/PC - FC/APC SM OC 0.3 M (Normalization Optical Cable)	1005-0954	1
FC/PC - FC/APC SM OC 1.0 M (Wavelength Reference Cable)	1005-0953	1
FC/PC Connector Adapter	08154-61702	2
Angled to Flat Connector Converter	1005-0089	1
FC/PC Dust Cap	1401-0291	2
English User’s Guide Manual	86082-90001	1
Quick Start Card	86082-90005	1
CD-ROM	86082-90004	1

Table 1-2. Options and Accessories

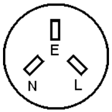
Option	Description	Product Number	Item Quantity
Instrument System Options			
111	C-Band System (with C-band TLS MF) TLS module: TNBL LSR MD-1550	8164A 81680A	1 1
112	L-Band System (with L-Band TLS MF) TLS module: TNBL LSR MD-1660	8164A 81640A	1 1
201	Receiver Only (Delete TLS) Software for TLS Upgrade	None 86082-90004	1 1
Connector Accessories			
010	HP/HMS-10 Connector Adapter	08154-61701	3
011	HMS-10 Dust Cap	1401-0291	3
012	FC/PC Connector Adapter FC/PC Dust Cap Angled to Flat, FC/PC Adapter	08154-61702 1401-0291 1250-3175	3 3 1
013	DIN Optical Connector Adapter DIN Dust Cap	08154-61703 1401-0291	3 3
014	ST Optical Connector Adapter ST Dust Cap	08154-61704 1401-0291	3 3
017	SC Optical Connector Adapter SC Dust Cap	08154-61708 1401-0291	3 3
	Reference Reflector	81000BR	
Power Selection			
900	Power Cord (United Kingdom)	8120-8705	1
901	Power Cord (Australia, New Zealand, China)	8120-0696	1
902	Power Cord (Europe)	8120-1692	1
903	Power Cord (United States)	8120-1521	1
906	Power Cord (Switzerland)	8120-2296	1
912	Power Cord (Denmark)	8120-2957	1
917	Power Cord (South Africa, India)	8120-4600	1
918	Power Cord (Japan)	8120-4754	1
919	Power Cord (Israel)	8120-5181	1

Option	Description	Product Number	Item Quantity
Recommended Accessories			
1CN	Kit - Front Handle (TLS) Kit - Front Handle (Receiver)	5063-9227 5063-9229	1 1
AX4	Kit - Rack Mount (TLS) Kit - Rack Mount (Receiver)	5063-9214 5063-9216	1 1
AXE	Kit - Rack Mount and Handles (TLS) Kit - Rack Mount and Handles (Receiver)	5063-9221 5063-9223	1 1
Documentation and Manuals			
ABA	English User’s Guide Service Guide English: CD ROM Manual Set Quick Start	86082-90001 86082-90002 86082-90004 86082-90005	1
ABJ	Japanese User’s Guide	86082-90006	1
AB2	Simplified Chinese User’s Guide	86082-90007	1
Certification of Calibration			
1BM	Standard Commercial Calibration Certificate	---	1
UK6	Commercial Calibration Certificate with Test Data	N/A	1

AC Line-Power Cords

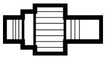
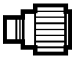
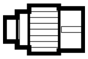
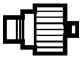
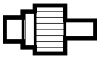
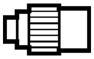
Plug Type	Cable Part No.	Plug Description	Length (in/cm)	Color	Country
250V 	8120-1351C 8120-8705	Straight *BS1363A 90°	90/228 90/228	Gray Mint Gray	United Kingdom, Cyprus, Nigeria, Zimbabwe, Singapore
250V 	8120-1369C 8120-0696	Straight *NZSS198/ ASC 90°	79/200 87/221	Gray Mint Gray	Australia, New Zealand
250V 	8120-1689C 8120-1692	Straight *CEE7-Y11 90°	79/200 79/200	Mint Gray Mint Gray	East and West Europe, Saudi Arabia, So. Africa, India (unpolarized in many nations)
125V 	8120-1378C 8120-1521	Straight *NEMA5-15P 90°	90/228 90/228	Jade Gray Jade Gray	United States, Canada, Mexico, Philippines, Taiwan
250V 	8120-2104C 8120-2296	Straight *SEV1011 1959-24507 Type 12 90°	79/200 79/200	Mint Gray Mint Gray	Switzerland
220V 	8120-2956C 8120-2957	Straight *DHCK107 90°	79/200 79/200	Mint Gray Mint Gray	Denmark
250V 	8120-4211 8120-4600	Straight SABS164 90°	79/200 79/200	Jade Gray	Republic of South Africa India
100V 	8120-4753C 8120-4754	Straight MITI 90°	90/230 90/230	Dark Gray	Japan

* Part number shown for plug is the industry identifier for the plug only. Number shown for cable is the Agilent part number for the complete cable including the plug.

Plug Type	Cable Part No.	Plug Description	Length (in/cm)	Color	Country
250V 	8120-5181	Straight	79/200	Gray	Israel

* Part number shown for plug is the industry identifier for the plug only. Number shown for cable is the Agilent part number for the complete cable including the plug.

Front-Panel Fiber-Optic Adapters

Front Panel Fiber-Optic Adapter	Description	Agilent Part Number
	Diamond HMS-10	08154-61701
	FC/PC[a]	08154-61702
	SC	08154-61708
	DIN	08154-61703
	ST	08154-61704
	Biconic	08154-61705

a. The FC/PC is the default front-panel optical connector.

Agilent 86082A Front and Rear Panels

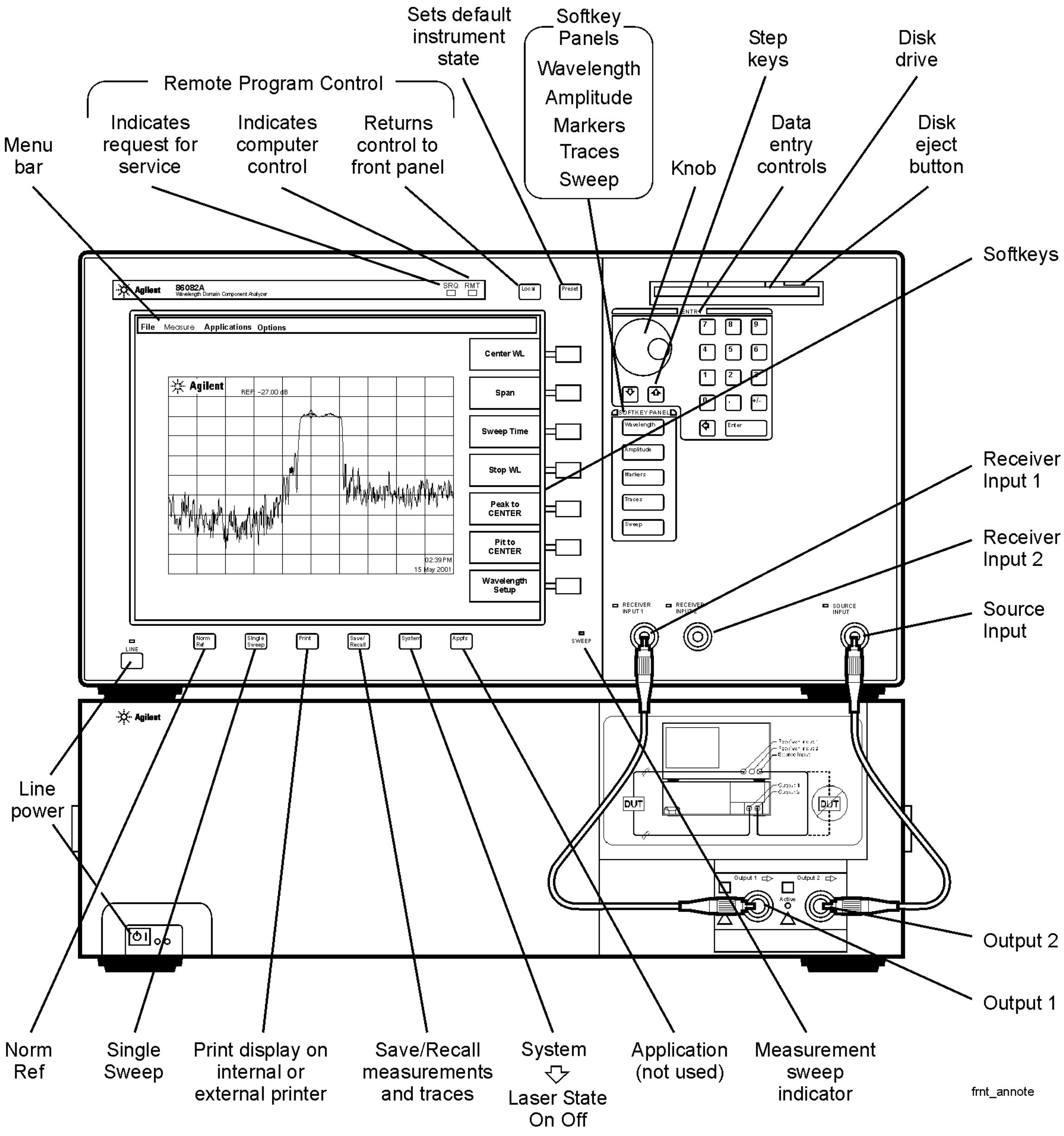


Figure 1-1. Front Panel

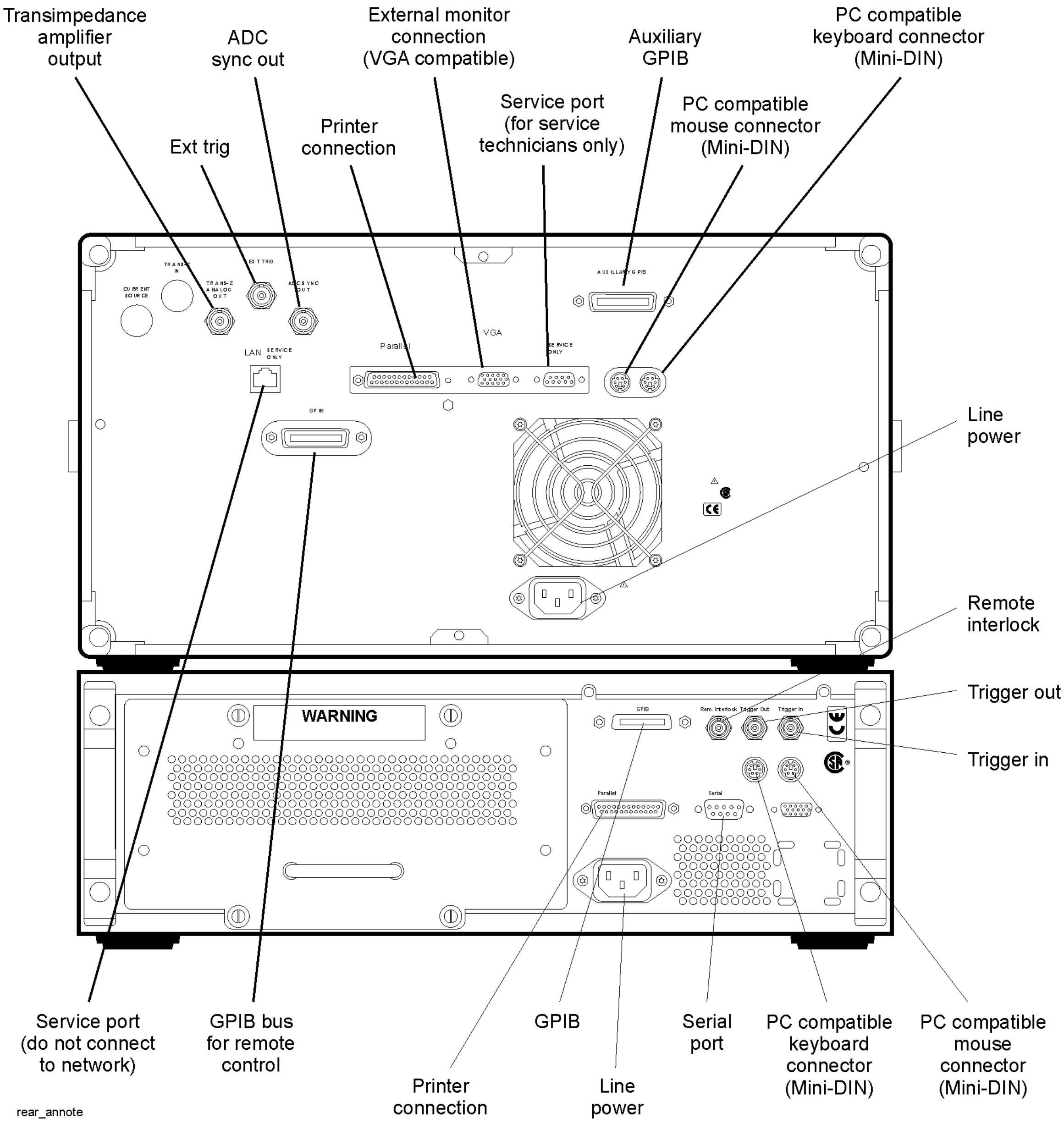
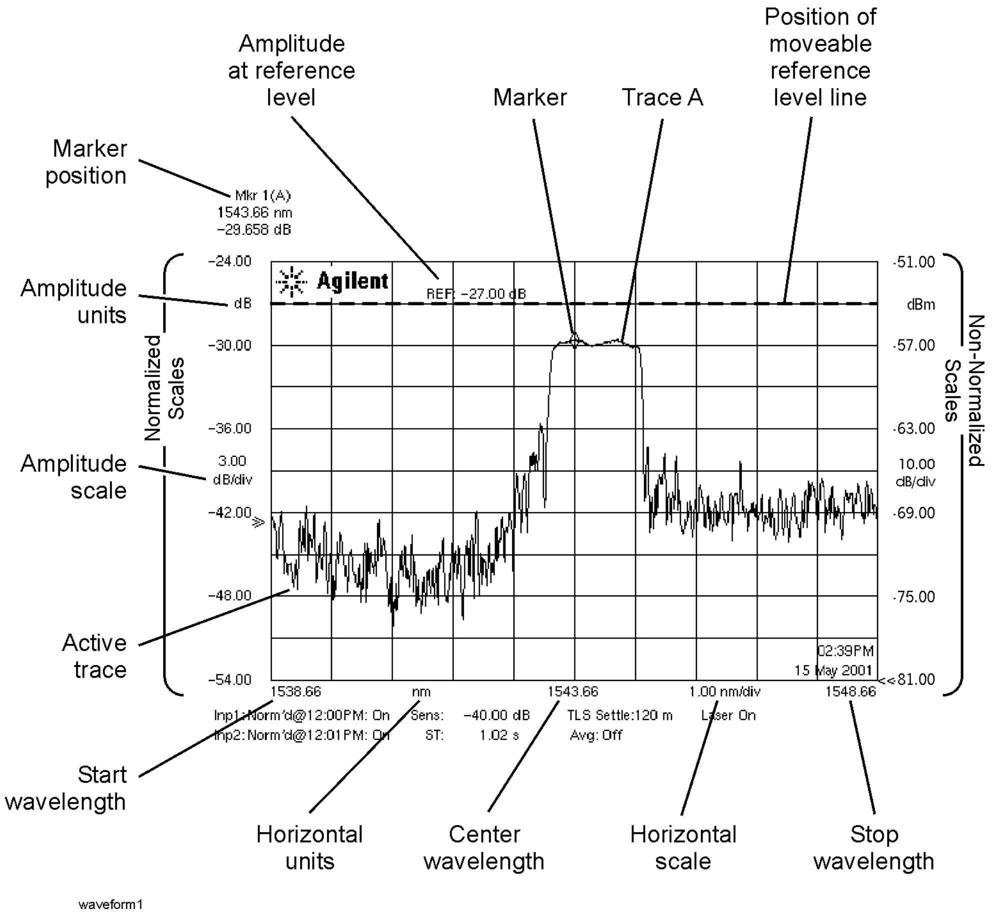

Transimpedance amplifier output
ADC sync out
External monitor connection (VGA compatible)
Auxiliary GPIB
PC compatible keyboard connector (Mini-DIN)
Service port (for service technicians only)
PC compatible mouse connector (Mini-DIN)
Ext trig
Printer connection
CURRENT SOURCE
TRANS-Z IN
TRANS-Z ANALOG OUT
EXT TRIG
ADC SYNC OUT
AUXILLARY GPIB
VGA
Parallel
SERVICE ONLY
LAN
SERVICE ONLY
GPIB
Line power
Remote interlock
Trigger out
Trigger in
WARNING
GPIB
Rem. Interlock
Trigger Out
Trigger In
Parallel
Serial
Service port (do not connect to network)
GPIB bus for remote control
GPIB
Serial port
PC compatible keyboard connector (Mini-DIN)
PC compatible mouse connector (Mini-DIN)
Printer connection
Line power
rear_annote

Figure 1-2. Rear Panel

WDCA Display



Mkr 1(A)
1543.66 nm
−29.658 dB

REF: −27.00 dB

Inp1: Norm'd@12:00PM: On
Inp2: Norm'd@12:01PM: On

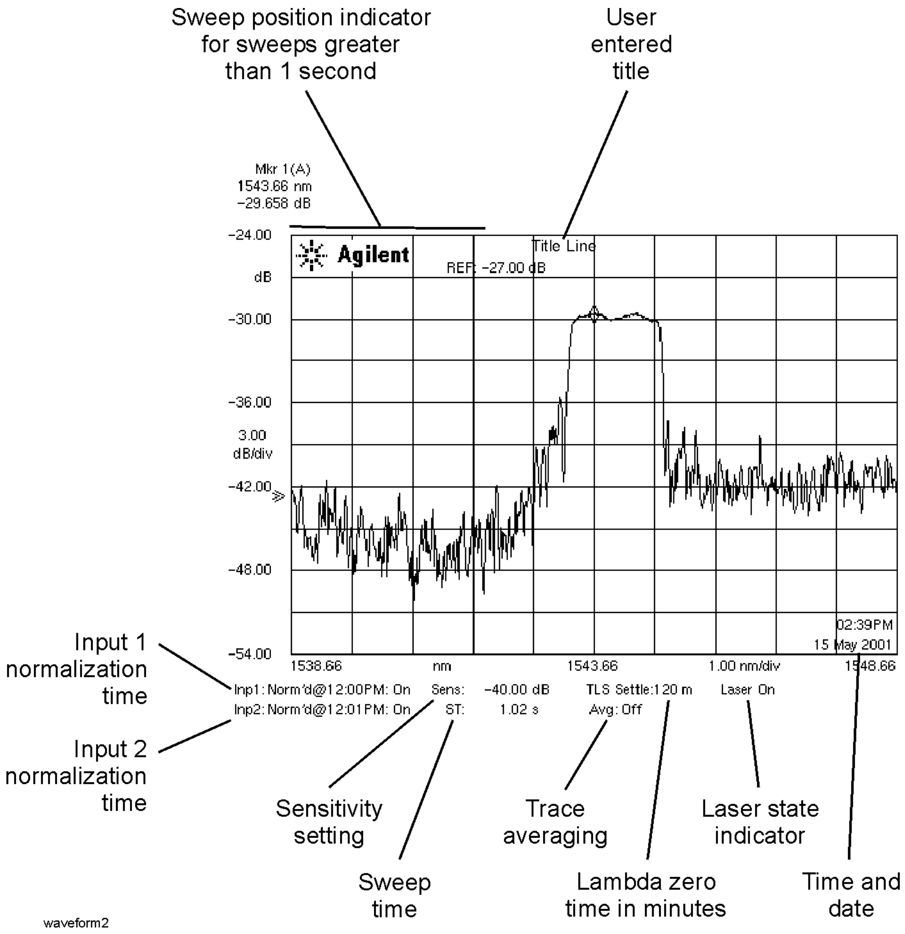
Sens: −40.00 dB
ST: 1.02 s

TLS Settle:120 m
Avg: Off

Laser On

02:39PM
15 May 2001

Figure 1-3. WDCA Display



Mkr 1(A)
1543.66 nm
−29.658 dB

Title Line
REF: −27.00 dB

02:39PM
15 May 2001

Inp1: Norm'd@12:00PM: On Sens: −40.00 dB TLS Settle:120 m Laser On
Inp2: Norm'd@12:01PM: On ST: 1.02 s Avg: Off

Figure 1-4. Display Annotations

Accessing the WDCA Functions

Instrument settings can be accessed from either the menu bar or the softkey panel.

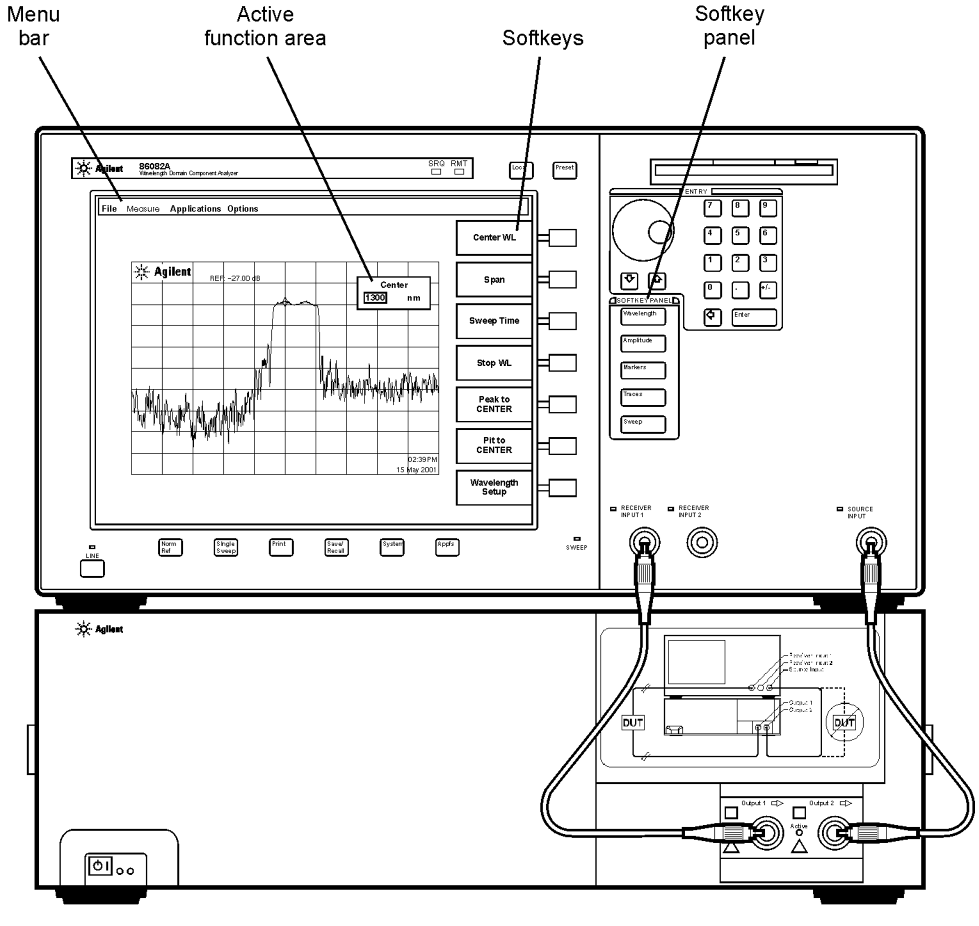
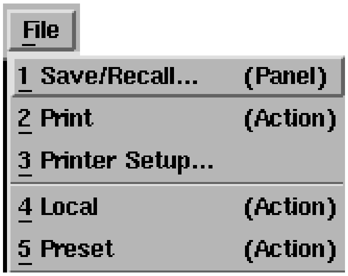


Figure 1-5. Accessing WDCA Functions

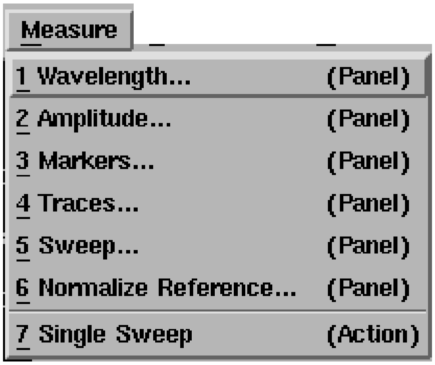
The Menu Bar

The Menu bar includes the File, Measure, Application, and Options drop-down menus. Each menu selection includes a descriptive label. Access to the Menu bar is accomplished with either a mouse or keyboard. You must restart the WDCA after installing a mouse or keyboard to activate.

File Menu

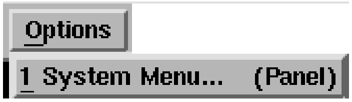


Measure Menu



Applications Menu Not used at this time

Options Menu

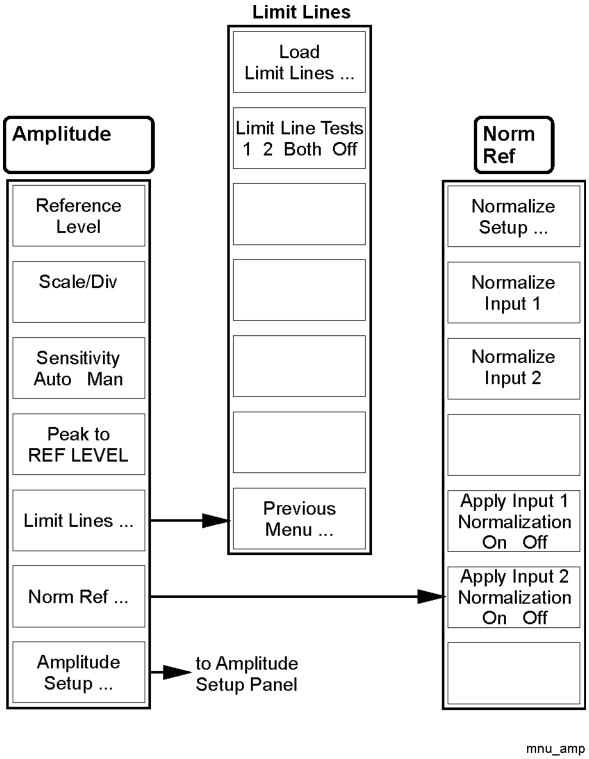


The Softkey Panels

Softkey panels are used to access the menus associated with each front-panel key. This section includes brief descriptions of the following menus. See Chapter 2, “Function Reference” for additional information on each of the analyzer functions.

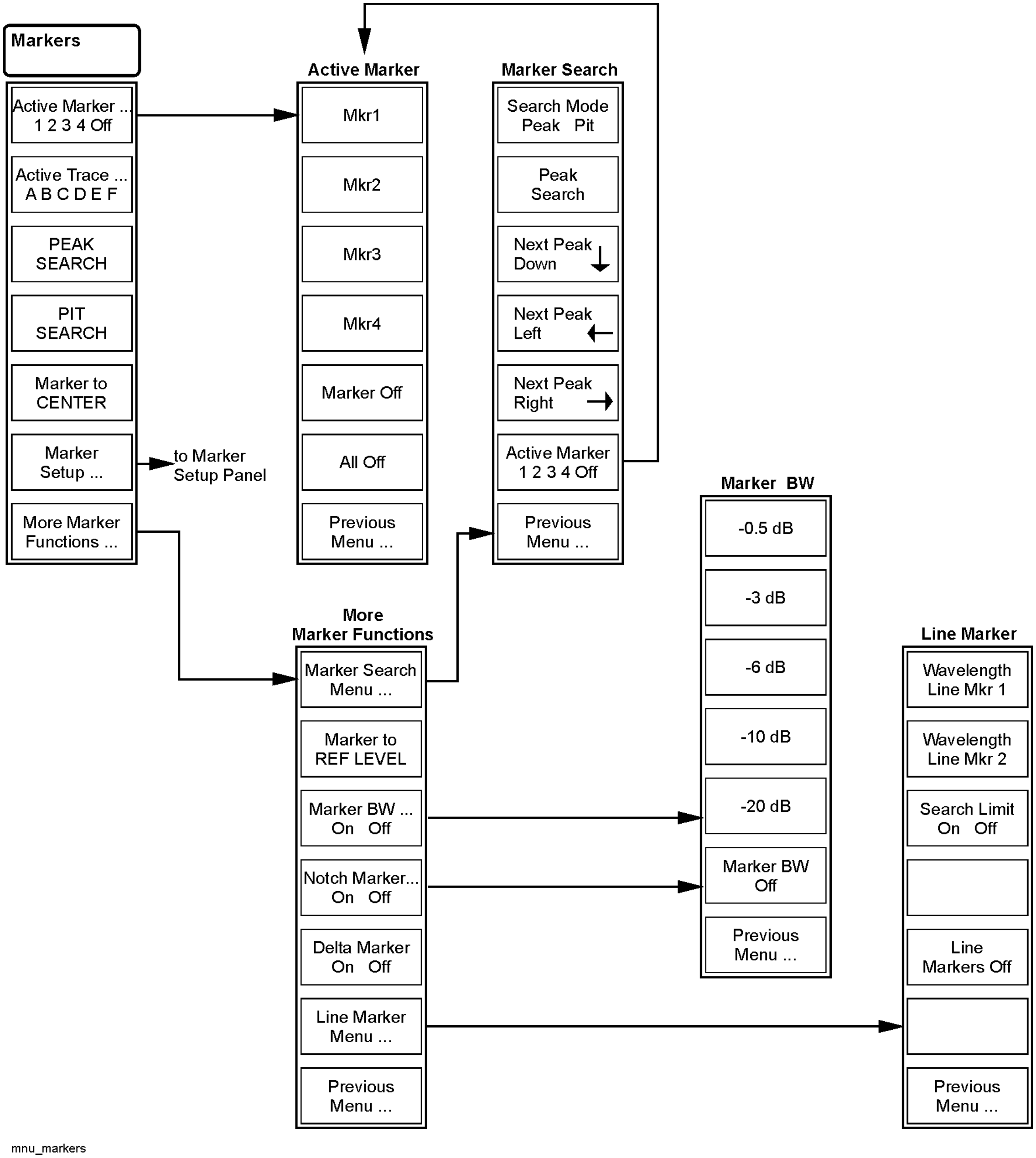
The Amplitude Menus

The Amplitude softkeys are accessed by using the front-panel Amplitude key or the Measure menu Amplitude selection on the menu bar.



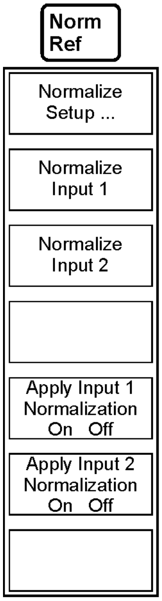
The Markers Menus

The Markers softkeys are accessed using the front-panel Markers key or the Measure menu Markers selection on the menu bar.


Markers
Active Marker ... 1 2 3 4 Off
Active Trace ... A B C D E F
PEAK SEARCH
PIT SEARCH
Marker to CENTER
Marker Setup ...
More Marker Functions ...
to Marker Setup Panel
Active Marker
Mkr1
Mkr2
Mkr3
Mkr4
Marker Off
All Off
Previous Menu ...
Marker Search
Search Mode Peak Pit
Peak Search
Next Peak Down
Next Peak Left
Next Peak Right
Active Marker 1 2 3 4 Off
Previous Menu ...
More Marker Functions
Marker Search Menu ...
Marker to REF LEVEL
Marker BW ... On Off
Notch Marker... On Off
Delta Marker On Off
Line Marker Menu ...
Previous Menu ...
Marker BW
-0.5 dB
-3 dB
-6 dB
-10 dB
-20 dB
Marker BW Off
Previous Menu ...
Line Marker
Wavelength Line Mkr 1
Wavelength Line Mkr 2
Search Limit On Off
Line Markers Off
Previous Menu ...
mnu_markers

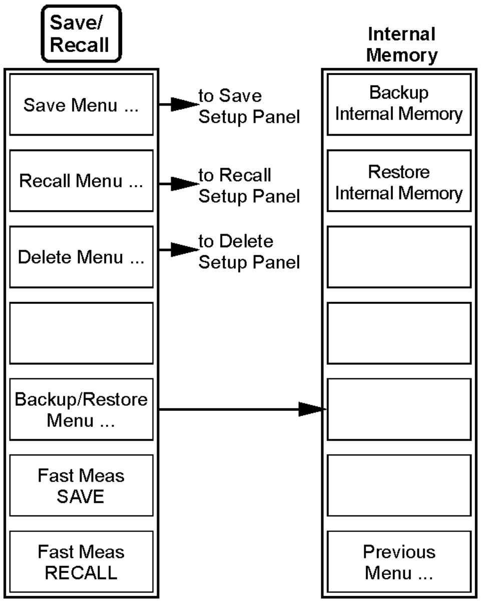
The Normalization Reference Menu

The Norm Ref softkey and setup panels are accessed by using the drop-down Measure menu Norm Ref selection, from the front-panel Norm Ref key, or from the front-panel Amplitude, Norm Ref.


Norm Ref
Normalize Setup ...
Normalize Input 1
Normalize Input 2
Apply Input 1 Normalization On Off
Apply Input 2 Normalization On Off

The Save/Recall Menus

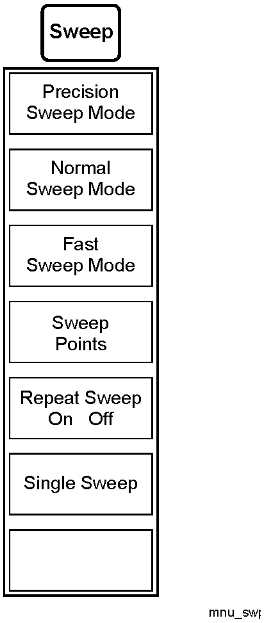
The Save/Recall softkeys and setup panels are accessed by using the drop-down File menu Save/Recall selection or the front-panel Save/Recall key. Use these functions to save, recall, and print the measurement results.



mnu_save

The Sweep Menu

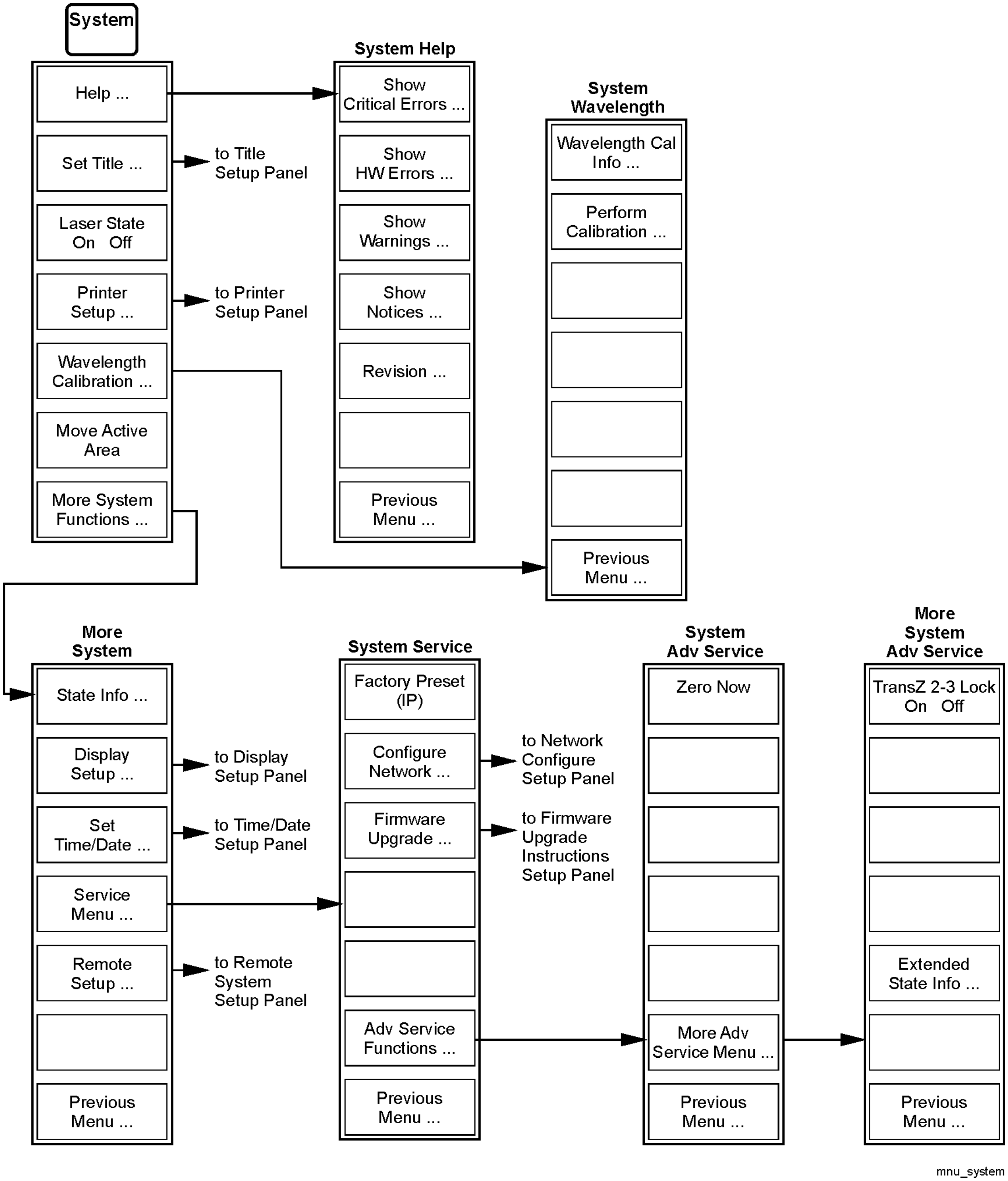
The Sweep softkeys are accessed using the front-panel Sweep key or the Measure menu Sweep selection on the menu bar.



This page left intentionally blank.

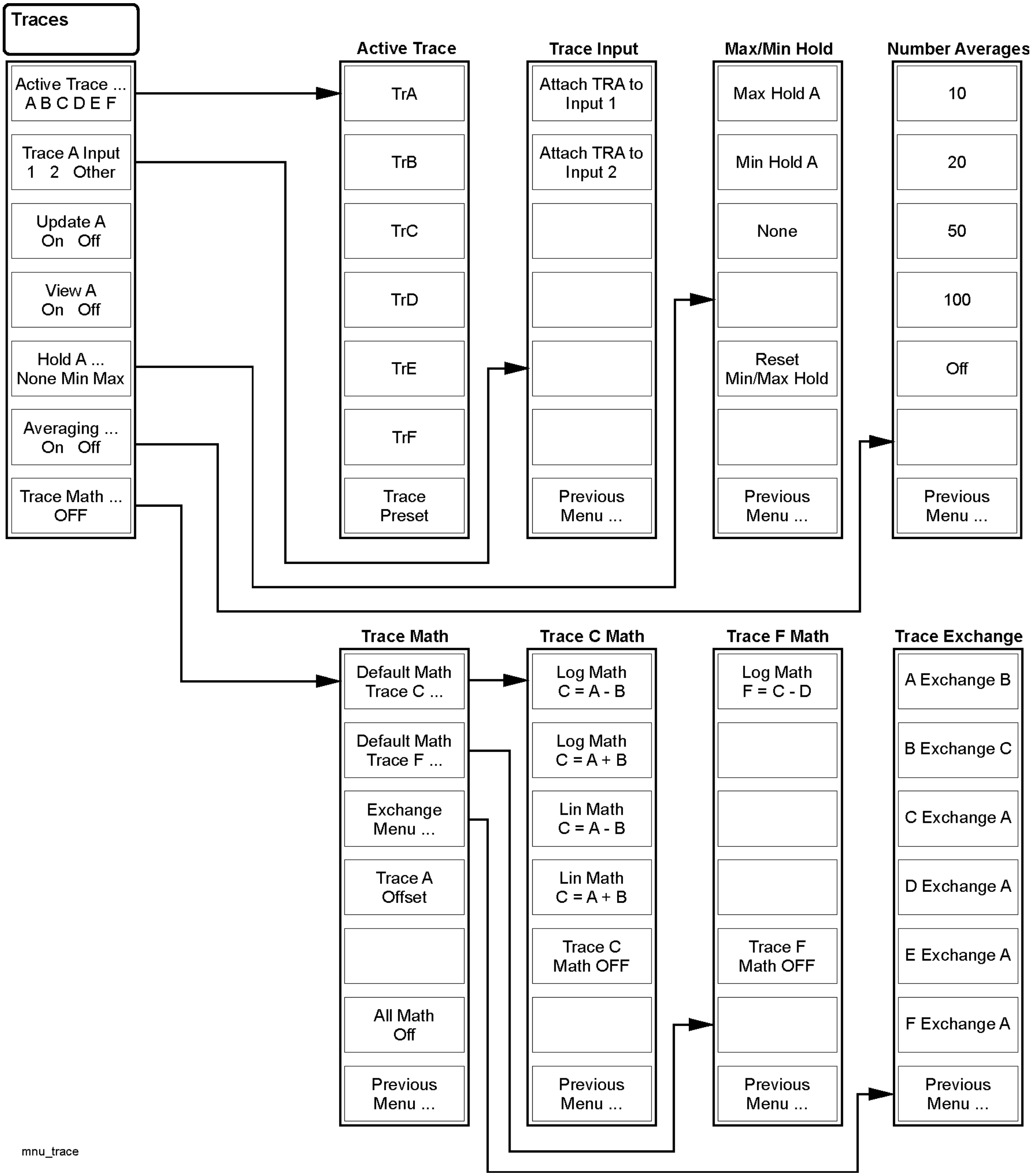
The System Menus

The System softkeys are accessed using the front-panel System key or the Options menu System selection on the menu bar.


System
Help ...
Set Title ...
to Title Setup Panel
Laser State On Off
Printer Setup ...
to Printer Setup Panel
Wavelength Calibration ...
Move Active Area
More System Functions ...
System Help
Show Critical Errors ...
Show HW Errors ...
Show Warnings ...
Show Notices ...
Revision ...
Previous Menu ...
System Wavelength
Wavelength Cal Info ...
Perform Calibration ...
Previous Menu ...
More System
State Info ...
Display Setup ...
to Display Setup Panel
Set Time/Date ...
to Time/Date Setup Panel
Service Menu ...
Remote Setup ...
to Remote System Setup Panel
Previous Menu ...
System Service
Factory Preset (IP)
Configure Network ...
to Network Configure Setup Panel
Firmware Upgrade ...
to Firmware Upgrade Instructions Setup Panel
Adv Service Functions ...
Previous Menu ...
System Adv Service
Zero Now
More Adv Service Menu ...
Previous Menu ...
More System Adv Service
TransZ 2-3 Lock On Off
Extended State Info ...
Previous Menu ...
mnu_system

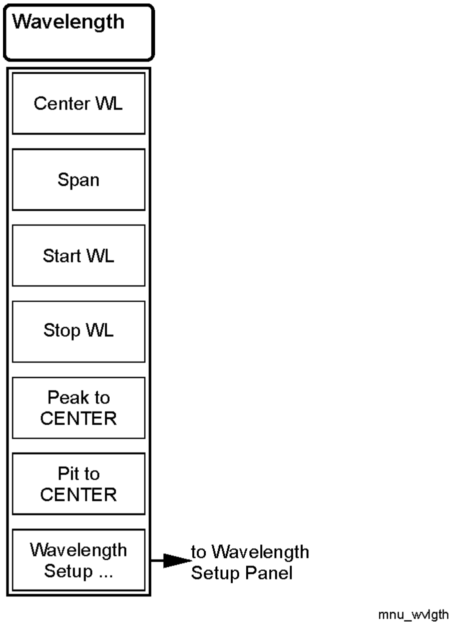
The Traces Menus

The Traces softkeys are accessed by using the front-panel Traces key or the Measure menu Traces selection on the menu bar.


Traces
Active Trace ... A B C D E F
Trace A Input 1 2 Other
Update A On Off
View A On Off
Hold A ... None Min Max
Averaging ... On Off
Trace Math ... OFF
Active Trace
TrA
TrB
TrC
TrD
TrE
TrF
Trace Preset
Trace Input
Attach TRA to Input 1
Attach TRA to Input 2
Previous Menu ...
Max/Min Hold
Max Hold A
Min Hold A
None
Reset Min/Max Hold
Previous Menu ...
Number Averages
10
20
50
100
Off
Previous Menu ...
Trace Math
Default Math Trace C ...
Default Math Trace F ...
Exchange Menu ...
Trace A Offset
All Math Off
Previous Menu ...
Trace C Math
Log Math C = A - B
Log Math C = A + B
Lin Math C = A - B
Lin Math C = A + B
Trace C Math OFF
Previous Menu ...
Trace F Math
Log Math F = C - D
Trace F Math OFF
Previous Menu ...
Trace Exchange
A Exchange B
B Exchange C
C Exchange A
D Exchange A
E Exchange A
F Exchange A
Previous Menu ...
mnu_trace

The Wavelength Menu

The Wavelength softkeys are accessed using the front-panel Wavelength key or the Measure menu Wavelength selection on the menu bar.



Laser Safety Considerations

NOTE Refer to the *Tunable Laser Modules User's Guide* for complete laser safety information.

Laser Safety

Laser radiation in the ultraviolet and far infrared parts of the spectrum can cause damage primarily to the cornea and lens of the eye. Laser radiation in the visible and near infrared regions of the spectrum can cause damage to the retina of the eye.

WARNING **Do NOT, under any circumstances, look into the optical output or any fiber/device attached to the output while the laser is in operation.**

This system should be serviced only by authorized personnel.

Do not enable the laser unless fiber or an equivalent device is attached to the optical output connector.

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

Electrical Safety

The electrical safety considerations are documented in the section "General Safety Considerations" on page -iii. Familiarize yourself with the safety markings and instructions before operating this system.

Setting Up the WDCA

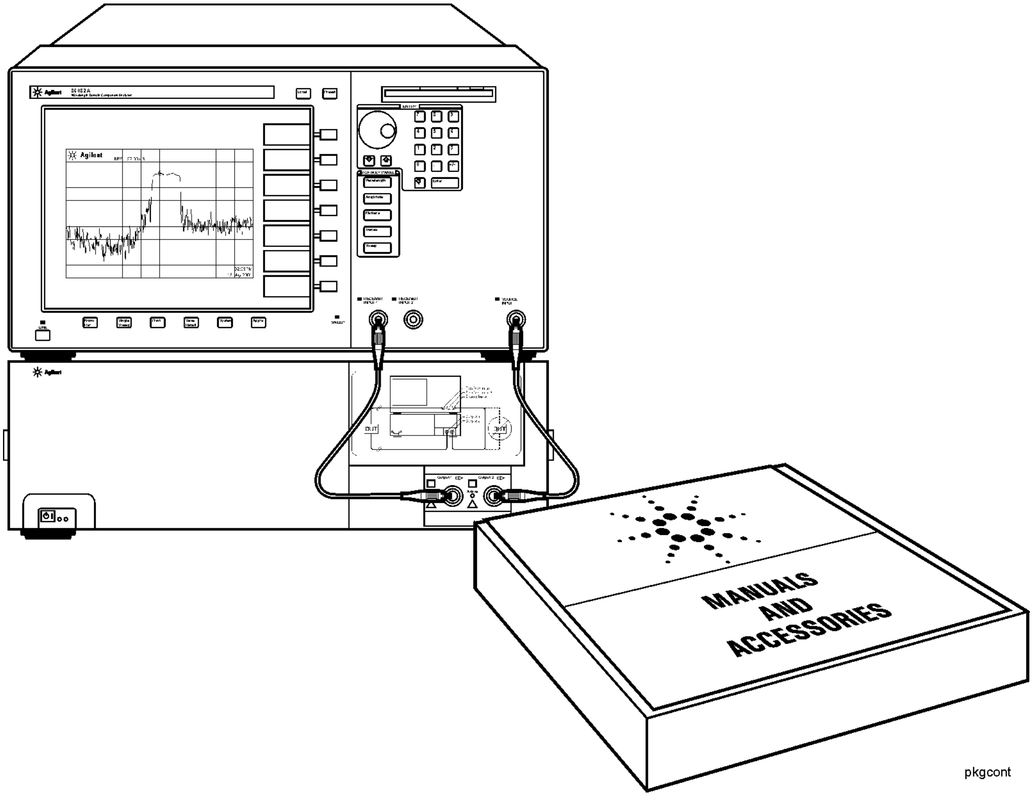


Figure 1-6. WDCA Shipment Contents

1 Unpack your shipment.

❐ Inspect the shipping containers for damage.

❐ Inspect the instrument(s).

❐ Verify that you received the options and accessories that you ordered.

Keep the shipping container and cushioning material until you have inspected the contents of the shipment for completeness and have checked the WDCA mechanically and electrically.

If anything is missing or defective, contact your nearest Agilent Technologies Sales Office. If the shipment was damaged, contact the carrier, then contact the nearest Agilent Technologies Sales office. Keep the shipping materials for the carrier's inspection. The Agilent Technologies Sales office will arrange for repair or replacement at Agilent Technologies' option without waiting for a claim settlement. Refer to "Returning the Instrument for Service" on page 8-24.

CAUTION

Fiber-optic connectors are easily damaged when connected to dirty or damaged cables and accessories. When you use improper cleaning and handling techniques, you risk expensive instrument repairs, damaged cables, and compromised measurements. Before you connect any fiber-optic cable to the WDCA, refer to "Cleaning Connections for Accurate Measurements" on page 8-11.

2 Clean and connect the cables and power cords as shown in Figure 1-7 and Figure 1-8.

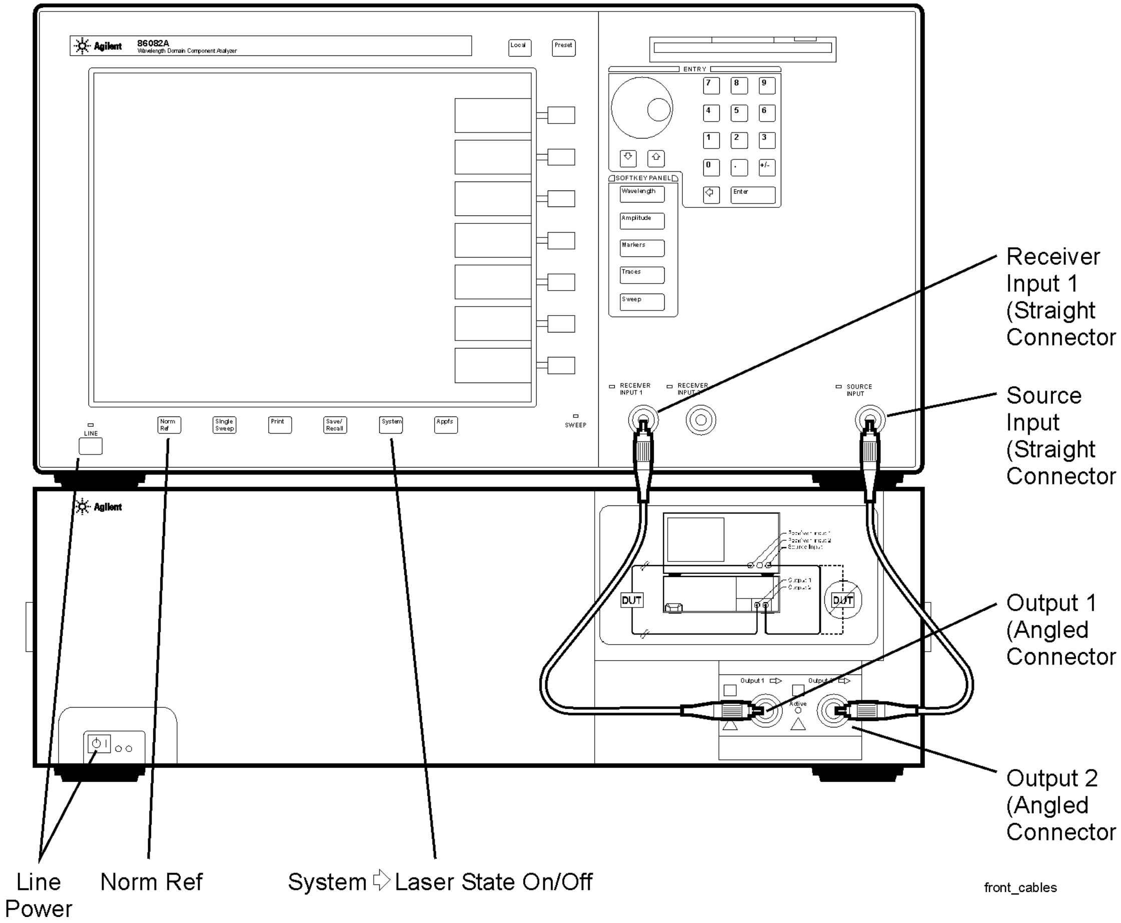


Figure 1-7. Front Panel Configuration

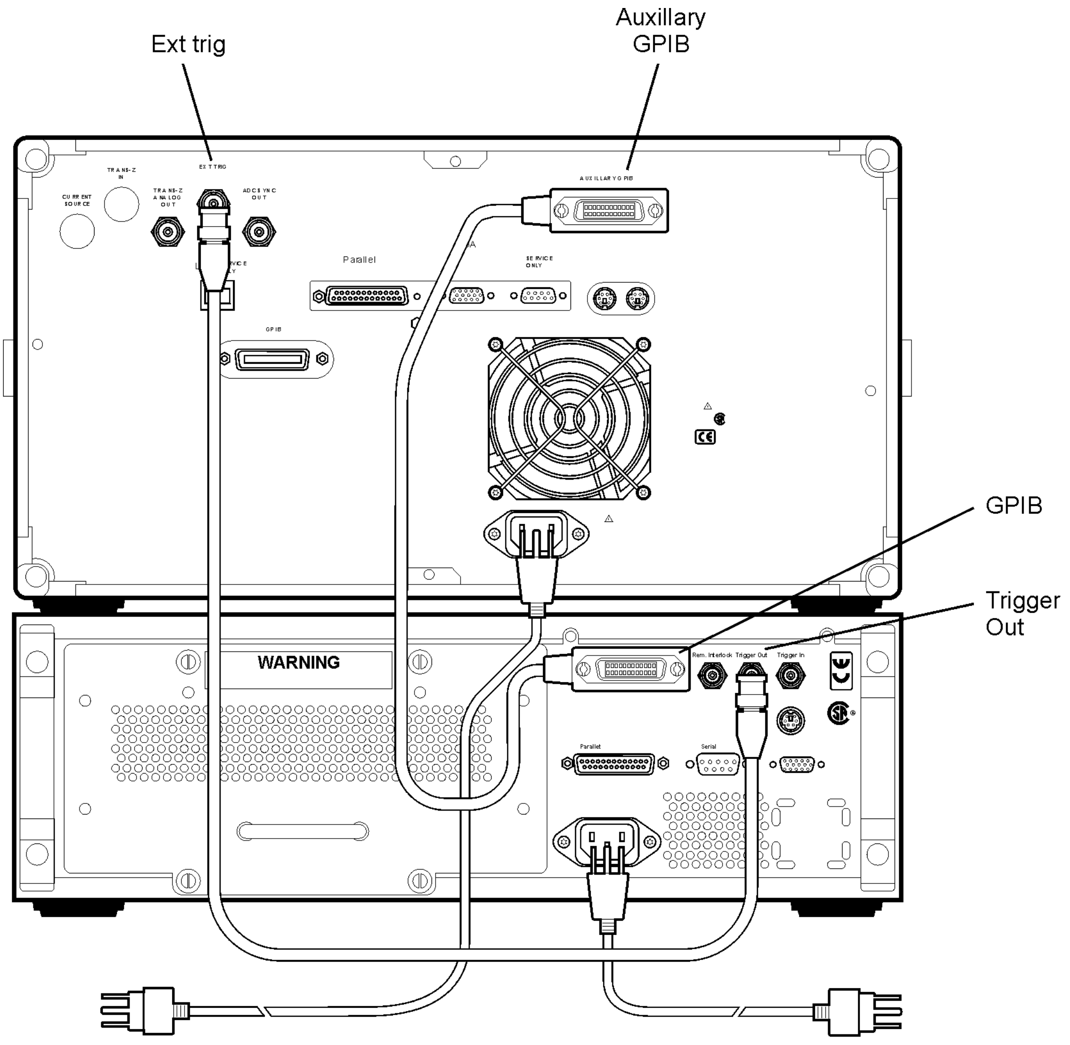


Figure 1-8. Rear Panel Configuration

3 Turn on both instruments. The start-up process takes approximately 15 minutes.

NOTE All specifications apply after the instrument's temperature has been stabilized after 1 hour of continuous operation.

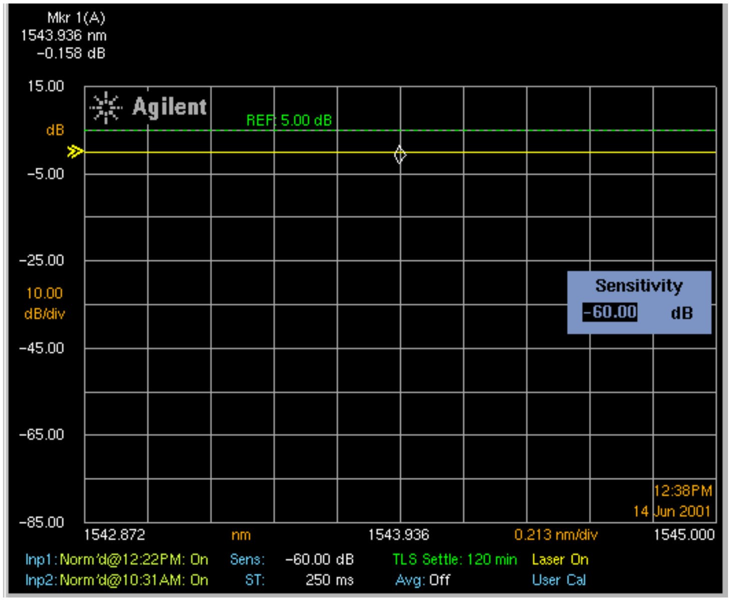
4 **For Option 201 only**, upgrade your TLS firmware using the firmware upgrade procedure provided on the 86082A CD-ROM.

5 Perform a wavelength calibration by pressing System > Wavelength Calibration > Perform Calibration. The calibration takes approximately 5 minutes.

6 Normalize Receiver Input 1 by pressing Norm Ref > Normalize Input 1.

7 When the normalization is complete, the software automatically takes a sweep.

8 Verify that the normalized trace is at 0 dB. You are now ready to start making measurements. Continue with "Using the Analyzer" on page 3-1.

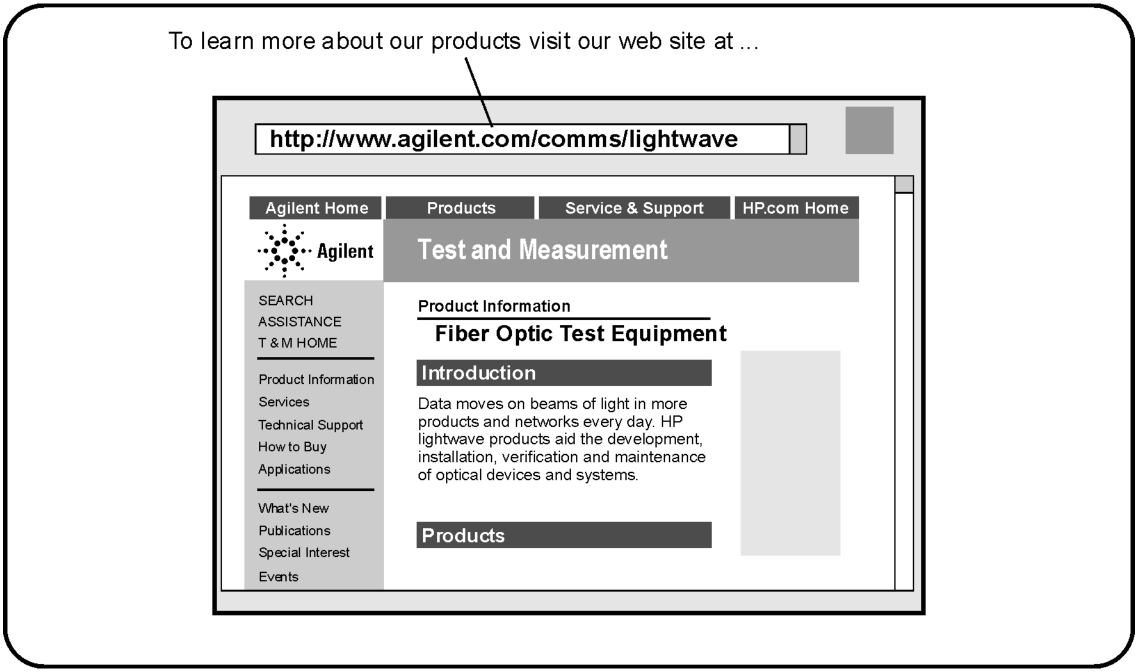


Mkr 1(A)
1543.936 nm
−0.158 dB

Sensitivity −60.00 dB

Inp1: Norm'd@12:22PM: On Sens: −60.00 dB TLS Settle: 120 min Laser On
Inp2: Norm'd@10:31AM: On ST: 250 ms Avg: Off User Cal

To Learn More About Agilent Products



To learn more about our products visit our web site at ...

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

Product Information

Fiber Optic Test Equipment

Introduction

Data moves on beams of light in more products and networks every day. HP lightwave products aid the development, installation, verification and maintenance of optical devices and systems.

Products

To learn more about this or any Agilent Technologies product, visit our web site at http://www.agilent.com.

To learn more about fiber optic test equipment, go to the Agilent Technologies home page listed above, and follow this path:

1 Click Communications.

2 Click Optical and Photonic Solutions.

This path will take you to the Optical and Photonic Solutions page. Alternately, you can enter the URL for this page directly:

http://www.agilent.com/cm/opticalgrp.shtml.

2

Function Reference

Function Reference

This chapter is an alphabetical reference of front panel keys, softkeys, and setup panel parameters. It is designed for quick information access. For example, during an operation you may find a key whose function is unfamiliar to you. Simply note the key name and look the key up in this chapter.

Active Function Area Assist

Sets the active function area to the first or top softkey function. When this function is on, the active area is automatically set. For example, when you press the front-panel Wavelength key, the active function area assist will set the active function to the Center WL. You can immediately enter the desired center wavelength without pressing the Center WL softkey. If this function is off, the active dialog box will not appear on the display. To access the individual dialog box, press the desired softkey function.

Key Path System > More System Functions > Display Setup

Related Functions Move Active Area

Active Marker

Accesses a menu to select an active marker. Selecting a marker always places that marker on the currently selected active trace at the center wavelength. Up to four markers can be simultaneously displayed. If multiple markers are displayed, the active marker is displayed as a white diamond and the other markers will be displayed as green diamonds.

See “General Tips for Using Traces and Markers” on page 3-29, “Moving the Active Marker from One Trace to Another” on page 3-30, and “Measuring the Delta between Two Traces” on page 3-30.

Key Path Markers > Active Marker

Related Functions Active Trace, Peak Search, Pit Search, Marker to Center

Remote Commands

```
CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:STATe
CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:AOFF
```

Active Trace

Accesses a menu to select an active trace for viewing, updating, or storing for Receiver Input 1 or Receiver Input 2. To put a marker on a trace, or to make changes to a trace, the trace must be active. There are six traces available.

Note There are two amplitude scales that can be viewed on the WDCA. The scale on the left side of the display is for normalized traces and will only be displayed when at least one active trace has normalization turned on. Active normalized traces are indicated with a “>>”. The scale on the right side of the display is for non-normalized traces and will only be displayed when at least one active trace is non-normalized. Active non-normalized traces are indicated with a “<<“. In the event that there is at least one normalized trace and one non-normalized trace, both amplitude scales with be displayed on the screen.

See “General Tips for Using Traces and Markers” on page 3-29, “Moving the Active Marker from One Trace to Another” on page 3-30, and “Measuring the Delta between Two Traces” on page 3-30.

Key Path Traces > Active Trace
Markers > Active Trace

Related Functions Trace Math, View (Trace)

Remote Commands

```
CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:TRACe
TRACe:FEED:INPute TRA,2
TRACe:FEED:CONTrol:TRA, ALWays
DISPlay:WINDow:TRACe TRA, ON
```

Adv Service Functions

Accesses the following functions:

- Extended State Info
- Trans Z 2-3 Lock
- Zero Now

Key Path System > More System Functions > Service Menu > Adv Service Functions

Agilent Logo (Display Setup Panel)

Accesses the panel to turn the Agilent logo on or off. When this function is on, the Agilent logo appears on the display's upper, left corner of the graticule and on printouts.

Key Path System > More System Functions > Display Setup

All Math Off

Turns all the math functions off.

Key Path Traces > More Trace Functions > Trace Math Off > All Math Off

All Off

Turns all the active markers off.

Key Path Markers > Active Marker > All Off

Remote Commands CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:AOFF

Amp Display Mode (Amplitude Setup Panel)

Specifies either a logarithmic or linear amplitude scale. The logarithmic scale reads in decibels. The linear scale is proportional to the input power. The bottom of the graticule line represents zero watts. The top of the graticule represents the reference level. Therefore, with 10 graticule lines, each division represents 10% of the input signal at the top of the screen.

Key Path Amplitude > Amplitude Setup > Amp Display Mode

Related Functions Amplitude Units

Remote Commands DISPlay:WINDow:TRACe:Y:SCALe LINear

Amplitude

Accesses a menu of keys that control the analyzer's sensitivity and amplitude scales. The Amplitude functions can be accessed from the front-panel Amplitude key or from the Measure menu Amplitude selection. The function keys include:

- amplitude setup
- limit lines
- normalize reference
- peak to ref level
- reference level
- scale/div
- sensitivity

Amplitude Setup

Accesses the Amplitude Setup panel used to specify amplitude related functions:

- amplitude display mode
- amplitude units
- auto ranging
- auto zero
- reference level position

Key Path Amplitude > Amplitude Setup...

Related Functions Reference Level, Sensitivity

Remote Commands

(Amplitude Display Mode):DISPlay:WINDow:TRACe:Y:SCALe LINear
DISPlay:WINDow:TRACe:Y:SCALe LOGarithmic
(Amplitude Units):UNIT:POWer
(Auto Zero):CALibration:ZERO
(Reference Level Position):DISPlay:WINDow:TRACe:Y:SCALe:RPOSition

Amplitude Units (Amplitude Setup Panel)

Selects the amplitude scale's units. Selecting Auto for amplitude units allows power to be shown in dB when using the logarithmic scale and in watts when using the linear scale. Selecting W forces the readout to be in watts, regardless of the amplitude scaling.

Note There are two amplitude scales that can be viewed on the WDCA. The scale on the left side of the display is for normalized traces and will only be displayed when at least one active trace has normalization turned on. Active normalized traces are indicated with a ">>". The scale on the right side of the display is for non-normalized traces and will only be displayed when at least one active trace is non-normalized. Active non-normalized traces are indicated with a "<<". In the event that there is at least one normalized trace and one non-normalized trace, both amplitude scales with be displayed on the screen.

Key Path Amplitude > Amplitude Setup > Amplitude Units

Related Functions Amp Display Mode, Apply Input 1/2 Normalization On Off

Remote Commands UNIT:POWer

Apply Input 1 Normalization On Off

Turns the trace normalization for Receiver Input 1 on or off. Refer to "Normalizing the Measurement" on page 3-11.

Key Path Norm Ref > Apply Input 1 Normalization On Off

Related Functions Normalize Input 1, Normalize Setup

Remote Commands SENSe:NORMalize:ENABle
SENSe:NORMalize:ENABle?

Apply Input 2 Normalization On Off

Turns the trace normalization for Receiver Input 1 on or off. Refer to "Normalizing the Measurement" on page 3-11.

Key Path Norm Ref > Apply Input 2 Normalization On Off

Related Functions Normalize Input 2, Normalize Setup

Remote Commands SENSe:NORMalize:ENABle
SENSe:NORMalize:ENABle?

Attach Trace A to Input 1

Associates the active trace to Receiver Input 1. You can assign up to six traces to one input. For example, you may choose to assign three traces to input 1. Trace A could have View on, but Update turned off. Trace B could have Update on with Max Hold on. Trace C could have Update and View on.

Key Path Traces > Trace A Input > Attach TRA to Input 1

Related Functions View (trace), Update (trace)

Remote Commands TRACe:FEED:INPut TRA, 1

Attach Trace A to Input 2

Associates the active trace to Receiver Input 2. You can assign up to six traces to one input. For example, you may choose to assign three traces to input 2. Trace A could have View on, but Update turned off. Trace B could have Update on with Max Hold on. Trace C could have Update and View on.

Key Path Traces > Trace Input A > Attach TRA to Input 2

Related Functions View (trace), Update (trace)

Remote Commands TRACe:FEED:INPut TRA, 2

Auto Ranging (Amplitude Setup Panel)

When on, allows the WDCA to achieve the desired sensitivity by taking several sweeps and switching between gain stages. The final trace data is a blended composite of each trace taken in the different gain stages.

Note The sweep time (ST) that is displayed in the lower portion of the display is the time for the WDCA to sweep over one gain stage. The WDCA may take up to three sweeps in three different gain stages to make the measurement. This depends on the settings for sensitivity, reference level, auto range and also the particular device being measured. The final data trace is a blended composite of each trace taken in the different gain stages. For more information, refer to “Increase in Sweep Time” on page 3-32.

Setting Auto Range to off will force the WDCA to use a single gain stage and achieve the desired sensitivity by applying a digital filter.

Key Path Amplitude > Amplitude Setup > Auto Range

Auto Zero (Amplitude Setup Panel)

Turns Auto Zero on and off. Auto Zero on enables the internal amplifiers to be zeroed between sweeps. In this mode, the analyzer compensates for temperature-related current drift between each sweep. Although this “zeroing” increases amplitude accuracy, it also increases the time between sweeps. If amplitude accuracy is not critical to your measurement, turning off auto zero will improve throughput. Pressing the front-panel Preset key resets this function to on.

Tip: The analyzer performs a more complete zeroing when the instrument is first turned on. You can run this more complete routine anytime by pressing the Zero Now softkey.

Key Path Amplitude > Amplitude Setup > Auto Zero

Related Functions Zero Now

Remote Commands CALibration:ZERO

Averaging

Selects the number of measurement sweeps to be averaged. When on is selected, the number of sweeps to be averaged is selected from the softkeys. Alternately, averaging values other than those displayed can be entered by using the numeric keypad or the knob when the Avg Count active function area is displayed. 1 to 1000 averages can be set.

When the number of sweeps taken is less than the count, the following formula is used to calculate the data:

$$AVG = \frac{sum\ of\ current\ sweeps}{number\ of\ averages\ selected}$$

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

$$\left(New\ average = \frac{count - 1}{count} \times last\ average + \frac{new\ measurement}{count} \right)$$

Key Path Traces > Averaging

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

Backup Internal Memory

Accesses the analyzer backup utility. A screen appears asking you to insert a formatted floppy disk into the floppy drive. The existing contents of the internal memory is then saved on the disk. The Backup Internal Memory function overwrites the floppy disk with a new image. Any existing files or catalogs on the floppy disk will be destroyed. The disk will not be viewable on a PC and no trace or measurement files can be saved onto the disk until it is reformatted. Any successive backup operations will overwrite the previous backup information, so only the latest backup information can be recovered through the Restore Internal Memory operation.

Key Path Save/Recall > Backup/Restore Menu >Backup Internal Memory

Backup/Restore Menu

Accesses the analyzer utilities used to backup or restore internal memory.

Key Path Save/Recall > Backup/Restore Menu

Bandwidth Marker Interpolation (Marker Setup Panel)

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

Key Path Markers > Marker Setup > Bandwidth Marker Interpolation

Related Functions Active Marker, Delta Marker

Remote Commands CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BANDwidth
:INTerpolation

BW Marker Units (Marker Setup Panel)

Sets the bandwidth marker X-axis readout for frequency or wavelength when the instrument is in a non-zero span. The available selections are nm, µm, Ang, GHz, and THz. The default unit is nm (nanometers). This setting controls all four bandwidth markers. The normal and delta offset markers have their own settings.

Key Path Markers > Marker Setup > BW Marker Units

Related Functions Marker BW

Remote Commands CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BANDwidth:READout

Center Wavelength (WL)

Sets the center wavelength using the knob, step keys, or numeric keys. The span remains constant. The center wavelength and the start and stop wavelength settings are related as follows:

$$\textit{Center Wavelength} = \frac{\textit{stop wavelength} + \textit{start wavelength}}{2}$$

Key Path Wavelength > Center WL

Related Functions Start WL, Stop WL, Span

Remote Commands SENSe:WAVelength:CENTer

Center Wavelength Step Size (Wavelength Setup Panel)

Specifies the center wavelength step size. This is used for incrementing center wavelength using the ↑ and ↓ keys.

Key Path Wavelength > Wavelength Setup > Center Wavelength Step Size

Remote Commands SENSe:WAVelength:CENTer:STEP:INCRement

Configure Network

Starts a utility to enable and configure networking. Enter this menu only if you have all the necessary networking parameters. Refer to “Information and Equipment Required for the Configuration Process” on page 4-2.

Key Path System > More System Functions > Service Menu > Configure Network

Current GPIB Address (Remote Setup Panel)

Displays the GPIB address currently being used. To change the address, use the numeric keys or knob to enter the new GPIB address information in the Remote Setup panel.

Key Path System > More System Functions > Remote Setup

Default Math Trace C

Defines the math expression to be used and turns the math operation on: $C=A_{LOG}$-B, $C=A_{LOG}$+B, $C=A_{LIN}$-B, or $C=A_{LIN}$+B. The result is placed in Trace C. The math operation is performed in linear units.

See “Upper and lower limit lines showing pass/fail test” on page 3-39 and See “Using Trace Math to Measure Wavelength Drift” on page 3-41.

Key Path Traces > More Trace Functions > Trace Math Off > Default Math Trace C

Remote Commands CALCulate3:MATH:STATE OFF|ON|0|1

Default Math Trace F

Defines the math expression to be used and turns the math operation on: $F=C_{LOG}-D$. The result is placed in Trace F. The math operation is performed in linear units.

See “Upper and lower limit lines showing pass/fail test” on page 3-39 and See “Using Trace Math to Measure Wavelength Drift” on page 3-41.

Key Path Traces > More Trace Functions > Trace Math Off > Default Math Trace F

Remote Commands CALCulate6:MATH:STATE OFF|ON|0|1

Delete File From (Delete Setup Panel)

Selects whether to delete a file from a floppy disk or from internal memory.

Key Path Save/Recall > Delete Menu > Delete File From

Remote Commands MMEMory:DELete

Delete Menu

Accesses the Delete Setup panel. This is used to delete files from a floppy disk or from internal memory.

Key Path Save/Recall > Delete Menu

Remote Commands MMEMory:DELete

Delta Marker On/Off

This toggle function fixes the position of the reference marker and activates the delta marker. This measures the difference between the active (reference) marker and the delta marker. The front knob or keypad can be used to move the delta marker to the desired location.

If the delta function is turned on and no marker is active, Marker #1 will be turned on and placed at the center wavelength as a reference for the delta marker.

Key Path

Markers > More Marker Functions > Delta Marker

Remote Commands

CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:DELTa:STATe

Delta Marker Units (Marker Setup Panel)

Sets the delta marker X-axis readout for frequency or wavelength when the instrument is in a non-zero span. The available selections are nm, µm, Ang, GHz, and THz. The default units are nm (nanometers). This setting controls all four delta markers. The normal and bandwidth markers have their own settings.

Key Path

Markers > Marker Setup > Delta Marker Units

Related Functions

Delta Marker

Remote Commands

CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:DELTa:X:READout

Display Setup

Accesses the display setup panel selections which include:

- active function area assist
- Agilent logo
- date/time
- title

Key Path

System > More System Functions > Display Setup

Remote Commands

DISPlay:WINDow:TEXT:DATA <string>

Exchange Menu

Accesses the exchange selections which include:

- A Exchange B
- B Exchange C
- C Exchange A
- D Exchange A
- E Exchange A
- F Exchange A

Exchanges the X- and Y-axis data of the two traces. The only trace pairs that can be exchanged are Trace A with any trace, and Trace B with Trace C.

Key Path

Traces > More Trace Functions > Trace Math > Exchange Menu

Remote Commands

TRACe:EXCHange

Extended State Information

Displays the Hi Gain TransZ and the Lo Gain TransZ amplifier stage values.

Key Path System > More System Functions > Service Menu > Adv Service Functions > More Adv Service Menu > Extended State Info

Factory Preset (IP)

Restores auto align settings to original factory values and will clear the optional settings. See “Preset” on page 2-44.

Key Path System > More System Functions > Service Menu > Factory Preset (IP)

Related Functions Preset

Fast Meas Recall

Accesses the Fast Measure Recall function to recall the measurement state previously saved as FASTSAVE.dat by the Fast Measure Save function.

Key Path Save/Recall > Fast Meas Recall

Related Functions Recall

Fast Meas Save

Accesses the Fast Measure Save function to save the measurement to internal memory. The instrument saves the current measurement state to internal memory as FASTSAVE.dat. Only one FASTSAVE.dat file exists, so performing a Fast Meas Save will overwrite any currently existing Fast Save file.

Key Path Save/Recall > Fast Meas Save

Related Functions Save

Fast Sweep Mode

Selects the fastest sweep mode. Fast mode allows wider spans with faster sweep speed. In this mode you can measure filters with multiple channels or see if the rejection band holds over wide ranges. Fast sweep mode may not be appropriate to use if the filter has very sharp responses. The fast sweeping speed may not accurately display the signal.

Note The sweep time (ST) that is displayed in the lower portion of the display is the time for the WDCA to sweep over one gain stage. The WDCA may take up to three sweeps in three different gain stages to make the measurement. This depends on the settings for sensitivity, reference level, auto range and also the particular device being measured. The final data trace is a blended composite of each trace taken in the different gain stages. For more information, refer to “Increase in Sweep Time” on page 3-32.

Key Path Sweep > Fast Sweep Mode

Related Functions Normal Sweep Mode, Precision Sweep Mode

Remote Commands [SENSe]:SWEep:SPEed

File Name (Save Setup Panel)

Selects automatic or manual mode for saving a filename.

When Auto is selected, the measurement is automatically named and saved to an auto generated filename when the Auto Save softkey is pressed. Measurement data is saved to the filename ST_xxxxx.dat and Trace(s) only is saved to the filename TR_xxxxx.csv. “xxxxx” represents a numbering system which is incremented each time the measurement data is saved. For example, the first time you save Trace(s) only data, the filename TR_00001.csv will be assigned.

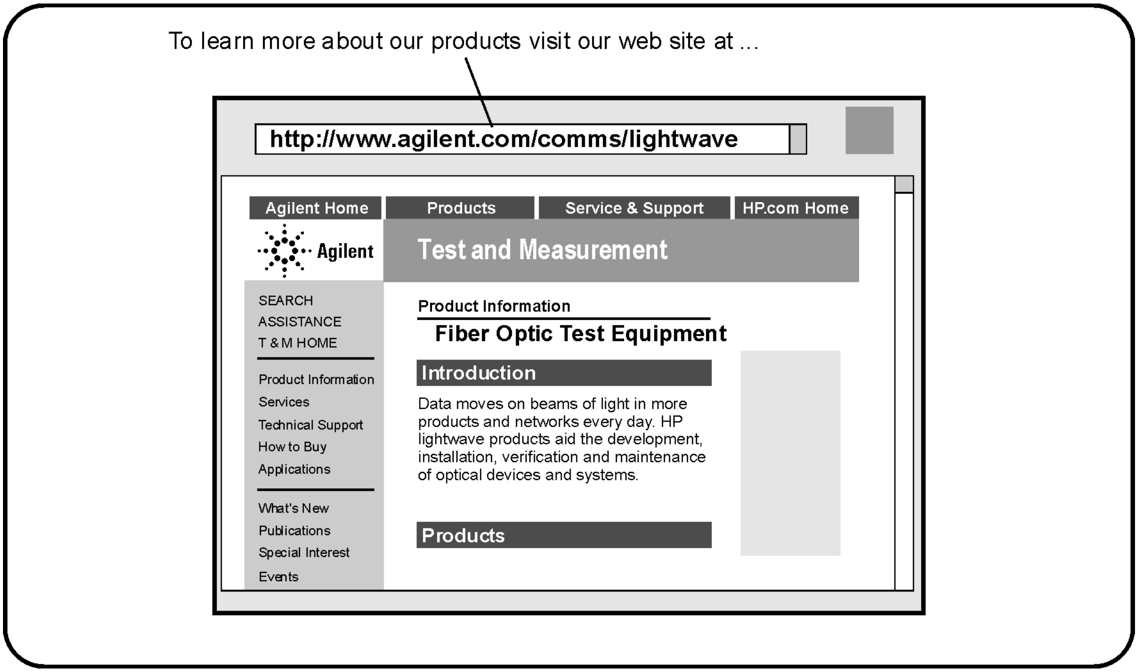
When Manual is selected, pressing the Choose File to Save softkey will access the File Name Menu setup panel. Use the front panel step keys, and arrow softkeys to highlight and then select each letter in the filename. When you finish entering the filename, press SAVE FILE.

Key Path Save/Recall > Save Menu > File Name

Remote Commands *SAV

Firmware Upgrade

The instrument will restart into a utility to upgrade the instrument firmware. Refer to the CD ROM which was included with your shipment or to the website below for upgrade instructions.



To learn more about our products visit our web site at ...

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

Data moves on beams of light in more products and networks every day. HP lightwave products aid the development, installation, verification and maintenance of optical devices and systems.

Key Path	System > More System Functions > Service Menu > Firmware Upgrade
Remote Commands	MMEMory:INITialize

Help

Opens a window displaying hardware errors, critical errors, notices, and warnings as well as displaying the analyzer's current firmware revision. You can print the queue, clear the queue, or page up and down to view the information. If no hardware errors, critical errors, notices, or warnings have been generated, these functions will be shaded.

Key Path

System > Help

Remote Commands

SYSTem:HELP:HEADer
SYSTem:ERRor?
*IDN

Hold A...F None Min Max (trace)

Hold Max compares the current amplitude value of each point on the active trace in the current sweep to the corresponding point detected during the previous sweep, then displays the maximum value.

Hold Min compares the current amplitude value of each point on a trace in the current sweep to the corresponding point detected during the previous sweep, then displays the minimum value.

Hold None turns the Hold function off.

Key Path

Traces > Hold None Min Max

Remote Commands

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

Laser State On Off

Sets the on/off state of the laser source.

Key Path

System > Laser State On Off

Remote Commands

```
OUTPut:STATe
```

Limit Line Tests 1 2 Both Off

Allows you to test the shape of the displayed amplitude response. Because limit lines allow you to perform repetitive testing of response shapes, they are ideal for pass/fail testing.

There are no restrictions on what type of limit line (upper or lower) is chosen for limit line 1 and 2. You may choose to designate limit line 1 as an upper limit line and limit line 2 as a lower limit line. Using the limit lines in this way will determine if the amplitude response passes or fails the limit line boundaries.

You could also choose to designate both limit lines 1 and 2 as upper limit lines. For this example, limit line 1 could be used as a hard specification and limit line 2 as a margin.

See “Creating a Limit Line to Establish Pass/Fail Criteria” on page 3-36.

Key Path

Amplitude > Limit Lines > Limit Line Tests 1 2 Both Off

Related Functions

Load Limit Lines

Remote Commands

```
:CALCulate[1|2|3|4|5|6]:LLINe: [1| 2]:DISPlay
:CALCulate[1|2|3|4|5|6]:LLINe: [1| 2]: FAIL?
:CALCulate[1|2|3|4|5|6]:LLINe: [1| 2]:STATe
:MMEMory:LOAD:LLINE 1|2, <file_name>, [INTernal|FLOPpy]
```

Lin Math C=A–B

Subtracts Trace B from Trace A point by point, then stores the results in Trace C in linear units. If Trace A is in View On mode, this function is continuous and occurs every sweep. Results of the subtraction are referenced to 0 dB. This often requires a reference level adjustment. For example, if the reference level is set at –40 dBm, Trace A is at –50 dBm, and Trace B is at –55 dBm, the C=A–B function places Trace C at –5 dB. Because this is off screen, the reference level will need to be adjusted.

Key Path Traces > More Trace Functions > Trace Math OFF > Default Math Trace C > Lin Math C=A–B

Remote Commands CALCulate3:MATH:EXPRession(TRA–TRB)

Lin Math C=A+B

Adds Trace A and Trace B point by point, then stores the results in Trace C in linear units. The results of this function often require a reference level adjustment. Trace A is placed in View On mode. Trace math is mainly used to normalize the display during stimulus-response measurements.

Key Path Traces > More Trace Functions > Trace Math OFF > Default Math Trace C > Lin Math C=A+B

Remote Commands CALCulate3:MATH:EXPRession(TRA+TRB)

Line Markers Off

Turns off all line markers and line marker functions.

Key Path Markers > More Marker Functions > Line Marker Menu > Line Markers Off

Remote Commands CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:SRANge:OFF

Load Limit Line 1/2

Allows you to select from the available limit lines for limit line testing. Each limit line file is designated as either upper or lower limit lines for use in the limit line test. An upper limit line is identified by a “u” at the beginning of the file name. A lower limit line is identified by an “l” at the beginning of the file name.

See “Creating a Limit Line to Establish Pass/Fail Criteria” on page 3-36.

There are no restrictions on what type of limit line (upper or lower) is chosen for limit line 1 and 2. You may choose to designate limit line 1 as an upper limit line and limit line 2 as a lower limit line. Using the limit lines in this way will determine if the amplitude response passes or fails the limit line boundaries.

You could also choose to designate both limit lines 1 and 2 as upper limit lines. For this example, limit line 1 could be used as a hard specification and limit line 2 as a margin.

A limit line floppy disk is supplied with your WDCA. You can choose from existing limit lines or create a new limit line that best suits your needs. Refer to “Creating a Limit Line to Establish Pass/Fail Criteria” on page 3-36 for additional information.

Tip: When choosing a file name, you may want to indicate the limit line as either upper or lower. Try to limit the file name to eight characters as only eight characters will be displayed in the Limit Line Catalog.

Key Path Amplitude > Limit Lines > Load Limit Lines > Load Limit Line 1

Related Functions Limit Line Test 1 2 Both Off

Remote Commands (To load limit definitions): MMEMory:LOAD:LLINe 1|2, <file_name>, [INTernal|FLOPpy]
(To transfer limit files over the GPIB):MMEMory:DATA
:CALCulate[1|2|3|4|5|6]:LLINe: [1| 2]:DISPlay [?] {ON|OFF|0|1}
:CALCulate[1|2|3|4|5|6]:LLINe: [1| 2]: FAIL?
:CALCulate[1|2|3|4|5|6]:LLINe: [1| 2]:STATe[?] {ON|OFF|1|0}

Load Limit Lines

Accesses a menu to specify limit lines 1 and 2.

See “Creating a Limit Line to Establish Pass/Fail Criteria” on page 3-36.

Key Path Amplitude > Limit Lines > Load Limit Lines

Related Functions Load Limit Line 1/2

Remote Commands

(To load limit definitions):MMEMory:LOAD:LLINE 1|2, <file_name>, [INTernal|FLOPpy]
(To transfer limit files over the GPIB):MMEMory:DATA
:CALCulate[1|2|3|4|5|6]:LLINe: [1| 2]:DISPlay
:CALCulate[1|2|3|4|5|6]:LLINe: [1| 2]: FAIL?
:CALCulate[1|2|3|4|5|6]:LLINe: [1| 2]:STATe

Local

Restores front-panel control of the instrument. Whenever the instrument is in Remote mode, the RMT message is displayed on the instrument's screen and all keys are disabled except for the front-panel Local key.

Key Path

Local

Remote Commands

GPIB GoTo Local Command

Log Math C=A–B

Subtracts Trace B from Trace A point by point, then stores the results in Trace C in logarithmic units. If Trace A is in View On mode, this function is continuous and occurs every sweep. Results of the subtraction are referenced to 0 dB. This often requires a reference level adjustment. For example, if the reference level is set at –40 dBm, Trace A is at –50 dBm, and Trace B is at –55 dBm, the C=A–B function places Trace C at –5 dB. Because this is off screen, the reference level will need to be adjusted.

Key Path

Traces > More Trace Functions > Trace Math OFF > Default Math Trace C > Log Math C=A–B

Related Functions

Log Math C=A+B, Log Math F=C–D

Remote Commands

CALCulate3:MATH:EXPRession(TRA/TRB)

Log Math C=A+B

Adds Trace A and Trace B point by point, then stores the results in Trace C in logarithmic units. The results of this function often require a reference level adjustment. Trace A is placed in View On mode. Trace math is mainly used to normalize the display during stimulus-response measurements.

Key Path Traces > More Trace Functions > Trace Math OFF > Default Math Trace C > Log Math C=A+B

Related Functions Log Math C=A–B, Log Math F=C–D

Remote Commands CALCulate3:MATH:EXPRession(TRA*TRB)

Log Math F=C–D

Subtracts Trace D from Trace C point by point, then stores the results in Trace F in logarithmic units. If Trace C is in View On mode, this function is continuous and occurs every sweep. Results of the subtraction are referenced to 0 dB. This often requires a reference level adjustment. For example, if the reference level is set at –40 dBm, Trace D is at –50 dBm, and Trace C is at –55 dBm, the F=C–D function places Trace F at –5 dB. Because this is off screen, the reference level will need to be adjusted.

Key Path Traces > More Trace Functions > Trace Math OFF > Default Math Trace C > Log Math F=C–D

Related Functions Log Math C=A–B, Log Math C=A+B

Remote Commands CALCulate6:MATH:EXPRession(TRC/TRD)

Marker BW

Used to measures the passband of the signal. To enter the bandwidth amplitude of the bandwidth markers, make a selection from the softkeys (–0.5, –3 dB, –6 dB, or –10 dB) or use the knob, step keys, or numeric keys.

If a peak search is not performed, then pressing Marker BW finds the bandwidth around the currently active marker. If the bandwidth markers cannot be placed at the selected value, an error message, such as `“BW not found”` will be displayed.

See “Zooming in on a portion of a response” on page 3-21.

Key Path Markers > More Marker Functions > Marker BW

Related Functions Peak Search

Remote Commands (Marker BW On|Off) CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion :BANDwidth:STATe
CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BANDwidth:INTerpolate
(-0.5 dB|-3 dB|-6 dB|-10 dB|-20 dB)CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4] :FUNCtion:BWIDwidth:NDB
CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BANDwidth:RESult?

Marker Off

Turns the current active marker off.

Key Path Markers >Active Marker > Marker Off

Remote Commands CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:STATe OFF
CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:AOFF

Marker Search Menu

Accesses the following marker peak and pit search functions:

- Select peak or pit search mode
- Perform peak or pit search
- Select next peak down, left or right
- Select next pit up, left or right
- Select active markers

The active markers can also be turned on or off from this menu, which avoids returning to the Marker menu to change markers.

Key Path Markers > More Marker Functions > Marker Search Menu

Marker Search Threshold Value (Marker Setup Panel)

Sets the amplitude threshold value used for limiting the marker search function. Any peaks below the threshold value will be ignored. When the Marker Search Threshold is on, a dotted line is shown on the display at the threshold level. The Use Marker Search Threshold function in the Marker Setup Panel must be enabled or the Marker Search Threshold Value will be greyed out.

Key Path Markers > Marker Setup > Marker Search Threshold Value

Related Functions Pit Excursion, Peak Excursion, Marker Search Threshold

Remote Commands

```
CALCulate:THReshold
CALCulate:THReshold:STATe
```

Marker Setup

Accesses a menu for making changes to default marker settings. From the setup panel you can make changes to marker units, interpolation methods, peak and pit excursion, marker search threshold, and peak search at end of each sweep.

Key Path Markers > Marker Setup

Related Functions
Bandwidth/Marker Interpolation On|Off
BW Marker Units
Normal/Delta marker Interpolation On|Off
Normal Marker Units
Peak Excursion
Pit Excursion
Threshold Value
Use Marker Search Threshold

Marker to Center

Changes the center wavelength to the wavelength of the active marker.

Key Path Markers > Marker to Center

Remote Commands CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:SCENter

Marker to Ref Level

Changes the reference level to the amplitude of the active marker.

Key Path Markers > More Marker Functions > Marker to Ref Level

Remote Commands CALCulate[1|2|3|4|5|6]:MARKer:SRLEVel

Markers

Accesses a menu for selecting and controlling trace markers. The Markers keys can be accessed using the front panel Markers key or the Measure menu Markers selection.

See “General Tips for Using Traces and Markers” on page 3-29, “Moving the Active Marker from One Trace to Another” on page 3-30, and “Measuring the Delta between Two Traces” on page 3-30.

More Marker Functions

Accesses the following marker search functions:

- Delta Marker
- Line Marker menu
- Marker BW
- Marker Search menu
- Marker to Ref Level
- Notch Marker

Key Path Markers > More Marker Functions

More System Functions

Accesses the following functions:

- display setup
- remote setup
- service menu
- set time/date
- state info

Key Path System > More System Functions

Move Active Area

Moves the active function area to one of eight on-screen locations. To change the active function area location, press the Active Function Area softkey until you are satisfied with the position.

Key Path System > Move Active Area

New GPIB Address (Remote Setup Panel)

Allows you to enter a new GPIB address. To change the address, use the numeric entry keys or knob. Press the Defaults softkey to reset the address to the factory preset default.

Key Path System > More System Functions > Remote Setup

Related Functions Current GPIB Address

Next Peak Down ↓

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

Key Path Markers > More Marker Functions > Marker Search Menu > Next Peak ↓

Remote Commands :CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MAXimum:NEXT

Next Peak Left ←

Places the marker on the next peak located at a lower X-axis value (usually wavelength) than the current marker position. This peak must meet the peak excursion and threshold criteria. If the specified marker is off, it will be turned on and placed at the center wavelength or frequency.

Key Path Markers > More Marker Functions > Marker Search Menu >Next Peak ←

Remote Commands :CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MAXimum:LEFT

Next Peak Right →

Places the marker on the next peak located at a higher X-axis value (usually wavelength) than the current marker position. This peak must meet the peak excursion and threshold criteria. If the specified marker is off, it will be turned on and placed at the center wavelength or frequency.

Key Path Markers > More Marker Functions > Marker Search Menu > Next Peak →

Remote Commands :CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MAXimum:RIGHt

Next Pit Left ←

Places the marker on the next pit located at a lower X-axis value (usually wavelength) than the current marker position. This pit must meet the pit excursion and threshold criteria. If the specified marker is off, it will be turned on and placed at the center wavelength or frequency.

Key Path Markers > More Marker Functions > Marker Search Menu > Next Pit ←

Remote Commands :CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MINimum:LEFT

Next Pit Right →

Places the marker on the next pit located at a higher X-axis value (usually wavelength) than the current marker position. This pit must meet the pit excursion and threshold criteria. If the specified marker is off, it will be turned on and placed at the center wavelength or frequency.

Key Path Markers > More Marker Functions > Marker Search Menu > Next Pit →

Remote Commands :CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MINimum:RIGHt

Next Pit Up ↑

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

Key Path Markers > More Marker Functions > Marker Search Menu > Next Pit ↑

Remote Commands CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MINimum:NEXT

Normal/Delta Marker Interpolation (Marker Setup Panel)

Turns the normal/delta marker interpolation on or off. When on, the normal/ delta markers will be placed at the exact wavelength or frequency value selected if it is within the trace range. The marker will linearly interpolate between two trace data points. When off, the marker will be placed on the nearest trace data point.

This setting enables interpolation for all four markers when in normal or delta mode.

Key Path Markers > Marker Setup > Normal/Delta Marker Interpolation

Related Functions Active Marker, Delta Marker

Remote Commands CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:INTerpolation

Normal Marker Units (Marker Setup Panel)

Sets the marker X-axis readout for frequency or wavelength when the analyzer is in a non-zero span. The X-axis default units is nm (nanometers). This setting controls only the normal marker X-axis readout and the delta reference readout. This setting controls all four normal markers. The bandwidth and delta offset markers have their own setting.

Key Path Markers > Marker Setup > Normal Marker Units

Related Functions Active Marker

Remote Commands CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:X:READout

Normal Sweep Mode

Normal mode provides excellent amplitude and wavelength accuracy. At a 2 nm span, two sweeps per second can be achieved.

Note The sweep time (ST) that is displayed in the lower portion of the display is the time for the WDCA to sweep over one gain stage. The WDCA may take up to three sweeps in three different gain stages to make the measurement. This depends on the settings for sensitivity, reference level, auto range and also the particular device being measured. The final data trace is a blended composite of each trace taken in the different gain stages. For more information, refer to “Increase in Sweep Time” on page 3-32

Key Path Sweep > Normal Sweep Mode

Related Functions Fast Sweep Mode, Precision Sweep mode

Remote Commands [SENSe]:SWEep:SPEed

Normalize Input 1

Initiates and sets the normalization trace on input1. Normalization is used to observe changes to a displayed response. For example, you might want to normalize your test setup prior to measuring the device. When the device is measured the response will be corrected for any losses due to the test setup. Normalization should be performed after each 8 hours of operation.

Tip: If you accidently start the normalization before connecting a patch cord between the TLS output and receiver input, you can abort by pressing the Preset key. The instrument will take approximately 10 seconds to respond.

Note Normalization data is valid only for the sweep mode used during normalization. For example, if Normal Sweep Mode is the active sweep mode during normalization, the normalized data will be valid only for normal sweep mode.

Refer to “Normalizing the Measurement” on page 3-11.

Key Path Normalize Reference > Normalize Input 1

Related Functions Normalize Input 2, Normalize Setup

Remote Commands INITiate:NORMalize

Normalize Input 2

Initiates and sets the normalization trace on input 2. Normalization is used to observe changes to a displayed response. For example, you might want to normalize your test setup prior to measuring the device. When the device is measured the response will be corrected for any losses due to the test setup. Normalization should be performed after each 8 hours of operation.

Tip: If you accidently start the normalization before connecting a patch cord between the TLS output and receiver input, you can abort by pressing the Preset key. The instrument will take approximately 10 seconds to respond.

Note Normalization data is valid only for the sweep mode used during normalization. For example, if Normal Sweep Mode is the active sweep mode during normalization, the normalized data will be valid only for normal sweep mode.

Refer to “Normalizing the Measurement” on page 3-11.

Key Path Normalize Reference > Normalize Input 2

Related Functions Normalize Input 1, Normalize Setup

Remote Commands INITiate:NORMalize

Normalize Reference

Accesses the following trace normalization functions:

- Normalize Setup
- Normalize Input 1
- Normalize Input 2
- Apply Input 1 Normalization
- Apply Input 2 Normalization

Normalize Setup

Accesses the Normalization Receiver Input Setup panel used to specify the normalized wavelength range.

Key Path Normalize Reference > Normalize Setup

Related Functions Normalize Input 1, Normalize Setup

Remote Commands Start wavelength: SENSe:NORMalize:WAVelength:STARt
Stop wavelength: SENSe:NORMalize:WAVelength:STOP

Normalize Start Wavelength (Normalize Setup Panel)

Sets the starting point of the normalized wavelength measurement range. The normalization takes multiple sweeps on linked together traces to make a normalization trace.

Tip: Make sure that you choose a normalized wavelength range that is equal to or greater than the measurement wavelength range.

Key Path Normalize Reference > Normalize Setup > Start Wavelength

Related Functions Normalize Input 1, Normalize Setup

Remote Commands SENSe:NORMalize:WAVelength:STARt

Normalize Stop Wavelength (Normalize Setup Panel)

Sets the stopping point of the normalized wavelength measurement range. The normalization takes multiple sweeps on linked together traces to make a normalization trace.

Tip: Make sure that you choose a normalized wavelength range that is equal to or greater than the measurement wavelength range.

Key Path Normalize Reference > Normalize Setup > Stop Wavelength

Related Functions Normalize Input 1, Normalize Setup

Remote Commands SENSe:NORMalize:WAVelength:STOP

Notch Marker On/Off

Measures the bandwidth of a notch filter. To enter the bandwidth amplitude for the notch marker, make a selection from the softkeys (–0.5, –3 dB, –6 dB, or –10 dB) or enter the desired value using the knob, step keys, or numeric keys. The notch bandwidth measurement will be displayed at the top of the screen.

When the notch marker is turned on, the active marker (if not currently on) is turned on and placed at the center wavelength. The reference power or peak power point is found and the selected bandwidth is measured down from the reference (or peak) power.

Note The pit excursion value (Specified in the Marker Setup panel.) determines whether or not a pit in the trace will be considered for the notch bandwidth measurement. In the event that more than one notch is displayed on the active trace, the notch markers will be placed on the notch closest to the active marker. To ensure that you measure the desired notch on the trace, use the knob to place the active marker near the center of the notch.

Key Path Markers > More Marker Functions > Marker BW

Remote Commands

```
CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BANDwidth:TYPE
CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BANDwidth:STATe
CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BANDwidth:RESult?
(-0.5 dB|-3 dB|-6 dB|-10 dB|-20 dB) CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]
:FUNCtion:BANDwidth:NDB
(Notch Marker BW Off) CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion
:BANDwidth:STATe
```

Peak to Center

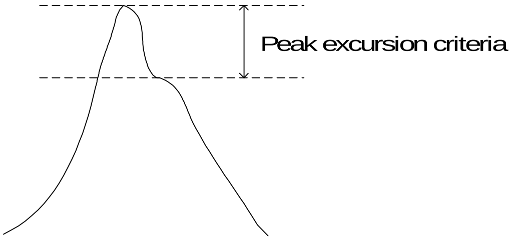
Finds the highest amplitude trace point and sets the center wavelength to that wavelength.

Key Path Wavelength > Peak to Center

Remote Commands CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MAXimum
CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:SCENter
SENSe:WAVelength:CENTer

Peak Excursion (Marker Setup Panel)

Sets the peak excursion value for the marker search functions.



For marker search functions, a signal peak is defined as a rise and fall in the displayed response by at least the peak excursion value. If peak excursion is set too high, legitimate peaks may not be discerned as signals. With lower values more signals will be discerned, but peaks in the noise floor may also be discerned as signals. To reduce the noise floor variance to a value less than the peak excursion rate, try using video averaging and/or lowering the sensitivity.

Key Path Markers > Marker Setup > Peak Excursion

Related Functions Marker Search Threshold, Next Peak Down ↓, Next Peak Right →, Next Peak Left ←

Remote Commands CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:PEXCursion:PEAK

Peak to REF LEVEL

Sets the value of the reference level equal to the value of the highest point on the active trace.

Key Path Amplitude > Peak to REF LEVEL

Related Functions Reference Level

Remote Commands

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

Peak Search

Places a marker on the highest amplitude trace point. If no marker is on, Marker #1 will be used for the peak search.

See “General Tips for Using Traces and Markers” on page 3-29

Key Path

Markers > Peak Search
Markers > More Marker Functions > Marker Search Menu > Search Mode Peak > Peak Search

Remote Commands

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

Peak Search at End of Each Sweep (Marker Setup Panel)

Finds the peak value of the trace and moves the marker to the peak at the end of each sweep. This function operates on normal, delta, and bandwidth markers.

Key Path

Markers > Marker Setup > Peak Search at End of Each Sweep

Perform Calibration

Performs a user wavelength calibration. The calibration process takes approximately 2 minutes for a wavelength calibration and 3.5 minutes for a wavelength calibration and Lambda Zeroing. The wavelength calibration should be performed at least once a shift to ensure accurate measurements. A wavelength calibration will automatically be performed when the operating temperature of the TLS has changed so that a Lambda Zero alignment of the TLS will occur. Refer to “Performing a User Wavelength Calibration” on page 3-17 and “Status Reporting for Laser Wavelength Zeroing” on page 3-13 for additional information.

Key Path System > Wavelength Calibration > Perform Calibration

Remote Commands CALibration:WAVelength:INTernal

Pit to Center

Places the lowest trace point at the center wavelength.

Key Path Wavelength > Pit to Center

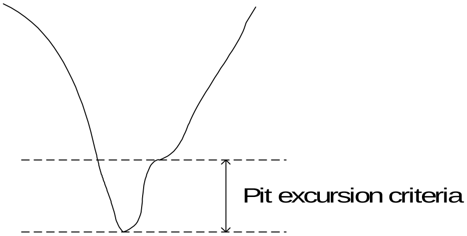
Related Functions Pit Excursion

Remote Commands

```
CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MINimum
CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:SCENter
```

Pit Excursion (Marker Setup Panel)

Sets the pit excursion value for the marker search routines.



The pit excursion value is used to determine whether or not a local minimum in the trace is to be considered a pit.

For marker search functions, a signal pit is defined as having a fall and then a rise in the displayed response of at least the pit excursion value. Reducing the pit excursion to values less than 3 dB may cause the marker-pitting functions to identify noise spikes as pits. To reduce the noise floor variance to a value less than the pit excursion rate, try changing the reference level at the top of the graticule and increasing sensitivity.

Key Path Markers > Marker Setup > Pit Excursion

Related Functions Marker Search Threshold

Remote Commands CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:PEXCursion:PIT

Pit Search

Places a marker on the lowest amplitude trace point which meets the pit excursion criteria. If no marker is on, Marker #1 will be used for the pit search.

Key Path Markers > More Marker Functions > Marker Search Menu > Search Mode Pit > Pit Search

Remote Commands CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MINimum

Precision Sweep Mode

Precision mode offers the most accurate sweep setting to use if you are concerned about wavelength and amplitude accuracy for final filter test applications. However, the increased accuracy also increases the sweep time.

Note The sweep time (ST) that is displayed in the lower portion of the display is the time for the WDCA to sweep over one gain stage. The WDCA may take up to three sweeps in three different gain stages to make the measurement. This depends on the settings for sensitivity, reference level, auto range and also the particular device being measured. The final data trace is a blended composite of each trace taken in the different gain stages. For more information, refer to “Increase in Sweep Time” on page 3-32

Key Path Sweep > Precision Sweep Mode

Related Functions Fast Sweep Mode, Normal Sweep Mode

Remote Commands [SENSe]:SWEep:SPEed

Preset

Restarts and sets the analyzer to a known preset state. Selecting preset aborts any current operations and clears the GPIB output queue. Preset leaves some settings in place, for example, the title on the display.

Table 2-3. Default values

Function	Preset Value	Function	Preset Value
Start wavelength	1545 nm	Stop wavelength	1555 nm
Wavelength offset	0	Wavelength step size	1 nm
Wavelength units	nm	Power reference level position	9 Divisions
Power reference level	0 dBm	Ratio reference level	0 dB
Power scale	10 dB/div	Ratio scale	10 dB/div
Ratio reference level position	9 Divisions	Ratio units[a]	Auto
Amplitude units	Auto	ADC sync out	Off
Sensitivity	Auto	Trace length	1001
Auto range	On	Repetitive sweep	On
Graticule	On	Annotation	On
Current marker	MKR 1	Normal marker interpolation	Off
Bandwidth marker interpolation	On	Peak search threshold	–90 dBm
Threshold enable	Off	Peak search excursion	3 dB
Pit search excursion	3 dB	Peak search on end-of-sweep	Off
Marker 1–4	Off	Line markers	Off
Marker 1–4 bandwidth amplitude	–3 dB	Marker 1–4 bandwidth	Off
Current trace	TRA	Marker 1–4 delta	Off
Trace mean	Off	Marker 1–4 delta marker units	nm
Trace mean limit[a]	Off	Sweep Mode	Normal
Trace data format[a]	ASCII	TRA visibility	On
Trace average count	100	TRA update	On
TRB–TRF update	Off	TRA–TRF min hold	Off
TRB–TRF visibility	Off	TRA–TRF averaging	Off
TRA–TRF max hold	Off	TRA–TRF data	3–point trace
TRA–TRF trace math	Off		

[a] Accessible only via the remote interface.

Key Path Preset

Related Functions Factory Preset (IP)

Remote Commands SYSTem:PRESet

Print

The print function can be accessed by the front panel print key or by using the drop-down File menu print selection. Use this function to print a copy of the display. Print location is determined by the Printer Setup panel.

Remote Commands HCOPy:IMMediate

Printer Setup

Accesses the printer setup panel to select either the internal printer or external printer as the print destination.

Key Path System > Printer Setup

Remote Commands HCOPy:DESTination

Recall (Recall Setup Panel)

Selects whether all measurement data or only trace data will be recalled.

When Measurement (All Visible + State) is selected, all instrument conditions will be recalled. When the file is recalled, the instrument state will be set to the same state as when the file was saved.

When Trace(s) Only is selected, the trace data is displayed under the current instrument conditions.

Note To insure accurate measurements, a wavelength calibration should be performed each time measurement or trace data is recalled from memory.

Key Path Save/Recall > Recall Menu > Recall

Remote Commands (Measurement) *RCL
(Trace Only):MMEMory:LOAD:TRACe

Recall From (Recall Setup Panel)

Selects whether to recall data from a floppy disk or from internal memory.

Key Path Save/Recall > Recall Menu > Recall From

Remote Commands *RCL FLOPpy | INTernal

Recall Menu

Accesses the Recall Setup panel for the following settings:

- Recall mode (All measurement data or trace data only)
- Whether to recall from a floppy disk or from the internal drive

Note To insure accurate measurements, a wavelength calibration should be performed each time measurement or trace data is recalled from memory.

Key Path Save/Recall > Recall Menu

Remote Commands MMEMory:LOAD:TRACe

Reference Level

Specifies the reference level value, the maximum expected power to be measured, at the reference level position. The reference level position is indicated on the display by a dashed green line and the REF annotation on the display. The default position is one major graticule division from the top of the display. The reference level is entered in dB. Use the Amplitude Setup panel to change the reference level position.

Key Path Amplitude > Reference Level

Related Functions Amplitude Setup (Reference Level Position), Scale/Div

Remote Commands DISPlay:WINDow:TRACe:Y:SCALe:RLEVel

Reference Level Position (Amplitude Setup Panel)

Determines the position of the reference level on the graticule. Setting this value to zero divisions places the reference level on the very bottom of the grid. Setting the reference level to 10 divisions places the reference level at the top of the grid. The default (or preset) position is 9 divisions, which is one major graticule down from the top of the display.

The reference level position can be changed using the knob, step keys, or numeric keypad.When using the knob or step keys, the value will automatically change. When using the numeric keypad, a terminator must be selected from the softkey panel.

Key Path Amplitude > Amplitude Setup > Reference Level Position

Related Functions Amplitude Units, Amp Display Mode, Reference Level, Scale/Div

Remote Commands DISPlay:WINDow:TRACe:Y:SCALe:RLPOS

Remote Setup

Accesses the remote setup panel. The current GPIB address information is indicated. To change the address, use the numerical entry keys or knob to enter the new GPIB address information. If changes are made, press the Select softkey. Press the Defaults softkey to reset the address to the factory preset default.

Key Path System> More System Functions > Remote Setup

Repeat Sweep

Sweeps the spectrum continuously and updates the measurement data after each sweep. Repeat Sweep ensures evenly timed sweeps for a stable display of the current tuning range. Sweeps will continuously repeat as long as the instrument is in repeat sweep mode. The SWEEP indicator light on the front panel of the analyzer is on when the sweep is in progress. To minimize wear on the laser, repeat sweep will shut off automatically after 7000 sweeps.

Key Path Sweep > Repeat Sweep

Related Functions Single Sweep

Remote Commands INITiate:CONTinuous

Reset Min/Max Hold

Resets trace hold data and returns to the Traces function keys where you can select an active trace, trace input, update, view, and hold functions.

Key Path Traces > Hold (trace) > Reset Min/Max Hold

Remote Commands

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

Restore Internal Memory

Accesses the analyzer Restore Utility. This operation will remove all existing files from internal memory and replace them with files from a floppy disk. These files are created using the Backup Internal Memory function.

Key Path

Save/Recall > Backup/Restore Menu > Restore Internal Memory

Revision

Displays the analyzer’s firmware revision information.

Key Path

System > Help > Revision

Remote Commands

```
*IDN?
```

Save (Save Setup Panel)

Selects whether all measurement data or only trace data will be saved.

When Measurement (All Visible + State) is selected, all instrument conditions will be saved. The measurement data is saved in binary format (*.dat* file). This includes the traces and all measurement conditions. The .dat file format can only be read by the instrument. You will not be able to view this file on your PC. When the file is recalled, the instrument state will be set to the same state as when the file was saved.

The Trace(s) Only files are saved in comma separated variable (*.csv*) format. When the file is recalled, the trace data is displayed under the current instrument conditions.

Measurement data is saved to the filename ST_xxxxx.dat and Trace(s) only is saved to the filename TR_xxxxx.csv. “xxxxx” represents a numbering system which is incremented each time the measurement data is saved. For example, the first time you save Trace(s) only data, the filename TR_00001.csv will be assigned.

Key Path Save/Recall > Save Menu > Save:

Remote Commands (Measurement) *SAV
(Traces Only):MMEMory:STORe:TRACe

Save Graphics (Save Setup Panel)

Saves graphic data in CGM or GIF format.

The CGM (Computer Graphics Metafile) format is a vector graphics format that describes pictures and graphical elements in geometric terms.

The GIF (Graphics Interchange format) is a cross platform graphic standard. GIF formats are commonly used on many different platforms and readable by many different types of software.

This selection is only valid when saving to the floppy drive.

Key Path Save/Recall > Save Menu > Save Graphics

Remote Commands HCOPy:DEVice:LANGuage PLC | CGM

Save Menu

Accesses the Save Setup panel to configure the following settings:

- Save mode (All measurement data or trace data only)
- Which trace(s) to save
- Whether to save graphics
- Whether to save to a floppy disk or to the internal drive
- Whether to create a file name automatically or manually

Key Path Save/Recall > Save Menu

Remote Commands
(Graphics format) HCOPy:DEVice:LANGuage
(Measurement) *SAV
(Trace only) MMEMory:STORe:TRACe

Save/Recall

The Save/Recall menus and setup panels are accessed by the front panel Save/ Recall or the drop-down File menu Save/Recall selection. These functions are used to save and recall measurement results. This menu accesses the following functions:

- Save Menu
- Recall Menu
- Delete Menu
- Backup/Restore Menu
- Fast Measure Save
- Fast Recall Save

Save To (Save Setup Panel)

Selects saving data to a floppy disk or to internal memory.

Key Path Save/Recall > Save Menu > Save To

Remote Commands *SAV FLOPpy | INTernal

Save Traces (Save Setup Panel)

Selects the traces to be saved. You can select individual traces or all the current trace data.

Key Path Save/Recall > Save Menu > Save Traces

Remote Commands MMEMory:STORe:TRACe

Scale/Div

Specifies the dB per division of the vertical (that is, amplitude) scale. If Auto is specified, the measurement function sets the vertical scale based on the dynamic range of the signal. If a particular vertical scale is desired, clear the Auto check box and put the desired dB/div in the dB dialog box. The amplitude scale can be set from 0.01 to 20 dB per division. The Preset value is 10 dB per division. The default selection is Auto.

Key Path

Amplitude > Scale/Div

Related Functions

Display Mode Log Linear, Reference Level

Remote Commands

DISPlay:WINDow:TRACe:Y:SCALe:PDIVision

Search Limit On/Off

When the search limit function is on, all the marker peak/pit searches will apply only to the range specified by the two wavelength line markers.

Key Path

Markers > More Marker Functions > Line Marker Menu > Search Limit On/Off

Remote Commands

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

Search Mode Peak/Pit

Allows you to select between peak and pit search modes. The selections on the Marker Search Menu will change depending on which of the two modes is selected.

Key Path

Markers > More Marker Functions > Marker Search Menu > Search Mode

Remote Commands

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

Sensitivity

Requests the lowest amplitude signal that can be measured relative to the power at “top of screen”. It is defined as the signal that is six times the RMS noise. The minimum setting is –100 dB. An error will be reported for values outside of this range and the sensitivity will round to the nearest valid sensitivity.

Manual allows manual input of sensitivities and enables auto gain ranging. The “top of screen” and the sensitivity setting determines the requested dynamic range. The system will sweep once per gain stage and may require up to three sweeps to achieve the requested dynamic range.

Auto automatically chooses a sensitivity and a single gain range based on “top of screen”. This will result in approximately 40 dB of dynamic range.

Note

The sweep time (ST) that is displayed in the lower portion of the display is the time for the WDCA to sweep over one gain stage. The WDCA may take up to three sweeps in three different gain stages to make the measurement. This depends on the settings for sensitivity, reference level, auto range and also the particular device being measured. The final data trace is a blended composite of each trace taken in the different gain stages. For more information refer to “Optimizing the Sensitivity Setting” on page 3-32 and to “Increase in Sweep Time” on page 3-32.

Key Path

Amplitude > Sensitivity

Related Functions

Amplitude Setup (Auto Ranging), Reference Level

Remote Commands

(Automatic) SENSe: POWer:DC:RANGe:LOWer:AUTO
(Manual) SENSe: POWer:DC:RANGe:LOWer

Set Time/Date

Accesses the time/date setup panel. The selections are current time (24-hour format), current date, and time zone. To make changes:

- use the navigation keys to select the dialog box
- use the numeric entry keys to change the time and date
- use the step keys to change the time zone and month
- press Set Time/Date when the desired changes are completed.

Changing the time, date, or time zone causes the analyzer to reboot with the current information.

Key Path System > More System Functions > Set Time/Date

Set Title

Accesses the Title Setup panel to add a title to the display. Use the softkeys to change or select the items from the setup panel.

Key Path System > Set Title

Remote Commands DISPlay[:WINDow[1]]:TEXT:DATA

Show Critical Errors

Opens a window displaying critical errors. You can print the queue, clear the queue, or page up and down to view the information. If no errors are generated, the function will be grayed out.

Key Path System > Help > Show Critical Errors

Related Functions Show Hardware Errors, Show Notices, Show Warnings, or Revision

Remote Commands SYSTem:ERRor?

Show HW Errors

Opens a window displaying hardware errors. You can print the queue, clear the queue, or page up and down to view the information. If no errors are generated, the function will be grayed out.

Key Path System > Help > Show HW Errors

Related Functions Show Critical Errors, Show Notices, Show Warnings, or Revision

Remote Commands SYSTem:ERRor?

Show Notices

Opens a window displaying notices. You can print the queue, clear the queue, or page up and down to view the information. If no notices are generated, the function will be grayed out.

Key Path System > Help > Show Notices

Related Functions Show Hardware Errors, Show Critical Errors, Show Warnings, or Revision

Remote Commands SYSTem:ERRor?

Show Warnings

Opens a window displaying warnings. You can print the queue, clear the queue, or page up and down to view the information. If no warnings are generated, the function will be grayed out.

Key Path

System > Help > Show Warnings

Related Functions

Show Hardware Errors, Show Critical Errors, Show Notices, or Revision

Remote Commands

SYSTem:ERRor?

Single Sweep

Initiates one sweep of the measurement range. Use this function to update the displayed measurement data.

Trigger conditions must be met in order for a single sweep to occur. The SWEEP indicator light on the front panel is on when the sweep is in progress. The indicator is off when the sweep is complete.

Tip: Single sweep mode is especially useful when programming the instrument and under the following circumstances:

- insuring the trace reflects current measurement range settings
- capturing traces before processing them with math commands
- capturing traces before positioning markers

Key Path

Sweep > Single Sweep
Single Sweep

Related Functions

Repeat Sweep

Remote Commands

INITiate:IMMediate

Span

Defines the wavelength measurement range for viewing the spectrum.

The span is set symmetrically about the center wavelength. The resolution of the wavelength readout decreases with an increase in the span setting. When the span is set to 0, the display's horizontal axis represents time instead of wavelength. A span of 0 (called zero span mode) configures the analyzer as a fixed tuned receiver.

Note Zero span mode requires that sensitivity be in Auto mode.

The wavelength measurement range can also be set using the Start WL and Stop WL functions.

Note If you increase the span around a center wavelength beyond one of the end wavelength limits of the TLS, the center wavelength will change to a value that will allow the span to increase. For example, if the center wavelength is set to 1553 nm and you increase the span to 100 nm, the center wavelength changes to 1532 nm in order to be able to accommodate the 100 nm span.

See "Measuring the response of a filter" on page 3-18.

Key Path Wavelength > Span

Related Functions Start WL, Stop WL, Center WL

Remote Commands SENSe:WAVelength:SPAN

Start WL

Sets the start wavelength. The center wavelength and span are adjusted so that:

$$Start\ =\ Center - \frac{Span}{2}$$

Use the knob, step keys, or numeric keys to enter the desired value.

If the instrument is in zero span, this command sets the center wavelength to the value specified.

The default setting for start wavelength is 1545 nm.

The minimum value for the start wavelength will vary depending on the TLS used in the system.

Option	Band	Start wavelength	Stop wavelength
111	C Band	1480 nm	1575 nm
112	L Band	1520 nm	1620 nm

Key Path Wavelength > Start WL

Related Functions Span, Stop WL, Center WL

Remote Commands SENSe:WAVelength:STARt

State

Displays the state information. The state information includes:

- model #
- serial #
- options
- software revision
- center wavelength
- span
- start wavelength
- stop wavelength
- wavelength step
- wavelength offset

You can print this information using the internal or external printer.

Key Path System > More System Functions > State Info

Remote Commands HCOPy:DESTination
HCOPy:IMMediate

Stop WL

Specifies the stop wavelength. The center wavelength and span are adjusted so that:

$$Stop = Center + \frac{Span}{2}$$

Use the knob, step keys, or numeric keys to enter the desired value.

If the instrument is in zero span, this command sets the center wavelength to the value specified.

The default for stop wavelength is 1555 nm.

The maximum value for the stop wavelength will vary depending on the TLS used in the system.

Option	Band	Start wavelength	Stop wavelength
111	C Band	1480 nm	1575 nm
112	L Band	1520 nm	1620 nm

Key Path Wavelength > Stop WL

Related Functions Span, Start WL, Center WL

Remote Commands SENSe:WAVelength:STOP

Sweep

Accesses a menu for changing sweep settings. These functions can also be accessed from the Measure menu Sweep selection. The function keys include:

- Fast Sweep Mode
- Normal Sweep Mode
- Precision Sweep Mode
- Repeat Sweep
- Single Sweep
- Sweep Points

Sweep Points

Specifies the number of data points taken for a sweep. The more data points the better the trace resolution, but the longer the sweep time. You can select from 3 to 9601 points. Enter the number of data points using the step keys, numeric key pad, or knob. The default is 1001 points.

Key Path

Sweep > Sweep Points

Remote Commands

SENSe:SWEep:POINts

System

Accesses menus for changing, displaying, selecting, and viewing the analyzer's functions. Some of the functions available are:

- shows HW and critical errors, warnings and notices
- sets a title to the display
- displays the firmware revision
- sets printer setup
- moves the active function area
- displays the analyzer state information
- changes the display setup
- changes the time and dates
- selects the power-on state
- performs a factory preset
- upgrades the firmware
- zeros the instrument
- sets the wavelength limit
- sets the TransZ 2-3 lock
- changes the auto measure defaults
- sets the GPIB address
- turns laser on and off

Time/Date (Display Setup Panel)

Turns the date and time on the display on or off. When on, the date and time will appear on the display's lower, right corner of the display and on printouts.

Key Path System > More System Functions > Display Setup

Related Functions SYSTem:TIMe

Title (Display Setup Panel)

Turns the title on or off the display. When the title selection is on, the title will appear on the display's top, center of the graticule, and on printouts.

Key Path System > More System Functions > Display Setup

Remote Commands DISPlay:WINDow:TEXT:DATA

Trace (A,B,C,D,E,F) Input

Associates the active trace to Receiver Input 1 or Receiver Input 2. You can assign up to six traces to one input. For example, you may choose to assign three traces to input 1. Trace A could have View on, but Update turned off. Trace B could have Update on with Max Hold on. Trace C could have both Update and View on.

Key Path Traces > Trace A Input

Trace C Math Off

Turns off Trace C math processing.

Key Path

Traces > More Trace Functions > Trace Math Off > Default Math Trace C > Trace C Math Off

Remote Commands

CALCulate3:MATH:STATe

Trace F Math Off

Turns off Trace F math processing.

Key Path

Traces > More Trace Functions > Trace Math Off > Default Math Trace F > Trace F Math Off

Remote Commands

CALCulate6:MATH:STATe
CALCulate6:MATH:EXPRession:DEFine
TRACe:EXCHange

Trace Math

Accesses a menu to manipulate and compare traces by adding or subtracting trace-amplitude and display-line data. Trace Math characterizes changes due to environmental stimulus and displays the cumulative effect of multiple devices. When using trace math, the traces must be obtained using identical scale and amplitude values. Any difference in reference level, amplitude units-per-division, or amplitude units invalidates any data resulting from trace math.

The following functions can be accessed:

- Default Math Trace C
- Default Math Trace F
- Exchange Menu
- Trace A Offset
- All Math Off

See “Upper and lower limit lines showing pass/fail test” on page 3-39 and See “Using Trace Math to Measure Wavelength Drift” on page 3-41.

Key Path Traces > More Trace Functions > Trace Math

Remote Commands

```
CALCulate[1|2|3|4|5|6]:MATH:STATe
CALCulate[1|2|3|4|5|6]:MATH:EXPRession:DEFine
TRACe:EXCHange
```

Trace OffSet

Offsets the active trace by the user-specified value.

Key Path Traces > More Trace Functions > Trace Math Off > Trace A Offset

Remote Commands

```
CALCulate[1|2|3|4|5|6]:OFFSet
```

Traces

Accesses a menu for selecting and controlling traces. Some of the functions available are:

- Active Trace
- Averaging
- Hold Trace
- Trace Input
- Trace Math
- Update Trace
- View Trace

The analyzer displays up to six traces: A, B, C, D, E, and F. When the analyzer is first turned on, trace A is the active trace.

TransZ 2-3 Lock

Prohibits the analyzer from using a transimpedance gain higher than the 10k ohm stage. The default state for transZ 2-3 lock is off.

Key Path

Systems> More System Functions > Service Menu > Adv Service Functions > More Adv Service Menu > TransZ 2-3 Lock

Update A...F On|OFF

When on, the selected trace is updated after each sweep. When off, the trace data will not be updated.

Key Path

Traces > Update

Related Functions

View A...F

Remote Commands

```
SENSe:SWEep:POINts
TRACe:FEED:CONTrol TRA|TRB|TRC|TRD|TRE|TRF,ALWays
TRACe:FEED:CONTrol TRA|TRB|TRC|TRD|TRE|TRF,NEVer
```

Use Marker Search Threshold On/Off (Marker Setup Panel)

When on, the marker search function ignores peaks below the threshold value. A dotted line is shown on the display at the threshold level.

Key Path

Markers > Marker Setup > Use Marker Search Threshold

Related Functions

Pit Excursion, Peak Excursion, Marker Threshold Value

Remote Commands

```
CALCulate:THReshold:STATe
```

User Wavelength Cal Date

Shows the date and time of the last successful user-performed wavelength calibration.

Key Path

Wavelength > Wavelength Setup > User Wavelength Cal Date

Related Functions

Wavelength Calibration

Remote Commands

None

View (trace)

Allows trace A, B, C, D, E, or F data to be viewed. Viewed traces are not updated as sweeps occur unless the Update function is on. The data is stored in a temporary trace memory location.

Key Path

Traces > View

Related Functions

Active Trace, Update

Remote Commands

DISPlay:WINDow:TRACe TRA|TRB|TRC|TRD|TRE|TRF,ON

Wavelength

Accesses a menu to set:

- center wavelength
- peak to center
- pit to center
- span
- start wavelength
- stop wavelength
- wavelength setup

Wavelength Calibration

Measures the TLS wavelength accuracy against a physical standard. The differences between the gas cell and the TLS are applied in the firmware.

When the message `TLS Settle: xxxm (`located in the lower right of the display) turns from green to either yellow or red, then initiating a User Wavelength Calibration will also perform a Lambda Zero. If a User Wavelength Calibration is initiated when the message is green, then only a Wavelength Calibration will be performed. The Lamda Zero process takes approximately two minutes and the Wavelength Calibration takes approximately 1.5 minutes.

Refer to “Status Reporting for Laser Wavelength Zeroing” on page 3-13 and to “Performing a User Wavelength Calibration” on page 3-17.

Key Path System > Wavelength Calibration

Related Functions Wavelength > Wavelength Setup > Wavelength Calibration

Remote Commands

CALibrate:WAVelength:STATe ON
CALibrate:WAVelength:STATe OFF
CALibrate:WAVelength:INTernal

Wavelength Cal Info

Displays the date of the last factory calibration and the last user wavelength calibration.

Key Path System > Wavelength Calibration > Wavelength Cal Info

Related Functions Wavelength > Wavelength Setup > Wavelength Calibration

Remote Commands CALibrate:WAVelength:DATE?

Wavelength Line Mkr 1/2

Allows you to set the positions of the line markers. Wavelength Line Marker 1 is always to the left of Wavelength Line Marker 2. When either one of the line markers is accessed, the reduced section of the wavelength is highlighted in blue. The line marker can then be set to the desired position using the knob, step keys, or numeric keypad.

Key Path

Markers > More Marker Functions > Line Marker Menu > Wavelength Line Marker 1 | 2

Remote Commands

(Wavelength Line Marker 1):CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]
:SRANge:LOWer
(Wavelength Line Marker 2):CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]
:SRANge:UPPer
(Sweep Limit):SENSe:WAVelength:SRANge:UPPer

Wavelength Offset

Specifies the wavelength offset. This is an offset between the measured wavelength and the displayed wavelength. You can calibrate the TLS with a wave meter and correct for any offset. Entering a value in the wavelength offset has no affect on the trace.

Key Path

Wavelength > Wavelength Setup > Wavelength Offset

Remote Commands

SENSe:WAVelength:OFFSet

Wavelength Setup

Accesses the Wavelength Setup panel:

- Center Wavelength Step Size
- User Wavelength Cal Date
- Wavelength Calibration
- Wavelength Offset
- Wavelength Units

Key Path

Wavelength > Wavelength Setup

Remote Commands

```
CALibrate:WAVelength:STATe ON
CALibrate:WAVelength:STATe OFF
SENSe:WAVelength:OFFSet
SENSe:WAVelength:CENTer:STEP:INCRement
```

Wavelength Step Size

Allows you to set the step size.

Key Path

Wavelength > Wavelength Setup > Wavelength Step Size

Remote Commands

```
SENSe:WAVelength:CENTer:STEP:INCRement
```

Wavelength Units

Sets the display wavelength units to nm, µm, or Ang.

Angtrom (Ang) is a unit of measurement of wavelength of light and other radiation equal to one ten-thousandth of a micron or one hundred-millionth of a centimeter.

Key Path

Wavelength > Wavelength Setup > Wavelength Units

Remote Commands

None

Zero Now

Causes the auto zero function to run immediately. This allows you to manually zero the analyzer when the auto zero function is off. The photodetector is measured one time, and then the resulting correction is applied to all subsequent measurements, for improved amplitude accuracy. Zero now does ADC calibration, sets the offset DAC, and zeros each TransZ amp for all three diodes. Auto zero or retrace zero zeros only the TransZ amp being used. Adjustment for the TransZ amps optical to electrical conversion helps characterize diodes.

In Auto Zero mode, the analyzer compensates for temperature-related current drift between each sweep. Although this “zeroing” increases amplitude accuracy, it also increases the time between sweeps. If amplitude accuracy is not critical to your measurement, turning off zeroing will improve throughput.

The analyzer performs a more complete zeroing when the analyzer is first turned on. You can run this more complete routine anytime by pressing the Zero Now softkey.

Key Path

System > More System Functions > Service Menu > Adv Service Functions > Zero Now

Remote Commands

CALibration:ZERO:AUTO ONCE

3

The Basics

Measurement Techniques

Using the Analyzer

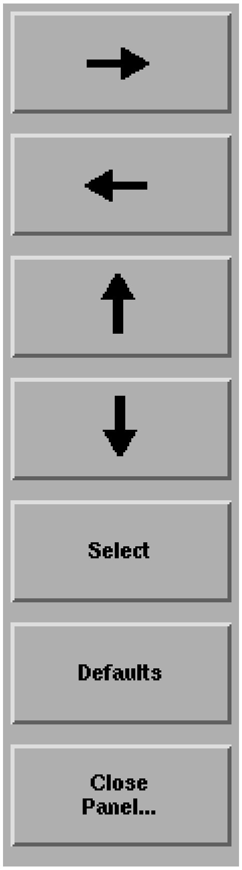
The Basics

The following procedures show you how to setup and use the basic analyzer functions.

To Fill In a Setup Panel

Setup panels allow you to adjust setup conditions which are not frequently changed.

Using the softkeys



Arrows allow you to navigate from field to field in the dialog box. The highlighted parameter can be changed. The front-panel number keys, step keys, and knob allows the user to enter a numeric value in the highlighted field.

Select selects or deselects the highlighted parameter.

Defaults resets the parameters to their default condition.

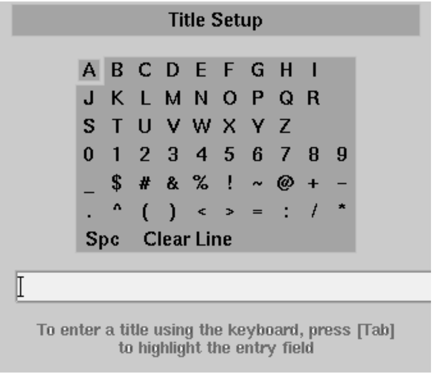
Close Panel saves the current setup and returns you to the previous menu.

Using the Navigation Softkeys on a Setup Panel

1. Use the arrow softkeys to highlight the settings on the setup panel.
2. Use the **Select** softkey to toggle the selection boxes on and off.
3. Enter values in the numeric fields using the front-panel knob or numeric entry pad.
4. To return the setup values to the instrument’s preset settings, press the **Defaults** softkey.
5. When you are satisfied with your selections, press the **Close Panel....** softkey to enter your selections and close the setup panel.

Adding a Title to the Display

1. Press the front-panel **System** key.
2. Press the **Set Title....** softkey. The Title Setup panel appears.



To enter a title using the keyboard, press [Tab] to highlight the entry field

The Title Setup panel

3. Refer to “To Fill In a Setup Panel” on page 3-2 for information on changing and selecting items in the setup panel.
4. To view the title on the display, press **More System Functions** >**Display Setup** and check the **Title On** box.

Backing Up or Restoring the Internal Memory

1 Press the front-panel **Save/Recall** key.

2 Press the **Backup/Restore Menu....** softkey.

Softkey panel selections

Backup Internal Memory
A WDCA Backup Utility screen appears asking you to insert a formatted floppy disk in the external drive. The Backup Internal Memory function overwrites the floppy disk with a new image. Any existing files or catalogs on the floppy disk will be destroyed. The disk will not be viewable on a PC and no trace or measurement files can be saved onto the disk until it is reformatted. Any successive backup operations will overwrite the previous backup information, so only the latest backup information can be recovered through the Restore Internal Memory operation.

Restore Internal Memory
A WDCA Restore Utility screen appears. This operation will remove all files from internal memory and replace them with files from backup floppy disks.

Moving the Active Function Area

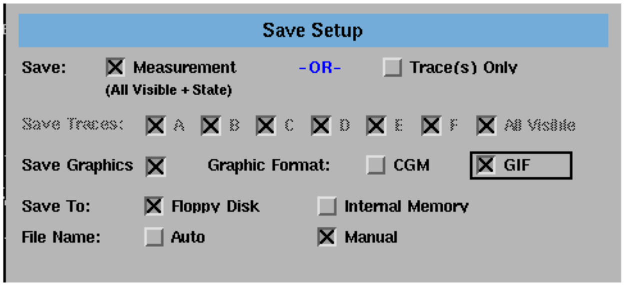
The active function area on the display can be moved to eight different locations. This will allow you to place the active area in a location that will not interfere with the trace information.

1 Press the front-panel **System** key.

2 Press the **Move Active Area** softkey. Each press of the softkey moves the active function area to one of eight onscreen locations.

Saving Measurement and Trace Data

1. Press the front-panel **Save/Recall** key.
2. Press the **Save Menu....** softkey.
3. The Save Setup panel opens. Refer to “To Fill In a Setup Panel” on page 3-2 for information on changing and selecting items in the setup panel.



Save Setup

Save: ☒ Measurement (All Visible + State) -OR- ☐ Trace(s) Only

Save Traces: ☒ A ☒ B ☒ C ☒ D ☒ E ☒ F ☒ All Visible

Save Graphics ☒ Graphic Format: ☐ CGM ☒ GIF

Save To: ☒ Floppy Disk ☐ Internal Memory

File Name: ☐ Auto ☒ Manual

The Save Setup panel

Setup panel selections

Save

Saves the measurement data in a binary format (*.dat* file). This includes the traces and all measurement conditions. The .dat file format can only be read by the WDCA. You will not be able to view this file on your PC. When the file is recalled, the instrument state will be set to the same state as when the file was saved.

The Trace(s) Only files are saved in comma separated variable (*.csv*) format and are auto named starting with TR_00000.csv. State files are auto named starting with ST_00000.csv.

When the file is recalled, the trace data will be displayed *under the current instrument settings*.To view the instrument settings, press System > More System Functions > State Info.

Save Traces

Selects the traces to be saved.

Save Graphics

Allows you to save graphic data in one of two formats:

> **CGM,** Computer Graphics Metafile format, is a vector graphics format that describes pictures and graphical elements in geometric terms. The file is saved with a .cgm extension. This selection is valid only when saving to the floppy drive.
>
> **GIF,** Graphics Interchange Format, is a cross-platform graphic standard. GIF formats are commonly used on many different platforms and readable by many different kinds of software. The file is saved with a .gif extension. GIF supports up to 8-bit color (256 colors).

Save to

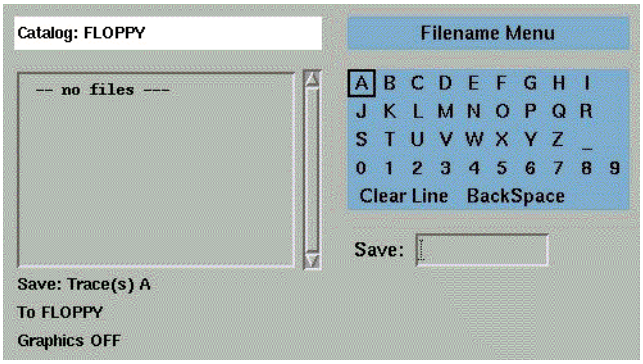
Selects saving data to a floppy disk or to internal memory.

File Name

Selects manual or automatic mode for choosing a file name.

4 If you have chosen Auto to select the file name, press the **Auto Save** softkey. The WDCA will generate a filename and save the file.

5 If you have chosen Manual to select the file name, press the **Choose File to Save** softkey. The Filename Menu setup panel opens.



Catalog: FLOPPY

-- no files ---

Save: Trace(s) A
To FLOPPY
Graphics OFF

Save:

The Filename Menu setup panel

Entering a Filename Using the Arrow Keys

1 Use the front-panel step keys (⇑ and ⇓) and the arrow softkeys (→ and ←) to highlight each letter of the filename.

2 When the desired letter or function is selected, press the **Select** softkey.

3 Select the **BackSpace** function to delete individual letters.

4 Select the **Clear Line** function to delete the entire filename.

5 When you finish entering the filename, press the **SAVE FILE** softkey.

Entering a Filename using an External Keyboard

There must be a PS-2 keyboard installed on the WDCA prior to bootup.

1 Press [TAB] on the keyboard to highlight the entry field.

2 Enter the filename using the keyboard.

3 When you finish entering the filename, press the *SAVE FILE* softkey.

Saving Data in Fast Meas Save Mode

1 Press the front-panel **Save/Recall** key.

2 Press the **Fast Meas SAVE** softkey.

3 The instrument saves the current measurement state to internal memory as FASTSAVE.dat. Only one FASTSAVE.dat file exists, so performing a Fast Meas Save will overwrite any currently existing Fast Save file.

Recalling Measurement and Trace Data

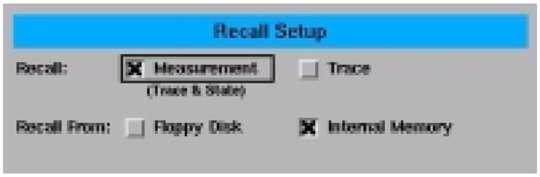
Note

To insure accurate measurements, a wavelength calibration should be performed each time measurement or trace data is recalled from memory.

1 Press the front-panel **Save/Recall** key.

2 Press the **Recall Menu....** softkey.

3 The Recall Setup panel opens.



Recall Setup

Recall: ☒ Measurement (Trace & State) □ Trace

Recall From: □ Floppy Disk ☒ Internal Memory

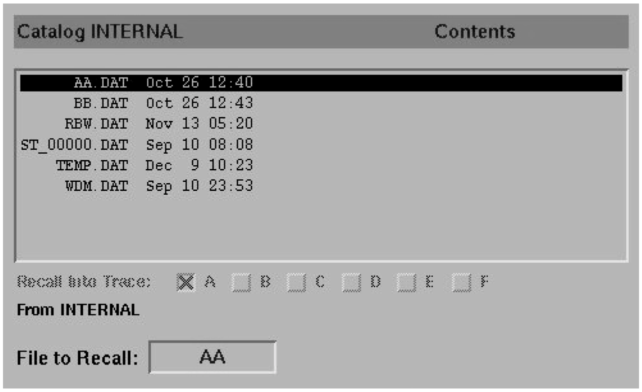
The Recall Menu setup panel

Setup panel selections

Recall
Selects whether a measurement or trace will be recalled.

Recall From
Selects whether to recall from a floppy disk or from internal memory.

4 When you are satisfied with your selections, press the **Choose File to Recall** softkey. The Catalog setup panel opens.



Catalog INTERNAL Contents

```
    AA.DAT  Oct 26 12:40
    BB.DAT  Oct 26 12:43
   RBW.DAT  Nov 13 05:20
ST_00000.DAT  Sep 10 08:08
  TEMP.DAT  Dec  9 10:23
   WDM.DAT  Sep 10 23:53
```

Recall into Trace: ☒ A □ B □ C □ D □ E □ F

From INTERNAL

File to Recall: AA

The Catalog setup panel

5 Use the arrow keys or **Prev File**, **Next File** softkeys to highlight the desired file. Press **RECALL FILE** to load the selected file.

Recalling Data in Fast Meas Recall Mode

1 Press the front-panel **Save/Recall** key.

2 Press the **Fast Meas RECALL** softkey.

3 The instrument recalls the measurement state previously saved as FASTSAVE.dat by the Fast Meas Save function.

Measurement Techniques

The WDCA is a stimulus response instrument that has a source (a TLS) and a receiver that work together to measure passive components.

The TLS provides the input signal for the device under test (DUT) and controls the sweep speed of the instrument. It is the primary contributor for the critical specifications of resolution bandwidth, dynamic range, and wavelength range.

Because the source is the primary contributor to the measurement speed, the sweep speed is not user controlled, rather, sweep time is mostly affected by the wavelength span setting. Other functions that affect the sweep speed are the receiver's sensitivity setting and the selected sweep mode (Fast, Normal, or Precision). The faster the TLS sweeps the less time for amplitude settling. Normal Sweep Mode is a good choice for most measurement applications. Use Precision Sweep Mode when a little extra amplitude accuracy is required. See "Wavelength Specifications" on page 9-9.

To get the best accuracy from the WDCA, you need to normalize the source and receiver with a patch cord (that is, thru cable) to obtain the best performance.

The following procedures will show you how to use the WDCA to make timely, accurate measurements of your DUT.

Normalizing the Measurement

It is very important to normalize your test setup in order to make accurate measurements. Normalization removes amplitude changes caused by your test setup and provides a known reference for comparison. This assures that measurement results reflect only the characteristics of your device. Because TLS power varies with the wavelength span, normalization also improves the amplitude accuracy of the WDCA. This is accomplished by adjusting the TLS power based on the specified normalization start and stop wavelengths. This power will remain unchanged until another normalization is performed over a different wavelength span.

Note

Normalization data is valid only for the sweep mode used during normalization. For example, if Normal Sweep mode is the active sweep mode during normalization, the normalized data will be valid only for Normal Sweep mode.

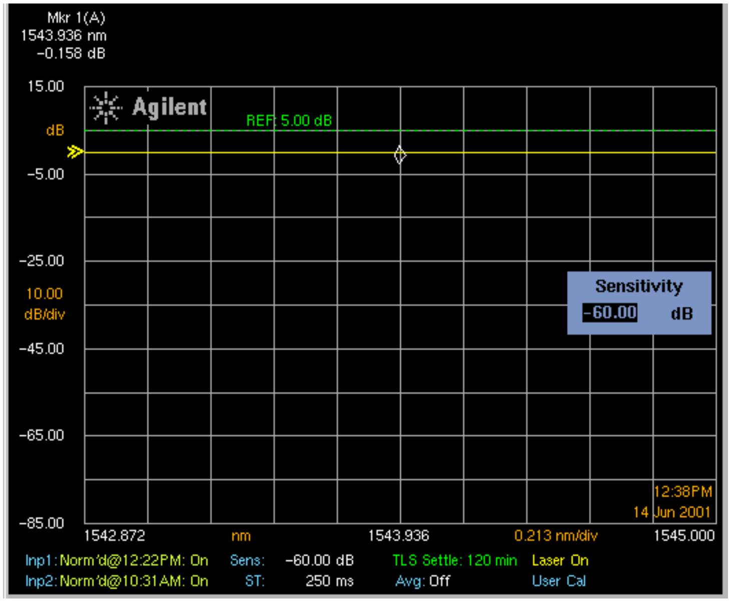
It is recommended that a normalization is performed every 8 hours.

1. Connect a fiber between the Source Output 1 and Receiver Input 1.
2. From the WDCA front panel, press **Norm Ref** to access the Normalization menu.
3. Press ***Normalize Setup*** and make any changes applicable for your device.

   From the setup panel, you can choose the start and stop normalized frequencies as applicable for your device.
4. Press ***Close Panel*** and then press ***Normalize Input 1***.

   The message, “`Normalizing receiver input 1, please wait`” will be displayed at the bottom of the screen. The normalization will take a few moments. This time varies depending on the selected wavelength range.

   Tip: If you accidently start the normalization before connecting a patch cord between the TLS output and receiver input, you can abort by pressing the Preset key. The instrument will take approximately 10 seconds to respond.
5. When the normalization is complete, the software automatically takes a sweep.
6. Verify that the normalized trace is at 0 dB.



Mkr 1(A)
1543.936 nm
-0.158 dB

Sensitivity
-60.00 dB

Inp1:Norm'd@12:22PM: On Sens: -60.00 dB TLS Settle: 120 min Laser On
Inp2:Norm'd@10:31AM: On ST: 250 ms Avg: Off User Cal

Figure 3-1. Normalized trace at 0 dB

Tip: The normalized wavelength range is independent of the displayed measurement wavelength range. This allows you to calibrate a wider wavelength range and then perform a measurement on a subset of the normalized range.

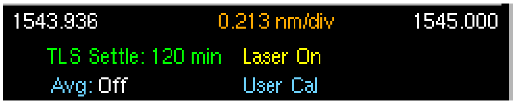
However, if the wavelength range selected extends outside the normalized wavelength range, only the normalized portion of the trace will be accurately displayed.

The normalized trace will compensate for any losses due to the measurement setup. For this setup, it would be just losses in the fiber and connections on either end.

Status Reporting for Laser Wavelength Zeroing

The Wavelength Domain Component Analyzer (WDCA) relies on the 81640 and 81680 tunable laser sources (TLS) to provide high resolution and dynamic range. These lasers have a mechanically tuned cavity that is sensitive to temperature change, requiring the laser to track the internal temperature and occasionally do an internal realignment that takes about 2 minutes. In the language of the laser, this is called Lambda Zero. Lambda Zero is also known as TLS settle. The WDCA uses the temperature inside the laser to predict when the next Lambda Zero or TLS settle will be needed.

A status message, `TLS Settle: xxxm` is located at the bottom of the display. `The` "xxx" refers to the estimated time in minutes before a Lambda Zero is required. Under normal temperature conditions, the status message may show 120 minutes as the estimate before the next Lambda Zero is required (120 minutes is the maximum time that the instrument can predict). This status message is updated every minute. It is possible for the status message to continue to read 120 minutes if the ambient temperature remains stable. However, if the ambient temperature changes rapidly, or if the WDCA receiver is turned on before the TLS has been allowed to properly warm-up (approximately 2 hours), then a Lambda Zero may be required sooner.



```
1543.936          0.213 nm/div          1545.000
      TLS Settle: 120 min   Laser On
      Avg: Off              User Cal
```

Figure 3-2. TLS settle message lower right side of display

Note

It is important that you always have a patchcord connected from Output 2 on the TLS to the Source Input on the WDCA receiver to insure a valid calibration.

A Lambda Zero can be performed in two ways:

- When the message `TLS Settle: xxxm` turns from green to yellow or red, then initiating a User Wavelength Calibration will also perform a Lambda Zero. If a User Wavelength Calibration is initiated when the message is green, then only a Wavelength Calibration will be performed. The Lamda Zero process takes approximately two minutes and the Wavelength Calibration takes approximately 1.5 minutes.
- The message `TLS Settle: xxxm` will typically turn from yellow to red 4 minutes prior to a Lambda Zero being executed automatically. However, de-

pending on current ambient temperature conditions, a Lambda Zero may be executed automatically in less than 4 minutes. For this reason, once the TLS Settle: xxxm message turns red, a Lambda Zero should be expected at any time. Once a Lambda Zero has been executed, the front panel and remote interface will be disabled until it has been completed. This takes approximately 3.5 minutes.

Theory of Operation

For the WDCA to detect the need for a Lambda Zero, three status bits, divided between two registers are used:

- PendingLambdaZero, bit 2 in the Operation Status register
- LambdaZeroInProgress, bit 3 in the Operation Status register
- LambdaZeroComplete, in the Operation Questionable register

PendingLambdaZero uses the temperature inside the laser to predict when the next Lambda Zero will be needed. Under normal temperature conditions, the prediction works well. If the ambient temperature is changing rapidly, or if the WDCA receiver is turned on before the TLS has properly warmed up to reach the ambient temperature (approximately 2 hours), the predication is less reliable.

LambdaZeroInProgress is set when the Lambda Zero begins, and will remain high until the Lambda Zero is completed. You can configure the WDCA so that a positive transition of this bit is latched and summarized in the Status Byte bit 7. This can then be read with a serial poll.

LambdaZeroComplete is the inverse of LambdaZeroInProgress. It is placed in a separate register so that the end of the Lambda Zero can be detected with a serial poll. You can configure the WDCA so that positive transition of this bit is latched and summarized in Status Byte bit 2.

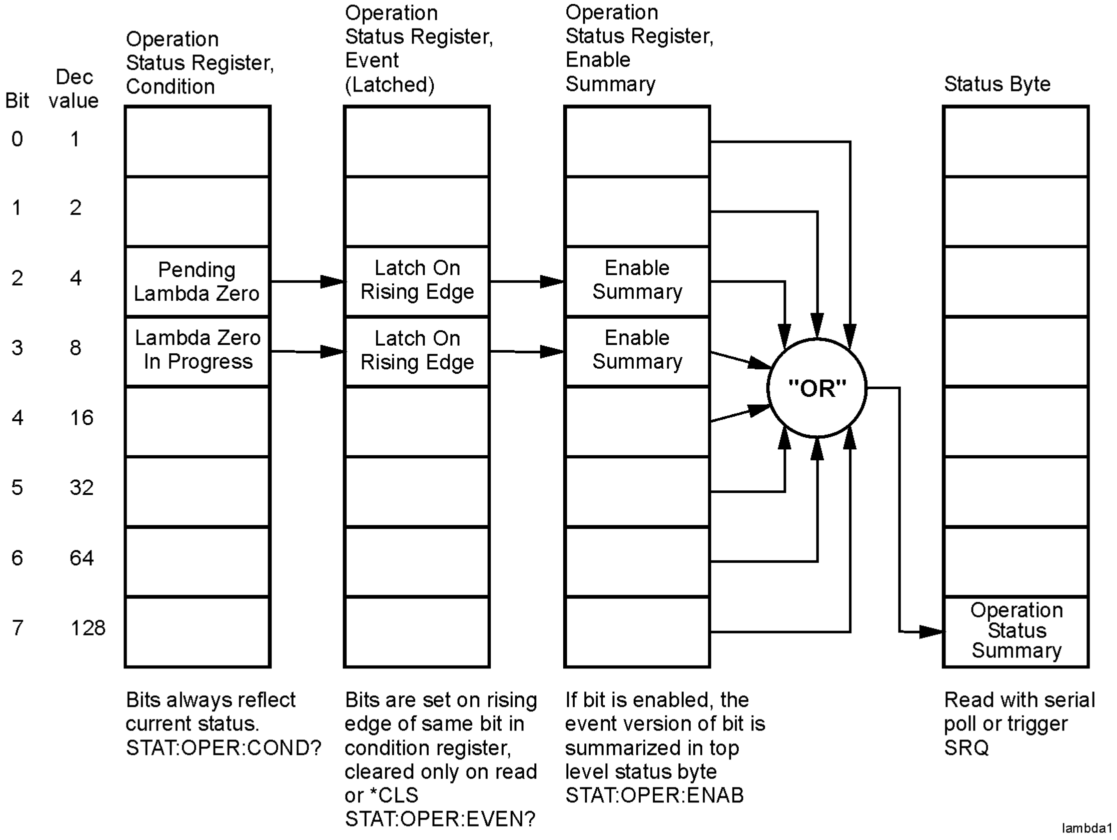

Bit
Dec value
0 1
1 2
2 4
3 8
4 16
5 32
6 64
7 128
Operation Status Register, Condition
Pending Lambda Zero
Lambda Zero In Progress
Operation Status Register, Event (Latched)
Latch On Rising Edge
Latch On Rising Edge
Operation Status Register, Enable Summary
Enable Summary
Enable Summary
"OR"
Status Byte
Operation Status Summary
Bits always reflect current status. STAT:OPER:COND?
Bits are set on rising edge of same bit in condition register, cleared only on read or *CLS STAT:OPER:EVEN?
If bit is enabled, the event version of bit is summarized in top level status byte STAT:OPER:ENAB
Read with serial poll or trigger SRQ
lambda1

Figure 3-3. Lambda Zero Status Bit Registers

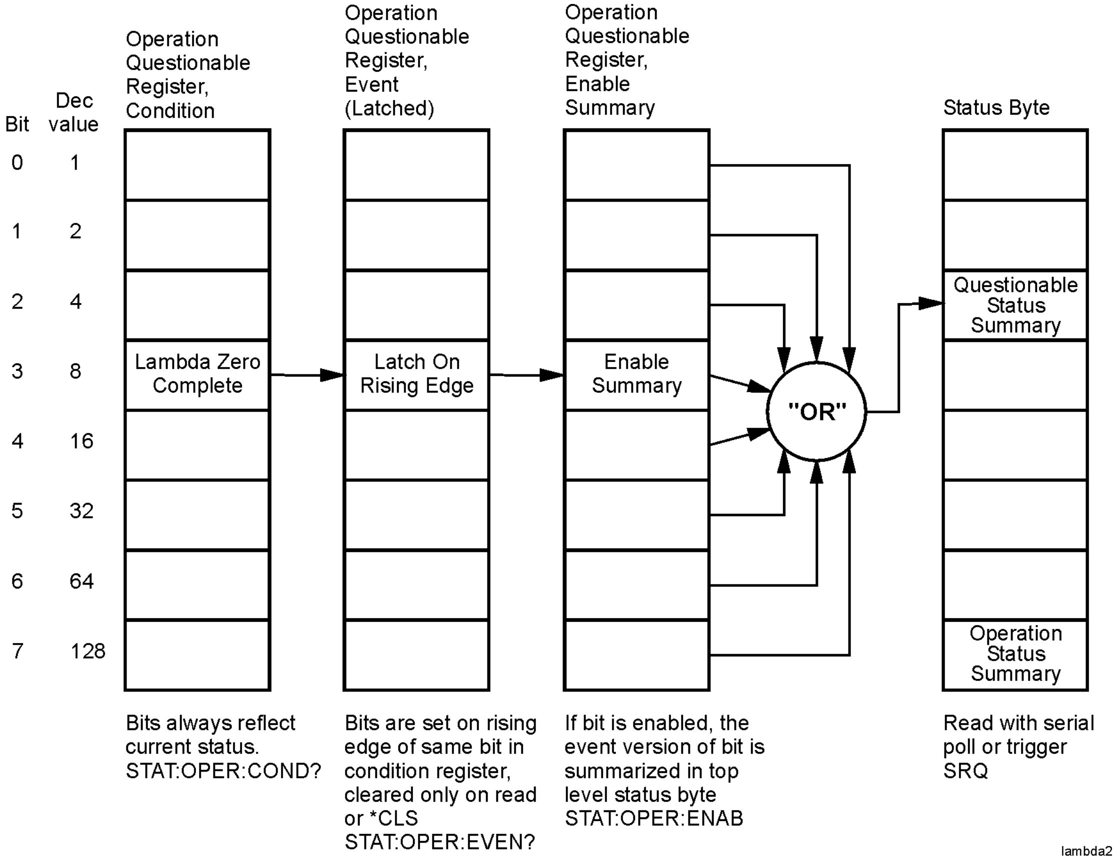

Bit
Dec value
0 1
1 2
2 4
3 8
4 16
5 32
6 64
7 128
Operation Questionable Register, Condition
Lambda Zero Complete
Operation Questionable Register, Event (Latched)
Latch On Rising Edge
Operation Questionable Register, Enable Summary
Enable Summary
"OR"
Status Byte
Questionable Status Summary
Operation Status Summary
Bits always reflect current status. STAT:OPER:COND?
Bits are set on rising edge of same bit in condition register, cleared only on read or *CLS STAT:OPER:EVEN?
If bit is enabled, the event version of bit is summarized in top level status byte STAT:OPER:ENAB
Read with serial poll or trigger SRQ
lambda2

Figure 3-4. Lambda Zero Complete Status Bit Registers

Performing a User Wavelength Calibration

1. From the front panel, press **System** > **Wavelength Calibration** > **Perform Calibration**.
2. Clean and connect a patch cord from the TLS Output 2 to the WDCA Source Input.
3. Press **Execute Calibration**. The calibration process takes approximately 2 minutes for a wavelength calibration and 3.5 minutes for a wavelength calibration and Lambda Zeroing.

   The date of the last calibration will be displayed on the Wavelength Calibration Information panel.

Measuring a Filter Response

This procedure will show you how to display the response of the filter using the normalized trace.For more information, refer to “Normalizing the Measurement” on page 3-11

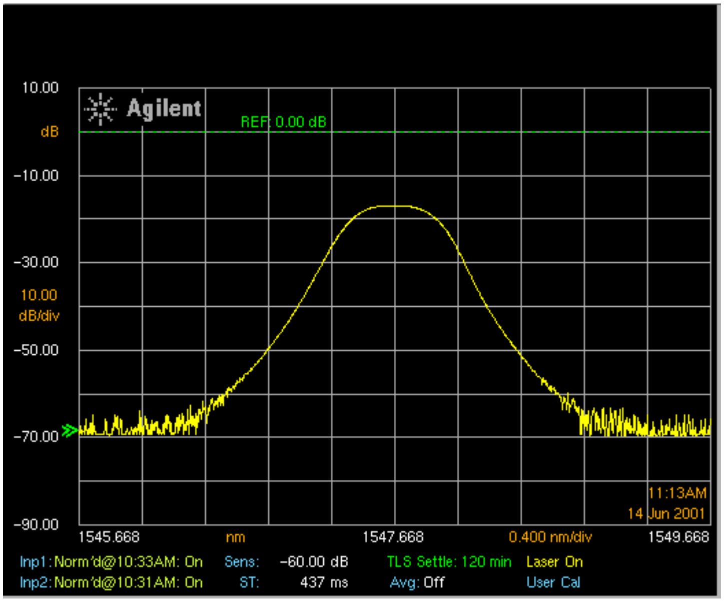
This procedure assumes that you have normalized your test setup.

1 Press ***Wavelength*** and enter the desired range to characterize your device.

2 Disconnect the fiber from Receiver Input 1 and connect it to the input of the filter. Connect another fiber between Receiver Input 1 and the output of the filter.

3 Press ***Single Sweep*** to display the response of the filter.

Because the trace has been normalized, the losses associated with the setup have been compensated out of the measured response. Therefore, you are seeing the response of the filter only.



Inp1: Norm'd@10:33AM: On Sens: -60.00 dB TLS Settle: 120 min Laser On
Inp2: Norm'd@10:31AM: On ST: 437 ms Avg: Off User Cal

Figure 3-5. Measuring the response of a filter

This page left intentionally blank.

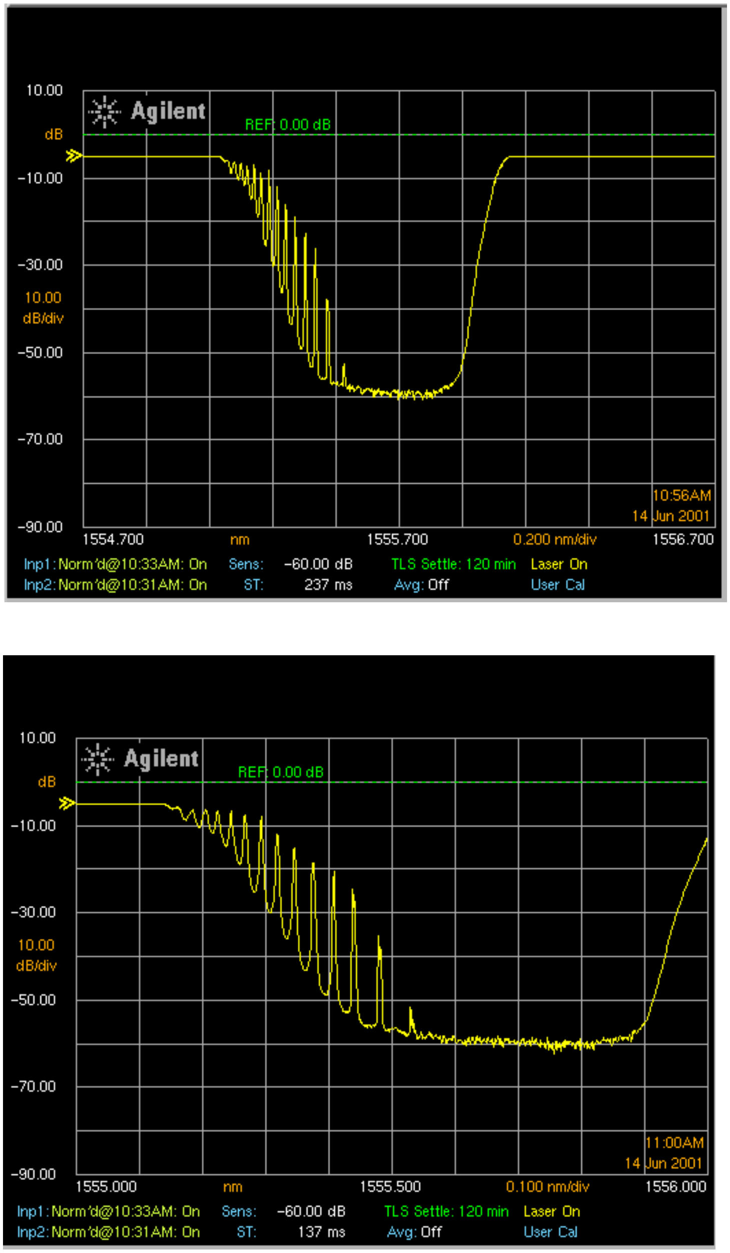
Using Span to Zoom in on a Portion of the Response

To see a more detailed view of the device’s response, decrease the span to expand the trace. This will enable you to precisely focus in on the desired measurement area.

This procedure assumes that you have normalized your test setup and positioned the response on the display. For more information on these processes, refer to “Normalizing the Measurement” on page 3-11 and refer to “Status Reporting for Laser Wavelength Zeroing” on page 3-13.

- Press **Wavelength** > ***Span*** and reduce the span to ***2*** nm.

Reducing the span allows you to zoom in on desired portions of the response.



Inp1: Norm'd@10:33AM: On Sens: −60.00 dB TLS Settle: 120 min Laser On
Inp2: Norm'd@10:31AM: On ST: 237 ms Avg: Off User Cal

Inp1: Norm'd@10:33AM: On Sens: −60.00 dB TLS Settle: 120 min Laser On
Inp2: Norm'd@10:31AM: On ST: 137 ms Avg: Off User Cal

Figure 3-6. Zooming in on a portion of a response

Measuring the Bandwidth of a Filter

The filter's bandwidth may be measured at -3 dB, -6 dB, -10 dB, -20 dB, or set to the value required for your filter's specification.

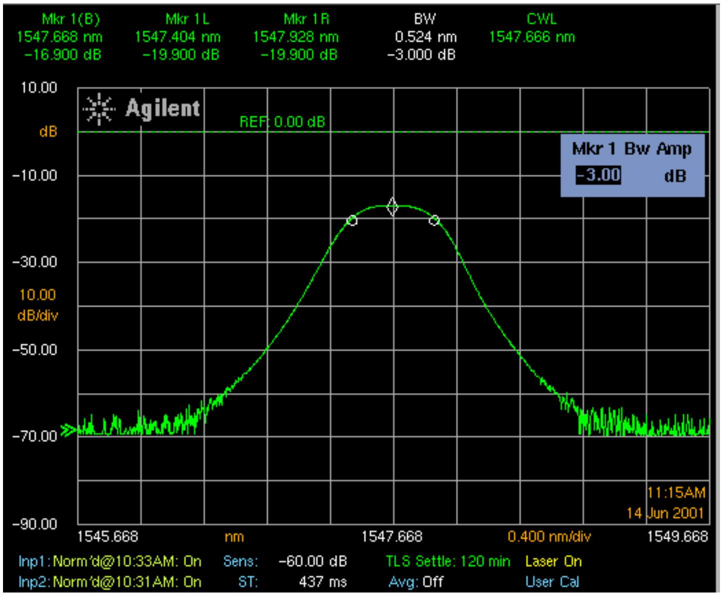
This procedure assumes that you have normalized your test setup and positioned the response on the display. For more information on these processes, refer to "Normalizing the Measurement" on page 3-11 and refer to "Status Reporting for Laser Wavelength Zeroing" on page 3-13.

1 From the front panel, press **Markers** > ***Peak Search*** to find the highest peak in the displayed response.

2 Press **More Marker Functions** > ***Marker BW On*** > ***– 3 dB*** to measure the 3 dB bandwidth point of the peak.

The measurement results are displayed at the top of the screen.

- Mkr 1 shows the peak wavelength and amplitude.
- Mkr 1L displays the rising edge of the 3 dB wavelength and amplitude.
- Mkr 1R displays the falling edge of the 3 dB wavelength and amplitude.
- BW displays the 3 dB wavelength width of the response.
- CWL displays the center wavelength of the 3 dB response. This is computed by adding the rising and falling edges, 3 dB wavelengths and dividing by 2.



Mkr 1(B) 1547.668 nm −16.900 dB

Mkr 1L 1547.404 nm −19.900 dB

Mkr 1R 1547.928 nm −19.900 dB

BW 0.524 nm −3.000 dB

CWL 1547.666 nm

Mkr 1 Bw Amp −3.00 dB

Inp1: Norm'd@10:33AM: On Sens: −60.00 dB TLS Settle: 120 min Laser On

Inp2: Norm'd@10:31AM: On ST: 437 ms Avg: Off User Cal

Figure 3-7. Measuring –3 dB bandwidth of a filter

Measuring Insertion Loss of a Device

Insertion loss is the difference between the power at the input of the device and the power at the output of a device. Since this is a ratio type measurement, insertion loss (or attenuation) is expressed in dB. This procedure assumes that once a normalization has been performed, the normalized trace will be on and a sweep will be taken.

1 Connect a fiber between the Source Output 1 and Receiver Input 1.

2 From the front panel, press **Wavelength** and enter the desired start and stop wavelength values.

Tip: The normalized wavelength range is independent of the displayed measurement wavelength range. This allows you to calibrate a wider wavelength range and then perform a measurement on a subset of the normalized range.

However, if the wavelength range selected overlaps the normalized wavelength range, only the normalized portion of the trace will be displayed.

3 Press **Norm Ref** to access the Normalization menu.

4 Press ***Normalize Setup*** and make changes as applicable for your device.

5 Press ***Close Panel*** and then press ***Normalize Input 1***.

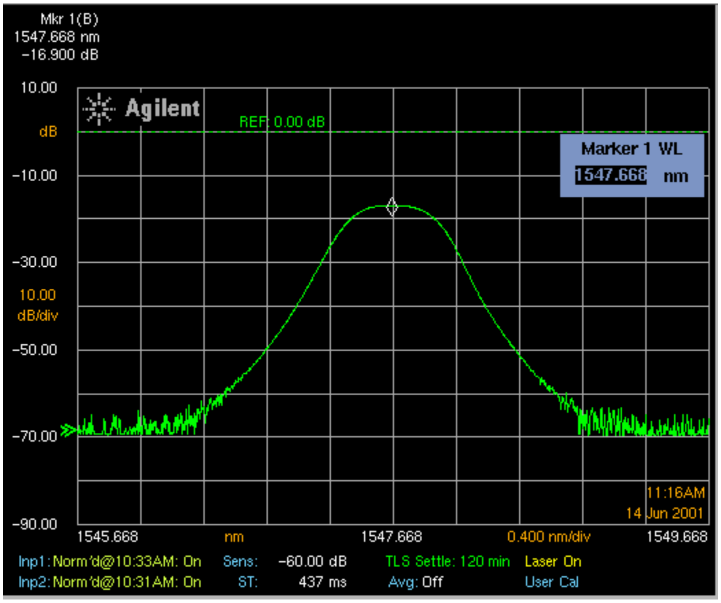
The message, “`Normalizing receiver input 1, please wait`” will be displayed at the bottom of the screen. The normalization will take a few moments. This time varies depending on the selected wavelength range.

When the normalization is finished, notice that the scale on the right side of the display is in dB and the normalized trace is set to a reference of 0 dB.

6 Disconnect the fiber from the Source Output and connect it to the input of the Device Under Test (DUT). Connect another fiber between Receiver Input 1 and output of the DUT.

7 Press Single Sweep to update the trace.

8 From the front panel, press **Markers** > ***Peak Search*** to find the insertion loss of the DUT.



Mkr 1(B)
1547.668 nm
−16.900 dB

Marker 1 WL
1547.668 nm

REF 0.00 dB

Inp1: Norm'd@10:33AM: On   Sens: −60.00 dB   TLS Settle: 120 min   Laser On
Inp2: Norm'd@10:31AM: On   ST: 437 ms   Avg: Off   User Cal

11:16AM
14 Jun 2001

Figure 3-8. Measuring insertion loss

Measuring Insertion Loss and Return Loss Simultaneously

By using a coupler you can display both the insertion loss and return loss of a filter.

The first step is to normalize the return loss path. The DUT will be replaced with an Agilent 81000BR reference reflector. The 81000BR is a gold-plated HMS-10 connector for use in measuring return loss of optical connectors. It allows you to establish a precise reference for reflection measurements. Return loss is specified at 0.18 dB +/-0.1 dB (96% +/-2%).

1 Connect the equipment as shown in Figure 3-9.

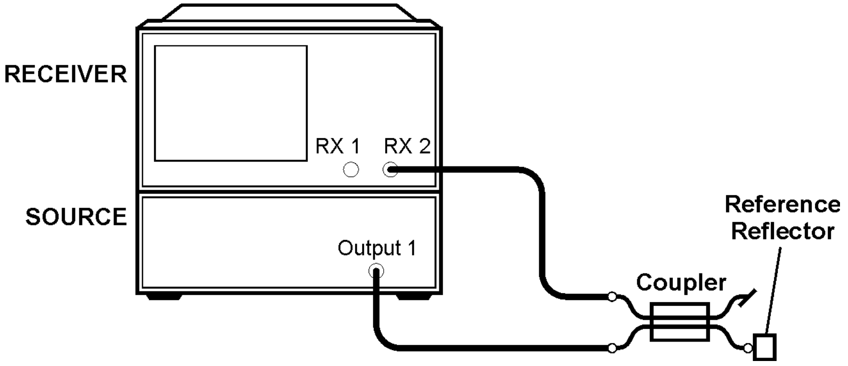


Figure 3-9. Normalize Receiver Input 2 setup

2 Press **Normalize Ref** to access the Normalization menu.

3 Press ***Normalize Setup*** and make changes as applicable for your device.

4 Press ***Normalize Input 2*** to normalize the return loss path for the device.

Connecting the coupler output to the source output is similar to placing a mirror at the coupler input (100% reflections), therefore we have a reflection of a known value.

5 Connect the equipment as in Figure 3-10.

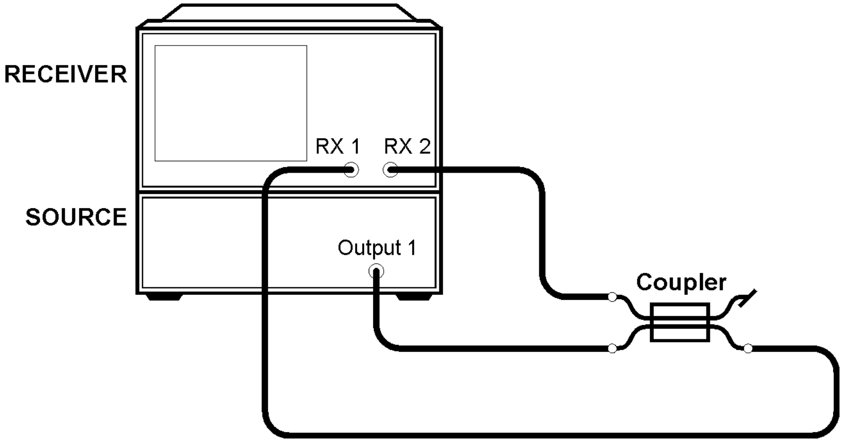


Figure 3-10. Normalize Receiver Input 1 setup

6 Press ***Normalize Input 1***.

7 Connect the equipment as shown in Figure 3-11.

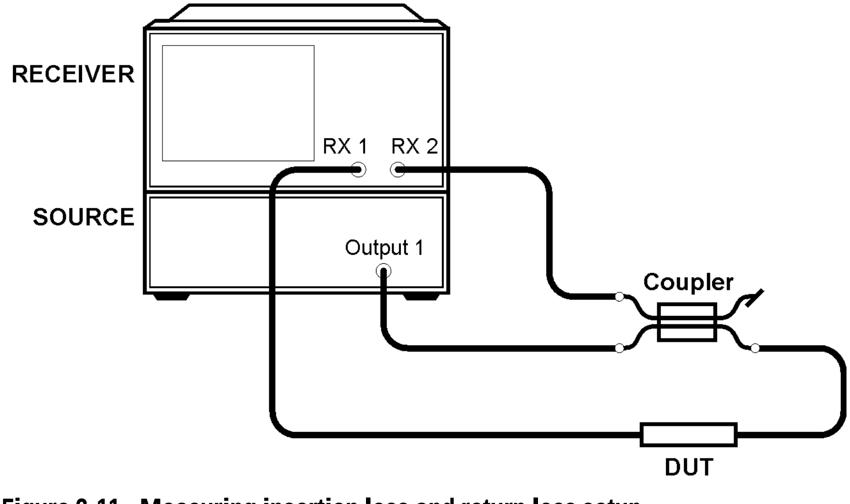


Figure 3-11. Measuring insertion loss and return loss setup

8 Press **Wavelength** and set the ***Center WL*** to the wavelength of the DUT.

9 Press ***Span*** and choose a narrow span. For example, 2 nm.

10 Press **Traces** > ***Active Trace*** > ***TrA*** to turn on trace A (the yellow trace).

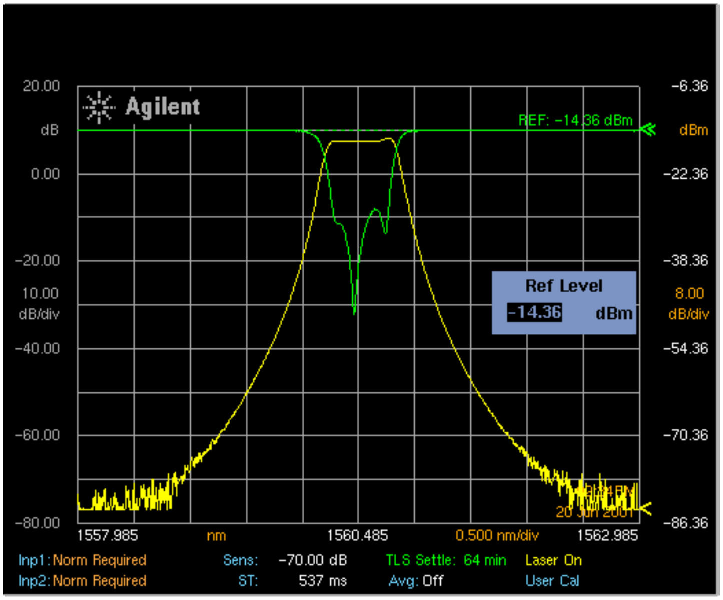
11 Press ***Trace A Input 1*** > **Attach TRA to Input 1** to make trace A the active trace for Receiver Input 1.

12 Press ***Active Trace*** > ***TrB*** to turn on trace B (the green trace).

13 Press ***Trace B Input 2*** > **Attach TRB to Input 2** to make trace B the active trace for Receiver Input 2.

14 Press **Sweep** > ***Single Sweep***.

Input 1 (the yellow trace) shows the insertion loss of the filter. Input 2 (the green trace) shows the return loss of the filter.



Inp1: Norm Required Sens: -70.00 dB TLS Settle: 64 min Laser On
Inp2: Norm Required ST: 537 ms Avg: Off User Cal

Figure 3-12. Measuring return loss and insertion loss simultaneously

Spectrum re-printed courtesy of Hoya Corporation USA

General Tips for Using Traces and Markers

The WDCA provides the ability to display up to six traces with up to four markers. Using multiple markers on multiple traces can sometimes seem complicated. Knowing a few tips makes trace and marker manipulation much easier.

- Markers are always placed on the currently selected active trace. Therefore, use the Active Trace function to activate the desired trace, then select a marker to be placed on that trace.
- When multiple markers are currently used on multiple traces, the Marker Status area (located at the top of the display) makes it easy to identify the state of each marker.

  Information provided for each marker includes:

  - Wavelength
  - Amplitude
  - The trace associated with the marker.

    For example, if marker 1 is on Trace A then the annotation will show Mkr1 (A).

    In addition, if there are markers on two different traces, then the delta of the wavelength and amplitude for the two different markers is also displayed. For example, Mkr (2-1) 0.206 nm, -0.002 dB.

  The color of the annotation denotes different characteristics of the markers:

  - White annotation denotes the status of the currently active marker.
  - Green annotation denotes the status of all currently used markers.
  - Red annotation denotes that some type of an error occurred with the marker measurement.

Moving the Active Marker from One Trace to Another

The following procedure shows you how to move the active marker (marker 1) from Trace A to Trace B.

This procedure assumes that you have normalized your test setup and positioned the response on the display. For more information on these processes, refer to “Normalizing the Measurement” on page 3-11 and refer to “Status Reporting for Laser Wavelength Zeroing” on page 3-13.

1 From the front panel, press **Markers** > **Active Trace** > **TrB** to make Trace B the active trace.

2 Press **Active Marker** > **Mkr 1**.

Measuring the Delta between Two Traces

The following procedure shows you how to find the amplitude and wavelength difference between the maximum peaks of two different traces.

This procedure assumes that you have normalized your test setup and positioned the response on the display. For more information on these processes, refer to “Normalizing the Measurement” on page 3-11 and refer to “Status Reporting for Laser Wavelength Zeroing” on page 3-13.

1 From the front panel, press **Markers** > **Active Trace** and select the first trace to place a marker.

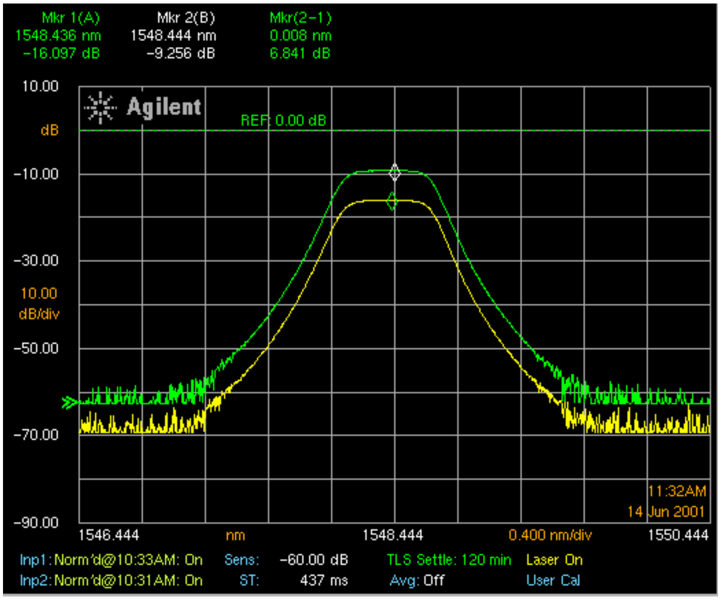
2 Press **Active Marker** > **Mkr 1** > **Peak Search** to place the marker on the highest peak of the active trace.

3 Press **Active Trace** and select the second trace to place a marker.

4 Press **Active Marker** > **Mkr 2** > Peak Search to place the marker on the highest peak of the second trace.

5 View the results of the measurement from the marker annotation at the top of the display.

The wavelength and amplitude of each trace marker is shown, and also the amplitude and wavelength difference of the peaks of the two traces.



Mkr 1(A) 1548.436 nm −16.097 dB

Mkr 2(B) 1548.444 nm −9.256 dB

Mkr(2−1) 0.008 nm 6.841 dB

Inp1: Norm'd@10:33AM: On Sens: −60.00 dB TLS Settle: 120 min Laser On

Inp2: Norm'd@10:31AM: On ST: 437 ms Avg: Off User Cal

Figure 3-13. Wavelength and amplitude difference between two signals

Optimizing the Sensitivity Setting

Sensitivity is the difference (in dB) between the top graticule of the display and the lowest amplitude feature of the trace to be measured. The WDCA will use switch gain stages and digital filtering to reduce the rms noise floor 7.8 dB below the stated sensitivity (a linear factor of 6).

If the range between sensitivity and the top graticule of the display requires more than one gain stage, the WDCA will take one sweep per gain stage, resulting in multiple sweeps per trace update. Each gain stage covers about 40 dB range. While it is possible to stretch a gain range downward with digital filtering, it is generally faster to switch gain stages.

A signal that just touches the top display graticule line is the largest signal to be measured. This is referred to the top of screen. The normalization data is used to convert the top of screen value to dBm, and that value is used to select the gain range. Don’t confuse the reference level with the largest signal to be measured: the reference line is normally one division below top of screen. You can put the reference line at the top of screen by going into **Amplitude** > **Amplitude Setup** and setting the **Reference Level Position** to 10 divisions. This has the advantage that **Amplitude** > **Peak To REF LEVEL** will put the top of the trace in its optimum position.

Like the top of screen, the instrument uses the normalization data to convert sensitivity back to an absolute signal level in dBm. The WDCA has a specified receiver noise floor of –85 dBm, so sensitivities that convert to powers below –85 dBm result in very long sweep times. The instrument will not allow you to enter a sensitivity that converts to a power below –91 dBm.

Increase in Sweep Time

As you increase the sensitivity, you will notice that the sweep time increases:

- If another gain range is required, the instrument will take multiple sweeps for each trace update. You may see the sweep light blink several times. If the trace update bar is visible, it will show several full sweeps before the visible trace updates. In practice, most measurement conditions will be covered by 2 sweeps, with 3 sweeps being the most ever used.
- If the new gain range has high enough gain, the sweep will slow down to accommodate the amplifier.
- Digital filtering will be used if the requested sensitivity is high enough, further slowing the sweep down.

The WDCA sweep rate is limited by the sweep rates available from the laser source. The precision, normal and fast sweep modes use 5, 10 and 40 nm/s sweep rates as a default. If increased sensitivity requires a slower sweep, the instrument will use progressively lower sweep rates. For example, entering a high sensitivity in the fast sweep mode can cause it to use the 5 nm/s sweep rate used by precision sweep mode. As you adjust sensitivity, you might notice that the sweep time has significant discontinuities that occur when the laser sweep rate is changed. Part of good sensitivity selection is observing the discontinuity and staying above it.

Accuracy Unspecified

Only the 5, 10 and 40 nm/s rates have wavelength and amplitude specifications, the instrument turns on the *Acc Unspec* (Accuracy Unspecified) notation when sweep rates below 5 nm/s are used. The Acc Unspec notation also appears when the instrument needs a slower sweep rate than is available from the laser source, and hence is forced to sweep too fast.

If you want to go to higher sensitivity but avoid the Acc Unspec notation, consider going to the Fast Sweep Mode. This mode has less stringent amplitude and wavelength specifications than precision or normal, so it can use more filtering without slowing the sweep down. In addition, because it starts at the 40 nm/s laser source sweep rate, it can reduce the sweep rate down to 5 nm/s before the Acc Unspec notation appears.

For very long sweeps, it may not make sense to sweep twice in order to cover multiple gain ranges. The WDCA allows the user to turn off automatic gain ranging and instead use digital filtering to achieve the desired sensitivity. As with any filter, halving the bandwidth slows the sweep down by half and only improves sensitivity by 3 dB. To turn off the automatic gain ranging, first enter the desired sensitivity using **Amplitude** > **Sensitivity**. Then go into the amplitude setup panel by pressing **Amplitude** > **Amplitude Setup** and selecting **Auto Ranging: Off**.

Note

Sensitivity [Auto] implies **Auto Ranging Off**, and will change the value in the amplitude setup panel. Similarly, **Sensitivity [Man]** implies **Auto Ranging On**, and will change the value in the setup panel when first activated.

This procedure assumes that you have normalized your test setup and positioned the response on the display. For more information on these processes, refer to “Normalizing the Measurement” on page 3-11 and refer to “Status Reporting for Laser Wavelength Zeroing” on page 3-13.

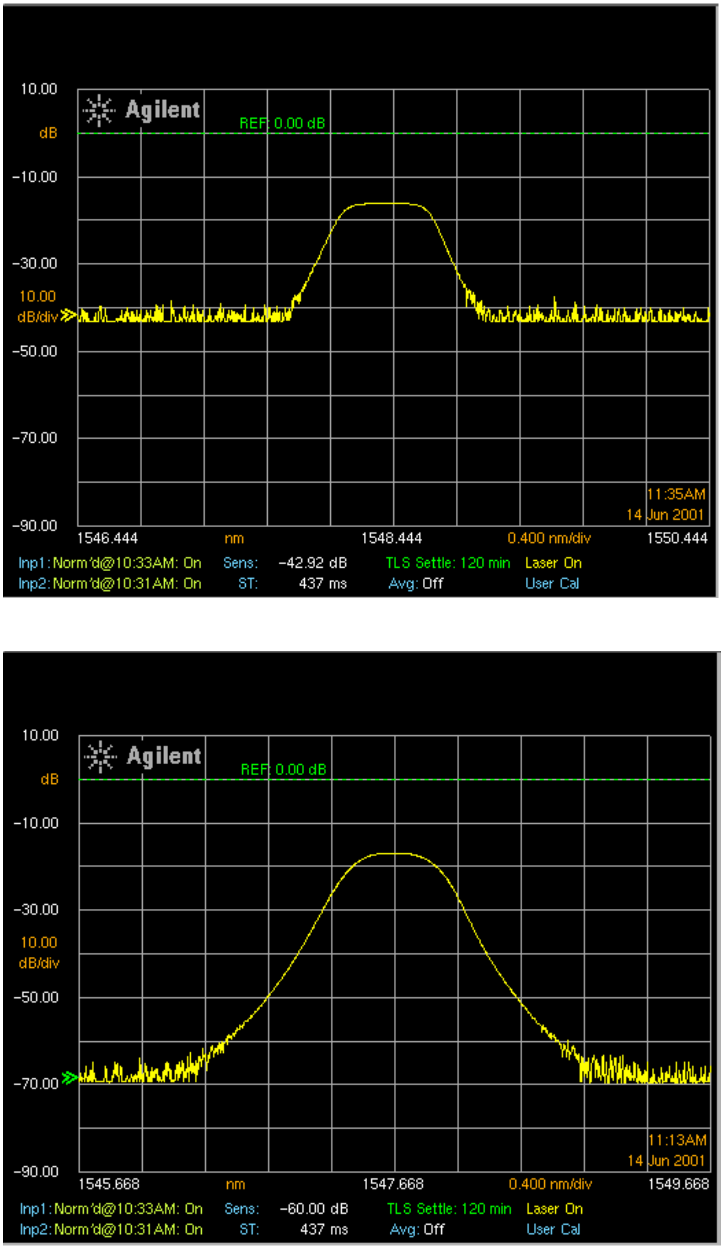
1 From the front panel press **Amplitude** > **Reference Level.** Adjust the reference level so the highest value in the trace just intersects the top graticule (grid line) of the display.

2 Adjust **Scale/Div** so the desired response is visible and covers more than half the display.

3 Press **Sweep** > **Repeat Sweep On**. Make sure that auto-sensitivity is on: **Amplitude** > **Sensitivity** [Auto].

4 Place a marker on the lowest amplitude feature of the trace you would like to characterize using Markers. Note the sweep-to-sweep variation.

5 If the sweep-to-sweep variation is too high for your needs, increase the sensitivity until the sweep-to-sweep variation is acceptable. The sweep will slow down and multiple sweeps per trace update will start to occur as sensitivity is increased.



Inp1: Norm'd@10:33AM: On Sens: −42.92 dB TLS Settle: 120 min Laser On
Inp2: Norm'd@10:31AM: On ST: 437 ms Avg: Off User Cal

Inp1: Norm'd@10:33AM: On Sens: −60.00 dB TLS Settle: 120 min Laser On
Inp2: Norm'd@10:31AM: On ST: 437 ms Avg: Off User Cal

Figure 3-14. Auto sensitivity versus manual sensitivity set to –60 dB

Creating a Limit Line to Establish Pass/Fail Criteria

Limit lines provide the capability to compare a response to a specified limit and determine if the response passes or fails. A limit line disk is provided with the WDCA; however, you may want to create your own custom limit lines to suit your measurement needs.

Below are some guidelines to follow when creating limit lines.

- Each limit line must be identified as either an upper or lower limit line. This is for automatic limit line testing.
- Data points must be entered as wavelength and amplitude pairs.
- A comma “,” must be used as a separator between the wavelength and amplitude data points.
- Wavelength (x-axis values) must be entered in meters.
- Amplitude (y-axis values) must be entered in dB.
- Each data point pair must indicate whether the current data point will be connected to the previous data point.
- Each data point pair must be on a separate line.
- Limit lines are only available for relative (dB) mode not for absolute (dBm) mode.

Tip: For your convenience, instructional text (denoted by number signs “#” surrounding the text) has been added to each of the supplied limit line files. This will make it easy to remember the guidelines required for creating a limit line file.

The following example shows you how to modify an existing limit line to create a upper limit line for a notch filter.

1 Insert the WDCA Limit Line floppy disk into the disk drive of your PC.

2 Launch a text editor such as Word Pad and open one of the limit line files. For this example, we will use L1548`bpf.lim`.

Note Some text editors may not recognize the`.lim` extension. Since the limit line files are text files, you can change the extension to `.txt`.

```
###########################################################
# Limit Line Type [Upper 1 | Lower 0]
###########################################################

TYPE = 0

###########################################################
# Point data
# x-value in meters,
# y-value in dB,
# connect to previous point [Yes 1 | No 0]
###########################################################

POINTS =
1.54810e-6,-20,1
1.54812e-6,-3,1
1.54864e-6,-3,1
1.54866e-6,-20,1
```

Figure 3-15. L1548bpf.lim limit line file

3 Using the keyboard's arrow keys, scroll down to TYPE = 0. Replace "0" with either "1" or "Upper". This indicates that this limit line will be used to test the upper limits of the response.

4 Scroll down to the current data point entries. Click and drag all data entries to select, then press the Delete key.

5 To specify the first data point at a wavelength of 1538 nm and an amplitude of –18.8 db, type 1.538e-6, -18.8, 1.

The "1" indicates that this data point will be connected to the previous data point. For the first data point only, the Connect to Previous Point parameter is only used as a place holder. All other data points require this parameter to be specified.

6 Repeat Step 5 for the remaining data points. Remember to use a comma to separate each data point and a Connect to Previous Point parameter.

Table 3-1. Limit Line Data Points

Wavelength (in meters)	Amplitude (in dB)	Connect to Previous Point
1.5437-6	-18.8	1
1.5437-6	-67	1
1.5445-6	-67	1
1,5445-6	-18	1
1.548	-18	1

7 From the text editors File menu > click Save As. Enter a descriptive file name (maximum of eight characters). For example, unotch.lim.

Tip: When choosing a file name, you may want to indicate the limit line as either upper or lower. Try to limit the file name to eight characters as only eight characters will be displayed in the Limit Line Catalog.

8 Remove the disk from the PC and insert it into the disk drive of the WDCA.

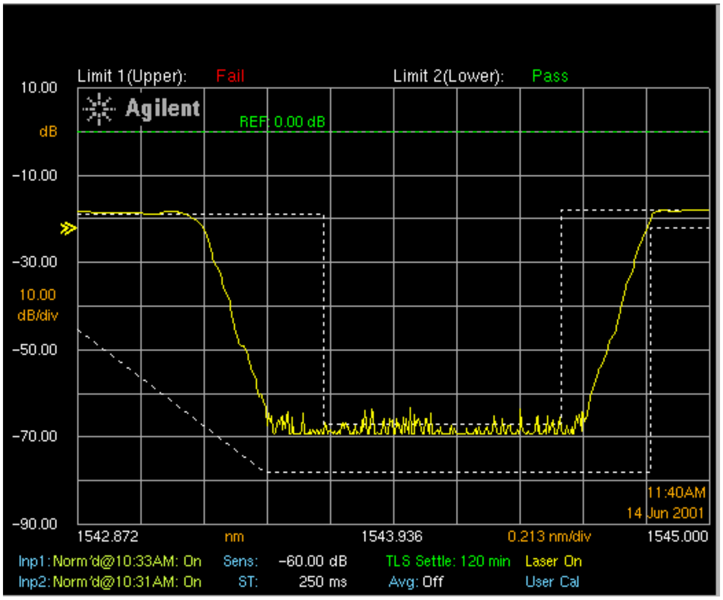
9 Setup the WDCA to display the response of your device. Refer to “Status Reporting for Laser Wavelength Zeroing” on page 3-13 for more information.

10 From the front panel, press **Amplitude** > **Limit Lines** > **Load Limit Lines** > **Load Limit Line 1**.

11 From the Limit Line catalog, select the newly created limit line and then press **Load File**.

12 Press **Previous Menu** > **Limit Line Tests 1**.

The results of the limit line test will be shown at the top of the display.



Limit 1(Upper): Fail Limit 2(Lower): Pass

Inp1: Norm'd@10:33AM: On Sens: -60.00 dB TLS Settle: 120 min Laser On
Inp2: Norm'd@10:31AM: On ST: 250 ms Avg: Off User Cal

Figure 3-16. Upper and lower limit lines showing pass/fail test

Using Trace Math to Measure the Efficiency of the DUT

By adding the insertion loss to the return loss of a DUT you can determine the efficiency of the DUT.

1. Set up your DUT for “Measuring Insertion Loss and Return Loss Simultaneously” on page 3-26.
2. From the front panel, press **Traces** > **Trace Math** > **Default Math Trace C** > **Log Math C = A + B** to add the insertion loss and return loss traces and display the results in Trace C.
3. If the DUT is 100% efficient, then Trace C will be at 0 dB. Any variation from 0 dB will show you the losses of the DUT.

Using Trace Math to Measure Wavelength Drift

This procedure assumes that you have normalized your test setup and positioned the response on the display. For more information on these processes, refer to “Normalizing the Measurement” on page 3-11 and refer to “Status Reporting for Laser Wavelength Zeroing” on page 3-13.

1 Setup your device so that you can view the response on the display.

2 From the front panel, press **Traces** > **Active Trace** > **TrA**.

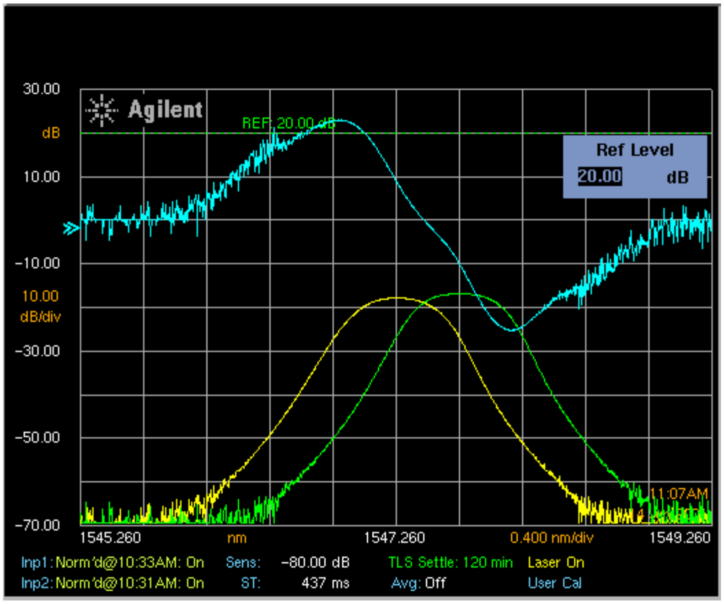
3 Press Single Sweep to update Trace A then press **Update A off**.

4 Press **Active Trace** > **TrB**.

5 Press **Sweep** > **Repeat Sweep On** to continuously update the measured response on Trace B.

6 Press **Traces** > **Trace Math Off** > **Default Math Trace C** > **Log Math C = A – B**.

You can now monitor the wavelength drift of your DUT over time.



Ref Level
20.00 dB

Inp1:Norm'd@10:33AM: On Sens: -80.00 dB TLS Settle: 120 min Laser On
Inp2:Norm'd@10:31AM: On ST: 437 ms Avg: Off User Cal

Figure 3-17. Top trace shows wavelength drift over time

4

Remote Front Panel Operation

Remote Front Panel

The Remote Front Panel capability provides a means to allow the front panel of the analyzer to be operated remotely from a PC with an X Windowing emulator or a UNIX workstation with X Windows.

With the exception for update time, which is limited by the speed of the underlying network, there should be no visible difference between what would be displayed on the analyzer locally and the remote display.

Information and Equipment Required for the Configuration Process

1 Before beginning the analyzer network configuration process, obtain the following required information from your Information Technology (IT) department.

In the analyzer network configuration process, an IP address and the associated hostname need to be assigned to the analyzer and added to the DNS utilized by your PC or UNIX workstation. Gateway and Net Mask information will also need to be specified in order for the analyzer to communicate with the LAN.

The new **Internet Protocol (IP) address** is used to uniquely identify the network address of the analyzer. An IP address is a set of four decimal numbers, separated by periods, like 192.170.128.21.

The **Hostname** associated with the IP address or the analyzer.

Net Mask (Subnetwork Mask)# is used to let the Internet Protocol separate the network ID from the full IP address and thus determine whether the source and destination IP address are on the same network.

Gateway address is used to communicate with devices on a different physical network. The gateway address is the address of a routing device that connects your analyzer’s LAN with other LANs. In some cases, this may be the same address as the IP address of the analyzer client.

Worksheet for your IT department

Internet Protocol (IP) address:_______________________________________

Host Name associated with the IP address:_____________________________

Net Mask (Subnetwork Mask)#:_______________________________________

Gateway address:____

The Network Configuration procedure does not require a local domain name, DNS, or NIS configuration. However, your PC or UNIX workstation will use the DNS to lookup the analyzer location by the host name. Any DNS, local domain name, and NIS information entered will not be used in configuring the analyzer network for Remote Front Panel operation.

2 A keyboard and mouse must be attached to the analyzer to complete the procedure. Restart the analyzer *after* connecting the keyboard and mouse.

3 Before using the remote front panel, the analyzer must be connected to the local area network (LAN). There must be no Proxies between the analyzer and the remote system. Continuous, normal operation of the analyzer is dependent on the speed of the underlying network.

Setting Up the Analyzer for Remote Operation

Note

Attach a keyboard and mouse to the rear panel of the analyzer before starting the following procedure.

1 From the analyzer's front panel, press **System** > **More System Functions** > **Service Menu** > **Configure Network**.

2 When the Network Configuration screen is displayed, press **OK**.

The analyzer will now restart the operating system and load the Network Configure utility.

3 From the Network Configure window, press **Configure**.

During the configuration process, you will be prompted to enter each of the networking parameters as described below. If networking is currently configured, you have the option of accepting the current configuration or updating the networking parameters.

Tip: This is where you can verify the current network settings.

Pressing Unconfig will disable the current network configuration, if any.

4 The HP-UX Set Systems Parameters panel is displayed. You will use this panel to supply the information needed to configure the analyzer on the network. Press **Yes** to continue.

5 Enter the **Host Name** for your analyzer, then press **OK**.

6 You will now be prompted to enter the time zone settings for the analyzer. It is important that the analyzer time and time zone are consistent with the network where it resides.

7 Select the **Time Zone** region in which the analyzer resides, then press **OK**.

8 Select the time and date for the analyzer time zone then press **OK**.

9 A window is displayed showing the current time and date of the time zone selected, press *Yes* if correct, or No to make corrections.

10 In the **System Internet Address** panel, enter the IP address then press **OK**.

11 When prompted to enter additional network parameters, press **Yes**.

12 Enter the **Subnetwork Mask** and **Gateway IP Address** then press **OK**.

13 When prompted for a local domain name and DNS IP address, press **Cancel**. These are not required for this application.

14 When prompted for the NIS Domain Name and NIS Server Name, press **Cancel**. This is not required for this application.

15 When prompted to Restart the analyzer, press **Yes Reboot**. The analyzer will take approximately five minutes to restart.

Connecting to the Analyzer over the Network

You can remap the analyzer front panel over the network using a PC with an X windowing emulator or a UNIX workstation running X Windows. The following X windowing emulators have been tested to ensure correct analyzer remote front panel operation.

- Reflection X revision 8.0.2
- X Win 32 revision 5.0
- UNIX with X Windows

Refer below for specific setup instructions.

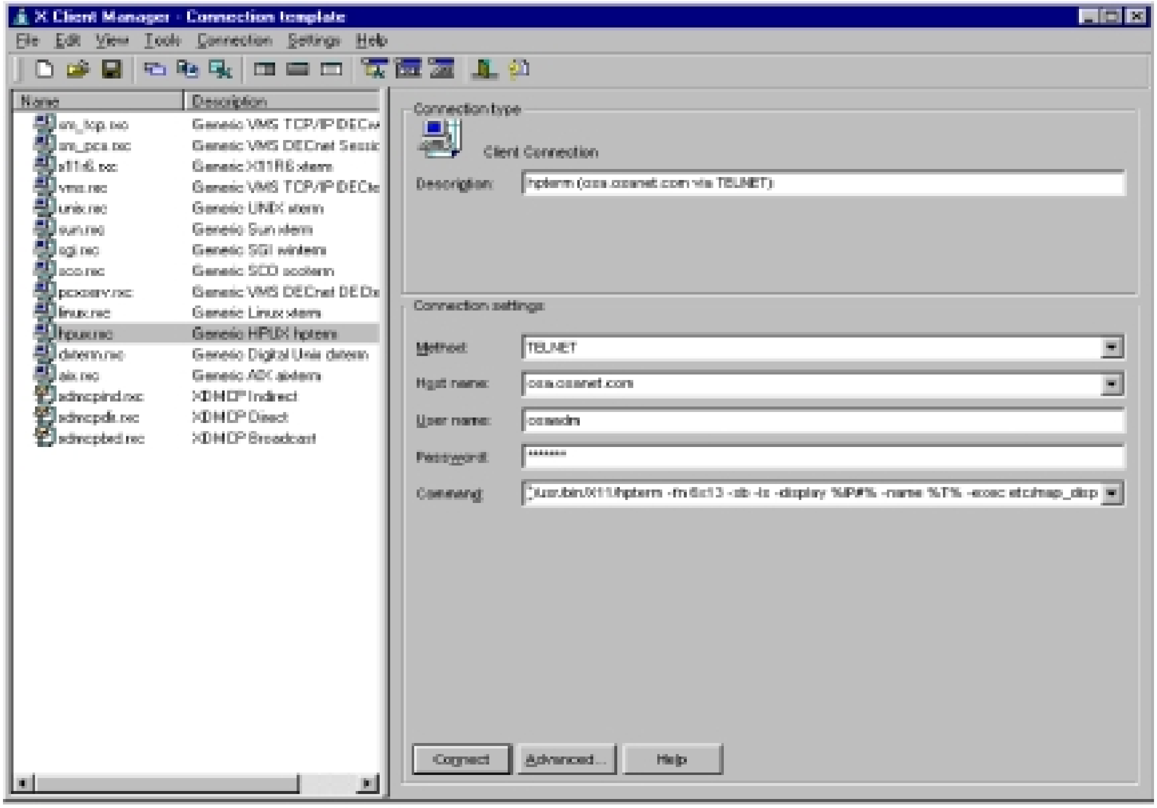
Before using the remote front panel ensure:

- That the analyzer is connected to the local area network (LAN).
- There are no Proxies between the analyzer and the remote system.
- That the underlying network operates fast enough to allow for reasonable operation of the analyzer.
- *If using a PC to run the remote front panel operation*, an X Windowing emulator is available on the PC.

With the exception of the power switch and knob, all front panel analyzer functionality is available when using the remote front panel. You can access the analyzer functionality via the main menu bar and make any data entries using the PC's keyboard.

Using the Reflection X Emulator to Run the Remote Front Panel

Below are the basic steps for setting up the Reflection X emulator to run the analyzer remote front panel. Refer to the Reflection X documentation for further information.



Name	Description
sn_tcp.rxc	Generic VMS TCP/IP DECw
sn_pcs.rxc	Generic VMS DECnet Sessic
x11r6.rxc	Generic X11R6 xterm
vms.rxc	Generic VMS TCP/IP DECte
unix.rxc	Generic UNIX xterm
sun.rxc	Generic Sun xterm
sgi.rxc	Generic SGI winterm
sco.rxc	Generic SCO scoterm
pcxserv.rxc	Generic VMS DECnet DEDx
linux.rxc	Generic Linux xterm
hpux.rxc	Generic HPUX hpterm
dxterm.rxc	Generic Digital Unix dxterm
aix.rxc	Generic AIX aixterm
xdmcpind.rxc	XDMCP Indirect
xdmcpdir.rxc	XDMCP Direct
xdmcpbcd.rxc	XDMCP Broadcast

Connection type

Client Connection

Description: hpterm (cca.ccanet.com via TELNET)

Connection settings

Method: TELNET

Host name: cca.ccanet.com

User name: ccaadm

Password: *******

Command: /usr/bin/X11/hpterm -fn 6x13 -sb -ls -display %IP#% -name %T% -exec etc/map_disp

1 From the PC **Start** menu, locate and run the Reflection X emulator.

2 From the emulator window, enter the following information:

   a In the Client Connection area (left side of the window), select **hpux.rxc**.

   b In the Method field, select **Telnet**.

   c In the Host Name field, select the name assigned to the analyzer.

   d In the User Name field, enter **osaadm.**

   e In the Password field, enter **osaosa1**.

   f In the Command field, you will need to modify the existing command as follows:

(usr/bin/x11/hpterm -fn 6x13 -sb -ls -display %IP#% -name %T% **-exec etc/map_display &**)

   g Click **Connect** to start the remote front panel controller.

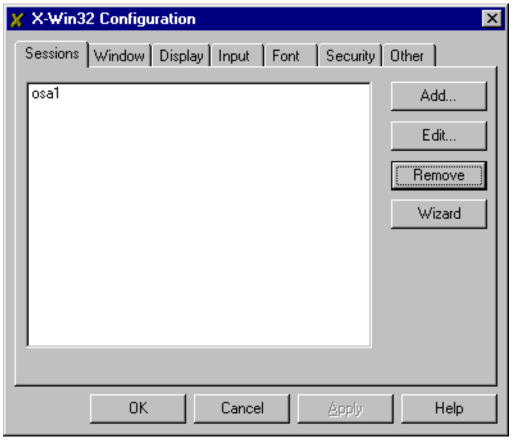
3 Continue to “Using the Remote Front Panel” on page 4-14.

Using the X Win 32 Emulator to Run the Remote Front Panel

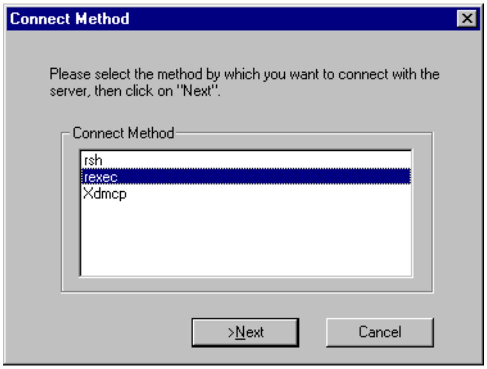
Below are the basic steps for setting up the X Win 32 emulator to run the analyzer remote front panel. Refer to the X Win 32 documentation for further information.

1 From the PC **Start** menu, click **X-Win32** > **X Win Config**.

2 From the X-Win32 Configuration window, click **Add**.

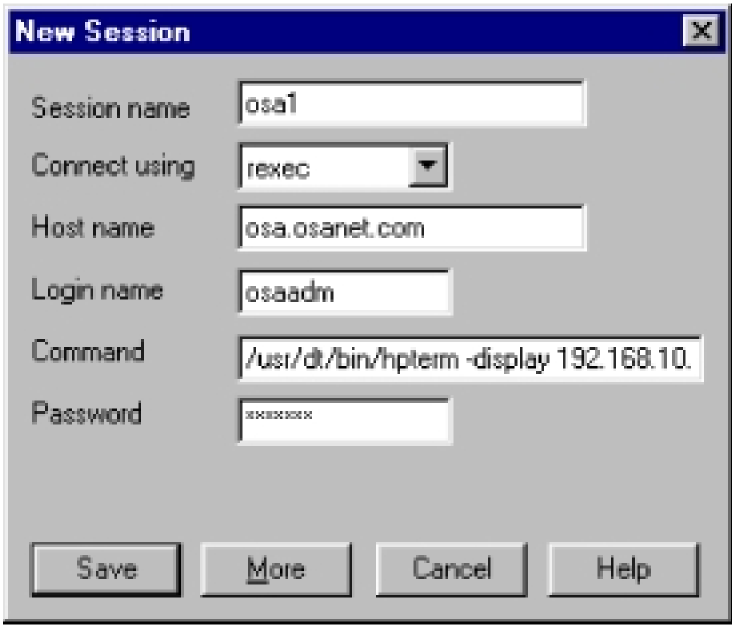

X-Win32 Configuration

3 In the Connect Method window, select **rexec** and then click **Next**.


Connect Method

Please select the method by which you want to connect with the server, then click on "Next".

4 In the New Session enter the following information:


New Session

Session name: osa1

Connect using: rexec

Host name: osa.osanet.com

Login name: osaadm

Command: /usr/dt/bin/hpterm -display 192.168.10.

Password: *******

a In the Session name field, enter a name to uniquely identify the analyzer.

Note

An advantage when using the X Win 32 emulator is that you can run multiple X applications on your desktop in separate windows. For this reason, it is recommended that you name the session after the host system you want to connect to in order to easily identify it when selecting it from a list of sessions.

b In the Host Name field, enter the host name assigned to the analyzer. The host name can either be the name or IP address of the remote host you want to connect to (for example, analyzer.analyzernet.com or 247.109.121.43).

c In the Login name field, enter **osaadm**.

d In the Command field, you will need to enter the command as follows:

/usr/dt/bin/hpterm -display **141.121.12.345**:0.0 -exec etc/map_display

where: **141.121.12.345** is the IP address for the PC that you are using now.

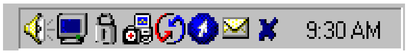
Commands are script files executed on the remote host that open X applications on the PC.

e In the Password field, enter **osaosa1**.

f Click **Save** and then **OK** to save the newly defined session.

5 From the PC, **Start** menu, click **X-Win 32** > **X Win 32**.

You will notice that an X icon will appear in your windows tray -- usually located in the lower right-hand part of your display.



6 Click on the X icon and select the desired analyzer session to run the remote front panel operation.

7 Continue to “Using the Remote Front Panel” on page 4-14.

Using a UNIX Workstation to Run the Remote Front Panel

In order to access the remote front panel from your UNIX workstation, the X server must be set up to allow connection to the analyzer. xhost is the service access control program which allows this access for X Windows.

The xhost program is used to add and delete host names to the list allowed to make connections to the X server. This provides a rudimentary form of privacy control and security.

1 From the command prompt type, **xhost + hostname**. Where hostname is the host name assigned to the analyzer.

2 From the command prompt type, **telnet hostname** and then press **Enter** to telnet to the analyzer. Remember that the hostname is the name that is assigned to the analyzer.

3 When prompted for the login name type, **osaadm** and then press **Enter**.

4 When prompted for the password type, **osaosa1** (or your new password if you have already changed it) and then press **Enter**.

5 At the prompt type, **etc/map_display** and then press **Enter** to start the remote front panel controller.

6 A Welcome screen is displayed and you are given three command choices,

- **Display** accesses a diagnostic tool to show the display parameter setup
- **Enable** accesses the remote front panel. This command is used to stop the analyzer from operating in normal mode and start operating in the Remote Front Panel mode. The annotation, “Remote Front Panel On” will be displayed on the hardware analyzer display.
- **Disable** turns the remote front panel off. The remote front panel will close on the PC display and the analyzer will restart and return and to normal mode.

A fourth command (which is not displayed) is **Passwd**.

- **Passwd** allows you to change the default password to one that is unique for you. It is recommended that you change the default password (osaosa1) at this time. Simply type `Passwd` at the prompt and enter a new password. The new password must differ from the old password by at least three characters.

Note Once you change the password, you will need to remember it for future use. *You will not be able to access the remote front panel capabilities without the new password.*

7 From the command prompt, type **Enable** and then press **Enter**.

8 When prompted, `Enter the IP number for the system`, enter the IP address for the UNIX workstation that you are using now and then press **Enter**.

The remote front panel command will be displayed on the analyzer and the analyzer will be restarted. You will be able to monitor the restarting process on the PC display. Once the analyzer has finished restarting, the front panel will be activated on your PC display.

9 You can now use the remote front panel just like you would if you were sitting in front of the analyzer. Remember to use the main menu bar to access analyzer functions and to enter data via the keyboard number keys.

10 When you are finished using the remote front panel, in the Map Display window, type `Disable`. The analyzer will reboot into normal mode with the remote front panel turned off. The analyzer display will close on the PC.

Using the Remote Front Panel

1 In the Map Display window, a Welcome screen is displayed and you are given three command choices,

- **Display** accesses a diagnostic tool to show the display parameter setup
- **Enable** accesses the remote front panel. This command is used to stop the analyzer from operating in normal mode and start operating in the Remote Front Panel mode. The annotation, “Remote Front Panel On” will be displayed on the hardware analyzer display.
- **Disable** turns the remote front panel off. The remote front panel will close on the PC display and the analyzer will restart and return and to normal mode.

A fourth command (which is not displayed) is **Passwd**.

- **Passwd** allows you to change the default password to one that is unique for you. It is recommended that you change the default password (osaosa1) at this time. Simply type `Passwd` at the prompt and enter a new password. The new password must differ from the old password by at least three characters.

Note Once you change the password, you will need to remember it for future use. *You will not be able to access the remote front panel capabilities without the new password.*

2 From the command prompt, type `Enable`. The remote front panel command will be displayed on the analyzer and the analyzer will be restarted. You will be able to monitor the restarting process on the PC display. Once the analyzer has finished restarting, the front panel will be activated on your PC display.

3 You can now use the remote front panel just like you would if you were sitting in front of the analyzer. Remember to use the main menu bar to access analyzer functions and to enter data via the keyboard number keys.

4 When you are finished using the remote front panel, in the Map Display window, type `Disable`. The analyzer will reboot into normal mode with the remote front panel turned off. The analyzer display will close on the PC.

5

Remote Operation

Remote Operation

This chapter provides information on remote programming of the Agilent 86082A WDCA. The easiest way to program the instrument is by using the VXI*plug&play* universal instrument drivers. The VXI*plug&play* universal instrument drivers allow you to develop programs using the following applications: Agilent VEE, LabVIEW[1], LabWindows/CVI, C, C++, and Microsoft®[2] Visual Basic[3].

If you desire more control of the instrument, you can use the GPIB programming commands that are documented in this chapter. The programming commands are compatible with the Standard Commands for Programmable Instruments (SCPI) standard.

If you've controlled instruments over GPIB but not on an 86082A WDCA, *Summary for Experienced GPIB Programmers* on page 5-5 is the place to start. Finally, once you have a remote application running, you will want to review *Techniques to Improve Throughput* on page 5-28 to improve your measurement throughput.

In this chapter a special font is used for GPIB commands which can be sent to the instrument

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

- sendMessage xxxxx
  means 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
  means to insert a call to a routine that reads an ASCII response from the in-

[1] LabVIEW and LabWindows/CVI are products of National Instruments Corporation.
[2] Microsoft® is a U.S. registered trademark of Microsoft Corp.
[3] Microsoft Visual Basic is a product of Microsoft Corp.

strument 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.

VISA (Virtual Instrument Software Architecture) is an I/O library designed according to the VXI plug and play system alliance.

SICL (Standard Instrument Control Library) is an I/O library developed by Hewlett Packard and Agilent Technologies that is portable across many systems and I/O interfaces.

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*, 1999.

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.

Table 5-1. Front-Panel Features

RMT □	Indicates that the instrument is operating under GPIB control.
SRQ □	Indicates that the instrument has requested service from the computer. Refer to “Monitoring the Instrument” on page 5-20.
Local	Pressing this button activates the front-panel keys after a computer has had control of the instrument.

Summary for Experienced GPIB Programmers

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

Wavelength Calibration

The WDCA wavelength can drift with temperature. The instrument has a built-in gas absorption cell that is used to periodically calibrate the instrument. You can request a calibration using CALibration:WAVelength:INTernal. It is recommended that the WDCA be calibrated whenever the temperature changes ±4° C from the last calibration. Refer to “Performing a User Wavelength Calibration” on page 3-17 for more information.

Lambda Zero

The WDCA employs a tunable laser with mechanical components that must be periodically realigned. The instrument will automatically execute the realignment as needed. This is called Lambda Zero. In addition, the instrument provides a prediction (in minutes) when the next re-alignment will occur. Once a realignment begins, the WDCA will not respond to any GPIB commands with the exception of a serial poll. Refer to “Performing a User Wavelength Calibration” on page 3-17 for a detailed explanation.

It is recommended that:

- Status reporting be configured to report “Lambda Zero in progress” and “Lambda Zero complete”.
- Before writing to or reading from the instrument, you should check for “Lambda Zero in progress”.
- Monitor the “Lambda Zero Pending” status bit and force a wavelength calibration at a convenient time. If “TLS Settle” time is less than 30 minutes, a wavelength calibration will also include Lambda Zero.

Normalization

The WDCA measures insertion loss relative to a normalization done with a through connection. The normalization can be taken over a broad span and interpolated back to smaller spans. Use SENSe:NORMalize:WAVelength STARt and STOP to set normalization span, INITiate:NORMalize to perform normlaization.

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 WDCA sees the EOI, it closes the buffer and begins processing the commands. The WDCA 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 @WDCA key words, and associating the END keyword with the session.

Use device clear

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

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

Use single sweep mode

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

```
INITiate:CONTinuous OFF
```

The instrument defaults to continuous sweep off.

Sweep Mode

The WDCA employs a tunable laser with fixed sweep rates. The faster the laser sweeps, the more uncertain the wavelength is and noisier the amplitude is. The instrument provides three sweep modes: Precision, Normal, and Fast. Use the fastest mode that will meet your accuracy requirements. Refer to “[:SENSe]:SWEep:SPEed PRECision|NORMal|FAST|0|1|2” on page 6-86.

Command synchronization

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

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, but 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 723 or SPOLL @WDCA.

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

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

Maximum query rate

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

Most commonly used commands

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

Table 5-2. Most commonly used GPIB commands

Command	Usage
selected device clear	This is not a text command - it is a GPIB hardware message you should use to start off your application.
*RST	The second thing you should do in your application to return the instrument to a known state and set single sweep mode.
CENT 1550 nm [SENSe:][WAVelength:]CENTer	Program the center wavelength, in nm.
SPAN, STARt, STOP	Other wavelength oriented commands.
POW:RANG:LOW -45 dB [SENSe:]POWer:[DC:]RANGE:LOWer	Tell the instrument the lowest power you expect to see, causing the instrument to adjust its sensitivity. Impacts sweep time.
POW:RANG:AUTO OFF [SENSe:]POWer:[DC:]RANGE:AUTO ON|OFF	Turn off auto-ranging - the automatic changing of gain range during a sweep.
SENS:POW:RANG:LOW -80 dB [SENSe:]POWer:[DC:]RANGE:LOWer	Sets the sensitivity. The sensitivity is the smallest amplitude to be measured relative to the top graticule of the display.
DISP:TRAC:Y:RLEV-10 dBm DISPlay:[WINDow[1]:]TRACe:Y[1,2]:[SCALe:]RLEVel	Set the display reference level - the maximum power expected to display.
INIT INITiate:[IMMediate]	Take a sweep.
INIT:NORM INITiate:NORMalize	Performs a system normalization. Insure that a patch cord is connected between the source output and receiver input.
NORM:WAV:STAR 1520 [SENSe:]NORMalize:WAVelength:STARt	Sets the start of the normalization span.
NORM:WAV:STOP 1520 [SENSe:]NORMalize:WAVelength:STOP	Sets the end of the normalization span.
CAL:WAV:INT CALibration:WAVelength:INTernal	Performs a wavelength calibration using the internal absorption cell. If TLS settling time is less than 30 minutes, a Lambda Zero is also performed.
SENS:SWE:SPE PREC SENSe:SWEep:SPEed	Selects the sweep mode. The trade-off is speed versus accuracy.

Table 5-2. Most commonly used GPIB commands

Command	Usage
SOUR:WAV:CORR:ZERO:TIME? SOURe:WAVelength:CORRection:ZERO:TIME?	Returns estimated minutes until Lambda Zero required. If the TLS settling time is less than 30 minutes, a wavelength calibration will also include a Lambda Zero.
*OPC?	Instrument returns a “1” when all GPIB commands received prior to the *OPC? are complete.
*WAI	Instrument stops processing new GPIB commands until all GPIB commands receiver prior to *WAI are complete.
CALC:MARK1:MAX CALCulate:MARKer[1|2|3|4]:MAXimum	Positions marker 1 to the maximum amplitude point.
CALC:MARK1:X? CALCulate:MARKer[1|2|3|4]:X?	Read back marker 1 wavelength.
CALC:MARK1:Y? CALCulate:MARKer[1|2|3|4]:Y?	Read back marker 1 amplitude.
TRAC? TRA TRACe:[DATA:][Y]? TRA|TRB|TRC|TRD|TRE|TRF	Transfer entire trace, according to format set with the FORM command.
*IDN?	Return string describing model number, serial number and software version.
SYST:ERR?	Read error at top of queue.
DISP OFF DISPlay:[WINDow[1]] ON|OFF	Turn the display off, without affecting the instrument state. Takes about 14 seconds.

Introduction to Controlling an WDCA

One of the easiest ways to learn how to write programs to control the instrument is to look at simple examples. In “Example Programs” on page 5-33, you’ll find several useful example programs. Although they are written using the Microsoft Visual Basic language, you can easily convert them to the language that you are using. The Agilent 86082A’s GPIB address is configured at the factory to a value of 23. You must set the output and input functions of your programming language to send the commands to this address. Pressing the green Preset key does not change the GPIB address.

To change the GPIB address

1 Press the front-panel System key.

2 Press the *More System Functions*.... softkey.

3 Press the *Remote Setup*.... softkey, and change the GPIB address.

Remote mode and front-panel lockout

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

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

Consult the documentation for your programming environment to determine which commands are used to put an instrument in the remote and local lockout modes. These are not Agilent 86082A commands; they are GPIB interface control level commands.

Synchronization Commands

The WDCA will not necessarily execute the commands in the order they are received. That is, it will not necessarily wait until one command completes before starting the next one. Two synchronization commands are provided, *OPC (operation complete) and *WAI (wait), that allow you to tell the instrument when to execute as fast as possible and when to stop and wait for a command to complete. In particular, it is important to wait for a sweep to complete before reading markers or trace data.

**OPC Command*

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

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

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

```
sendCommand inst, ":SENSe:STARt 1520 nm;:SENSe:STOP 1525 nm;:INITiate:IMMediate;
*OPC?"
value = readIntReply(inst)
sendCommand inst, ":CALCulate:MARKer1:MAXimum;:CALCulate:MARKer1:X?"
value = readDoubleReply(inst)
```

where:

- sendCommand is a routine in your favorite language using your favorite interface to send a command to the WDCA.
- inst is a handle to the WDCA appropriate to the interface you are using.
- readIntReply is a routine in your favorite language using your favorite interface to read a reply from the WDCA and convert it to an integer.
- readDoubleReply is a routine in your favorite language using your favorite interface to read a reply from the WDCA and convert it to a double precision real number.

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

Status byte command form

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

An example using serial poll is:

```
’ enable the OPC bit to reflect through to main status byte
sendCommand inst, "*ESE 1"
sendCommand inst, ":SENSe:STARt 1520 nm;:SENSe:STOP 1525 nm;:INITiate:IMMediate;
*OPC"
’ Talk to other instruments, and so on, until you want to check on sweep
value = serialPoll(inst)
’ if bit 5 is set, the sweep is done
    if (value & 32) then
    ‘sweep is done, read status register to clear condition
    sendCommand inst, "*ESR?")
    value = readIntReply(inst)
end if
```

where:

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

***WAI Command**

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

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

```
’ configure ascii binary transfer. FORM BIN,32 is faster, but harder to convert
sendCommand inst, "FORM ASCII"
sendCommand inst,
":SENSe:STARt 1520 nm;:SENSe:STOP 1525 nm;:INITiate:IMMediate;*WAI;TRACe:DATA:
Y? TRA"
Call readAsciiTrace(inst, trace)
```

where:

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

SCPI Syntax Rules

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

```
HCOPy:DATA?
MMEMory:DATA
TRACe:DATA:Y:POWER
TRACe:DATA:Y:RATio
TRACe:DATA:Y?
MEMory:STATe:EXTended?
```

Table 5-3. Syntax Notation Conventions

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

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

SCPI commands are grouped in subsytems

In accordance with IEEE 488.2, the instrument’s commands are grouped into “subsystems.” Commands in each subsystem perform similar tasks. The first page of Chapter 6, “Programming Commands” lists where each subsystem is documented.

Sending a command

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

```
sendMessaage”CALCulate:MARKer1:MAXimum”
```

Use either short or long forms

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

```
sendMessaage":SENSe:WAVelength:STARt?"
```

And this is the short form of the same command:

```
sendMessaage":SENS:WAV:STAR?"
```

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

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

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

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

Table 5-4. Examples of Short Forms

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

You can use upper or lowercase letters

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

Instrument responses, however, are always returned in uppercase.

Combine commands in the same subsystem

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

```
sendMessage":SENSe:WAVelength:STARt 1550NM"
sendMessage":SENSe:WAVelength:STOP 1560NM"
```

can be combined into one line:

```
sendMessage":SENSe:WAVelength:STARt 1550NM;STOP 1560NM"
```

The semicolon separates the two functions.

Combine commands from different subsystems

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

```
sendMessage":SENSe:WAVelength:SPAN:FULL;:CALCulate:MARKer1:MAXimum"
```

Sending common commands

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

```
sendMessage":SENSe:WAVelength:STARt 1550NM;*CLS;STOP 1550NM"
```

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

Adding parameters to a command

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

```
sendMessage":SENSe:WAVelength:STARt 1550NM"
```

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

```
sendMessage":DISPlay:WINDow:TRACe:STATe TRB, ON"
```

White space

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

Numbers

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

Table 5-5. Multipliers

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

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

```
*RST
:CALCulate1:MARKer1:X?
```

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

Program message terminator

The string of instructions sent to the instrument is executed after the instruction terminator is received. The terminator must be the End-Or-Identify (EOI) line asserted, either on the last character or on a new line. 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).

Querying data

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

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

```
sendMessage”:CALCulate:MARKer1:X?”
```

places the wavelength of marker 1 in the output queue.

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

Monitoring the Instrument

Your programs can monitor the Agilent 86082A for its operating status, including querying execution or command errors and determining whether or not measurements have been completed. Several status registers and queues are provided to accomplish these tasks as shown in Figure 5-1 on page 5-21. The status structures shown in the figure consist of condition registers, event registers, event enable registers, and, in the case of the Operation Status Structure, transition filters. For example, there exists the Standard Status *Condition* Register, the Standard Status *Event* Register, and the Standard Status *Event Enable* Register. Condition registers show the *current* condition of the status lines. Event registers show that an event has occurred. Once latched, these registers stay set until cleared. Event enable registers are masks that you can use to enable or disable the reporting of individual bits from an event register. For example, you can disable the reporting of the Sweeping bit in the Operation Status structure so that, even though it goes high in the registers, it can never set the summary bit 7 in the Status Byte high.

Querying a register always returns the value as a decimal-weighted sum of all set bits. Refer to Table 5-6 on page 5-20. For example, if the value returned was 528, this would indicate that bits 4 and 9 were set. Mask registers are set using these same values. For example, the *ESE 60 command sets bits 2 through 5 of the Standard Status Event Enable Register. Whenever any one of bits 2 through 5 of the Standard Status Event Register goes high, bit 5 of the status byte will be set.

Table 5-6. Decimal Values of Event Enable Register Bits

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

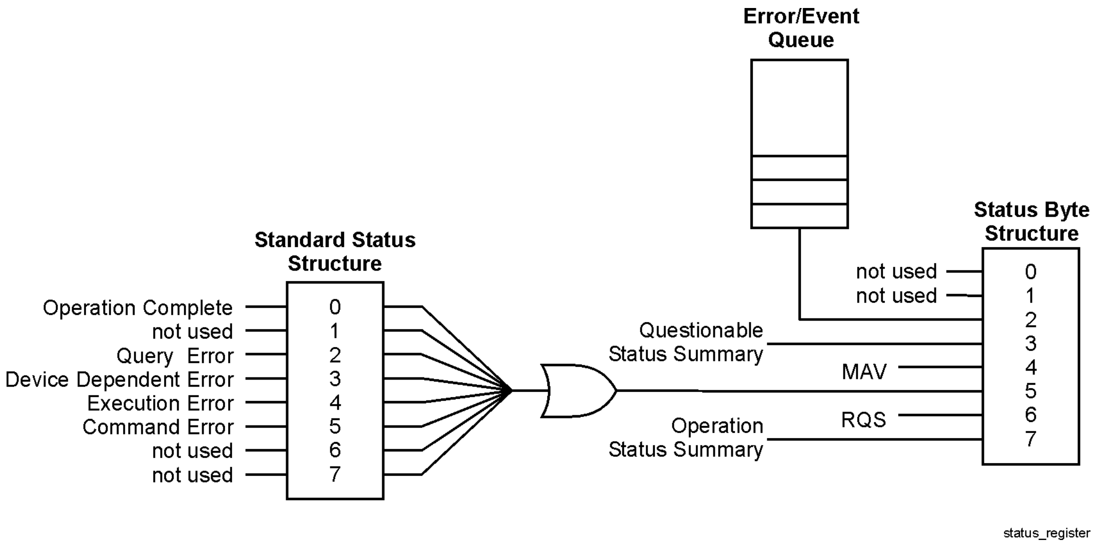


Figure 5-1. Status Registers

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

Status Byte

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

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

*SRE common command	Sets or reads the event enable register value (mask).

Standard Status Structure

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

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

Operation Status Structure

Contains bits that report on the instrument’s normal operation. Additional transition filters give you the ability to select the logic transitions which set the event register bits. For example, you can define the Measuring bit of the Operation Status Structure to be set when a status line transitions from false to true. This is a positive transition. You can also specify a negative transition where bits are set when a status line transitions from true to false.

Command	Use
STAT:OPER:COND?	Returns the value of the condition register.
STAT:OPER:EVEN?	Returns the value of the event register.
STAT:OPER:ENAB	Sets or returns the value of the event enable register (mask).
STAT:OPER:NTR	Sets or returns the value of the negative transition register.
STAT:OPER:PTR	Sets or returns the value of the positive transition register.

Table 5-7. Bits in Operation Status Structure

Bit	Definition
2	Pending Lambda Zero. Indicates a Lambda Zero is required and will be initiated within 3 minutes.
3	Lambda Zero In Progress. Instrument is performing a wavelength calibraton. It will not respond to GPIB commands until the calibration is completed.

See “Status Reporting for Laser Wavelength Zeroing” on page 3-13 for a discussion of how to use these status bits.

Questionable Status Structure

Contains bits that report on several questionable instruments conditions.

Command	Use
STAT:QUES:COND?	Returns the value of the condition register.
STAT:QUES:EVEN?	Returns the value of the event register.
STAT:QUES:ENAB	Sets or returns the value of the event enable register (mask).

See “Status Reporting for Laser Wavelength Zeroing” on page 3-13 for a discussion of how to use these status bits.

Table 5-8. Bits in Questionable Status Structure

Bit	Definition
3	Lambda Zero is complete.

Output Queue

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

Error Queue

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

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

Normalizing the Measurement

It is very important to normalize your test setup in order to make accurate measurements. Normalization removes amplitude changes caused by your test setup and provides a known reference for comparison. This assures that measurement results reflect only the characteristics of your device. Because TLS power varies with the wavelength span, normalization also improves the amplitude accuracy of the WDCA. This is accomplished by adjusting the TLS power based on the specified normalization start and stop wavelengths. This power will remain unchanged until another normalization is performed over a different wavelength span.

Note

Normalization data is valid only for the sweep mode used during normalization. For example, if Normal Sweep Mode is the active sweep mode during normalization, the normalized data will be valid only for normal sweep mode.

It is recommended that you perform a normalization every 8 hours.

Tip: The normalized wavelength range is independent of the displayed measurement wavelength range. This allows you to calibrate a wider wavelength range and then perform a measurement on a subset of the normalized range.

However, if the wavelength range selected extends outside the normalized wavelength range, only the normalized portion of the trace will be displayed accurately.

The full normalization commands

The normalization arrays do not necessarily line up with the current sweep. You are allowed to configure a large normalization span and let the WDCA interpolate to the currently measured span. If the measurement wavelength span extends beyond the normalization span, wavelengths less than the normalization span will use the first wavelength normalization point and wavelengths greater than the normalization span will use the last normalization point.

Since the WDCA stores only one normalization array per channel, you may want to archive the normalization data in the computer and transfer the data back as needed. Only transfer data in that has been acquired by the WDCA. Trace data can be used, but must be taken in 5 pm steps. To calculate the number of points the instrument expects, use the formula:

$$number\ of\ points = \frac{stop\ wavelength - start\ wavelength}{0.005nm} + 1$$

Note The following examples assume that the WDCA is configured for the default trace transfer format (ASCII).

The full normalization commands:

```
:INITiate:NORMalize 0|1
[:SENSe]:NORMalize:ENABle
[:SENSe]:NORMalize:WAVelength:STARt <wavelength>
[:SENSe]:NORMalize:WAVelength:STOP <wavelength>
```

To perform a normalization, connect a short cable from the laser to receiver input l:

```
[:SENSe]:NORMalize:WAVelength:STARt 1540 nm
[:SENSe]:NORMalize:WAVelength:STOP 1543 nm
:INITiate:NORMalize 1
```

To turn normalization on:

```
[:SENSe]:NORMalize:ENABle 1, ON
```

To read settings back:

```
[:SENSe]:NORMalize:WAVelength:STARt?
[:SENSe]:NORMalize:WAVelength:STOP?
TRACe:POINts? NORM 1
```

To read data back:

```
TRACe:DATA:Y? NORM 1
-5.32065062D+000,-5.32445866D+000,-5.33022176E+000,-5.32928816E+000,-5.3325715E+000...
```

To upload a different normalized trace:

```
{:SENSe]:NORMalize:WAVelength:STARt 1530 nm
{:SENSe]:NORMalize:WAVelength:STOP 1532 nm
TRACe:DATA:Y NORM 1, <value>, <value>...
NORMalize:ENABle 1, ON
```

NOTE

For faster speed, use binary transfer, only sending back in data acquired by the WDCA. You can send back in trace data taken with normalization off and send back normalization data read out of the instrument at a previous time. The data must be acquired with 5 pm wavelength step.

WDCA Techniques to Improve Throughput

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

Take control of the sweep

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

Turn the display off

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

To turn the display off:

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

This takes about 14 seconds and does not affect the instrument state. You can turn the display back on whenever you want to, either by sending DISP ON or by pressing the local key on the instrument front panel. It again will take about 14 seconds. Note that you cannot turn off the display from the front panel.

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

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

Combine commands into a single message

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

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

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

```
sendMessage INIT:*OPC?      // take a sweep, instrument tells us when done
receiverInteger             // wait for reply
```

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

Transfer traces when more than 2 marker operations

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

Use 32 bit binary trace transfers

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

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

```
sendMessage FORMat REAL,32
```

See “Example 6, Acquire Binary Trace” on page 5-61 for an example of how to transfer and interpret a binary trace.

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

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

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

Table 5-9. Binary vs. ASCII Downloads

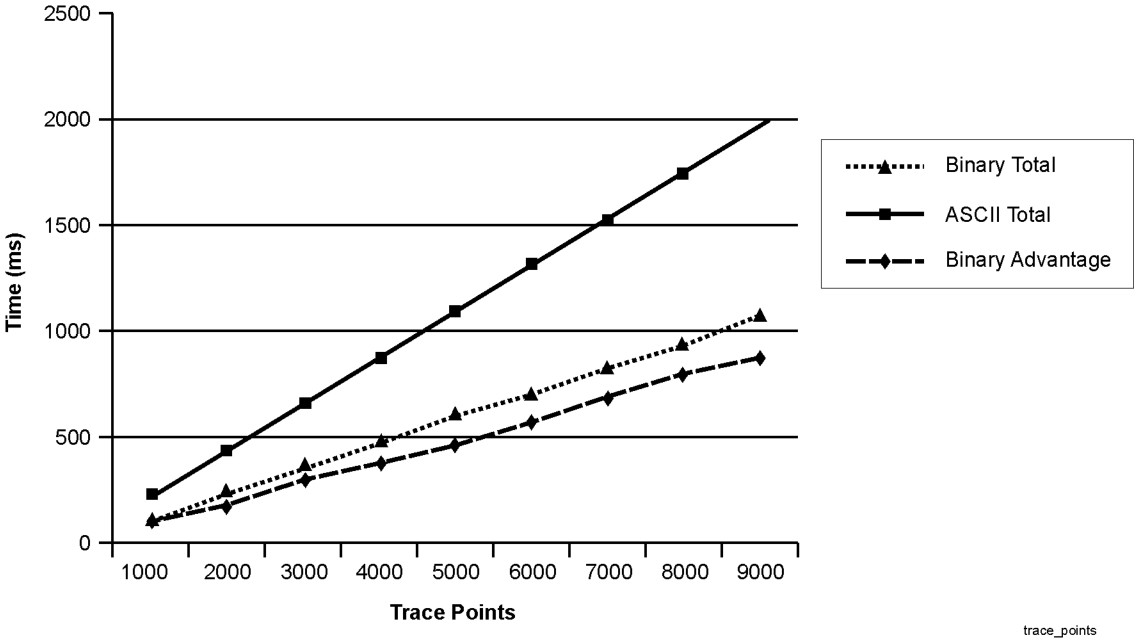


Figure 5-2. Binary vs. ASCII Downloads

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

```
int cnt;
char tempByte;
char *traceBytes;
// allocate buffer and transfer trace into traceBytes

for (cnt=0;cnt<numBytes;cnt+=4)
{
    // swap bytes 3 and 0
    tempByte         = traceBytes[cnt];
    traceBytes[cnt]   = traceBytes[cnt + 3];
    traceBytes[cnt+3] = tempByte;

    // swap bytes 2 and 1
    tempByte         = traceBytes[cnt+1];
    traceBytes[cnt+1] = traceBytes[cnt+2];
    traceBytes[cnt+2] = tempByte;

}
```

Optimize Sensitivity, Sweep Mode, and Wavelength Span

Sensitivity is the smallest detectable signal relative to the top of the display. As sensitivity increases so does the sweep time. Therefore, use the minimum sensitivity that allows you to discern the signals in your measurement. Using auto sensitivity optimizes the sensitivity setting allowing for the fastest sweep time for the selected instrument mode. If your test application can tolerate slightly higher amplitude and wavelength uncertainty, try using Fast Sweep mode instead of Precision or Normal Sweep modes. The wavelength span also affects sweep time, the greater the span the longer the sweep time.

Example Programs

These programs are provided to give you examples of using Agilent 86082A WDCA remote programming commands in typical applications. They are not meant to teach general programming techniques or provide ready-to-use solutions. They should allow you to see how measurements are performed and how to return data to the computer. The program examples are written in Visual Basic and VC++ for Windows.

The following example programs are provided in this section:

Example 1, Acquire Trace

Description

This program demonstrates how to use Visual Basic to take a measurement and download a trace in ASCII format. The program also demonstrates how to change several settings such as center wavelength, span, and input receiver channel. It should be noted that the settings are not sent to the instrument until the Acquire Trace button is pressed.

Program

```
Private Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)

Private Const OPC_TMO = 180
Private Const LZ_TMO = 360

Private Sub cmdAcquireTrace_Click()
Dim TimeStart As Double

'--  VISA session variables
Dim DefRM As Long                                          'VISA session
Dim DevSessionNum As Long                                  'VISA session to device
Dim ReadBuffer As String * 65500                           'Buffer to read back data
Dim STB As Integer                                         'Status byte
Dim Status As Long: Status = 0
Dim ErrMsg As String: ErrMsg = ""

  On Error GoTo ErrorHandler

  MousePointer = vbHourglass

  '--  Create a VISA session
  Status = viOpenDefaultRM(DefRM)
  If Status <> 0 Then GoTo ErrorHandler

  '--  Open a session to the WDCA
  Status = viOpen(DefRM, "GPIB" & txtGPIB_IN.Text & "::" & txtGPIB_PA.Text, 0, 0, DevSessionNum)
  If Status <> 0 Then GoTo ErrorHandler

  '--  Clear the instrument
  Status = viClear(DevSessionNum)
  If Status <> 0 Then GoTo ErrorHandler

  '--  Set write buffer to flush on access
  Status = viSetAttribute(DevSessionNum, VI_ATTR_WR_BUF_OPER_MODE, VI_FLUSH_ON_ACCESS)
  If Status <> 0 Then GoTo ErrorHandler

  '--  Set read buffer to flush on access
  Status = viSetAttribute(DevSessionNum, VI_ATTR_RD_BUF_OPER_MODE, VI_FLUSH_ON_ACCESS)
  If Status <> 0 Then GoTo ErrorHandler

  '--  Clear status registers
  Status = viVPrintf(DevSessionNum, "*CLS" & Chr$(10), 0)
  If Status <> 0 Then GoTo ErrorHandler
```

```
'--  Map OPC to bit 5 of the Standard Event Status Enable Register
Status = viVPrintf(DevSessionNum, "*ESE 1" & Chr$(10), 0)
If Status <> 0 Then GoTo ErrorHandler

'--  Ensure that the instrument isn't in repeat sweep mode
Status = viVPrintf(DevSessionNum, "INIT:CONT 0" & Chr$(10), 0)
If Status <> 0 Then GoTo ErrorHandler

'--  Check to see if source is about to LambaZero.  If so, force it to happen.
Status = CheckLambdaZero(DevSessionNum, ErrMsg)
If Status <> 0 Or ErrMsg <> "" Then GoTo ErrorHandler

'--  Format data as ASCII
Status = viVPrintf(DevSessionNum, "FORM:DATA ASCII" & Chr$(10), 0)
If Status <> 0 Then GoTo ErrorHandler

'--  Set Center Wavelength
Status = viVPrintf(DevSessionNum, "SENS:WAV:CENT " & Val(txtCWL.Text) & "nm" & Chr$(10), 0)
If Status <> 0 Then GoTo ErrorHandler

'--  Set number of points for WDCA to acquire during sweep
Status = viVPrintf(DevSessionNum, "SENS:SWE:POIN 1001" & Chr$(10), 0)
If Status <> 0 Then GoTo ErrorHandler

'--  Set Span
Status = viVPrintf(DevSessionNum, "SENS:WAV:SPAN " & Val(txtSpan.Text) & "nm" & Chr$(10), 0)
If Status <> 0 Then GoTo ErrorHandler

'--  Set sensitivity
If chkSensitivityLock.Value = 0 Then
   '--  Unlock Sensitivity
   Status = viVPrintf(DevSessionNum, "SENS:POW:DC:RANG:LOW:AUTO OFF" & Chr$(10), 0)
   If Status <> 0 Then GoTo ErrorHandler

   '--  Set Sensitivity
   Status = viVPrintf(DevSessionNum, "SENS:POW:DC:RANG:LOW " & Val(txtSensitivity.Text) & _
            Chr$(10), 0)
   If Status <> 0 Then GoTo ErrorHandler

Else
   '--  Lock Sensitivity
   Status = viVPrintf(DevSessionNum, "SENS:POW:DC:RANG:LOW:AUTO ON" & Chr$(10), 0)
   If Status <> 0 Then GoTo ErrorHandler

End If

'--  Sweep Mode
Status = viVPrintf(DevSessionNum, "SENS:SWE:SPE " & cmbSweepMode.Text & Chr$(10), 0)
If Status <> 0 Then GoTo ErrorHandler

'--  Trace settings
   '--  Attach trace to input
  Status = viVPrintf(DevSessionNum, "TRAC:FEED:INP " & cmbTrace.Text & "," & cmbChannel.Text & _
            Chr$(10), 0)
   If Status <> 0 Then GoTo ErrorHandler
```

```
    ’--  Turn trace update on
    Status = viVPrintf(DevSessionNum, "TRAC:FEED:CONT " & cmbTrace.Text & ",ALW" & Chr$(10), 0)
    If Status <> 0 Then GoTo ErrorHandler

    ’--  Make trace visible on the display
   Status = viVPrintf(DevSessionNum, "DISP:WIND:TRAC:STAT " & cmbTrace.Text & ",1" & Chr$(10), 0)
    If Status <> 0 Then GoTo ErrorHandler

’--  Take a sweep and send OPC
Status = viVPrintf(DevSessionNum, "INIT:IMM;*OPC" & Chr$(10), 0)
If Status <> 0 Then GoTo ErrorHandler

’--  Wait for completion

    TimeStart = Now * 86400     ’Used to track elapsed time

    Do

        Call Sleep(5)   ’Delay so the instrument isn’t tied up with responding to STB
        DoEvents

        ’--  Check bit 5 of the status byte register to see if the operation is complete
        Status = viReadSTB(DevSessionNum, STB)
        If Status <> 0 Then GoTo ErrorHandler

        ’--  Check bit 7 to see if a lambda zero is taking place
        If (STB And 128) = 128 Then

            ’--  If an OPC_TMO + 5 minutes goes by, throw a timeout
            If (Now * 86400) - TimeStart > OPC_TMO + 300 Then GoTo OPCTimeOut

        Else

            ’--  If a minute goes by, throw a timeout
            If (Now * 86400) - TimeStart > OPC_TMO Then GoTo OPCTimeOut

        End If

    Loop While (STB And 32) <> 32

    ’--  Clear status information
    Status = viVPrintf(DevSessionNum, "*CLS" & Chr$(10), 0)
    If Status <> 0 Then GoTo ErrorHandler

’--  Download results
    ’--  Send query to acquire Trace A
    Status = viVPrintf(DevSessionNum, "TRAC? TRA" & Chr$(10), 0)
    If Status <> 0 Then GoTo ErrorHandler

    ’--  Read the trace from the buffer
    Status = viVScanf(DevSessionNum, "%t", ReadBuffer)
    If Status <> 0 Then GoTo ErrorHandler
```

```
        'Enter code to translate the comma-delimitted data
        Call MsgBox("Done!  Small sample of points:" & vbCrLf & Left(ReadBuffer, 33) & "...", _
                vbOKOnly, "Trace Acquired")

EndSession:
    On Error Resume Next

    '--  Close VISA session to the WDCA
    Call viClose(DevSessionNum)

    '--  End VISA session
    Call viClose(DefRM)

    MousePointer = vbDefault

    Exit Sub

OPCTimeOut:
    MousePointer = vbDefault
    Call MsgBox("Timeout while waiting for OPC.  Make sure all cables are securely fastened.", _
                vbOKOnly, "Timeout")
    GoTo EndSession

ErrorHandler:
    If Err.Number = 0 Then

        If Status <> 0 Then
        '--  VISA error

            MousePointer = vbDefault

            '--  Translate error number into a message
            Call viStatusDesc(DevSessionNum, Status, ReadBuffer)

            '--  Trim off nulls
            ErrMsg = TrimVISAStr(ReadBuffer)

            '--  Post message and reset vars
            Call MsgBox(Status & vbCrLf & ErrMsg, vbOKOnly, "Error")

            Status = 0

        Else
        '--  Non-VISA error from another function using VISA
            Call MsgBox(ErrMsg, vbOKOnly, "Error")
        End If

        ErrMsg = ""

    Else
        Call MsgBox(Err.Number & vbCrLf & Err.Description, vbOKOnly, "Error")
    End If

    GoTo EndSession

End Sub
```

```
Private Function TrimVISAStr(ByVal VISAStr As String) As String
Dim Pos As Long

    On Error GoTo ErrorHandler

    Pos = (InStr(VISAStr, Chr(0))) - 1

    If Pos <> -1 Then
        TrimVISAStr = Left(VISAStr, Pos)
    Else
        TrimVISAStr = Trim(VISAStr)
    End If

    Exit Function

ErrorHandler:
    TrimVISAStr = VISAStr

End Function
```

Example 2, User Calibration

Description

A noticeable change in temperature will cause the WDCA to perform a wavelength calibration. However, there may be times in which it is important to perform a calibration before testing a device. This Visual Basic program demonstrates how to perform a wavelength calibration.

Program

```
Private Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)

Private Const OPC_TMO = 180
Private Const LZ_TMO = 360


Private Sub cmdCalibration_Click()
Dim TimeStart As Double

'--  VISA session variables
Dim DefRM As Long                                    'VISA session
Dim DevSessionNum As Long                            'VISA session to device
Dim ReadBuffer As String * 65500                     'Buffer to read back data
Dim STB As Integer                                   'Status byte
Dim Status As Long: Status = 0
Dim ErrMsg As String: ErrMsg = ""

  On Error GoTo ErrorHandler

  Call MsgBox("Connect Output 2 of the TLS to the Source Input", vbOKOnly)

  MousePointer = vbHourglass

  '--  Create a VISA session
  Status = viOpenDefaultRM(DefRM)
  If Status <> 0 Then GoTo ErrorHandler

  '--  Open a session to the WDCA
  Status = viOpen(DefRM, "GPIB" & txtGPIB_IN.Text & "::" & txtGPIB_PA.Text, 0, 0, DevSessionNum)
  If Status <> 0 Then GoTo ErrorHandler

  '--  Clear the instrument
  Status = viClear(DevSessionNum)
  If Status <> 0 Then GoTo ErrorHandler

  '--  Set write buffer to flush on access
  Status = viSetAttribute(DevSessionNum, VI_ATTR_WR_BUF_OPER_MODE, VI_FLUSH_ON_ACCESS)
  If Status <> 0 Then GoTo ErrorHandler

  '--  Set read buffer to flush on access
  Status = viSetAttribute(DevSessionNum, VI_ATTR_RD_BUF_OPER_MODE, VI_FLUSH_ON_ACCESS)
  If Status <> 0 Then GoTo ErrorHandler

  '--  Clear status registers
```

```
    Status = viVPrintf(DevSessionNum, "*CLS" & Chr$(10), 0)
    If Status <> 0 Then GoTo ErrorHandler

    '--  Map OPC to bit 5 of the Standard Event Status Enable Register
    Status = viVPrintf(DevSessionNum, "*ESE 1" & Chr$(10), 0)
    If Status <> 0 Then GoTo ErrorHandler

    '--  Ensure that the instrument isn't in repeat sweep mode
    Status = viVPrintf(DevSessionNum, "INIT:CONT 0" & Chr$(10), 0)
    If Status <> 0 Then GoTo ErrorHandler

    '--  Send Calibration command OPC
    Status = viVPrintf(DevSessionNum, "CAL:WAV:INT;*OPC" & Chr$(10), 0)
    If Status <> 0 Then GoTo ErrorHandler

    '--  Wait for completion

        TimeStart = Now * 86400     'Used to track elapsed time

        Do

            '--  If a minute goes by, throw a timeout
            If (Now * 86400) - TimeStart > OPC_TMO Then GoTo OPCTimeOut

            '--  Check bit 5 of the status byte register to see if the operation is complete
            Status = viReadSTB(DevSessionNum, STB)
            If Status <> 0 Then GoTo ErrorHandler

            Call Sleep(5)   'Delay so the instrument isn't tied up with responding to STB
            DoEvents

        Loop While (STB And 32) <> 32

        '--  Clear status information
        Status = viVPrintf(DevSessionNum, "*CLS" & Chr$(10), 0)
        If Status <> 0 Then GoTo ErrorHandler

        Call MsgBox("Calibration Completed!", vbOKOnly, "Done")

EndSession:
    On Error Resume Next

    '--  Close VISA session to the WDCA
    Call viClose(DevSessionNum)

    '--  End VISA session
    Call viClose(DefRM)

    MousePointer = vbDefault

    Exit Sub

OPCTimeOut:
    MousePointer = vbDefault
    Call MsgBox("Timeout while waiting for OPC.  Make sure all cables are securely fastened.", _
                vbOKOnly, "Timeout")
    GoTo EndSession

ErrorHandler:
    If Err.Number = 0 Then

        MousePointer = vbDefault

        '--  Translate error number into a message
        Call viStatusDesc(DevSessionNum, Status, ReadBuffer)

        '--  Trim off nulls
        ErrMsg = TrimVISAStr(ReadBuffer)

        '--  Post message and reset vars
        Call MsgBox(Status & vbCrLf & ErrMsg, vbOKOnly, "Error")
        Status = 0
        ErrMsg = ""
    Else
        Call MsgBox(Err.Number & vbCrLf & Err.Description, vbOKOnly, "Error")
    End If

    GoTo EndSession

End Sub


Private Function TrimVISAStr(ByVal VISAStr As String) As String
Dim Pos As Long

    On Error GoTo ErrorHandler

    Pos = (InStr(VISAStr, Chr(0))) - 1

    If Pos <> -1 Then
        TrimVISAStr = Left(VISAStr, Pos)
    Else
        TrimVISAStr = Trim(VISAStr)
    End If

    Exit Function

ErrorHandler:
    TrimVISAStr = VISAStr

End Function
```

Example 3, Normalization

Description

Normalization should be performed to ensure quality characterization of a DUT. This Visual Basic program demonstrates how to perform normalization remotely.

Program

```
Private Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)

Private Const OPC_TMO = 180
Private Const LZ_TMO = 360


Private Sub cmdStart_Click()
Dim TimeStart As Double

’--  VISA session variables
Dim DefRM As Long                                       ’VISA session
Dim DevSessionNum As Long                               ’VISA session to device
Dim ReadBuffer As String * 65500                        ’Buffer to read back data
Dim STB As Integer                                      ’Status byte
Dim Status As Long: Status = 0
Dim ErrMsg As String: ErrMsg = ""

    On Error GoTo ErrorHandler

    ’--  Prompt user to make the proper connection
    If MsgBox("Connect Output 1 to Input " & cmbInput.Text & "." & vbCrLf & "Press OK to continue.", _
            vbOKCancel, "Make connection") = vbCancel Then Exit Sub

    MousePointer = vbHourglass

    ’--  Call function to initialize a VISA session to the WDCA
    Status = InitVISASession(txtGPIB_IN.Text, txtGPIB_PA.Text, DefRM, DevSessionNum, ErrMsg)
    If Status <> 0 Or ErrMsg <> "" Then GoTo ErrorHandler

    ’--  Wavelength Start
    Status = viVPrintf(DevSessionNum, "SENS:NORM:WAV:START " & Val(txtWavelengthStart.Text) _
             & "nm" & Chr$(10), 0)
    If Status <> 0 Then GoTo ErrorHandler

    ’--  Wavelength Stop
    Status = viVPrintf(DevSessionNum, "SENS:NORM:WAV:STOP " & Val(txtWavelengthStop.Text) & _
             "nm" & Chr$(10), 0)
    If Status <> 0 Then GoTo ErrorHandler

’--  Trace settings
        ’--  Attach trace to input
        Status = viVPrintf(DevSessionNum, "TRAC:FEED:INP " & cmbTrace.Text & "," & cmbInput.Text & _
                Chr$(10), 0)
        If Status <> 0 Then GoTo ErrorHandler

        ’--  Turn trace update on
    Status = viVPrintf(DevSessionNum, "TRAC:FEED:CONT " & cmbTrace.Text & ",ALW" & Chr$(10), 0)
    If Status <> 0 Then GoTo ErrorHandler

    ’--  Make trace visible on the display
   Status = viVPrintf(DevSessionNum, "DISP:WIND:TRAC:STAT " & cmbTrace.Text & ",1" & Chr$(10), 0)
    If Status <> 0 Then GoTo ErrorHandler

’--  Start normalization for the appropriate channel
Status = viVPrintf(DevSessionNum, "INIT:NORM " & cmbInput.Text & ";*OPC" & Chr$(10), 0)
If Status <> 0 Then GoTo ErrorHandler

’--  Wait for completion

    TimeStart = Now * 86400     ’Used to track elapsed time

    Do

        ’--  If a minute goes by, throw a timeout
        If (Now * 86400) - TimeStart > OPC_TMO Then GoTo OPCTimeOut

        ’--  Check bit 5 of the status byte register to see if the operation is complete
        Status = viReadSTB(DevSessionNum, STB)
        If Status <> 0 Then GoTo ErrorHandler

        Call Sleep(10)   ’Delay so the instrument isn’t tied up with responding to STB
        DoEvents

    Loop While (STB And 32) <> 32

    ’--  Clear status information
    Status = viVPrintf(DevSessionNum, "*CLS" & Chr$(10), 0)
    If Status <> 0 Then GoTo ErrorHandler

’--  Enable normalization for the selected input
Status = viVPrintf(DevSessionNum, "SENS:NORM:ENABLE " & cmbInput.Text & ",ON" & Chr$(10), 0)
If Status <> 0 Then GoTo ErrorHandler

’--  Take a sweep and send OPC
    Status = viVPrintf(DevSessionNum, "INIT:IMM;*OPC" & Chr$(10), 0)
    If Status <> 0 Then GoTo ErrorHandler

    ’--  Wait for completion

        TimeStart = Now * 86400     ’Used to track elapsed time

        Do

            Call Sleep(5)   ’Delay so the instrument isn’t tied up with responding to STB
            DoEvents

            ’--  Check bit 5 of the status byte register to see if the operation is complete
            Status = viReadSTB(DevSessionNum, STB)
            If Status <> 0 Then GoTo ErrorHandler

            ’--  Check bit 7 to see if a lambda zero is taking place
            If (STB And 128) = 128 Then
```

```
               ’--  If an OPC_TMO + 5 minutes goes by, throw a timeout
               If (Now * 86400) - TimeStart > OPC_TMO + 300 Then GoTo OPCTimeOut

            Else

               ’--  If a minute goes by, throw a timeout
               If (Now * 86400) - TimeStart > OPC_TMO Then GoTo OPCTimeOut

            End If

         Loop While (STB And 32) <> 32

         ’--  Clear status information
         Status = viVPrintf(DevSessionNum, "*CLS" & Chr$(10), 0)
         If Status <> 0 Then GoTo ErrorHandler

   Call MsgBox("Normalization complete!", vbOKOnly, "Done")

EndSession:
   On Error Resume Next

   ’--  Close VISA session to the WDCA
   Call viClose(DevSessionNum)

   ’--  End VISA session
   Call viClose(DefRM)

   MousePointer = vbDefault

   Exit Sub

OPCTimeOut:
   MousePointer = vbDefault
   Call MsgBox("Timeout while waiting for OPC.  Make sure all cables are securely fastened.", _
               vbOKOnly, "Timeout")
   GoTo EndSession

ErrorHandler:
   If Err.Number = 0 Then

      If Status <> 0 Then
      ’--  VISA error

         MousePointer = vbDefault

         ’--  Translate error number into a message
         Call viStatusDesc(DevSessionNum, Status, ReadBuffer)

         ’--  Trim off nulls
         ErrMsg = TrimVISAStr(ReadBuffer)

         ’--  Post message and reset vars
         Call MsgBox(Status & vbCrLf & ErrMsg, vbOKOnly, "Error")

         Status = 0
```

```
        Else
        '--  Non-VISA error from another function using VISA
            Call MsgBox(ErrMsg, vbOKOnly, "Error")
        End If

        ErrMsg = ""

    Else
        Call MsgBox(Err.Number & vbCrLf & Err.Description, vbOKOnly, "Error")
    End If

    GoTo EndSession

End Sub


Private Function CheckLambdaZero(DevSessionNum As Long, Optional ErrMsg As String = "") _
                  As Long
'--  This function will check to see if a TLS LambdaZero is about to occur.  If so, force it to
'--  happen to prevent timeouts and ensure data integrity.
'--
Dim TimeStart As Double

Dim STB As Integer
Dim Status As Long

    On Error GoTo ErrorHandler

    '--  Summarize Lambda Zero Pending to the status byte register (bit 7, decimal 128)
    Status = viVPrintf(DevSessionNum, "STAT:OPER:ENAB 4" & Chr$(10), 0)
    If Status <> 0 Then GoTo ErrorHandler

    '--  Summarize Lambda Zero Complete to the status byte register (bit 3, decimal 8)
    Status = viVPrintf(DevSessionNum, "STAT:QUES:ENAB 8" & Chr$(10), 0)
    If Status <> 0 Then GoTo ErrorHandler

    '--  Check to see if a Lambda Zero is pending
    Status = viReadSTB(DevSessionNum, STB)
    If Status <> 0 Then GoTo ErrorHandler

    '--  If a Lambda Zero is pending, force it to occur
    If (STB And 128) = 128 Then

        '--  Start Lambda Zero
        Status = viVPrintf(DevSessionNum, "CAL:WAV:INT" & Chr$(10), 0)
        If Status <> 0 Then GoTo ErrorHandler

        TimeStart = Now * 86400     'Used to track elapsed time

        '--  Wait for Lambda Zero to complete
        Do

            Call Sleep(5)
            DoEvents
```

```
         ’--  Read STB to determine when Lambda Zero is complete
         Status = viReadSTB(DevSessionNum, STB)
         If Status <> 0 Then GoTo ErrorHandler

         ’--  Check for timeout (this could take over 5 minutes)
         If ((Now * 86400) - TimeStart) > (LZ_TMO) Then GoTo ErrorHandler

      Loop While (STB And 8) <> 8

   End If

   ’--  Clear the Status Byte
   Status = viVPrintf(DevSessionNum, "*CLS" & Chr$(10), 0)
   If Status <> 0 Then GoTo ErrorHandler

   ’--  Summarize Lambda Zero In Progress to the status byte (bit 7, decimal 128).
   ’--  This can be used later to detect if an operation timed out due to a LambdaZero
   Status = viVPrintf(DevSessionNum, "STAT:OPER:ENAB 8" & Chr$(10), 0)
   If Status <> 0 Then GoTo ErrorHandler

   ErrMsg = ""
   CheckLambdaZero = 0
   Exit Function

LZ_Timeout:
   ErrMsg = "Timeout while waiting for Lambda Zero to complete.  Make sure Output 2 is connected" & _
            " to Source Input!"
   CheckLambdaZero = 0
   Exit Function

ErrorHandler:
   If Err.Number = 0 Then
      CheckLambdaZero = Status
   Else
      ErrMsg = Err.Number & vbCrLf & Err.Description
      CheckLambdaZero = 0
   End If

End Function


Private Function TrimVISAStr(ByVal VISAStr As String) As String
Dim Pos As Long

   On Error GoTo ErrorHandler

   Pos = (InStr(VISAStr, Chr(0))) - 1

   If Pos <> -1 Then
      TrimVISAStr = Left(VISAStr, Pos)
   Else
      TrimVISAStr = Trim(VISAStr)
   End If

   Exit Function
```

```
ErrorHandler:
    TrimVISAStr = VISAStr

End Function
```

Example 4, Markers

Description

The WDCA is capable of performing bandwidth measurements of the DUT. This Visual Basic program will demonstrate how easy it is to have the instrument calculate the bandwidth and return the result.

Program

```
Private Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)

Private Const OPC_TMO = 180
Private Const LZ_TMO = 360

Private Sub cmdMarker_Click()
Dim TimeStart As Double

'--  VISA session variables
Dim DefRM As Long                                   'VISA session
Dim DevSessionNum As Long                           'VISA session to device
Dim ReadBuffer As String * 65500                    'Buffer to read back data
Dim STB As Integer                                  'Status byte
Dim Status As Long: Status = 0
Dim ErrMsg As String: ErrMsg = ""

On Error GoTo ErrorHandler

  MousePointer = vbHourglass

  '--  Call function to initialize a VISA session to the WDCA
  Status = InitVISASession(txtGPIB_IN.Text, txtGPIB_PA.Text, DefRM, DevSessionNum, ErrMsg)
  If Status <> 0 Or ErrMsg <> "" Then GoTo ErrorHandler

  '--  Set Center Wavelength
  Status = viVPrintf(DevSessionNum, "SENS:WAV:CENT " & Val(txtCWL.Text) & "nm" & Chr$(10), 0)
  If Status <> 0 Then GoTo ErrorHandler

  '--  Set Span
  Status = viVPrintf(DevSessionNum, "SENS:WAV:SPAN " & Val(txtSpan.Text) & "nm" & Chr$(10), 0)
  If Status <> 0 Then GoTo ErrorHandler

  '--  Set sensitivity
  If chkSensitivityLock.Value = 0 Then
    '--  Unlock Sensitivity
    Status = viVPrintf(DevSessionNum, "SENS:POW:DC:RANG:LOW:AUTO OFF" & Chr$(10), 0)
    If Status <> 0 Then GoTo ErrorHandler

    '--  Set Sensitivity
    Status = viVPrintf(DevSessionNum, "SENS:POW:DC:RANG:LOW " & Val(txtSensitivity.Text) & _
          Chr$(10), 0)
    If Status <> 0 Then GoTo ErrorHandler

  Else
    '--  Lock Sensitivity
    Status = viVPrintf(DevSessionNum, "SENS:POW:DC:RANG:LOW:AUTO ON" & Chr$(10), 0)
    If Status <> 0 Then GoTo ErrorHandler

End If

'-- Sweep Mode
Status = viVPrintf(DevSessionNum, "SENS:SWE:SPE " & cmbSweepMode.Text & Chr$(10), 0)
If Status <> 0 Then GoTo ErrorHandler

'-- Trace A settings
    '-- Attach trace A to input 1
   Status = viVPrintf(DevSessionNum, "TRAC:FEED:INP " & cmbTrace.Text & "," & cmbChannel.Text & _
              Chr$(10), 0)
    If Status <> 0 Then GoTo ErrorHandler

    '-- Turn trace A update on
    Status = viVPrintf(DevSessionNum, "TRAC:FEED:CONT " & cmbTrace.Text & ",ALW" & Chr$(10), 0)
    If Status <> 0 Then GoTo ErrorHandler

    '-- Make trace A visible on the display
   Status = viVPrintf(DevSessionNum, "DISP:WIND:TRAC:STAT " & cmbTrace.Text & ",1" & Chr$(10), 0)
    If Status <> 0 Then GoTo ErrorHandler

'-- Take a sweep and send OPC
Status = viVPrintf(DevSessionNum, "INIT:IMM;*OPC" & Chr$(10), 0)
If Status <> 0 Then GoTo ErrorHandler

'-- Wait for completion

    TimeStart = Now * 86400     'Used to track elapsed time

    Do

        '-- If a minute goes by, throw a timeout
        If (Now * 86400) - TimeStart > OPC_TMO Then GoTo OPCTimeOut

        '-- Check bit 5 of the status byte register to see if the operation is complete
        Status = viReadSTB(DevSessionNum, STB)
        If Status <> 0 Then GoTo ErrorHandler

        Call Sleep(5)   'Delay so the instrument isn't tied up with responding to STB
        DoEvents

    Loop While (STB And 32) <> 32

    '-- Clear status information
    Status = viVPrintf(DevSessionNum, "*CLS" & Chr$(10), 0)
    If Status <> 0 Then GoTo ErrorHandler

'-- ** Markers **
    '-- Set the marker to a trace
   Status = viVPrintf(DevSessionNum, "CALC:MARK" & cmbMarker.Text & ":TRAC " & cmbTrace.Text, _
              ErrMsg)
    If Status <> 0 Then GoTo ErrorHandler

    '-- Do a peak search
    Status = viVPrintf(DevSessionNum, "CALC:MARK" & cmbMarker.Text & ":MAX", ErrMsg)
    If Status <> 0 Then GoTo ErrorHandler

    '--  Set the BW marker to num DB
    Status = viVPrintf(DevSessionNum, "CALC:MARK" & cmbMarker.Text &":FUNC:BAND:NDB " & _
            txtNDB.Text, ErrMsg)
    If Status <> 0 Then GoTo ErrorHandler

    '--  Turn on the bandwidth marker
    Status = viVPrintf(DevSessionNum, "CALC:MARK" & cmbMarker.Text & _
                                              ":FUNC:BAND:STAT_ON",ErrMsg)
    If Status <> 0 Then GoTo ErrorHandler

    '--  Retreive result (NOTE:  Result is also displayed at the top of the screen under BW)
    Status = viVPrintf(DevSessionNum, "CALC:MARK" & cmbMarker.Text & ":FUNC:BAND:RES?", _
                                              ErrMsg)
    If Status <> 0 Then GoTo ErrorHandler

    Status = viVScanf(DevSessionNum, "%t", ReadBuffer)
    If Status <> 0 Then GoTo ErrorHandler

    Call MsgBox("Bandwidth = " & TrimVISAStr(ReadBuffer), vbOKOnly, "Done")


EndSession:
  On Error Resume Next

  '--  Close VISA session to the WDCA
  Call viClose(DevSessionNum)

  '--  End VISA session
  Call viClose(DefRM)

  MousePointer = vbDefault

  Exit Sub

OPCTimeOut:
  MousePointer = vbDefault
  Call MsgBox("Timeout while waiting for OPC.  Make sure all cables are securely fastened.", _
          vbOKOnly, "Timeout")
  GoTo EndSession

ErrorHandler:
  If Err.Number = 0 Then

    If Status <> 0 Then
    '--  VISA error

      MousePointer = vbDefault

      '--  Translate error number into a message
      Call viStatusDesc(DevSessionNum, Status, ReadBuffer)

      '--  Trim off nulls
      ErrMsg = TrimVISAStr(ReadBuffer)
```

```
        '--  Post message and reset vars
        Call MsgBox(Status & vbCrLf & ErrMsg, vbOKOnly, "Error")

        Status = 0

      Else
      '--  Non-VISA error from another function using VISA
        Call MsgBox(ErrMsg, vbOKOnly, "Error")
      End If

      ErrMsg = ""

    Else
      Call MsgBox(Err.Number & vbCrLf & Err.Description, vbOKOnly, "Error")
    End If

    GoTo EndSession

End Sub


Private Function CheckLambdaZero(DevSessionNum As Long, Optional ErrMsg As String = "") _
          As Long
'--  This function will check to see if a TLS LambdaZero is about to occur.  If so, force it to
'--  happen to prevent timeouts and ensure data integrity.
'--
Dim TimeStart As Double

Dim STB As Integer
Dim Status As Long

    On Error GoTo ErrorHandler

    '--  Summarize Lambda Zero Pending to the status byte register (bit 7, decimal 128)
    Status = viVPrintf(DevSessionNum, "STAT:OPER:ENAB 4" & Chr$(10), 0)
    If Status <> 0 Then GoTo ErrorHandler

    '--  Summarize Lambda Zero Complete to the status byte register (bit 3, decimal 8)
    Status = viVPrintf(DevSessionNum, "STAT:QUES:ENAB 8" & Chr$(10), 0)
    If Status <> 0 Then GoTo ErrorHandler

    '--  Check to see if a Lambda Zero is pending
    Status = viReadSTB(DevSessionNum, STB)
    If Status <> 0 Then GoTo ErrorHandler

    '--  If a Lambda Zero is pending, force it to occur
    If (STB And 128) = 128 Then

      '--  Start Lambda Zero
      Status = viVPrintf(DevSessionNum, "CAL:WAV:INT" & Chr$(10), 0)
      If Status <> 0 Then GoTo ErrorHandler

      TimeStart = Now * 86400     'Used to track elapsed time

      '--  Wait for Lambda Zero to complete
```

```
        Do

            Call Sleep(5)
            DoEvents

            '--  Read STB to determine when Lambda Zero is complete
            Status = viReadSTB(DevSessionNum, STB)
            If Status <> 0 Then GoTo ErrorHandler

            '--  Check for timeout (this could take over 5 minutes)
            If ((Now * 86400) - TimeStart) > (LZ_TMO) Then GoTo ErrorHandler

        Loop While (STB And 8) <> 8

    End If

    '--  Clear the Status Byte
    Status = viVPrintf(DevSessionNum, "*CLS" & Chr$(10), 0)
    If Status <> 0 Then GoTo ErrorHandler

    '--  Summarize Lambda Zero In Progress to the status byte (bit 7, decimal 128).
    '--  This can be used later to detect if an operation timed out due to a LambdaZero
    Status = viVPrintf(DevSessionNum, "STAT:OPER:ENAB 8" & Chr$(10), 0)
    If Status <> 0 Then GoTo ErrorHandler

    ErrMsg = ""
    CheckLambdaZero = 0
    Exit Function

LZ_Timeout:
    ErrMsg = "Timeout while waiting for Lambda Zero to complete.  Make sure Output 2 is connected" & _
            " to Source Input!"
    CheckLambdaZero = 0
    Exit Function

ErrorHandler:
    If Err.Number = 0 Then
        CheckLambdaZero = Status
    Else
        ErrMsg = Err.Number & vbCrLf & Err.Description
        CheckLambdaZero = 0
    End If

End Function


Private Function InitVISASession(ByVal GPIB_IN As String, ByVal GPIB_PA As String, _
                                DefRM As Long, DevSessionNum As Long, _
                                Optional ErrMsg As String = "") As Long

'--  VISA session variables
Dim ReadBuffer As String * 65500    'Buffer to read back data
Dim Status As Long: Status = 0

    On Error GoTo ErrorHandler
    '--  Create a VISA session
    Status = viOpenDefaultRM(DefRM)
    If Status <> 0 Then GoTo ErrorHandler

    '--  Open a session to the WDCA
    Status = viOpen(DefRM, "GPIB" & GPIB_IN & "::" & GPIB_PA, 0, 0, DevSessionNum)
    If Status <> 0 Then GoTo ErrorHandler

    '--  Clear the instrument
    Status = viClear(DevSessionNum)
    If Status <> 0 Then GoTo ErrorHandler

    '--  Set write buffer to flush on access
    Status = viSetAttribute(DevSessionNum, VI_ATTR_WR_BUF_OPER_MODE, VI_FLUSH_ON_ACCESS)
    If Status <> 0 Then GoTo ErrorHandler

    '--  Set read buffer to flush on access
    Status = viSetAttribute(DevSessionNum, VI_ATTR_RD_BUF_OPER_MODE, VI_FLUSH_ON_ACCESS)
    If Status <> 0 Then GoTo ErrorHandler

    '--  Ensure that the instrument isn't in repeat sweep mode
    Status = viVPrintf(DevSessionNum, "INIT:CONT 0" & Chr$(10), 0)
    If Status <> 0 Then GoTo ErrorHandler

    '--  Clear status registers
    Status = viVPrintf(DevSessionNum, "*CLS" & Chr$(10), 0)
    If Status <> 0 Then GoTo ErrorHandler

    '--  Map OPC to bit 5 of the Standard Event Status Enable Register
    Status = viVPrintf(DevSessionNum, "*ESE 1" & Chr$(10), 0)
    If Status <> 0 Then GoTo ErrorHandler

    '--  Check for Lambda Zero
    Status = CheckLambdaZero(DevSessionNum, ErrMsg)
    If Status <> 0 Then GoTo ErrorHandler

    '--  Format data as ASCII
    Status = viVPrintf(DevSessionNum, "FORM:DATA ASCII" & Chr$(10), 0)
    If Status <> 0 Then GoTo ErrorHandler

    Status = 0
    ErrMsg = ""
    Exit Function

ErrorHandler:
    If Err.Number = 0 Then

        '--  Translate error number into a message
        Call viStatusDesc(DevSessionNum, Status, ReadBuffer)

        '--  Trim off nulls
        ErrMsg = TrimVISAStr(ReadBuffer)
        InitVISASession = Status
    Else
        ErrMsg = Err.Number & vbCrLf & Err.Description
    End If
End Function

Private Function TrimVISAStr(ByVal VISAStr As String) As String
Dim Pos As Long

    On Error GoTo ErrorHandler

    Pos = (InStr(VISAStr, Chr(0))) - 1

    If Pos <> -1 Then
        TrimVISAStr = Left(VISAStr, Pos)
    Else
        TrimVISAStr = Trim(VISAStr)
    End If

    Exit Function

ErrorHandler:
    TrimVISAStr = VISAStr

End Function
```

Example 5, Zero Span

Description

Zero span allows you to monitor one wavelength while taking measurements over time. This Visual Basic program demonstrates how to put the instrument into zero span.

Program

```
Private Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)

Private Const OPC_TMO = 180
Private Const LZ_TMO = 360

Private Sub cmdRepeatSweep_Click()
Dim TimeStart As Double

'--  VISA session variables
Dim DefRM As Long                                   'VISA session
Dim DevSessionNum As Long                           'VISA session to device
Dim ReadBuffer As String * 65500                    'Buffer to read back data
Dim STB As Integer                                  'Status byte
Dim Status As Long: Status = 0
Dim ErrMsg As String: ErrMsg = ""

  On Error GoTo ErrorHandler

  MousePointer = vbHourglass

  '--  Call function to initialize a VISA session to the WDCA
  Status = InitVISASession(txtGPIB_IN.Text, txtGPIB_PA.Text, DefRM, DevSessionNum, ErrMsg)
  If Status <> 0 Or ErrMsg <> "" Then GoTo ErrorHandler

  '--  Set Center Wavelength
  Status = viVPrintf(DevSessionNum, "SENS:WAV:CENT " & Val(txtCWL.Text) & "nm" & Chr$(10), 0)
  If Status <> 0 Then GoTo ErrorHandler

  '--  Set sensitivity
  If chkSensitivityLock.Value = 0 Then
    '--  Unlock Sensitivity
    Status = viVPrintf(DevSessionNum, "SENS:POW:DC:RANG:LOW:AUTO OFF" & Chr$(10), 0)
    If Status <> 0 Then GoTo ErrorHandler

    '--  Set Sensitivity
    Status = viVPrintf(DevSessionNum, "SENS:POW:DC:RANG:LOW " & Val(txtSensitivity.Text) & _
          Chr$(10), 0)
    If Status <> 0 Then GoTo ErrorHandler

  Else
    '--  Lock Sensitivity
    Status = viVPrintf(DevSessionNum, "SENS:POW:DC:RANG:LOW:AUTO ON" & Chr$(10), 0)
    If Status <> 0 Then GoTo ErrorHandler

  End If
```

```
 '--  Trace settings
    '--  Attach trace to input
   Status = viVPrintf(DevSessionNum, "TRAC:FEED:INP " & cmbTrace.Text & "," & cmbChannel.Text & _
             Chr$(10), 0)
    If Status <> 0 Then GoTo ErrorHandler

    '--  Turn trace update on
    Status = viVPrintf(DevSessionNum, "TRAC:FEED:CONT " & cmbTrace.Text & ",ALW" & Chr$(10), 0)
    If Status <> 0 Then GoTo ErrorHandler

    '--  Make trace visible on the display
   Status = viVPrintf(DevSessionNum, "DISP:WIND:TRAC:STAT " & cmbTrace.Text & ",1" & Chr$(10), 0)
    If Status <> 0 Then GoTo ErrorHandler

 '--  Set Span
 Status = viVPrintf(DevSessionNum, "SENS:WAV:SPAN 0nm;*OPC" & Chr$(10), 0)
 If Status <> 0 Then GoTo ErrorHandler

 '--  Wait for completion

    TimeStart = Now * 86400     'Used to track elapsed time

    Do

       '--  If a minute goes by, throw a timeout
       If (Now * 86400) - TimeStart > OPC_TMO Then GoTo OPCTimeOut

       '--  Check bit 5 of the status byte register to see if the operation is complete
       Status = viReadSTB(DevSessionNum, STB)
       If Status <> 0 Then GoTo ErrorHandler

       Call Sleep(5)   'Delay so the instrument isn't tied up with responding to STB
       DoEvents

    Loop While (STB And 32) <> 32

    '--  Clear status information
    Status = viVPrintf(DevSessionNum, "*CLS" & Chr$(10), 0)
    If Status <> 0 Then GoTo ErrorHandler

 '--  Start sweeping
 Status = viVPrintf(DevSessionNum, "INIT:CONT 1" & Chr$(10), 0)
 If Status <> 0 Then GoTo ErrorHandler

EndSession:
 On Error Resume Next

 '--  Close VISA session to the WDCA
 Call viClose(DevSessionNum)

 '--  End VISA session
 Call viClose(DefRM)

 MousePointer = vbDefault

 Exit Sub
```

```
OPCTimeOut:
   MousePointer = vbDefault
   Call MsgBox("Timeout while waiting for OPC.  Make sure all cables are securely fastened.", _
               vbOKOnly, "Timeout")
   GoTo EndSession

ErrorHandler:
   If Err.Number = 0 Then

      If Status <> 0 Then
      '--  VISA error

         MousePointer = vbDefault

         '--  Translate error number into a message
         Call viStatusDesc(DevSessionNum, Status, ReadBuffer)

         '--  Trim off nulls
         ErrMsg = TrimVISAStr(ReadBuffer)

         '--  Post message and reset vars
         Call MsgBox(Status & vbCrLf & ErrMsg, vbOKOnly, "Error")

         Status = 0

      Else
      '--  Non-VISA error from another function using VISA
         Call MsgBox(ErrMsg, vbOKOnly, "Error")
      End If

      ErrMsg = ""

   Else
      Call MsgBox(Err.Number & vbCrLf & Err.Description, vbOKOnly, "Error")
   End If

   GoTo EndSession

End Sub


Private Function CheckLambdaZero(DevSessionNum As Long, Optional ErrMsg As String = "") _
              As Long
'--  This function will check to see if a TLS LambdaZero is about to occur.  If so, force it to
'--  happen to prevent timeouts and ensure data integrity.
'--
Dim TimeStart As Double

Dim STB As Integer
Dim Status As Long

   On Error GoTo ErrorHandler

   '--  Summarize Lambda Zero Pending to the status byte register (bit 7, decimal 128)
   Status = viVPrintf(DevSessionNum, "STAT:OPER:ENAB 4" & Chr$(10), 0)
   If Status <> 0 Then GoTo ErrorHandler

   '--  Summarize Lambda Zero Complete to the status byte register (bit 3, decimal 8)
   Status = viVPrintf(DevSessionNum, "STAT:QUES:ENAB 8" & Chr$(10), 0)
   If Status <> 0 Then GoTo ErrorHandler

   '--  Check to see if a Lambda Zero is pending
   Status = viReadSTB(DevSessionNum, STB)
   If Status <> 0 Then GoTo ErrorHandler

   '--  If a Lambda Zero is pending, force it to occur
   If (STB And 128) = 128 Then

      '--  Start Lambda Zero
      Status = viVPrintf(DevSessionNum, "CAL:WAV:INT" & Chr$(10), 0)
      If Status <> 0 Then GoTo ErrorHandler

      TimeStart = Now * 86400     'Used to track elapsed time

      '--  Wait for Lambda Zero to complete
      Do

         Call Sleep(5)
         DoEvents

         '--  Read STB to determine when Lambda Zero is complete
         Status = viReadSTB(DevSessionNum, STB)
         If Status <> 0 Then GoTo ErrorHandler

         '--  Check for timeout (this could take over 5 minutes)
         If ((Now * 86400) - TimeStart) > (LZ_TMO) Then GoTo ErrorHandler

      Loop While (STB And 8) <> 8

   End If

   '--  Clear the Status Byte
   Status = viVPrintf(DevSessionNum, "*CLS" & Chr$(10), 0)
   If Status <> 0 Then GoTo ErrorHandler

   '--  Summarize Lambda Zero In Progress to the status byte (bit 7, decimal 128).
   '--  This can be used later to detect if an operation timed out due to a LambdaZero
   Status = viVPrintf(DevSessionNum, "STAT:OPER:ENAB 8" & Chr$(10), 0)
   If Status <> 0 Then GoTo ErrorHandler

   ErrMsg = ""
   CheckLambdaZero = 0
   Exit Function

LZ_Timeout:
   ErrMsg = "Timeout while waiting for Lambda Zero to complete.  Make sure Output 2 is connected" & _
            " to Source Input!"
   CheckLambdaZero = 0
   Exit Function

ErrorHandler:
   If Err.Number = 0 Then
      CheckLambdaZero = Status
   Else
      ErrMsg = Err.Number & vbCrLf & Err.Description
      CheckLambdaZero = 0
   End If

End Function


Private Function InitVISASession(ByVal GPIB_IN As String, ByVal GPIB_PA As String, _
                              DefRM As Long, DevSessionNum As Long, _
                              Optional ErrMsg As String = "") As Long

'--  VISA session variables
Dim ReadBuffer As String * 65500    'Buffer to read back data
Dim Status As Long: Status = 0

   On Error GoTo ErrorHandler

   '--  Create a VISA session
   Status = viOpenDefaultRM(DefRM)
   If Status <> 0 Then GoTo ErrorHandler

   '--  Open a session to the WDCA
   Status = viOpen(DefRM, "GPIB" & GPIB_IN & "::" & GPIB_PA, 0, 0, DevSessionNum)
   If Status <> 0 Then GoTo ErrorHandler

   '--  Clear the instrument
   Status = viClear(DevSessionNum)
   If Status <> 0 Then GoTo ErrorHandler

   '--  Set write buffer to flush on access
   Status = viSetAttribute(DevSessionNum, VI_ATTR_WR_BUF_OPER_MODE, VI_FLUSH_ON_ACCESS)
   If Status <> 0 Then GoTo ErrorHandler

   '--  Set read buffer to flush on access
   Status = viSetAttribute(DevSessionNum, VI_ATTR_RD_BUF_OPER_MODE, VI_FLUSH_ON_ACCESS)
   If Status <> 0 Then GoTo ErrorHandler

   '--  Ensure that the instrument isn't in repeat sweep mode
   Status = viVPrintf(DevSessionNum, "INIT:CONT 0" & Chr$(10), 0)
   If Status <> 0 Then GoTo ErrorHandler

   '--  Clear status registers
   Status = viVPrintf(DevSessionNum, "*CLS" & Chr$(10), 0)
   If Status <> 0 Then GoTo ErrorHandler

   '--  Map OPC to bit 5 of the Standard Event Status Enable Register
   Status = viVPrintf(DevSessionNum, "*ESE 1" & Chr$(10), 0)
   If Status <> 0 Then GoTo ErrorHandler

   '--  Check for Lambda Zero
   Status = CheckLambdaZero(DevSessionNum, ErrMsg)
   If Status <> 0 Then GoTo ErrorHandler
```

```
    '--  Format data as ASCII
    Status = viVPrintf(DevSessionNum, "FORM:DATA ASCII" & Chr$(10), 0)
    If Status <> 0 Then GoTo ErrorHandler

    Status = 0
    ErrMsg = ""
    Exit Function

ErrorHandler:
    If Err.Number = 0 Then

        '--  Translate error number into a message
        Call viStatusDesc(DevSessionNum, Status, ReadBuffer)

        '--  Trim off nulls
        ErrMsg = TrimVISAStr(ReadBuffer)
        InitVISASession = Status
    Else
        ErrMsg = Err.Number & vbCrLf & Err.Description
    End If

End Function


Private Function TrimVISAStr(ByVal VISAStr As String) As String
Dim Pos As Long

    On Error GoTo ErrorHandler

    Pos = (InStr(VISAStr, Chr(0))) - 1

    If Pos <> -1 Then
        TrimVISAStr = Left(VISAStr, Pos)
    Else
        TrimVISAStr = Trim(VISAStr)
    End If

    Exit Function

ErrorHandler:
    TrimVISAStr = VISAStr

End Function
```

Example 6, Acquire Binary Trace

Description

In Visual Basic, greater performance can be achieved by downloading a trace as binary (Real32) data. This program benchmarks the advantage by downloading the trace as ASCII and Real32, then displays the time it took to accomplish each task. The work required to convert the trace from binary to numbers that Visual Basic can use is also demonstrated.

Program

```
Private Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)

Private Const OPC_TMO = 180
Private Const LZ_TMO = 360

Private Sub cmdAcquireTrace_Click()

’--  VISA session variables
Dim DefRM As Long                                   ’VISA session
Dim DevSessionNum As Long                           ’VISA session to device
Dim BinBuffer As String * 65500                     ’Buffer to read back data
Dim ReadBuffer As String * 65500
Dim AscBuffer As String                             ’Fixed length strings may not be large
                                                    ‘enough for ASCII downloads
Dim STB As Integer                                  ’Status byte
Dim Status As Long: Status = 0
Dim ErrMsg As String: ErrMsg = ""

’--  Performance timers for Binary Trace
Dim oBinTraceBench As clsStopWatch
Dim oBinCnvrtBench As clsStopWatch
Dim dBinTraceBench As Double
Dim dBinCnvrtBench As Double

’--  Performance timers for ASCII Trace
Dim oAscTraceBench As clsStopWatch
Dim oAscCnvrtBench As clsStopWatch
Dim dAscTraceBench As Double
Dim dAscCnvrtBench As Double

’--  Misc variables
Dim dBinTrace() As Double                           ’Holds binary trace values
Dim dAscTrace() As Double                           ’Holds ASCII trace values
Dim numPoints As Integer                            ’Number of points in a trace
Dim TimeStart As Double                             ’Used to track OPC serial poll timeout
Dim msgstr As String

    On Error GoTo ErrorHandler

    MousePointer = vbHourglass

    ’--  Create stopwatch objects
    Set oBinTraceBench = New clsStopWatch
Set oBinCnvrtBench = New clsStopWatch
Set oAscTraceBench = New clsStopWatch
Set oAscCnvrtBench = New clsStopWatch

’--  Call function to initialize a VISA session to the WDCA
Status = InitVISASession(txtGPIB_IN.Text, txtGPIB_PA.Text, DefRM, DevSessionNum, ErrMsg)
If Status <> 0 Or ErrMsg <> "" Then GoTo ErrorHandler

’--  Set Center Wavelength
Status = viVPrintf(DevSessionNum, "SENS:WAV:CENT " & Val(txtCWL.Text) & "nm" & Chr$(10), 0)
If Status <> 0 Then GoTo ErrorHandler

’--  Set number of points for WDCA to acquire during sweep
Status = viVPrintf(DevSessionNum, "SENS:SWE:POIN " & CInt(txtNumPoints.Text) & Chr$(10), 0)
If Status <> 0 Then GoTo ErrorHandler

’--  Set Span
Status = viVPrintf(DevSessionNum, "SENS:WAV:SPAN " & Val(txtSpan.Text) & "nm" & Chr$(10), 0)
If Status <> 0 Then GoTo ErrorHandler

’--  Set sensitivity
If chkSensitivityLock.Value = 0 Then
    ’--  Unlock Sensitivity
    Status = viVPrintf(DevSessionNum, "SENS:POW:DC:RANG:LOW:AUTO OFF" & Chr$(10), 0)
    If Status <> 0 Then GoTo ErrorHandler

    ’--  Set Sensitivity
    Status = viVPrintf(DevSessionNum, "SENS:POW:DC:RANG:LOW " & Val(txtSensitivity.Text) & _
            Chr$(10), 0)
    If Status <> 0 Then GoTo ErrorHandler

Else
    ’--  Lock Sensitivity
    Status = viVPrintf(DevSessionNum, "SENS:POW:DC:RANG:LOW:AUTO ON" & Chr$(10), 0)
    If Status <> 0 Then GoTo ErrorHandler

End If

’--  Sweep Mode
Status = viVPrintf(DevSessionNum, "SENS:SWE:SPE " & cmbSweepMode.Text & Chr$(10), 0)
If Status <> 0 Then GoTo ErrorHandler

’--  Trace settings
    ’--  Attach trace to input
   Status = viVPrintf(DevSessionNum, "TRAC:FEED:INP " & cmbTrace.Text & "," & cmbChannel.Text & _
            Chr$(10), 0)
    If Status <> 0 Then GoTo ErrorHandler

    ’--  Turn trace update on
    Status = viVPrintf(DevSessionNum, "TRAC:FEED:CONT " & cmbTrace.Text & ",ALW" & Chr$(10), 0)
    If Status <> 0 Then GoTo ErrorHandler

    ’--  Make trace visible on the display
   Status = viVPrintf(DevSessionNum, "DISP:WIND:TRAC:STAT " & cmbTrace.Text & ",1" & Chr$(10), 0)
    If Status <> 0 Then GoTo ErrorHandler
```

```
'--  Take a sweep and send OPC
Status = viVPrintf(DevSessionNum, "INIT:IMM;*OPC" & Chr$(10), 0)
If Status <> 0 Then GoTo ErrorHandler

'--  Wait for completion

   TimeStart = Now * 86400     'Used to track elapsed time

   Do

      '--  If a minute goes by, throw a timeout
      If (Now * 86400) - TimeStart > OPC_TMO Then GoTo OPCTimeOut

      '--  Check bit 5 of the status byte register to see if the operation is complete
      Status = viReadSTB(DevSessionNum, STB)
      If Status <> 0 Then GoTo ErrorHandler

      Call Sleep(5)   'Delay so the instrument isn't tied up with responding to STB
      DoEvents

   Loop While (STB And 32) <> 32

   '--  Clear status information.  NOTE:  This is required for *OPC to work again!
   Status = viVPrintf(DevSessionNum, "*CLS" & Chr$(10), 0)
   If Status <> 0 Then GoTo ErrorHandler

'----  Download results as Binary  ----

   '--  Format data as REAL32
   Status = viVPrintf(DevSessionNum, "FORM:DATA REAL,32" & Chr$(10), 0)
   If Status <> 0 Then GoTo ErrorHandler

   '--  Time the data transfer
   Call oBinTraceBench.StopWatch(swStart)

      '--  Send query to acquire Trace
      Status = viVPrintf(DevSessionNum, "TRAC? " & cmbTrace.Text & Chr$(10), 0)
      If Status <> 0 Then GoTo ErrorHandler

      '--  Read the trace from the buffer
      Status = viVScanf(DevSessionNum, "%t", BinBuffer)
      If Status <> 0 Then GoTo ErrorHandler

   Call oBinTraceBench.StopWatch(swStop, dBinTraceBench)

   '--  Time how long it takes to translate the data
   Call oBinCnvrtBench.StopWatch(swStart)

      Status = modConvert.BinASCToArray(BinBuffer, eREAL32, dBinTrace, ErrMsg)
      If Status <> 0 Then Call Err.Raise(Status, "BinASCToArray", ErrMsg)

   Call oBinCnvrtBench.StopWatch(swStop, dBinCnvrtBench)

'----  Download results as ASCII  ----

   '--  Format data as ASCII
```

```
        Status = viVPrintf(DevSessionNum, "FORM:DATA ASCII" & Chr$(10), 0)
        If Status <> 0 Then GoTo ErrorHandler

        '--  Time data transfer
        Call oAscTraceBench.StopWatch(swStart)

            '--  Make the ASCII buffer extra large for downloading a lot of trace points
            AscBuffer = Space(170000)

            '--  Send query to acquire Trace
            Status = viVPrintf(DevSessionNum, "TRAC? " & cmbTrace.Text & Chr$(10), 0)
            If Status <> 0 Then GoTo ErrorHandler

            '--  Read the trace from the buffer
            Status = viVScanf(DevSessionNum, "%t", AscBuffer)
            If Status <> 0 Then GoTo ErrorHandler

        Call oAscTraceBench.StopWatch(swStop, dAscTraceBench)

        '--  Time how long it takes to translate the data
        Call oAscCnvrtBench.StopWatch(swStart)

            '--  Send out query to get the number of sweep points
            Status = viVPrintf(DevSessionNum, "SENS:SWE:POIN?" & Chr$(10), 0)
            If Status <> 0 Then GoTo ErrorHandler

            '--  Read the number of trace points from the buffer
            Status = viVScanf(DevSessionNum, "%t", ReadBuffer)
            If Status <> 0 Then GoTo ErrorHandler

            '--  Shave off end nulls and convert to integer
            numPoints = CInt(TrimVISAStr(ReadBuffer))

            '--  Parse comma-delimited array string
            Status = modConvert.ASCToArray(AscBuffer, numPoints, dAscTrace, ErrMsg)
            If Status <> 0 Then Call Err.Raise(Status, "ASCToArray", ErrMsg)

        Call oAscCnvrtBench.StopWatch(swStop, dAscCnvrtBench)


    '--  Display benchmark results
    msgstr = "Binary Results:" & vbCrLf & _
            "Trace Xfer = " & dBinTraceBench & "ms" & vbCrLf & _
            "Trace Cnvrt = " & dBinCnvrtBench & "ms" & vbCrLf & vbCrLf & _
            "ASCII Results:" & vbCrLf & _
            "Trace Xfer = " & dAscTraceBench & "ms" & vbCrLf & _
            "Trace Cnvrt = " & dAscCnvrtBench & "ms" & vbCrLf & vbCrLf & _
            "Number of Points = " & numPoints

    Call MsgBox(msgstr, vbOKOnly, "Done")

EndSession:
    On Error Resume Next

    '--  Free objects
    Set oBinTraceBench = Nothing
    Set oBinCnvrtBench = Nothing
    Set oAscTraceBench = Nothing
    Set oAscCnvrtBench = Nothing

    '--  Close VISA session to the WDCA
    Call viClose(DevSessionNum)

    '--  End VISA session
    Call viClose(DefRM)

    MousePointer = vbDefault

    Exit Sub

OPCTimeOut:
    MousePointer = vbDefault
    Call MsgBox("Timeout while waiting for OPC.  Make sure all cables are securely fastened.", _
                vbOKOnly, "Timeout")
    GoTo EndSession

ErrorHandler:
    If Err.Number = 0 Then

        If Status <> 0 Then
        '--  VISA error

            MousePointer = vbDefault

            '--  Translate error number into a message
            Call viStatusDesc(DevSessionNum, Status, ReadBuffer)

            '--  Trim off nulls
            ErrMsg = TrimVISAStr(ReadBuffer)

            '--  Post message and reset vars
            Call MsgBox(Status & vbCrLf & ErrMsg, vbOKOnly, "Error")

            Status = 0

        Else
        '--  Non-VISA error from another function using VISA
            Call MsgBox(ErrMsg, vbOKOnly, "Error")
        End If

        ErrMsg = ""

    Else
        Call MsgBox(Err.Number & vbCrLf & Err.Description, vbOKOnly, "Error")
    End If

    GoTo EndSession

End Sub
```

```
Private Function CheckLambdaZero(DevSessionNum As Long, Optional ErrMsg As String = "") _
            As Long
'--  This function will check to see if a TLS LambdaZero is about to occur.  If so, force it to
'--  happen to prevent timeouts and ensure data integrity.
'--
Dim TimeStart As Double

Dim STB As Integer
Dim Status As Long

  On Error GoTo ErrorHandler

  '--  Summarize Lambda Zero Pending to the status byte register (bit 7, decimal 128)
  Status = viVPrintf(DevSessionNum, "STAT:OPER:ENAB 4" & Chr$(10), 0)
  If Status <> 0 Then GoTo ErrorHandler

  '--  Summarize Lambda Zero Complete to the status byte register (bit 3, decimal 8)
  Status = viVPrintf(DevSessionNum, "STAT:QUES:ENAB 8" & Chr$(10), 0)
  If Status <> 0 Then GoTo ErrorHandler

  '--  Check to see if a Lambda Zero is pending
  Status = viReadSTB(DevSessionNum, STB)
  If Status <> 0 Then GoTo ErrorHandler

  '--  If a Lambda Zero is pending, force it to occur
  If (STB And 128) = 128 Then

    '--  Start Lambda Zero
    Status = viVPrintf(DevSessionNum, "CAL:WAV:INT" & Chr$(10), 0)
    If Status <> 0 Then GoTo ErrorHandler

    TimeStart = Now * 86400     'Used to track elapsed time

    '--  Wait for Lambda Zero to complete
    Do

      Call Sleep(5)
      DoEvents

      '--  Read STB to determine when Lambda Zero is complete
      Status = viReadSTB(DevSessionNum, STB)
      If Status <> 0 Then GoTo ErrorHandler

      '--  Check for timeout (this could take over 5 minutes)
      If ((Now * 86400) - TimeStart) > (LZ_TMO) Then GoTo ErrorHandler

    Loop While (STB And 8) <> 8

  End If

  '--  Clear the Status Byte
  Status = viVPrintf(DevSessionNum, "*CLS" & Chr$(10), 0)
  If Status <> 0 Then GoTo ErrorHandler

  '--  Summarize Lambda Zero In Progress to the status byte (bit 7, decimal 128).
   '--  This can be used later to detect if an operation timed out due to a LambdaZero
```

```
    Status = viVPrintf(DevSessionNum, "STAT:OPER:ENAB 8" & Chr$(10), 0)
    If Status <> 0 Then GoTo ErrorHandler

    ErrMsg = ""
    CheckLambdaZero = 0
    Exit Function

LZ_Timeout:
    ErrMsg = "Timeout while waiting for Lambda Zero to complete.  Make sure Output 2 is connected" & _
             " to Source Input!"
    CheckLambdaZero = 0
    Exit Function

ErrorHandler:
    If Err.Number = 0 Then
        CheckLambdaZero = Status
    Else
        ErrMsg = Err.Number & vbCrLf & Err.Description
        CheckLambdaZero = 0
    End If


End Function


Private Function InitVISASession(ByVal GPIB_IN As String, ByVal GPIB_PA As String, _
                                DefRM As Long, DevSessionNum As Long, _
                                Optional ErrMsg As String = "") As Long

'--  VISA session variables
Dim ReadBuffer As String * 65500    'Buffer to read back data
Dim Status As Long: Status = 0

    On Error GoTo ErrorHandler

    '--  Create a VISA session
    Status = viOpenDefaultRM(DefRM)
    If Status <> 0 Then GoTo ErrorHandler

    '--  Open a session to the WDCA
    Status = viOpen(DefRM, "GPIB" & GPIB_IN & "::" & GPIB_PA, 0, 0, DevSessionNum)
    If Status <> 0 Then GoTo ErrorHandler

    '--  Clear the instrument
    Status = viClear(DevSessionNum)
    If Status <> 0 Then GoTo ErrorHandler

    '--  Set write buffer to flush on access
    Status = viSetAttribute(DevSessionNum, VI_ATTR_WR_BUF_OPER_MODE, VI_FLUSH_ON_ACCESS)
    If Status <> 0 Then GoTo ErrorHandler

    '--  Set read buffer to flush on access
    Status = viSetAttribute(DevSessionNum, VI_ATTR_RD_BUF_OPER_MODE, VI_FLUSH_ON_ACCESS)
    If Status <> 0 Then GoTo ErrorHandler

    '--  Ensure that the instrument isn't in repeat sweep mode
    Status = viVPrintf(DevSessionNum, "INIT:CONT 0" & Chr$(10), 0)
    If Status <> 0 Then GoTo ErrorHandler

    '--  Clear status registers
    Status = viVPrintf(DevSessionNum, "*CLS" & Chr$(10), 0)
    If Status <> 0 Then GoTo ErrorHandler

    '--  Map OPC to bit 5 of the Standard Event Status Enable Register
    Status = viVPrintf(DevSessionNum, "*ESE 1" & Chr$(10), 0)
    If Status <> 0 Then GoTo ErrorHandler

    '--  Check for Lambda Zero
    Status = CheckLambdaZero(DevSessionNum, ErrMsg)
    If Status <> 0 Then GoTo ErrorHandler

    '--  Format data as ASCII
    Status = viVPrintf(DevSessionNum, "FORM:DATA ASCII" & Chr$(10), 0)
    If Status <> 0 Then GoTo ErrorHandler

    Status = 0
    ErrMsg = ""
    Exit Function

ErrorHandler:
    If Err.Number = 0 Then

        '--  Translate error number into a message
        Call viStatusDesc(DevSessionNum, Status, ReadBuffer)

        '--  Trim off nulls
        ErrMsg = TrimVISAStr(ReadBuffer)
        InitVISASession = Status
    Else
        ErrMsg = Err.Number & vbCrLf & Err.Description
    End If
End Function


Private Function TrimVISAStr(ByVal VISAStr As String) As String
Dim Pos As Long

    On Error GoTo ErrorHandler

    Pos = (InStr(VISAStr, Chr(0))) - 1

    If Pos <> -1 Then
        TrimVISAStr = Left(VISAStr, Pos)
    Else
        TrimVISAStr = Trim(VISAStr)
    End If

    Exit Function

ErrorHandler:
    TrimVISAStr = VISAStr
```

```
End Function

Public Function ASCToArray(ByVal ASCTrace As String, ByVal numPoints As Integer, _
                    ByRef dResult() As Double, Optional ErrMsg As String = "") As Long
'--  This function will convert a comma-delimited ASCII trace to an array.
'--
Dim NxtComma As Long: NxtComma = 0
Dim PrvComma As Long: PrvComma = 0

Dim i As Integer

  On Error GoTo ErrorHandler

  '--  Size result array according to the number of points
  ReDim dResult(0 To numPoints - 1) As Double

  '--  Find the first comma
  NxtComma = InStr(ASCTrace, ",")

  If NxtComma <> 0 Then

    For i = 0 To numPoints - 1

      '--  Pull out the next value
      dResult(i) = Val(Mid(ASCTrace, PrvComma + 1, NxtComma - PrvComma - 1))

      '--  Increment string indexes
      PrvComma = NxtComma
      NxtComma = InStr(PrvComma + 1, ASCTrace, ",")

      '--  The last value wont have a comma
      If NxtComma = 0 Then NxtComma = Len(ASCTrace)

    Next i

  End If

  ErrMsg = ""
  ASCToArray = 0
  Exit Function

ErrorHandler:
  ErrMsg = Err.Description
  ASCToArray = Err.Number

End Function

Public Function BinASCToArray(ByVal BinASCTrace As String, ByVal eDataSize As enumDataSize, _
                    ByRef dResult() As Double, Optional ErrMsg As String = "") As Long
'--  The REAL format is a definite-length block of 64-bit floating-point binary numbers.  The
'--  definite-length block is defined by IEEE 488.2: a "#" character, followed by one digit
'--  (in ASCII) specifying the number of length bytes to follow, followed by the length
'--  (in ASCII), followed by length bytes of binary data.  The binary data is a sequence of
'--  4-byte (32-bit) or 8-byte (64-bit) floating point numbers.
'
Dim NumBytes As Integer: NumBytes = 4
Dim NumLengthBytes As Integer: NumLengthBytes = 4
Dim numPoints As Integer: numPoints = 1001

Dim i As Long
Dim ValStr As String
Dim StartPos As Long: StartPos = 7

Dim Status As Long: Status = 0

   On Error GoTo ErrorHandler

   '--  Find out if we have Real32 or Real64.  Real32 (4 bytes) is the suggested default
   '--  format
   If eDataSize = eREAL64 Then
      NumBytes = 8
   End If

   '--  How many length bytes are there
   NumLengthBytes = CInt(Mid(BinASCTrace, 2, 1))

   '--  Determine the number of points
   numPoints = CInt(Mid(BinASCTrace, 3, NumLengthBytes) / NumBytes)

   '--  Calculate the start position of the first value (2 for header, NumLengthBytes,
   '--  1 more for first value)
   StartPos = 3 + NumLengthBytes

   '--  Resize result array according to the number of points
   ReDim dResult(0 To numPoints - 1) As Double

   For i = 0 To numPoints - 1

      '--  Pull out the next value in the trace
      ValStr = Mid(BinASCTrace, (i * NumBytes) + StartPos, NumBytes)

      '--  Convert the binary value to a double and store it in the array
      dResult(i) = prvtBinASCToDouble(ValStr, eDataSize, Status, ErrMsg)
      If Status <> 0 Then Call Err.Raise(Status, "prvtBinASCToDouble", ErrMsg)

   Next i

   ErrMsg = ""
   BinASCToArray = 0
   Exit Function

ErrorHandler:
   ErrMsg = Err.Description
   BinASCToArray = Err.Number

End Function


Private Function prvtBinASCToDouble(ByVal BinASCVal As String, _
                        ByVal eDataSize As enumDataSize, _
                        Optional ErrNum As Long = 0, _
                        Optional ErrMsg As String = "") As Double
```

```
'--  This function will get the entire IEE-754 compliant binary number and convert it to
'--  a value that is usable to VB.
'

'--  Defaults
Dim Bias As Integer: Bias = 127
Dim ExpSize As Integer: ExpSize = 8

'--  Calculated variables
Dim BinaryStr As String
Dim SignBit As Integer: SignBit = 1     '1 or -1
Dim Exp As Double
Dim Mantissa As Double
Dim MantValLen As Integer
Dim MantStart As Integer
Dim LSOne As Integer

Dim Status As Long

Const EXP_START = 2

    On Error GoTo ErrorHandler

    '--  Data defaults to REAL32.  If REAL64, make adjustments accordingly
    If eDataSize = eREAL64 Then

        Bias = 1023
        ExpSize = 11

    End If

    '--  The binary string contains ASCII character codes.  Convert this to true binary
    BinaryStr = prvtBinASCToBinStr(BinASCVal)

    '--  Check sign bit
    If Left(BinaryStr, 1) = "1" Then SignBit = -1

    '--  Translate exponent to decimal number
    Exp = prvtBinStrToDec(Mid(BinaryStr, EXP_START, ExpSize), Status, ErrMsg)
    If Status <> 0 Then Call Err.Raise(Status, "prvtBinStrToDec", ErrMsg)

    '--  If Exp is not 0, the value is normalized
    If Exp <> 0 Then

        '--  Further Translate exponent:  2^(ExpValue - Bias)
        Exp = 2 ^ (Exp - Bias)

    End If

    '--  Find the least significant "1" in the string
    LSOne = InStrRev(BinaryStr, "1")
    MantStart = ExpSize + EXP_START

    '--  If a 1 was found, calculate the Mantissa, otherwise check for different values
    '--  according to IEEE-754 standards
    If (LSOne <> 0) And (LSOne > MantStart) Then
    '--  The mantissa size starts after the exponent and ends with the least significant 1
    MantValLen = LSOne - MantStart + 1

    If Exp <> 0 Then
    '--  Value is normalized.  Caclulate the value of the mantissa:
    '--  1 + (MantVal/(2^MantValLen))

      Mantissa = 1 + _
          (prvtBinStrToDec(Mid(BinaryStr, MantStart, MantValLen), Status, ErrMsg) / _
          (2 ^ MantValLen))

    Else
    '--  Value is denormalized.  There will not be an assumed 1 for the Mantissa.  Set Exp
    '--  to 1 so math below is still valid.  Formula:  (MantVal/(2^MantValLen))

      Mantissa = prvtBinStrToDec(Mid(BinaryStr, MantStart, MantValLen), Status, ErrMsg) _
            / (2 ^ MantValLen)
      Exp = 1

    End If

    '--  Either mantissa calculation could return an error.  Check status number.
    If Status <> 0 Then Call Err.Raise(Status, "prvtBinStrToDec", ErrMsg)

  Else

    '--  Mantissa is 0.  If exponent portion of binary string is 0, then the value is 0
    If prvtBinStrToDec(Mid(BinaryStr, EXP_START, ExpSize), Status, ErrMsg) = 0 Then
      Mantissa = 0
      Exp = 0
      SignBit = 0
    End If

  End If

  prvtBinASCToDouble = SignBit * Mantissa * Exp

  ErrMsg = ""
  ErrNum = 0
  Exit Function

ErrorHandler:
  ErrMsg = Err.Description
  ErrNum = Err.Source
  prvtBinASCToDouble = 0

End Function


Private Function prvtBinASCToBinStr(BinASC As String, Optional ErrNum As Long = 0, _
                          Optional ErrMsg As String = "") As String
'--  This function is designed to take an IEE-754 binary value and covert it to a pure
'--  binary string.
Dim TempHex As String: TempHex = ""
Dim TempBin As String: TempBin = ""
Dim NewBin As String

Dim i As Integer
Dim j As Integer

  '--  First convert to Hex
  For i = 1 To Len(BinASC)

    TempHex = Hex(Asc(Mid(BinASC, i, 1)))
    If Len(TempHex) < 2 Then TempHex = "0" & TempHex

    For j = 1 To 2

      Select Case Mid(TempHex, j, 1)
      Case "0"
        NewBin = "0000"
      Case "1"
        NewBin = "0001"
      Case "2"
        NewBin = "0010"
      Case "3"
        NewBin = "0011"
      Case "4"
        NewBin = "0100"
      Case "5"
        NewBin = "0101"
      Case "6"
        NewBin = "0110"
      Case "7"
        NewBin = "0111"
      Case "8"
        NewBin = "1000"
      Case "9"
        NewBin = "1001"
      Case "A"
        NewBin = "1010"
      Case "B"
        NewBin = "1011"
      Case "C"
        NewBin = "1100"
      Case "D"
        NewBin = "1101"
      Case "E"
        NewBin = "1110"
      Case "F"
        NewBin = "1111"
      End Select

      TempBin = TempBin & NewBin

    Next j

  Next i

  prvtBinASCToBinStr = TempBin
```

```
    ErrMsg = ""
    ErrNum = 0
    Exit Function

ErrorHandler:
    prvtBinASCToBinStr = ""
    ErrMsg = Err.Description
    ErrNum = Err.Number

End Function

Private Function prvtBinStrToDec(ByVal BinASC As String, Optional ErrNum As Long = 0, _
                                    Optional ErrMsg As String = "") As Double
'--  This function will take a binary string and convert it to a double
Dim TempVal As Double: TempVal = 0

Dim NextOne As Integer
Dim MaxLen As Integer

    On Error GoTo ErrorHandler

    '--  Find every 1 and calculate its value
    MaxLen = Len(BinASC)
    NextOne = InStr(BinASC, "1")

    '--  Make sure 0 wasn't passed in
    If NextOne <> 0 Then
        Do

            '--  Add up all the 1s (2^n)
            TempVal = TempVal + 2 ^ (MaxLen - NextOne)

            '--  Find the next 1
            NextOne = InStr(NextOne + 1, BinASC, "1")

        Loop While NextOne <> 0 '0 is returned when there are no more matches
    End If

    prvtBinStrToDec = TempVal

    ErrMsg = ""
    ErrNum = 0
    Exit Function

ErrorHandler:
    ErrMsg = Err.Description
    ErrNum = Err.Number
    prvtBinStrToDec = 0

End Function
```

Example 7, VC++ Example

Description

In VC++ a fast VISA formatting method can be used to acquire the trace. This method allows you to download the trace as binary (Real32) data and have the formatting function automatically convert and place the data into an array. This formatting method is demonstrated in the program below.

Program

```
#include "visa.h"                              //VISA library 2.2 specification
#include "visatype.h"                          //VISA type and macro declarations
#include <mmsystem.h>                          //for multimedia timer
#include <windows.h>                           //for sleep function

.
.
.

//--  VISA Declarations
ViSession DefRM;                               //Declare Resource manager
ViSession Inst;                                //Instrument session
ViStatus status = 0;
ViUInt16 STB = 0;

//--  VISA Return vars
float fData[9601];                             //9601 is the maximum number of points
short int NumPoints = 9601;
unsigned char PointsStr[5];

//--  Timer Vars (all values are in ms)
long StartTime;
long EndTime;
long ElapsedTime;

        //--  Open a session to the VISA Resource Manager
        status = viOpenDefaultRM(&DefRM);
        //Add code to check status for an error

        //--  Open a session to the instrument
        status = viOpen(DefRM,"GPIB0::23",VI_NULL,3000,&Inst);
        //Add code to check status for an error

        //--  Map OPC to bit 5 of STB
        status = viPrintf(Inst,"*ESE 1\n");
        //Add code to check status for an error

        //--  Trigger a sweep
        status = viPrintf(Inst,"INIT:IMM;*OPC\n");
        //Add code to check status for an error

        STB = 0;
        while (!(STB & 32))
        {
                status = viReadSTB(Inst,&STB);
                //Add code to check status for an error

                Sleep(10);                //Give instrument a chance to work
        }

        //--  Clear STB register.
        status = viPrintf(Inst,"*CLS\n");
        //Add code to check status for an error

        //--  Format data as REAL32 (binary)
        status = viPrintf(Inst,"FORM:DATA REAL,32\n");
        //Add code to check status for an error

        StartTime = timeGetTime();

        //--  Find out how many points are in the trace
                //--  Send query to get the number of points
                status = viPrintf(Inst,"SENS:SWE:POIN? TRA\n");
                //Add code to check status for an error

                //--  Read back the number of points
                        status = viScanf(Inst,"%t",PointsStr);
                        //Add code to check status for an error


        //--  Download the trace
                //--  Send query to download the trace
                status = viPrintf(Inst,"TRAC? TRA\n");
                //Add code to check status for an error

                //--  Read trace from buffer
                status = viScanf(Inst,"%9601zb",fData);
                //Add code to check status for an error

        //--  Calculate elapsed time
        EndTime = timeGetTime();
        ElapsedTime = (EndTime - StartTime);

        //--  Close inst session
        viClose(Inst);

        //--  Close the session to the VISA Resource Manager
        viClose(DefRM);
```

Example 8, Command Preset

Description

All the Visual Basic sample applications have a button to preset the instrument. This function returns the instrument to a known state.

Program

```
Private Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)

Private Const OPC_TMO = 180
Private Const LZ_TMO = 360

Private Sub cmdRST_Click()
Dim TimeStart As Double

'--  VISA session variables
Dim DefRM As Long                                          'VISA session
Dim DevSessionNum As Long                                  'VISA session to device
Dim ReadBuffer As String * 65500                           'Buffer to read back data
Dim STB As Integer                                         'Status byte
Dim Status As Long: Status = 0
Dim ErrMsg As String: ErrMsg = ""

  On Error GoTo ErrorHandler

  MousePointer = vbHourglass

  '--  Call function to initialize a VISA session to the WDCA
  Status = InitVISASession(txtGPIB_IN.Text, txtGPIB_PA.Text, DefRM, DevSessionNum, ErrMsg)
  If Status <> 0 Or ErrMsg <> "" Then GoTo ErrorHandler

  '--  Preset the instrument and send OPC
  Status = viVPrintf(DevSessionNum, "*RST;*OPC" & Chr$(10), 0)
  If Status <> 0 Then GoTo ErrorHandler

  '--  Wait for completion

    TimeStart = Now * 86400     'Used to track elapsed time

    Do

      '--  If a minute goes by, throw a timeout
      If (Now * 86400) - TimeStart > OPC_TMO Then GoTo OPCTimeOut

      '--  Check bit 5 of the status byte register to see if the operation is complete
      Status = viReadSTB(DevSessionNum, STB)
      If Status <> 0 Then GoTo ErrorHandler

      Call Sleep(5)   'Delay so the instrument isn't tied up with responding to STB
      DoEvents

    Loop While (STB And 32) <> 32

    '--  Clear status information
```

```
        Status = viVPrintf(DevSessionNum, "*CLS" & Chr$(10), 0)
        If Status <> 0 Then GoTo ErrorHandler

EndSession:
    On Error Resume Next

    '--  Close VISA session to the WDCA
    Call viClose(DevSessionNum)

    '--  End VISA session
    Call viClose(DefRM)

    MousePointer = vbDefault

    Exit Sub

OPCTimeOut:
    MousePointer = vbDefault
    Call MsgBox("Timeout while waiting for OPC.  Make sure all cables are securely fastened.", _
                vbOKOnly, "Timeout")
    GoTo EndSession

ErrorHandler:
    If Err.Number = 0 Then

        If Status <> 0 Then
        '--  VISA error

            MousePointer = vbDefault

            '--  Translate error number into a message
            Call viStatusDesc(DevSessionNum, Status, ReadBuffer)

            '--  Trim off nulls
            ErrMsg = TrimVISAStr(ReadBuffer)

            '--  Post message and reset vars
            Call MsgBox(Status & vbCrLf & ErrMsg, vbOKOnly, "Error")

            Status = 0

        Else
        '--  Non-VISA error from another function using VISA
            Call MsgBox(ErrMsg, vbOKOnly, "Error")
        End If

        ErrMsg = ""

    Else
        Call MsgBox(Err.Number & vbCrLf & Err.Description, vbOKOnly, "Error")
    End If

    GoTo EndSession

End Sub
```

```
Private Function InitVISASession(ByVal GPIB_IN As String, ByVal GPIB_PA As String, _
                    DefRM As Long, DevSessionNum As Long, _
                    Optional ErrMsg As String = "") As Long

'-- VISA session variables
Dim ReadBuffer As String * 65500                    'Buffer to read back data
Dim Status As Long: Status = 0

  On Error GoTo ErrorHandler

  '-- Create a VISA session
  Status = viOpenDefaultRM(DefRM)
  If Status <> 0 Then GoTo ErrorHandler

  '-- Open a session to the WDCA
  Status = viOpen(DefRM, "GPIB" & GPIB_IN & "::" & GPIB_PA, 0, 0, DevSessionNum)
  If Status <> 0 Then GoTo ErrorHandler

  '-- Clear the instrument
  Status = viClear(DevSessionNum)
  If Status <> 0 Then GoTo ErrorHandler

  '-- Set write buffer to flush on access
  Status = viSetAttribute(DevSessionNum, VI_ATTR_WR_BUF_OPER_MODE, VI_FLUSH_ON_ACCESS)
  If Status <> 0 Then GoTo ErrorHandler

  '-- Set read buffer to flush on access
  Status = viSetAttribute(DevSessionNum, VI_ATTR_RD_BUF_OPER_MODE, VI_FLUSH_ON_ACCESS)
  If Status <> 0 Then GoTo ErrorHandler

  '-- Ensure that the instrument isn't in repeat sweep mode
  Status = viVPrintf(DevSessionNum, "INIT:CONT 0" & Chr$(10), 0)
  If Status <> 0 Then GoTo ErrorHandler

  '-- Clear status registers
  Status = viVPrintf(DevSessionNum, "*CLS" & Chr$(10), 0)
  If Status <> 0 Then GoTo ErrorHandler

  '-- Map OPC to bit 5 of the Standard Event Status Enable Register
  Status = viVPrintf(DevSessionNum, "*ESE 1" & Chr$(10), 0)
  If Status <> 0 Then GoTo ErrorHandler

  '-- Check for Lambda Zero
  Status = CheckLambdaZero(DevSessionNum, ErrMsg)
  If Status <> 0 Then GoTo ErrorHandler

  '-- Format data as ASCII
  Status = viVPrintf(DevSessionNum, "FORM:DATA ASCII" & Chr$(10), 0)
  If Status <> 0 Then GoTo ErrorHandler

  Status = 0
  ErrMsg = ""
  Exit Function

ErrorHandler:
  If Err.Number = 0 Then
```

```
        '--  Translate error number into a message
        Call viStatusDesc(DevSessionNum, Status, ReadBuffer)

        '--  Trim off nulls
        ErrMsg = TrimVISAStr(ReadBuffer)
        InitVISASession = Status
    Else
        ErrMsg = Err.Number & vbCrLf & Err.Description
    End If

End Function


Private Function TrimVISAStr(ByVal VISAStr As String) As String
Dim Pos As Long

    On Error GoTo ErrorHandler

    Pos = (InStr(VISAStr, Chr(0))) - 1

    If Pos <> -1 Then
        TrimVISAStr = Left(VISAStr, Pos)
    Else
        TrimVISAStr = Trim(VISAStr)
    End If

    Exit Function

ErrorHandler:
    TrimVISAStr = VISAStr

End Function
```

Front Panel Functions to Remote Commands

This is a table of the front-panel functions of the WDCA and the corresponding remote commands.

Table 5-10. Front Panel Function to Remote Command for the Agilent86082A (1 of 6)

Front Panel Function	Remote Command
Amplitude	
Reference Level	DISPlay:WINDow:TRACe:Y:SCALe:RLEVel
Scale/Div	DISPlay:WINDow:TRACe:Y:SCALe:PDIVision
Sensitivity Auto | Man	Automatic mode command: SENSe:POWer:DC:RANGe:LOWer:AUTO Manual mode command: SENSe:POWer:DC:RANGe:LOWer
Peak to Reference Level	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MAXimum CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:SRLevel
Limit Lines	
Load Limit Lines	MMEMory:LOAD:LLINE 1|2, <file_name>, [INTernal|FLOPpy]
Limit Line Tests 1 | 2 | Both | Off	CALCulate[1|2|3|4|5|6]:LLINe: [1| 2]
Norm Ref	
Normalize Setup	
Start Wavelength	SENSe:NORMalize:WAVelength:STARt
Stop Wavelength	SENSe:NORMalize:WAVelength:STOP
Normalize Input 1	INITiate:NORMalize
Normalize Input 2	INITiate:NORMalize
Apply Input 1 Normalization	SENSe:NORMalize:ENABle
Apply Input 2 Normalization	SENSe:NORMalize:ENABle
Amplitude Setup	
Amplitude Units	UNIT:POWer
Auto Zero	CALibration:ZERO
Auto Range Enable On/Off	[SENSe:]POWer:[DC:]RANGe:AUTO
Reference Level Position	DISPlay:WINDow:TRACe:Y:SCALe:RPOSition
Amplitude Display Mode Log | Linear	DISPlay:WINDow:TRACe:Y:SCALe LINear DISPlay:WINDow:TRACe:Y:SCALe LOGarithmic
Local	GPIB “GoTo Local” Command

Table 5-10. Front Panel Function to Remote Command for the Agilent86082A (2 of 6)

Front Panel Function	Remote Command
Marker	
Active Marker 1 | 2 | 3 | 4 | Off	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:STATe CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:AOFF
Active Trace A | B | C | D | E | F	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:TRACe
Peak Search	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MAXimum
Pit Search	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MINimum
Marker to CENTER	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:SCENter
Marker Setup	
Bandwidth/Marker Interpolation On | Off	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BANDwidth:INTerpolate
BW Marker Units	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BANDwidth:READout
Normal/Delta Marker Interpolation On | Off	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:INTerpolate
Normal Marker Units	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:X:READout
Peak Excursion	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:PEXCursion:PEAK
Pit Excursion	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:PEXCursion:PIT
Marker Search Threshold Value	CALCulate[1|2|3|4|5|6]:THReshold
Use Marker Search Threshold	CALCulate[1|2|3|4|5|6]:THReshold:STATe
More Marker Functions	
Marker Search Menu	
Search Mode Peak	
Peak Search	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MAXimum
Next Peak Down ↓	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MAXimum:NEXT
Next Peak Left ←	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MAXimum:LEFT
Next Peak Right →	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MAXimum:RIGHt
Active Marker 1 | 2 | 3 | 4 | Off	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:STATe CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:AOFF
Search Mode Pit	
Pit Search	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MINimum
Next Pit Up ↑	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MINimum:NEXT
Next Pit Left ←	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MINimum:LEFT
Next Pit Right →	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MINimum:RIGHt
Active Marker 1 | 2 | 3 | 4 | Off	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:STATe CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:AOFF
Marker to REF LEVEL	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:SRLEVel
Marker BW On | Off	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BANDwidth:STATe
-0.5 dB | -3 dB | -6 dB | -10 dB | -20 dB	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth:NDB
Marker BW Off	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:STATe

Table 5-10. Front Panel Function to Remote Command for the Agilent86082A (3 of 6)

Front Panel Function	Remote Command
Notch Marker On | Off	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth:TYPE CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth:STATe
-0.5 dB | -3 dB | -6 dB | -10 dB | -20 dB	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth:NDB
Notch Marker BW Off	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth:STATe
Delta Marker On | Off	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:DELTa:STATe
Line Marker Menu	
Wavelength Line Marker 1	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:SRANge:LOWer
Wavelength Line Marker 2	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:SRANge:UPPer
Search Limit On | Off	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:SRANge:STATe
Line Markers Off	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:SRANge:OFF
Normalization Ref	
Normalize Setup	
Start Wavelength	SENSe:NORMalize:WAVelength:STARt
Stop Wavelength	SENSe:NORMalize:WAVelength:STOP
Normalize Input 1	INITiate:NORMalize
Normalize Input 2	INITiate:NORMalize
Apply Input 1 Normalization	SENSe:NORMalize:ENABle
Apply Input 2 Normalization	SENSe:NORMalize:ENABle
Preset	SYSTem:PRESet
Print	HCOPy:IMMediate
Save/Recall	
Recall Menu	
Measurement	*RCL
Recall From	
Trace	MMEMory:LOAD:TRACe
Save Menu	
File Name	
Measurement	*SAV
Save Graphics	
Graphics Format	HCOPy:DEVice:LANGuage
Trace Only	MMEMory:STORe:TRACe
Delete Menu	MMEMory:DELete
Backup/Restore Menu	
Fast Meas SAVE	
Fast Meas RECALL	

Table 5-10. Front Panel Function to Remote Command for the Agilent86082A (4 of 6)

Front Panel Function	Remote Command
Sweep	
Precision Mode	[SENSe]:SWEep:SPEed
Normal Mode	[SENSe]:SWEep:SPEed
Fast Mode	[SENSe]:SWEep:SPEed
Sweep Points	[SENSe]:SWEep:POInts
Repeat Sweep On | Off	INITiate:CONTinuous
Single Sweep	INITiate:IMMediate
System	
Help	
Show Critical Errors	SYSTem:ERRor?
Show HW Errors	SYSTem:ERRor?
Show Warnings	SYSTem:ERRor?
Show Notices	SYSTem:ERRor?
Revision	*IDN?
Set Title	DISPlay[:WINDow[1]]:TEXT:DATA
Laser State On Off	OUTPut:STATe
Printer Setup	
Printer Location	HCOPy:DESTination
Wavelength Calibration	
Perform Calibration	CALibration:WAVelength:INTernal
Move Active Area	
More System Functions	
State Info	
Display Setup	
Set Time/Date	
Remote Setup	
Service Menu	
Factory Preset	
Configure Network	
Firmware Upgrade	
Adv Service Functions	
Zero Now	CALibration:ZERO:AUTO ONCE
More Adv Service Menu	
TransZ Lock On | Off	
Traces	

Table 5-10. Front Panel Function to Remote Command for the Agilent86082A (5 of 6)

Front Panel Function	Remote Command
Active Trace A|B|C|D|E|F Update and view commands below will affect active trace	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:TRACe
Trace A...F Input 1 | 2 | Other	TRACe:FEED:INPut TRA|TRB|TRC|TRD|TRE|TRF
Update A...F On | Off	TRACe:FEED:CONTrol TRA|TRB|TRC|TRD|TRE|TRF,ALWays TRACe:FEED:CONTrol TRA|TRB|TRC|TRD|TRE|TRF,NEVer
View A...F On | Off	DISPlay:WINDow:TRACe:TRA|TRB|TRC|TRD|TRE|TRF
Hold A...F None | Max | Min	CALCulate[1|2|3|4|5|6]:MAXimum:STATe CALCulate[1|2|3|4|5|6]:MINimum:STATe
Reset Min/Max Hold	CALCulate[1|2|3|4|5|6]:MAXimum:CLEar CALCulate[1|2|3|4|5|6]:MINimum:CLEar
Averaging On | Off	CALCulate[1|2|3|4|5|6]:AVERage:COUNt CALCulate[1|2|3|4|5|6]:AVERage:STATe
Trace Math Off	CALCulate[1|2|3|4|5|6]:MATH:STATe
Default Math...	CALCulate[1|2|3|4|5|6]:MATH:EXPRession:DEFine
Lin Math C=A–B	CALCulate3:MATH:EXPRession (TRA–TRB)
Lin Math C=A+B	CALCulate3:MATH:EXPRession (TRA+TRB)
Log Math C=A–B	CALCulate3:MATH:EXPRession (TRA/TRB)
Log Math C=A+B	CALCulate3:MATH:EXPRession (TRA*TRB)
Trace C Math Off	CALCulate3:MATH;STATE
Trace F Math Off	CALCulate6:MATH;STATE
Trace Offset	CALCulate[1|2|3|4|5|6]:OFFset
Exchange Math	TRACe:EXCHange
Wavelength	
Center Wavelength	SENSe:WAVelength:CENTer
Span	SENSe:WAVelength:SPAN
Start Wavelength	SENSe:WAVelength:STARt
Stop Wavelength	SENSe:WAVelength:STOP
Peak to Center	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MAXimum CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:SCENter
Pit to Center	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MINimum CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:SCENter
Wavelength Setup	
Wavelength Units	
Wavelength Calibration User	CALibrate:WAVelength:STATe ON
Wavelength Calibration Factory	CALibrate:WAVelength:STATe OFF
Wavelength Offset	SENSe:WAVelength:OFFSet

Table 5-10. Front Panel Function to Remote Command for the Agilent86082A (6 of 6)

Front Panel Function	Remote Command
Wavelength Step Size	SENse:WAVelength:CENTer:STEP:INCRement

6

Programming Commands

Programming Commands

This chapter is the reference for all Agilent 86082A commands. In accordance with IEEE 488.2, the instrument’s commands are grouped into subsystems. Commands in each subsystem perform similar tasks. The following subsystems are provided:

Common commands
CALCulate
CALibration
DISPlay
FORMat
HCOPy
INITiate
MEMory
MMEMory
OUTPut
SENSe
STATus
SYSTem
TRACe
UNIT

Command Conventions

Table 6-1.

Convention	Description
< >	Angle brackets indicate text strings entered by the developer.
[ ]	Square brackets indicate that the keyword DEFAULT can be used instead of a value or a variable for that parameter. Refer to the actual command description for the behavior when the DEFAULT keyword is used for a parameter.
|	Indicates a choice of one element from a list.
{ }	Braces indicate a group of constants to select from. Each constant is separated by the | character.
name	Indicates the variable for which you provide a descriptive name. Any letter (Aa-Zz) followed by letters, digits (0-9) and underscore (_). Only the first 32 characters are significant.
spec_min	–infinity. The parameter *spec_min* cannot be a variable, only a constant or DEFAULT.
spec_max	+infinity. The parameter *spec_max* cannot be a variable, only a constant or DEFAULT.
from	Start wavelength or frequency of trace in nm (default) or THz.
to	Stop wavelength or frequency of trace in nm (default) or THz.
excursion	+excursion: means excursion dBs up (for example, from a pit). -excursion: means excursion dBs down (for example, from a peak).
ref_pt	The reference point to be used for a measurement keyword.

Front Panel Functions to Remote Commands

This is a table of the front-panel functions of the WDCA and the corresponding remote commands.

Table 6-2. Front Panel Function to Remote Command for the Measurement Applications (1 of 6)

Front Panel Function	Remote Command
Amplitude	
Reference Level	DISPlay:WINDow:TRACe:Y:SCALe:RLEVel
Scale/Div	DISPlay:WINDow:TRACe:Y:SCALe:PDIVision
Sensitivity Auto | Man	Automatic mode command: SENSe:POWer:DC:RANGe:LOWer:AUTO Manual mode command: SENSe:POWer:DC:RANGe:LOWer
Peak to Reference Level	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MAXimum CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:SRLevel
Limit Lines	
Load Limit Lines	MMEMory:LOAD:LLINE 1|2, <file_name>, [INTernal|FLOPpy]
Limit Line Tests 1 | 2 | Both | Off	CALCulate[1|2|3|4|5|6]:LLINe: [1| 2]:DISPlay
Norm Ref	
Normalize Setup	
Start Wavelength	SENSe:NORMalize:WAVelength:STARt
Stop Wavelength	SENSe:NORMalize:WAVelength:STOP
Normalize Input 1	INITiate:NORMalize
Normalize Input 2	INITiate:NORMalize
Apply Input 1 Normalization	SENSe:NORMalize:ENABle
Apply Input 2 Normalization	SENSe:NORMalize:ENABle
Amplitude Setup	
Amplitude Units	UNIT:POWer
Auto Zero	CALibration:ZERO
Reference Level Position	DISPlay:WINDow:TRACe:Y:SCALe:RPOS
Amplitude Display Mode Log | Linear	DISPlay:WINDow:TRACe:Y:SCALe LINear DISPlay:WINDow:TRACe:Y:SCALe LOGarithmic
Local	GPIB GoTo Local Command
Marker	

Table 6-2. Front Panel Function to Remote Command for the Measurement Applications (2 of 6)

Front Panel Function	Remote Command
Active Marker 1 | 2 | 3 | 4 | Off	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:STATe CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:AOFF
ActiveTrace A | B | C | D | E | F	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:TRACe
Peak Search	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MAXimum
Pit Search	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MINimum
Marker to CENTER	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:SCENter
Marker Setup	
Bandwidth/Marker Interpolation On | Off	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BANDwidth:INTerpolate
BW Marker Units	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BANDwidth:READout
Normal/Delta Marker Interpolation On | Off	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:INTerpolate
Normal Marker Units	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:X:READout
Peak Excursion	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:PEXCursion:PEAK
Pit Excursion	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:PEXCursion:PIT
Marker Search Threshold Value	CALCulate[1|2|3|4|5|6]:THReshold
Use Marker Search Threshold	CALCulate[1|2|3|4|5|6]:THReshold:STATe
More Marker Functions	
Marker Search Menu	
Search Mode Peak	
Peak Search	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MAXimum
Next Peak Down ↓	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MAXimum:NEXT
Next Peak Left ←	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MAXimum:LEFT
Next Peak Right →	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MAXimum:RIGHt
Active Marker 1 | 2 | 3 | 4 | Off	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:STATe CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:AOFF
Search Mode Pit	
Pit Search	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MINimum
Next Pit Up ↑	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MINimum:NEXT
Next Pit Left ←	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MINimum:LEFT
Next Pit Right →	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MINimum:RIGHt
Active Marker 1 | 2 | 3 | 4 | Off	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:STATe CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:AOFF
Marker to REF LEVEL	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:SRLEVel
Marker BW On | Off	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BANDwidth:STATe
-0.5 dB | -3 dB | -6 dB | -10 dB | -20 dB	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth:NDB
Marker BW Off	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:STATe

Table 6-2. Front Panel Function to Remote Command for the Measurement Applications (3 of 6)

Front Panel Function	Remote Command
Notch Marker On | Off	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:?BWIDth:TYPE CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth:STATe
-0.5 dB | -3 dB | -6 dB | -10 dB | -20 dB	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth:NDB
Notch Marker BW Off	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth:STATe
Delta Marker On | Off	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:DELTa:STATe
Line Marker Menu	
Wavelength Line Marker 1	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:SRANge:LOWer
Wavelength Line Marker 2	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:SRANge:UPPer
Search Limit On | Off	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:SRANge:STATe
Line Markers Off	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:SRANge:OFF
Normalization Ref	
Normalize Setup	
Start Wavelength	SENSe:NORMalize:WAVelength:STARt
Stop Wavelength	SENSe:NORMalize:WAVelength:STOP
Normalize Input 1	INITiate:NORMalize
Normalize Input 2	INITiate:NORMalize
Apply Input 1 Normalization	SENSe:NORMalize:ENABle
Apply Input 2 Normalization	SENSe:NORMalize:ENABle
Preset	SYSTem:PRESet
Print	HCOPy:IMMediate
Save/Recall	
Recall Menu	
Measurement	*RCL
Recall From	
Trace	MMEMory:LOAD:TRACe
Save Menu	
File Name	
Measurement	*SAV
Save Graphics	
Graphics Format	HCOPy:DEVice:LANGuage
Trace Only	MMEMory:STORe:TRACe
Delete Menu	MMEMory:DELete
Backup/Restore Menu	
Fast Meas SAVE	
Fast Meas RECALL	

Table 6-2. Front Panel Function to Remote Command for the Measurement Applications (4 of 6)

Front Panel Function	Remote Command
Sweep	
Precision Mode	[SENSe]:SWEep:SPEed
Normal Mode	[SENSe]:SWEep:SPEed
Fast Mode	[SENSe]:SWEep:SPEed
Sweep Points	[SENSe]:SWEep:POInts
Repeat Sweep On | Off	INITiate:CONTinuous
Single Sweep	INITiate:IMMediate
System	
Help	
Show Critical Errors	SYSTem:ERRor?
Show HW Errors	SYSTem:ERRor?
Show Warnings	SYSTem:ERRor?
Show Notices	SYSTem:ERRor?
Revision	*IDN?
Set Title	DISPlay[:WINDow[1]]:TEXT:DATA
Laser State On Off	OUTPut:STATe
Printer Setup	
Printer Location	HCOPy:DESTination
Wavelength Calibration	
Perform Calibration	CALibration:WAVelength:INTernal
Move Active Area	
More System Functions	
State Info	
Display Setup	
Set Time/Date	
Remote Setup	
Service Menu	
Factory Preset	
Configure Network	
Firmware Upgrade	
Adv Service Functions	
Zero Now	CALibration:ZERO:AUTO ONCE
More Adv Service Menu	
TransZ Lock On | Off	
Traces	

Table 6-2. Front Panel Function to Remote Command for the Measurement Applications (5 of 6)

Front Panel Function	Remote Command
Active Trace A|B|C|D|E|F Update and view commands below will affect active trace	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:TRACe
Trace A...F Input 1 | 2 | Other	
Update A...F On | Off	TRACe:FEED:CONTrol TRA|TRB|TRC|TRD|TRE|TRF,ALWays TRACe:FEED:CONTrol TRA|TRB|TRC|TRD|TRE|TRF,NEVer
View A...F On | Off	DISPlay:WINDow:TRACe:STATe
Hold A...F None | Max | Min	CALCulate[1|2|3|4|5|6]:MAXimum:STATe CALCulate[1|2|3|4|5|6]:MINimum:STATe
Reset Min/Max Hold	CALCulate[1|2|3|4|5|6]:MAXimum:CLEar CALCulate[1|2|3|4|5|6]:MINimum:CLEar
Averaging On | Off	CALCulate[1|2|3|4|5|6]:AVERage:COUNt CALCulate[1|2|3|4|5|6]:AVERage:STATe
Trace Math Off	CALCulate[1|2|3|4|5|6]:MATH:STATe
Default Math...	CALCulate[1|2|3|4|5|6]:MATH:EXPRession:DEFine
Lin Math C=A–B	CALCulate3:MATH:EXPRession (TRA–TRB)
Lin Math C=A+B	CALCulate3:MATH:EXPRession (TRA+TRB)
Log Math C=A–B	CALCulate3:MATH:EXPRession (TRA/TRB)
Log Math C=A+B	CALCulate3:MATH:EXPRession (TRA*TRB)
Trace C Math Off	CALCulate3:MATH;STATE
Trace F Math Off	CALCulate6:MATH;STATE
Trace Offset	CALCulate[1|2|3|4|5|6]:OFFset
Exchange Math	TRACe:EXCHange
Wavelength	
Center Wavelength	SENSe:WAVelength:CENTer
Span	SENSe:WAVelength:SPAN
Start Wavelength	SENSe:WAVelength:STARt
Stop Wavelength	SENSe:WAVelength:STOP
Peak to Center	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MAXimum CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:SCENter
Pit to Center	CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MINimum CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:SCENter
Wavelength Setup	
Wavelength Units	
Wavelength Calibration User	CALibrate:WAVelength:STATe ON
Wavelength Calibration Factory	CALibrate:WAVelength:STATe OFF
Wavelength Offset	SENSe:WAVelength:OFFSet

Table 6-2. Front Panel Function to Remote Command for the Measurement Applications (6 of 6)

Front Panel Function	Remote Command
Wavelength Step Size	SENse:WAVelength:CENTer:STEP:INCRement

Command Trees

Common Commands

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

:CALCulate

```
:AVERage
    :CLEar
    :COUNt <numeric_value>
    [:STATe] ON|OFF|1|0
:LLINe
    [1|2]:DISPlay [?] {ON|OFF|0|1}
    [1|2]: FAIL?
    [1|2]:STATe {ON|OFF|1|0}
    [1|2]:STATe[?]
    :ANALysis
        :STATe
:MARKer
    :AOFF
    :FUNCtion
        :BWIDth
            :TYPE PASSband|NOTCh
        :BWIDth|BANDwidth
            :INTerpolate ON|OFF|1|0
            :NDB?
            :NDB <param>
            :READout FREQuency|WAVelength|TIME
            :RESult?
            :STATe?
            [:STATe] OFF|ON|0|1
            [:TYPE] PASSband|NOTCh
            :X:CENTer?
            :X:LEFT?
            :X:REFerence?
        :X:RIGHt?
        :Y:REFerence?
    :DELTa
        :RESet
        [:STATe] ON|OFF|1|0
        :X:OFFSet?
            :FREQuency <numeric_value>
            :TIME <numeric_value>
            [:WAVelength] <numeric_value>
        :X:READout FREQuency|WAVelength|TIME
        :X:REFerence?
        :Y:OFFSet?
        :Y:REFerence?
    :PRESet
:INTerpolate ON|OFF|1|0
:MAXimum
    :LEFT
    :NEXT
    :RIGHt
:MINimum
    :LEFT
    :NEXT
    :RIGHt
:PEXCursion
    [:PEAK] <numeric_value>
    :PIT <numeric_value>
:SCENter
:SRANge
    :LOWer?
        :FREQuency <param>
        :TIME <param>
        [:WAVelength] <param>
    :UPPer?
        :FREQuency <param>
        :TIME <param>
        [:WAVelength] <param>
:SRLevel
[:STATe] ON|OFF|1|0
:TRACe TRA|TRB|TRC|TRD|TRE|TRF
:X?
    :FREQuency <numeric_value>
    :READout FREQuency|WAVelength|TIME
    :REFerence
    :TIME <numeric_value>
    [:WAVelength] <param>
:Y?
```

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

:CALibration

```
:DATE?
:STATe:OFF|ON|0|1
:WAVelength
     :DATE?
     :INTernal
     :STATe
ZERO[:AUTO] OFF|ON|0|1|ONCE
```

:DISPlay

```
[:WINDow[1]]
     :ANNotation[:ALL] OFF|ON|0|1
     :TEXT
          :CLEar
          :DATA <string> | <data_block>
     :TRACe
          :ALL[:SCALe][:AUTO]
               :MARKer OFF|ON|0|1
               :OPTimize OFF|ON|0|1
          :GRATicule:GRID[:STATe] OFF|ON|0|1
          :STATe <trace_name>, OFF|ON|0|1
```

```
                        :X[:SCALe]
                           :AUTO:SPAN <numeric_value>[M|UM|NM
                              :AUTO OFF|ON|0|1
                        :Y[SCALe]
                           :AUTO:PDIVision <numeric_value>[DB]
                        :Y1[SCALe]
                           :AUTO:PDIVision:AUTO OFF|ON|0|1
                           :LINear OFF|ON|0|1
                           :PDIVision <numeric_value>[DB]
                           :RLEVel <numeric_value>DB]
                           :RPOSition <numeric_value>

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

:HCOPy                  :DATA?
                        :DESTination "SYSTem:COMMunicate:INTernal"|"SYSTem:COMMunicate:CENTronics"
                        :DEVice:LANGuage <PCL|CGM>
                        [:IMMediate]

:INITiate               :CONTinuous OFF|ON|0|1
                        [:IMMediate]
                        :NORMalize 1|2

:MEMory                 :STATe[:EXTended]

:MMEMory                :CATalog? [INTernal|FLOPpy]
                        :DATA <file_name>, <data_block>
                        :DELete <file_name>, [INTernal|FLOPpy]
                        :LOAD
                           :CAT?
                           :DATA
                           :DELete
                           :LOAD
                              :LLINe 1|2, <file_name>, [INTernal |Floppy]
                           :TRACe <trace_name>, <file_name>, [INTernal|FLOPpy]
                        :STORe
                           :TRACe <trace_name>, <file_name>, [INTernal|FLOPpy]

:OUTPut                 :STATe OFF|ON|0|1

:SENSe                  :NORMalize
                           :ENABle 1|2; OFF|ON|0|1
                           :WAVelength:STARt <numeric_value> [M|UM|NM|PM|A|HZ|KHZ|MHZ|GHZ|THZ]
```

```
    :WAVelength:STOP <numeric_value> [M|UM|NM|PM|A|HZ|KHZ|MHZ|GHZ|THZ]
:POWer[:DC]:RANGe:LOWer
    :OFF|ON|0|1
:SWEep
    :POINts <numeric_value>
    :SPEed: PRECision|NORMal|FAST|0|1|2
[:WAVelength]
    :CENTer <numeric_value> [M|UM|NM|PM|A|HZ|KHZ|MHZ|GHZ|THZ]
        :STEP
            :AUTO OFF|ON|0|1
            [:INCRement] <numeric_value> [M|UM|NM|A]
    :OFFSet <numeric_value> [M|UM|NM|A]
    :SPAN <numeric_value> [M|UM|NM|A]
        :FULL
    :STARt <numeric_value> [M|UM|NM|PM|A|HZ|KHZ|MHZ|GHZ|THZ]
    :STOP <numeric_value> [M|UM|NM|PM|A|HZ|KHZ|MHZ|GHZ|THZ]
```

:STATus

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

:SYSTem

```
:COMM:GPIB[:SELF]:BUFFer OFF|ON|0|1
:DATE?
:DISPlay OFF|ON|0|1
:ERRor:[:NEXT]?
:HELP:HEADers
:PON[:TYPE] PRESet|LAST
:PRESet
:TIME?
:TZONe:NAME?
:VERSion?
```

:TRACe

```
[:DATA]
    :X
        :STARt? <trace_name>
        :STOP? <trace_name>
        :TIME:SSTop <trace_name>,<numeric_value>,<numeric_value>
```

```
          :TYPE? <trace_name>
          [:WAVelength]:SSTop <trace_name>,<numeric_value>,<numeric_value>
     [:Y] <trace_name>
          :RATio <trace_name>,<data_block>|<numeric_value>{,<numeric_value>}
     [:Y] [POWer]/nquery
     [:Y]? <trace_name>
          :nquery
:EXCHange <trace_1> | <trace_2>
:FEED
     :CONTrol <trace_name>, ALWays | NEVer
     :INPut TRA|TRB|TRC|TRD|TRE|TRF, 1|2
:POINts <trace_name>,<numeric_value>
:TIMestamp <trace_name>,year,month,date,hour,min,sec
```

:UNIT

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

Common Commands

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

*CLS (Clear Status)

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

Syntax

```
*CLS
```

Example

This example clears the status data structures of the WDCA.

```
10 OUTPUT 720;"*CLS"
20 END
```

*ESE (Event Status Enable)

Sets the bits in the Standard Event Enable register.

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

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

Table 6-3. Standard Event Status Enable Register Bits

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

Syntax

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

<integer> is a mask from 0 to 255.

Example

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

```
10 OUTPUT 720;"*ESE 64"
20 END
```

Query Response

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

*ESR? (Event Status Register)

Reads and clears the Standard Event Status register.

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

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

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

Syntax *ESR?

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

*IDN? (Identification Number)

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

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

The maximum length of the identification string is 50 bytes.

Syntax

*IDN?

Example

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

```
10 DIM Identify$[50] . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . !Dimension variable
20 OUTPUT 720;"*IDN?"
30 ENTER 720;Identify$
40 PRINT Identify$
50 END
```

Query Response

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

Related Key

Revision

*OPC (Operation Complete)

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

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

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

The *OPC query allows synchronization between the computer and the WDCA by using the message available (MAV) bit in the Status Byte, or by reading the output queue. Unlike the *OPC command, the *OPC query does not affect the OPC Event bit in the Standard Event Status Register.

Syntax

```
*OPC
*OPC?
```

Example

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

```
10 OUTPUT 720;":PRINT;*OPC"
20 END
```

Query Response

The *OPC? query places an ASCII character “1” in the WDCA's output queue when all pending selected device operations have finished.

*OPT? (Option)

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

Syntax

```
*OPT?
```

Example

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

```
10 DIM Options$[100]
20 OUTPUT 720;"*OPT?"
30 ENTER 720;Options$
40 PRINT Options$
50 END
```

Query Response

Returns a comma-separated list of strings that identifies the WDCA’s option configuration. A 0 indicates no options are present.

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

*RCL (Recall)

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

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

An integer, 0 through 9, specifies the save/recall register that contains the WDCA setup you want to recall.

Syntax

*RCL 0-9| “name”

Example

This example restores the WDCA to the WDCA setup stored in register 3.

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

Related Commands

*SAV. An error message appears on the WDCA display if nothing has been previously saved in the specified register.

Related Key

(Recall Menu) Measurement | Trace Data Only

*RST (Reset)

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

Executes a device reset and returns the instrument to a known state. This command is the same as the SYSTem:PRESet command.

Syntax

*RST

Related Commands

SYSTem:PRESet

*SAV (Save)

Saves instrument settings to the designated register or file; an example file name is: Output 723, “*SAV ‘data’”. An integer, 0 through 9, specifies which register to save the current WDCA setup.

Syntax

*SAV 0-9| “name”

Example

This example stores the current WDCA setup to register 3.

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

Related Commands

*RCL (Recall)

Related Key

(Save Menu) Measurement | Trace Data Only

*SRE (Service Request Enable)

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

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

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

Syntax

*SRE
*SRE?

Example

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

```
10 OUTPUT 720;"*SRE 16"
20 END
```

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

*STB? (Status Byte)

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

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

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

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

Syntax

*STB
*STB?

Example

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

```
10 OUTPUT 720;"*STB?"
20 ENTER 720;Value
30 PRINT Value
40 END
```

Query Response

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

Related Commands

*CLS

*TST? (Test)

The *TST query starts a self-test on the instrument.

The *TST? query causes the WDCA to perform a self-test, and places a response in the output queue indicating whether or not the self-test completed without any detected errors. This includes a self-test of the TLS (tunable laser source). Use the :SYSTem:ERRor command to check for errors. A zero indicates that the test passed and a non-zero indicates the self-test failed.

Disconnect Inputs First

You must disconnect all front-panel inputs before sending the *TST? query.

If a test fails, refer to the troubleshooting section of the service guide.

The Self-Test takes approximately 3 minutes to complete. When using timeouts in your program, 200 seconds duration is recommended.

Syntax

*TST?

Example

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

```
10 OUTPUT 720;"*TST?"
20 ENTER 720;Results
30 PRINT Results
40 END
```

Query Response Returns a 0 for pass; non-zero for fail.

*WAI (Wait-to-Continue)

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

The *WAI command ensures that overlapped commands are completely processed before subsequent commands, those sent after the *WAI command, are processed. *This command is not needed by the WDCA, since all commands are non-overlapped, but it is included for compatibility with existing programs that might use it.*

Syntax *WAI

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

```
10 OUTPUT 720;"SINGle;*WAI"
20 END
```

CALCulate Subsystem Commands

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

Note

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

The commands in this subsystem have the following command hierarchy:

:CALCulate

```
:AVERage
    :CLEar
    :COUNt <numeric_value>
    [:STATe] ON|OFF|1|0
:LLINe
    [1|2]:DISPlay [?]ON|OFF|1|0
    [1|2]:FAIL?
    [1|2]:STATe[?] ON|OFF|1|0
:MARKer [1|2|3|4]
    :AOFF
    :FUNCtion
        :BWIDth:TYPE PASSband|NOTCh
        :BWIDth|BANDwidth
            :INTerpolate ON|OFF|1|0
            :NDB <numeric_value>
            :READout FREQuency|WAVelength|TIME
            :RESult?
            :STATe ON|OFF|1|0
            [:TYPE] PASSband|NOTCh
            :X
                :CENTer?
                :LEFT?
                :REFerence?
                :RIGHt?
        :DELTa
            :RESet
```

```
          [:STATe] ON|OFF|1|0
              :X
                  :FREQuency <numeric_value>
                  :OFFSet?
                  :TIME <numeric_value>
                  [:WAVelength] <numeric_value>
              :X:READout FREQuency|WAVelength|TIME
              :Y
                  :OFFSet?
                  :REFerence?
          :PRESet
      :INTerpolate ON|OFF|1|0
      :MAXimum
          :LEFT
          :NEXT
          :RIGHt
      :MINimum
          :LEFT
          :NEXT
          :RIGHt
      :PEXCursion
          [:PEAK] <numeric_value>
          :PIT <numeric_value>
      :SCENter
      :SRANge
          :LOWer?
              :FREQuency <param>
              :TIME <param>
              [:WAVelength] <param>
          [:STATe] ON|OFF|1|0
          :UPPer?
              :FREQuency <param>
              :TIME <param>
              [:WAVelength] <param>
      :SRLevel
      [:STATe] ON|OFF|1|0
      :TRACe TRA|TRB|TRC|TRD|TRE|TRF
      :X?
          :FREQuency <numeric_value>
          :READout FREQuency|WAVelength|TIME
          :TIME <numeric_value>
          [:WAVelength] <param>
      :Y?
  :MATH
      [:EXPRession][:DEFine] (<expression>)
      :STATe ON|OFF|1|0
```

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

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

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

Syntax

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

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

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

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

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

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

Syntax

```
:CALCulate[1|2|3|4|5|6]:AVERage:COUNt <numeric_value>
:CALCulate[1|2|3|4|5|6]:AVERage:COUNt?
```

Related Key Averaging On|Off

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

Turns trace averaging on and off. When using this command, you must have the selected trace on. [1|2|3|4|5|6] refers to traces A, B, C, D, E, and F respectively.

If a math expression with the corresponding CALCulate subopcode is off, the SENSe:DATA is averaged. If the math expression is on, the result of the math expression is averaged.

The CALCulate[1|2|3|4|5|6]:AVERage, CALCulate[1|2|3|4|5|6]:MAXimum, and :CALCulate[1|2|3|4|5|6]:MINimum states are mutually exclusive. Each trace can have only one of these functions on at a time. Turning :CALCulate[1|2|3|4|5|6]:AVERage on will automatically turn CALCulate[1|2|3|4|5|6]:MAXimum and CALCulate[1|2|3|4|5|6]: MINimum off. Each trace can have different CALCulate blocks turned on.

Syntax

```
:CALCulate[1|2|3|4|5|6]:AVERage[:STATe] OFF|ON|0|1
:CALCulate[1|2|3|4|5|6]:AVERage[:STATe]?
```

Related Key Averaging On|Off

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

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

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

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

where:

λ_i is the wavelength of a single trace point

P_i is the power of a single trace point

P_o is total power as defined below

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

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

where:

P_i is the power of a single trace point

Syntax

```
:CALCulate[1|2|3|4|5|6]:CENTermass:STATe OFF|ON|0|1
```

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

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

Syntax

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

:CALCulate[1|2|3|4|5|6]:LLINe[1|2]:FAIL?

Returns a 0 if the limit line test passes or a 1 if the limit line test fails. A negative number is returned if the limit testing has not occurred. For example, –1 is returned if the limit test is not on or a –2 if a normalization is required.

The integer following the CALCulate mnemonic specifies the trace, with 1 corresponding to a trace A, 2 to trace B, and so on. The command defaults to trace A if not specified.

The integer following the LLINe mnemonic selects the limit line to query. The command defaults to limit line 1.

Note If you are using both limit lines, you must query the instrument twice to read the complete pass/fail status.

The query response indicates which of the selected trace fails the selected limit line

Syntax

```
:CALCulate[1|2|3|4|5|6]:LLINe[1|2]:FAIL?
```

Query Response

Indicates whether the selected trace fails the selected limit line.

Related Commands

```
:CALCulate[1|2|3|4|5|6]:LLINe[1|2]:DISPlay OFF|ON|0|1
:CALCulate[1|2|3|4|5|6]:LLINe[1|2]:STATe ON|1|OFF|0
:MMEMory:DATA (to transfer limit files over GPIB)
:MMEMory:LOAD:LLINe (to load limit definitions)
```

:CALCulate[1|2|3|4|5|6]:LLINe[1|2]:STATe ON|1|OFF|0

Turns the testing of the limit line segments on or off.

Syntax

```
:CALCulate[1|2|3|4|5|6]:LLINe[1|2]:STATe ON|1|OFF|0
:CALCulate[1|2|3|4|5|6]:LLINe[1|2]:STATe?
```

Query Response

Identifies whether the limit line is on or off.

Related Commands

```
:CALCulate[1|2|3|4|5|6]:LLINe[1|2] FAIL?
:CALCulate[1|2|3|4|5|6]:LLINe[1|2]:DISPlay OFF|ON|0|1
```

:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:AOFF

Turns off all markers and marker functions.

> **Note**
>
> Going to zero span will turn off all markers. This is because markers are referenced to a particular time or wavelength *not* a particular display position. Going out of zero span will restore the markers to the state they were in before going to zero span. Changing to or from zero span changes the fundamental units for the X-axis.
>
> If no marker number is given in the following marker commands, the command is interpreted as referring to marker number 1. For example, CALC:MARK ON is equivalent to CALC:MARK1 ON.

Syntax

```
:CALCulate[1|2|3|4|5|6]:Marker[1|2|3|4]:AOFF
```

Related Keys

Active Marker 1|2|3|4|OFF

:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth|BANDwidth:INTerpolate

Turns the bandwidth marker interpolation on or off.

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

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

Syntax

```
:CALCulate[1|2|3|4|5|6]:Marker[1|2|3|4]:FUNCtion:BWIDth|BANDwidth
  :INTerpolate OFF|ON|0|1
:CALCulate[1|2|3|4|5|6]:Marker[1|2|3|4]:FUNCtion:BWIDth
  |BANDwidth:INTerpolate?
```

Query Response Whether or not interpolation is turned on or off.

Related Key (Marker Setup) Bandwidth Interpolation On|Off

:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth |BANDwidth:NDB

Sets the desired vertical offset from the numbered marker of the bandwidth markers. Based on the type, sets the number of dB (NDB) for passband bandwidth marker or notch bandwidth marker. The parameter units are as specified in the UNIT:RATio command.

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

Syntax

```
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth
  |BANDwidth:NDB <numeric_value> Five Cases: UP,
  DOWN,MAXimum, MINimum,DEFine,VALue
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth
  |BANDwidth:NDB?
```

Related Commands

```
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth
  :TYPE PASS/NOTCh
```

Related Key (Marker BW On Off) -0.5 dB | -3 dB | -6 dB | -10 dB | -20 dB
(Notch Marker On Off) -0.5 dB | -3 dB | -6 dB | -10 dB | -20 dB

:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth |BANDwidth: READout

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

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

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

Syntax

```
:CALCulate[1|2|3|4|5|6]:Marker[1|2|3|4]:FUNCtion:BWIDth
  |BANDwidth:READout FREQuency|WAVelength|TIME
:CALCulate[1|2|3|4|5|6]:Marker[1|2|3|4]:FUNCtion:BWIDth
  |BANDwidth:READout?
```

Related Key (Marker Setup) BW Marker Units

:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth |BANDwidth:RESult?

Returns the difference in the X-axis values between the left and right bandwidth markers. The units returned are determined by the CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth|BANDwidth:X:READout state.

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

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

Syntax

```
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth
  |BANDwidth:RESult?
```

:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth |BANDwidth[:STATe] OFF|ON|0|1

Turns the bandwidth marker function on or off for a particular marker. Only one marker function is allowed to be on at a time for each individual marker. Turning on the bandwidth function for a marker will turn off any other marker function and turn on the bandwidth function. For example, turning on the bandwidth function for a marker that has the delta function on, will turn off the delta function and turn on the bandwidth function for the marker.

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

Syntax

```
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth
  |BANDwidth[:STATe] OFF|ON|0|1
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth
  |BANDwidth[:STATe]?
```

Related Commands

```
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth
  |BANDwidth:TYPE PASSband|NOTCh
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth
  |BANDwidth:TYPE
```

Related Key

Marker BW On Off

:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth|BANDwidth:TYPE PASSband|NOTCh

Selects the bandwidth marker type. The choice of passband or notch filter allows the correct bandwidth calculation to occur.

Syntax

```
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth
  |BANDidth:TYPE PASSband|NOTCh
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth
  |BANDwidth:TYPE?
```

Related Commands

```
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth
  :RESult?
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth
  |BANDwidth:X:RIGHt?
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth|
  |BANDwidth:X:LEFT?
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth
  |BANDwidth:X:CENTer?
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth
  |BANDwidth:STATe?
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth
  |BANDwidth:RESult?
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth
  |BANDwidth[:STATe]OFF|ON|0|1
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth
  |BANDwidth:NDB?
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth
  |BANDwidth:NDB <param>(Five Cases: UP, DOWN, MAX,
  MIN,DEF, Value)
```

Related Key

Marker BW On Off
Notch Marker On Off

:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth|BANDwidth:X:CENTer?

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

The units returned are determined by the :CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth|BANDwidth:X:READout state. For READout of FREQuency, the X value is returned in Hertz. For READout of WAVelength, the X value is returned in meters.

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

Syntax

```
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth
  |BANDwidth:X:CENTer?
```

:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth|BANDwidth:X:LEFT?

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

The units returned are determined by the :CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth|BANDwidth:X:READout state. For READout of FREQuency, the X value is returned in Hertz. For READout of WAVelength, the X value is returned in meters.

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

Syntax

```
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth
  |BANDwidth:X:LEFT?
```

:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth|BANDwidth:X:REFerence?

Returns the reference wavelength value for the notch marker. The reference point for the notch marker is defined as the highest power point in the trace.

Syntax

```
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth
  |BANDwidth:X:REFerence?
```

:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth|BANDwidth:X:RIGHt?

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

The units returned are determined by the :CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth|BANDwidth:X:READout state. For READout of FREQuency, the X value is returned in Hertz. For READout of WAVelength, the X value is returned in meters. For READout of WAVelength, the X value is returned in meters.

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

Syntax

```
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth
  |BANDwidth:X:RIGHt?
```

:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth|BANDwidth:Y:REFerence?

Returns the reference amplitude value for the notch bandwidth marker. The reference point for the notch bandwidth marker is defined as the highest amplitude point in the trace. This query generates a “settings conflict” error if the notch bandwidth function is off for the specific marker.

The units returned are determined by the :CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth|BANDwidth:X:READout state. For READout of FREQuency, the X value is returned in Hertz. For READout of WAVelength, the X value is returned in meters. For READout of WAVelength, the X value is returned in meters.

Syntax

```
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth
  |BANDwidth:Y:REFerence?
```

:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:DELTa:RESet

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

Syntax

```
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:DELTa
  :RESet
```

Related Commands

```
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:DELTa ON|OFF
```

:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:DELta[:STATe]

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

Individual markers can have only one marker function on at a time.

Turning the delta function for a marker on will turn any other marker function off. For example, turning the delta function on for a marker that has the bandwidth function on, will turn the bandwidth function off and turn the delta function on for the marker.

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

Read the offset: CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion

:DELTa:X:OFFSet?

Syntax

```
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:DELTa
  [:STATe]OFF|ON|0|1
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:DELTa
  [:STATe]?
```

Related Key Delta Marker On|Off

:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:DELTa:X:OFFSet?

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

The units of the value returned by the query are determined by the :CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:DELTa:X:READout state. For READout of FREQuency, the units are Hertz. For READout of WAVelength, the units are meters. For READout of TIME, the units are seconds.

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

Syntax

```
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:DELTa:X
  :OFFSet?
```

:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:DELTa:X:OFFSet:FREQuency

Allows you to set the marker offset in frequency units.

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

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

Syntax

```
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion
  :DELTa:X:OFFSet:FREQuency <numeric_value>
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion
  :DELTa:X:OFFSet?
```

:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:DELTa:X:OFFSet:TIME

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

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

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

Note: This command functions only in zero span. Use the following steps:

1. Put the instrument in 0 span.
2. Turn on the delta marker with the command:
   CALCulate[1|2|3|4|5|6]:MARKer1:FUNCtion:DELTa:STATe ON.
3. Set the time value:
   CALCulate[1|2|3|4|5|6]:MARKer1:FUNCtion:DELTa:X:OFFSet:TIME 30 ms
4. Read the offset:
   CALCulate[1|2|3|4|5|6]:MARKer1:FUNCtion:DELTa:X:OFFSet?

Syntax

```
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:DELTa:X
  :OFFSet:TIME <numeric_value>
```

:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:DELTa:X:OFFSet[:WAVelength]

Allows you to set the marker offset in wavelength units.

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

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

For example:

```
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion
  :DELTa:X:OFFSet:WAVelength 10NM when readout is
  WAVElength
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:DELTa
  :X:OFFSet:FREQuency 10THZ when readout is FREQuency
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:DELTa
  :X:OFFSet:WAVelength 1E-8M when readout is FREQuency
```

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

Syntax

```
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:DELTa
  :X:OFFSet[:WAVelength]<numeric_value>
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:DELTa
  :X:OFFSet?
```

:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:DELTa:X:READout

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

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

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

Syntax

```
:CALCulate[1|2|3|4|5|6]:Marker[1|2|3|4]:FUNCtion:DELTa
  :X:READout FREQuency|WAVelength|TIMe
:CALCulate[1|2|3|4|5|6]:Marker[1|2|3|4]:FUNCtion:DELTa
  :X:READout?
```

:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:DELTa:X: REFerence?

Returns the X-axis value of the reference marker.

The units of the returned value are determined by the :CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:DELTa:X:READout setting. For a READout of FREQuency, the return value is in Hertz. For a READout of WAVelength, the return value is in meters. For READout of TIME, the X value is returned in seconds.

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

Syntax

```
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:DELTa
  :X:REFerence?
```

:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:DELTa:Y:OFFset?

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

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

Syntax

```
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:DELTa
  :Y:OFFSet?
```

:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:DELTa:Y:REFerence?

Returns the Y-axis value of the reference marker.

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

Syntax

```
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:DELTa
  :Y:REFerence?
```

:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:PRESet

Turns off all marker functions for the specified marker. This command is provided as a convenient way to turn all marker functions off without having to check the state of each individual marker function. If the marker specified by this command is off, the marker will be turned on with all marker functions off, that is, the marker will be in the “normal” marker mode.

Syntax

```
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:PRESet
```

:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:INTerpolate

Turns the normal/delta marker interpolation on or off.

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

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

Syntax

```
:CALCulate[1|2|3|4|5|6]:Marker[1|2|3|4]:INTerpolate OFF|ON|0|1
:CALCulate[1|2|3|4|5|6]:Marker[1|2|3|4]:INTerpolate?
```

Related Key

Normal/Delta Marker Interpolation On|Off

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

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

The point does *not* have to meet the peak excursion and threshold criteria. The marker trace is determined by the :CALCulate[1|2|3|4|5|6]: MARKer[1|2|3|4]:TRACe command. If the specified marker is off, it will be turned on and placed on the highest point of the trace.

Syntax

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

Related Commands

```
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MAXimum:LEFT
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MAXimum:NEXT
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MAXimum:RIGHt
```

Related Key Peak Search

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

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

Syntax

```
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MAXimum:LEFT
```

Related Commands

```
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MAXimum
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MAXimum:NEXT
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MAXimum:RIGHt
```

Related Key Next Peak Left ←

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

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

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

Syntax

```
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MAXimum:NEXT
```

Related Commands

```
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MAXimum
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MAXimum:LEFT
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MAXimum:RIGHt
```

Related Key Next Peak Down ↓

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

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

Syntax

```
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MAXimum:RIGHt
```

Related Commands

```
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MAXimum
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MAXimum:LEFT
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MAXimum:NEXT
```

Related Key Next Peak Right →

:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MINimum

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

The point does *not* have to meet the pit excursion and threshold criteria. The marker trace is determined by the CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:TRACe command. If the specified marker is off, it will be turned on and placed on the lowest point of the trace.

Syntax

```
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MINimum
```

Related Commands

```
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MINimum:LEFT
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MINimum:NEXT
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MINimum:RIGHt
```

Related Key Pit Search

:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MINimum:LEFT

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

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

Syntax

```
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MINimum:LEFT
```

Related Commands

```
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MINimum
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MINimum:NEXT
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MINimum:RIGHt
```

Related Key Next Pit Left ←

:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MINimum:NEXT

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

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

Syntax

```
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MINimum:NEXT
```

Related Commands

```
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MINimum
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MINimum:LEFT
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MINimum:RIGHt
```

Related Key Next Pit Up ↑

:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MINimum:RIGHt

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

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

Syntax

```
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MINimum:RIGHt
```

Related Commands

```
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MINimum
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MINimum:LEFT
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MINimum:NEXT
```

Related Key Next Pit Right →

:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:PEXCursion[:PEAK]

Sets the peak excursion value for the marker search routines.

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

Syntax

```
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:PEXCursion[:PEAK]
  <numeric_value>
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:PEXCursion[:PEAK]?
```

Related Commands :CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:PEXCursion:PIT

Related Key Peak Excursion

:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:PEXCursion:PIT

Sets the pit excursion value for the marker search routines.

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

Syntax

```
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:PEXCursion:PIT
  <numeric_value>
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:PEXCursion:PIT?
```

Related Commands :CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:PEXCursion:[PEAK]

Related Key Pit Excursion

:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:SCENter

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

Syntax

```
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:SCENter
```

Related Commands

```
:CALCulate[1|2|3|4|5|6]:MARKer:[1|2|3|4]:MAXimum
:SENSe:WAVelength:CENTer
```

Related Key Marker to Center

:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:SRANge:LOWer?

Returns the lower limit for the marker search range.

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

Syntax

```
:CALCulate[1|2|3|4|5|6]:Marker[1|2|3|4]:SRANge:LOWer?
```

:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:SRANge:LOWer:FREQuency

Sets the lower limit for the marker search range.

Setting this value when :CALCulate[1|2|3|4|5|6]:Marker[1|2|3|4]:SRANge:STATe is off will automatically turn :CALCulate[1|2|3|4|5|6]:Marker[1|2|3|4]:SRANge:STATe on. The range used for the marker search range is the same range used for the trace mean range. Changing the range with this command will change ranges. Sending the command when the instrument is in a zero span will generate a “Settings conflict” error. The default units for the parameter is in Hertz.

Syntax

```
:CALCulate[1|2|3|4|5|6]:Marker[1|2|3|4]:SRANge:LOWer
  :FREQuency <param>
```

:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:SRANge:LOWer:TIME

Sets the lower limit for the marker search range.

Setting this value when CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:SRANge:STATe is off will automatically turn CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:SRANge:STATe on. The range used for the marker search range is the same range used for the trace mean range. Sending this command while span is not set to zero will results in a “Settings conflict” error. Default units for the parameter is in seconds.

Syntax

```
:CALCulate[1|2|3|4|5|6]:Marker[1|2|3|4]:SRANge:LOWer:TIME
  <param>
```

:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:SRANge:LOWer[:WAVelength]

Sets the lower limit for the marker search range.

Setting this value when :CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:SRANge:STATe is off will automatically turn :CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:SRANge:STATe on. The range used for the marker search range is the same range used for the trace mean range. Sending the command when the instrument is in a zero span will generate a "Settings conflict" error. Default units for the parameter is in meters; frequency units are allowed.

Syntax

```
:CALCulate[1|2|3|4|5|6]:Marker[1|2|3|4]:SRANge:LOWer
  [:WAVelength] <param>
```

Related Key Wavelength Marker 1

:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:SRANge:[:STATe] OFF|ON|0|1

Turns the search range on or off for all the markers. When the search range is on, all the marker maximum/minimum searches will be within the upper and lower wavelength range. Although these is a single range controlling the total power integration, the marker search range, the mean calculation, and the wavelength sweep range, there are four independent state settings for limiting the total power integration, the marker search, the mean calculation, and the wavelength sweep to the range. If all four states for the total power integration, the marker search, the mean calculation, and the wavelength sweep range are off, setting:CALCulate[1|2|3|4|5|6]:Marker[1|2|3|4]:SRANge:STATe to on will initialize the lower range to

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

and the upper range to

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

Syntax

```
:CALCulate[1|2|3|4|5|6]:Marker[1|2|3|4]:SRANge[:STATe]
  OFF|ON|0|1
```

Related Key Search Limit On Off

:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:SRANge:UPPer?

Returns the upper limit for the marker search range.

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

Syntax

```
:CALCulate[1|2|3|4|5|6]:Marker[1|2|3|4]:SRANge:UPPer?
```

Related Key Line Markers Off

:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:SRANge:UPPer:FREQuency

Sets the upper limit for the marker search range.

Setting this value when :CALCulate[1|2|3|4|5|6]:Marker[1|2|3|4]:SRANge:STATe is OFF will automatically turn :CALCulate[1|2|3|4|5|6]:Marker[1|2|3|4]:SRANge:STATe on. The range used for the marker search range is the same range used for the trace mean range. Sending the command when the instrument is in a zero span will generate a “Settings conflict” error. The default units for the parameter is in Hertz.

Syntax

```
:CALCulate[1|2|3|4|5|6]:Marker[1|2|3|4]:SRANge:UPPer
  :FREQuency <param>
```

:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:SRANge:UPPer:TIME

Sets the upper limit for the marker search range.

Setting this value when :CALCulate[1|2|3|4|5|6]:Marker[1|2|3|4]:SRANge:STATe is off will automatically turn :CALCulate[1|2|3|4|5|6]:Marker[1|2|3|4]:SRANge:STATe on. The range used for the marker search range is the same range used for the trace mean range. Sending this command while span is not set to zero will result in a “Settings conflict” error. Default units for the parameter is in seconds.

Syntax

```
:CALCulate[1|2|3|4|5|6]:Marker[1|2|3|4]:SRANge:UPPer:TIME
  <param>
```

:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:SRANge:UPPer[:WAVelength]

Sets the upper limit for the marker search range.

Setting this value when :CALCulate[1|2|3|4|5|6]:Marker[1|2|3|4]:SRANge:STATe is off will automatically turn :CALCulate[1|2|3|4|5|6]:Marker[1|2|3|4]: SRANge:STATe on. The range used for the marker search range is the same range used for the total power calculation, the trace mean range, and the wavelength sweep range. Sending the command when the instrument is in a zero span will generate a “Settings conflict” error. Default units for the parameter is in meters; frequency units are allowed.

Syntax

```
:CALCulate[1|2|3|4|5|6]:Marker[1|2|3|4]:SRANge:UPPer
  [:WAVelength] <param>
```

Related Key Wavelength Marker 2

:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:SRLevel

Sets the reference level to the amplitude of the marker.

Syntax

```
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:SRLevel
```

Related Key Marker to REF LEVEL

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

Turns a particular marker ON or OFF.

If no number is given for the MARKer node, 1 is assumed. (For example, :CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4] on will turn marker 1 on.) The marker will be placed on the trace determined by the :CALCulate[1|2|3|4|5|6]: MARKer[1|2|3|4]:TRACe command. If no trace is specified, the default trace is trace A. The marker will be placed at the center wavelength. Turning a marker off will turn off any marker function that was on for that particular marker. When the marker is turned on again, all the marker functions for that marker will be off.

Syntax

```
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4][:STATe] OFF|ON|0|1
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4][:STATe]?
```

Related Keys Active Marker 1|2|3|4|OFF

:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:TRACe

Places the marker on a particular trace.

Syntax

```
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:TRACe
  TRA|TRB|TRC|TRD|TRE|TRF
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:TRACe?
```

Related Key (Marker) ActiveTrace A|B|C|D|E|F

:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:X?

Returns the X-axis value of the normal marker.

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

The units of the value returned by the query is determined by the :CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:X:READout state. For READout of FREQuency, the units returned are in Hertz. For READout of WAVelength, the units returned are meters. For READout of TIME, the units are in seconds.

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

Syntax

```
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:X?
```

:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:X:FREQuency

Sets the X-axis value of the normal marker.

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

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

Syntax

```
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:X:FREQuency
<numeric_value>
```

:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:X:READout

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

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

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

Syntax

```
:CALCulate[1|2|3|4|5|6]:Marker[1|2|3|4]:X:READout
  :FREQuency|WAVelength|TIME
:CALCulate[1|2|3|4|5|6]:Marker[1|2|3|4]:X:READout?
```

Related Key Normal Marker Units

:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:X:TIME

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

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

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

Syntax

```
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:X:TIME
<numeric_value>
```

:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:X[:WAVelength]

Tunes the marker to the supplied value.

Syntax

```
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:X:WAVelength
  [M|UM|NM|PM|A]
```

:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:Y?

Returns the Y-axis value of the normal marker.

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

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

Syntax

```
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:Y?
```

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

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

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

Example Statements:

For the equivalent of the front panel C = A_{log} – B key:

```
OUTPUT 723 "CALCulate3:MATH:EXPRession (TRA / TRB)"
```

For the equivalent of the front panel C = A_{log} + B key:

```
OUTPUT 723 "CALCulate3:MATH:EXPRession (TRA * TRB)"
```

For the equivalent of the front panel C = A_{lin} – B key:

```
OUTPUT 723 "CALCulate3:MATH:EXPRession (TRA – TRB)"
```

For the equivalent of the front panel C = A_{lin} + B key:

```
OUTPUT 723 "CALCulate3:MATH:EXPRession (TRA + TRB)"
```

For the equivalent of the front panel F = C_{log} – D key:

```
OUTPUT 723 "CALCulate:MATH:EXPRession (TRC / TRD)"
```

Note

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

Syntax

```
:CALCulate[1|2|3|4|5|6]:MATH[:EXPRession][:DEFine]
  (<expression>)
:CALCulate[1|2|3|4|5|6]:MATH[:EXPRession][:DEFine]?
<expression> ::= <trace_name><operator><trace_name>[<opera-
tor><numeric_value>]
<operator> ::= +|–|*|/
```

Related Key

Default Math...

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

Determines whether or not math processing is done.

Syntax

```
:CALCulate[1|2|3|4|5|6]:MATH:STATe OFF|ON|0|1
:CALCulate[1|2|3|4|5|6]:MATH:STATe
```

Related Key Trace Math Off

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

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

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

Syntax

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

Related Key Reset Min/Max Hold

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

Turns maximum hold on a trace on or off.

The maximum hold operation compares the current amplitude value of each point on a trace in the current sweep to the corresponding point detected during the previous sweep, then stores the maximum value. The CALCulate[1|2|3|4|5|6]:AVERage, CALCulate[1|2|3|4|5|6]:MAXimum, and CALCulate[1|2|3|4|5|6]:MINimum states are mutually exclusive. Each trace can have only one of these functions on at a time. Turning CALCulate[1|2|3|4|5|6]: MAXimum on will automatically turn CALCulate[1|2|3|4|5|6]:AVERage and CALCulate[1|2|3|4|5|6]:MINimum off. Each trace is allowed to have a different CALCulate block turned on. For example, CALCulate1:MAXimum on, CALCulate2:MINimum on, CALCulate3:AVERage on will put trace A in maximum hold, trace B in minimum hold and trace C in trace average mode.

If the math expression with the corresponding CALCulate subopcode is off, then the SENSe:DATA is used for the maximum hold operation. If the math expression is on, the result of the math expression is used for the maximum hold operation.

Syntax

```
:CALCulate[1|2|3|4|5|6]:MAXimum[:STATe] OFF|ON|0|1
:CALCulate[1|2|3|4|5|6]:MAXimum[:STATe]?
```

Related Key Hold A None|Max|Min

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

Returns the arithmetic mean of the trace associated with the subopcode of the CALCulate node.

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

Syntax

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

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

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

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

Syntax

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

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

Sets the lower X-axis limit for the trace mean range calculation.

Setting this value when CALCulate[1|2|3|4|5|6]:MEAN:RANGe:STATe is off will automatically turn CALCulate[1|2|3|4|5|6]:MEAN:RANGe:STATe ON.

The range used for the trace mean range is the same range used for the marker search range. Changing the range with this command will change all four ranges. Default units for the parameters are Hertz.

Sending this command when the instrument is in a zero span will generate a “Settings conflict” error.

Syntax

```
:CALCulate[1|2|3|4|5|6]:MEAN:RANGe:LOWer:FREQuency
  <numeric_value>[HZ|KHZ|MHZ|GHZ|THZ]
```

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

Sets the lower X-axis limit for the trace mean range calculation.

Setting this value when CALCulate[1|2|3|4|5|6]:MEAN:RANGe:STATe is off will automatically turn CALCulate[1|2|3|4|5|6]:MEAN:RANGe:STATe on.

The range used for the trace mean range is the same range used for the marker search range. Changing the range with this command will change all four ranges. Default units for the parameters are seconds.

Sending this command while span is not set to zero will result in a “Settings conflict” error.

Syntax

```
:CALCulate[1|2|3|4|5|6]:MEAN:RANGe:LOWer:TIME
  <numeric_value>[NS|US|MS|S]
```

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

This command sets the lower X-axis limit for the trace mean range calculation.

Setting this value when CALCulate[1|2|3|4|5|6]:MEAN:RANGe:STATe is off will automatically turn CALCulate[1|2|3|4|5|6]:MEAN:RANGe:STATe ON.

The range used for the trace mean range is the same range used for the marker search range. Changing the range with this command will change all four ranges.

Sending this command when the instrument is in a zero span will generate a “Settings conflict” error. Default units for the parameter are meters. Frequency units are also allowed.

Syntax

```
:CALCulate[1|2|3|4|5|6]:MEAN:RANGe:LOWer[:WAVelength]
  <numeric_value>[M|UM|NM|PM|A]
```

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

Turns the trace mean calculation range ON or OFF for all traces.

Turning the calculation range ON will also turn the CALCulate[1|2|3|4|5|6]:MEAN:STATe ON for the specified trace (the trace is specified via its subopcode). There is a single range controlling the total power integration, the trace mean range, and the marker search range, but there are two independent state settings for limiting the trace mean calculation and the marker search.

Syntax

```
:CALCulate[1|2|3|4|5|6]:MEAN:RANGe[:STATe] OFF|ON|0|1
:CALCulate[1|2|3|4|5|6]:MEAN:RANGe[:STATe]?
```

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

Returns the upper X-axis limit for the trace mean range calculation.

The range used for the trace mean range is the same range used for the total power calculation and the marker search range.

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

Syntax

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

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

Sets the upper X-axis limit for the trace mean range calculation.

Setting this value when CALCulate[1|2|3|4|5|6]:MEAN:RANGe:STATe is off will automatically turn CALCulate[1|2|3|4|5|6]:MEAN:RANGe:STATe on. The range used for the trace mean range is the same range used for the total power calculation, and the marker search range. Changing the range with this command will change the two ranges. Default units for the parameter are Hertz.

Syntax

```
:CALCulate[1|2|3|4|5|6]:MEAN:RANGe:UPPer:FREQuency
  <numeric_value>[HZ|KHZ|MHZ|GHZ|THZ]
```

:CALCulate[1|2|3|4|5|6]:MEAN:RANGe:UPPer:TIME

Sets the upper X-axis limit for the trace mean range calculation.

Setting this value when CALCulate[1|2|3|4|5|6]:MEAN:RANGe:STATe is off will automatically turn CALCulate[1|2|3|4|5|6]:MEAN:RANGe:STATe on. The range used for the trace mean range is the same range used for the total power calculation, and the marker search range. Changing the range with this command will change the two ranges. Default units for the parameter are seconds.

Sending this command while span is not set to zero will result in a “Settings conflict” error.

Syntax

```
:CALCulate[1|2|3|4|5|6]:MEAN:RANGe:UPPer:TIME
  <numeric_value>[NS|US|MS|S]
```

:CALCulate[1|2|3|4|5|6]:MEAN:RANGe:UPPer[:WAVelength]

Sets the upper X-axis limit for the trace mean range calculation.

Setting this value when CALCulate[1|2|3|4|5|6]:MEAN:RANGe:STATe is off will automatically turn CALCulate[1|2|3|4|5|6]:MEAN:RANGe:STATe on. The range used for the trace mean range is the same range used for the total power calculation, and the marker search range. Changing the range with this command will change the two ranges. Default units for the parameter are meters.

Sending the command when the instrument is in a zero span will generate a "Settings conflict" error.

Syntax

```
:CALCulate[1|2|3|4|5|6]:MEAN:RANGe:UPPer[:WAVelength]
  <numeric_value>[M|UM|NM|PM|A]
```

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

Turns the mean power calculation for a trace ON or OFF.

Only one mean power calculation can be turned on at a time. For example, if a mean power calculation is being performed on trace A, turning a mean power calculation for trace B ON will turn the calculation for trace A OFF.

Syntax

```
:CALCulate[1|2|3|4|5|6]:MEAN:STATe OFF|ON|0|1
:CALCulate[1|2|3|4|5|6]:MEAN:STATe?
```

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

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

The trace will be initialized to the current value of the trace. If the specified trace is not in the minimum hold state, sending this command will have no effect.

Syntax

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

Related Key Reset Min/Max Hold

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

Turns minimum hold for a trace on or off.

The minimum hold operation compares the current amplitude value of each point on a trace in the current sweep to the corresponding point detected during the previous sweep, then stores the minimum value. The CALCulate[1|2|3|4|5|6]:AVERage, CALCulate[1|2|3|4|5|6]:MAXimum, and CALCulate[1|2|3|4|5|6]:MINimum states are mutually exclusive. Each trace can have only one of these functions on at a time. Turning CALCulate[1|2|3|4|5|6]: MINimum on will automatically turn CALCulate[1|2|3|4|5|6]:AVERage and CALCulate[1|2|3|4|5|6]:MAXimum off. Each trace can have a different CALCulate block turned on. For example, CALCulate1:MAXimum on, CALCulate2:MINimum ON, CALCulate3:AVERage ON will put trace A in maximum hold, trace B in minimum hold, and trace C in trace average mode.

If the math expression with the corresponding CALCulate subopcode is OFF, then the SENSe:DATA is used for the minimum hold operation. If the math expression is ON, the result of the math expression is used for the minimum hold operation.

Syntax

```
:CALCulate[1|2|3|4|5|6]:MINimum[:STATe] OFF|ON|0|1
:CALCulate[1|2|3|4|5|6]:MINimum[:STATe]?
```

Related Key Hold A None|Max|Min

:CALCulate[1|2|3|4|5|6]:OFFset <param>

Sets the trace offset for a trace.

The Trace Math menu includes a “per trace” offset for each trace. The offset for each trace can be set independently. The units are determined by the amplitude units setting. When the amplitude units are set to “auto”, the units will be in dB in log display mode and “X” (unitless ratio) in linear display mode.

Syntax

```
:CALCulate[1|2|3|4|5|6]:OFFset <param>
```

Related Key Trace A Offset

:CALCulate[1|2|3|4|5|6]:THReshold

Sets the value for the marker search threshold.

Syntax

```
:CALCulate[1|2|3|4|5|6]:THReshold <param>[W|MW|UW|DBM]
:CALCulate[1|2|3|4|5|6]:THReshold?
```

Related Key (Marker Setup) Threshold Value

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

Turns on the marker search threshold function.

When this threshold function is ON, marker peak searches will ignore peaks below the threshold value.

Syntax

```
:CALCulate[1|2|3|4|5|6]:THReshold:STATe ON|OFF|1|0
:CALCulate[1|2|3|4|5|6]:THReshold:STATe?
```

Related Key (Marker Setup) Use Marker Search Threshold

CALibration Subsystem Commands

This subsystem has the function of performing system calibration.

The commands in this subsystem have the following command hierarchy:

:CALIbration:

```
:DATE?
:STATe OFF|ON|0|1
:WAVelength
    :DATE?
    :INTernal
    :STATe ON|OFF|1|0
:ZERO[:AUTO] OFF|ON|0|1|ONCE
```

:CALibration:DATE?

Returns the date of the most recent calibration.

Syntax

```
:CALibration:DATE?
```

:CALibration:STATe OFF|ON|0|1

Specifies if the user calibration data is applied or not. Wavelength accuracy is only specified when calibration is on.

Syntax

```
:CALibration:STATe OFF|ON|0|1
:CALibration:STATe?
```

:CALibration:WAVelength:DATE?

Returns the date of the most recent wavelength calibration.

Syntax

```
:CALibration:WAVelength:DATE?
```

:CALibration:WAVelength:INTernal

Performs the wavelength calibration using the internal calibrator.

> **Note**
>
> Output 2 on the TLS front panel must be connected to the Source Input before sending this command.

Syntax

```
:CALibration:WAVelength:INTernal
```

Related Key Perform Calibration

:CALibration:WAVelength:STATe

Specifies whether or not the calibration wavelength data is applied. Wavelength accuracy is only specified with wavelength calibration on. Performs wavelength calibration and TLS settling routine to ensure absolute wavelength accuracy.

Syntax

```
:CALibrate:WAVelength:STATe ON|OFF|1|0
:CALibrate:WAVelength:STATe?
```

:CALibration:ZERO:[:AUTO]

Autozeroing measures and compensates for the dark current of the photodetector for improved amplitude accuracy. The ONCE parameter causes the dark current to be measured one time, and then the resulting correction is applied to *all* subsequent measurements. Auto Zero on causes the dark current to be measured every 60 seconds, and then the resulting correction is applied to the next sweep.

Syntax

```
:CALibration:ZERO[:AUTO] OFF|ON|0|1|ONCE
:CALibration:ZERO[:AUTO]?
```

Related Key

(Amplitude Setup) Auto Zero
Zero Now

DISPlay Subsystem Commands

The DISPlay controls the selection and presentation of textual, graphical, and TRACe information.

The commands in this subsystem have the following command hierarchy:

:DISPlay

```
[:WINDow]
[:WINDow[1]]
      :ANNotation[:ALL] OFF|ON|0|1
      :TEXT
          :CLEar
          :DATA <string> | <data_block>
      :TRACe
          :GRATicule:GRID[:STATe] OFF|ON|0|1
          :STATe <trace_name>, OFF|ON|0|1
          :X[:SCALe]
          :Y[SCALe]
               :LINear OFF|ON|0|1
          :Y1|Y2[SCALe]
               :PDIVision <numeric_value>[DB]
               :RLEVel <numeric_value>[DB]
               :RPOSition <numeric_value>
```

:DISPlay[:WINDow[1]]

Turns the display of the interface on or off.

Disabling the display speeds up the remote commands because the graphics no longer need to be updated. This is useful for manufacturing applications that use a PC to display test results.

The display is enabled by sending the DISPlay:WINDow:ON command. Turning the display on or off does not affect the instrument state. Note that there is not a front panel function for turning off the display.

Syntax

```
:DISPlay[:WINDow[1]] OFF|ON|0|1
```

:DISPlay[:WINDow[1]]:ANNotation[:ALL]

Turns the screen annotation on or off. This command affects only the X-axis and Y-axis labeling and labeling within the graticule.

Syntax

```
:DISPlay[:WINDow[1]]:ANNotation[:ALL] ON|OFF|0|1
:DISPlay[:WINDow[1]]:ANNotation[:ALL]?
```

:DISPlay[:WINDow[1]]:TEXT:CLEar

Erases all text on the display resulting from previous use of the DISPlay[:WINDow[1]]:TEXT:DATA command.

Syntax

```
:DISPlay[:WINDow[1]]:TEXT:CLEar
```

:DISPlay[:WINDow[1]]:TEXT:DATA

Writes text on the display in the Title area. Uses the <data_block> parameter to send extended ASCII characters such as control codes and symbols.

Syntax

```
:DISPlay[:WINDow[1]]:TEXT:DATA <string>|<data_block>
:DISPlay[:WINDow[1]]:TEXT:DATA?
```

Related Key Set Title

:DISPlay[:WINDow[1]]:TRACe:GRATicule:GRID[:STATe]

Turns the graticule on or off.

Syntax

```
:DISPlay[:WINDow[1]]:TRACe:GRATicule:GRID[:STATe]
  OFF|ON|0|1
:DISPlay[:WINDow[1]]:TRACe:GRATicule:GRID[:STATe]?
```

:DISPlay[:WINDow[1]]:TRACe[:STATe]:

Turns the trace display on or off. Specifying any trace other than the ones listed will generate an “Illegal parameter value” error.

Syntax

```
:DISPlay[:WINDow[1]]:TRACe[:STATe]
  TRA|TRB|TRC|TRD|TRE|TRF,OFF|ON|0|1
:DISPlay[:WINDow[1]]:TRACe[:STATe]?
  TRA|TRB|TRC|TRD|TRE|TRF
```

:DISPlay[:WINDow[1]]:TRACe:Y:SCALe LINear

Specifies whether the vertical scale is in linear units or in log units.

Syntax

```
:DISPlay[:WINDow[1]]:TRACe:Y:SCALe:LINear OFF|ON|0|1
:DISPlay[:WINDow[1]]:TRACe:Y:SCALe:LINear?
```

Related Key (Amplitude Setup) Amp Dsply Mode Log|Lin

:DISPlay[:WINDow[1]]:TRACe:Y1|Y2[:SCALe]:PDIVision

Specifies the dB per division of the vertical scale. Y1 refers to the dBm (power) scale, and Y2 refers to the dB (ratio) scale. Y and Y1 work for the dBm axis; Y2 works for the dB axis. To change the axis, normalization has to be either on Y2 or on Y1.

> The maximum value for the vertical scale is 20 dB per division for the power scale or the ratio scale. The minimum value is 0.01 dB per division. The Preset value is 10 dB per division.

Syntax

```
:DISPlay[:WINDow[1]]:TRACe:Y1|Y2[:SCALe]:PDIVision
  <numeric_value>[DB]
:DISPlay[:WINDow[1]]:TRACe:Y1|Y2[:SCALe]:PDIVision?
```

Related Key Scale/Div

:DISPlay[:WINDow[1]]:TRACe:Y1|Y2[:SCALe]:RLEVel

Specifies the power value of the reference level. Default units are set by the UNIT:POWer command for Y1 and the UNIT:RATio command for Y2. Y1 refers to the dBm (power) scale, and Y2 refers to the dB (ratio) scale.

Note: The units sent must match the specified Y axis, for example, dBm for Y1 and dB for Y2.

The maximum value for the power reference level is the equivalent of +300 dBm. The minimum value is –300 dBm. The Preset value for the power reference level is instrument dependent.
The maximum value for the ratio reference level is 270 dB. The minimum value is –330 dB. The Preset value for the ratio reference level is 0 dB.

Syntax

```
:DISPlay[:WINDow[1]]:TRACe:Y1|Y2[:SCALe]:RLEVel
  <numeric_value>[DB]
:DISPlay[:WINDow[1]]:TRACe:Y1|Y2[:SCALe]:RLEVel?
```

Related Key Reference Level

:DISPlay[:WINDow[1]]:TRACe:Y1|Y2[:SCALe]:RPOSition

Selects the position at which the reference level is displayed. The top and bottom graticule lines correspond to 10 and 0, respectively. The default is 9. Y1 refers to the dBm (power) scale, and Y2 refers to the dB scale. To change the axis, normalization has to be either on for Y2 or on Y1.

Syntax

```
:DISPlay[:WINDow[1]]:TRACe:Y1|Y2[:SCALe]:RPOSition
  <numeric value>
:DISPlay[:WINDow[1]]:TRACe:Y1|Y2[:SCALe]:RPOSition?
```

Related Key (Amplitude Level Position) Reference Level Position

FORMat Subsystem Commands

The FORMat subsystem sets a data format for transferring numeric and array information

The commands in this subsystem have the following command hierarchy:

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

:FORMat[:DATA]

Specifies the trace data format used during data transfer via GPIB.

This command affects data transfers for the TRACe[:DATA] subsystem. The ASCII format is a comma-separated list of numbers. The REAL format is a definite-length block of 32- or 64-bit floating-point binary numbers. The definite-length block is defined by IEEE 488.2: a "#" character, followed by one digit (in ASCII) specifying the number of length bytes to follow, followed by the length (in ASCII), followed by length bytes of binary data. The binary data is a sequence of 8-byte (32- or 64-bit) floating point numbers.

Syntax

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

HCOPy Subsystem Commands

The HCOPy subsystem controls the setup of printing to an external device.

The commands in this subsystem have the following command hierarchy:

:HCOPy

```
:DATA?
:DESTination “SYSTem:COMMunicate:INTernal”|”SYSTem:COMMunicate:CENTronics”
:DEVice
    :LANGuage <PCL|CGM>
:IMMediate
```

:HCOPy:DATA?

Returns the currently defined printer output as an indefinite length block. After removing the #0 prefix and newline suffix, this block can be saved by the controller and sent directly to a suitable printer.

Syntax

```
:HCOPy:DATA?
```

:HCOPy:DESTination

Selects the I/O port for hard copy output. This affects subsequent presses of the Print key and the HCOPy[:IMMediate] command.

Syntax

```
:HCOPy:DESTination “SYSTem:COMMunicate:INTernal”
  |”SYSTem:COMM:CENTronics”
:HCOPy:DESTination?
```

Related Key Printer Location

:HCOPy:DEVice:LANGuage

Sets the plot print format to either PCL (Printer Command Language) or CGM (Computer Graphics Metafile) mode.

Syntax

```
:HCOPy:DEVice:LANGuage <PCL|CGM>
:HCOPy:DEVice:LANGuage?
```

Query Response The query identifies whether the plot print format is in PCL or CGM mode.

Related Key (Save Menu) Graphic Format

:HCOPy:IMMediate

Prints out the test results to the port defined by the :HCOPy:DESTination command.

Syntax

```
:HCOPy:IMMediate
```

Related Key Print

INITiate Subsystem Commands

The commands in this subsystem have the following command hierarchy:

:INITiate

```
:CONTinuous OFF|ON|0|1
[:IMMediate]
:NORMalize 1|2
```

:INITiate:CONTinuous

Specifies repeat sweep.

Note

For real time alignment, continuous sweep should be turned on. Note that single sweep is the default instrument setting.

Syntax

```
:INITiate:CONTinuous OFF|ON|0|1
:INITiate:CONTinuous?
```

Query Response

Returns state of continuous sweep setting.

Related Key

Repeat Sweep On|Off

:INITiate[:IMMediate]

Takes a sweep with the current WDCA settings and displays a trace.

Syntax

```
:INITiate[:IMMediate]
```

Related Commands

```
:INITiate:CONTinuous OFF|ON|0|1
```

Related Key

Single Sweep

:INITiate:NORMalize

Initiates a normalization trace on the selected input.

After a through cable is connected, use this command to perform a normalization of the selected input.

Note

Normalization data is valid only for the sweep mode used during normalization. For example, if Normal Sweep Mode is the active sweep mode during normalization, the normalized data will be valid only for normal sweep mode.

Syntax

```
:INITiate:NORMalize 1|2
```

Related Commands

```
[:SENSe]:NORMalize:WAVelength:STARt [NM|PM]
[:SENSe]:NORMalize:WAVelength:STOP [NM|PM]
[:SENSe]:NORMalize:ENABle 1|2, ON|OFF|0|1
```

Related Key

Normalize Input 1
Normalize Input 2

MEMory Subsystem Commands

The purpose of the MEMory subsystem is to manage instrument memory and specifically excludes memory used for mass storage, which is defined in the MMEMory subsystem.

The commands in this subsystem have the following command hierarchy:

:MEMory :STATe[:EXTended]

:MEMory:STATe[:EXTended]?

Returns the extended state information as an indefinite length block.

Syntax

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

MMEMory Subsystem Commands

The MMEMory subsystem provides mass storage capabilities for the instrument.

The commands in this subsystem have the following command hierarchy:

:MMEMory

```
:CATalog? [INTernal|FLOPpy]
:DATA <file_name>, <data_block>
:DELete <file_name>, [INTernal|FLOPpy]
:LOAD
    :LLINe 1|2, <file_name>, [INTernal|FLOPpy]
    :TRACe <trace_name>, <file_name>, [INTernal|FLOPpy]
:STORe
    :TRACe <trace_name>, <file_name>, [INTernal|FLOPpy]
```

:MMEMory:CATalog?

The Catalog? query returns information on the current contents and state of the mass storage media.

Lists all files in the current directory. The return data will be formatted as:

<mem_used>,<mem_free> {,<file listing>}

Syntax

```
:MMEMory:CATalog? [INTernal|FLOPpy]
```

:MMEMory:DATA

Stores <data_block> in the memory location <file_name>.

The query response is the <data_block> stored in <file_name>, where <data_block> is an indefinite block.

Syntax

```
MMEMory:DATA <file_name>,<data_block>
MMEMory:DATA? <file_name>
```

Example

MMEMory:DATA ‘UPPER.LIMit’, #0 <limit file contents>

:MMEMory:DELete

Deletes the specified file. The file name extension must be specified as shown in the examples.

The DELete command removes a file from the specified mass storage device. The <file_name> parameter specifies the file_name to be removed.

Syntax

```
:MMEMory:DELete <file_name>, INTernal|FLOPpy
```

Example

For Traces: MMEMory:DELete “CBG.csv”, FLOPpy
For State (*SAVe): MMEMory:DELete “CBG.dat”, FLOPpy

Related Key

Delete Menu

:MMEMory:LOAD:LLINe 1|2, <file_name>, [INTernal|FLOPpy]

Loads a limit line file. The file has to conform to the limit line file format. Refer to "Creating a Limit Line to Establish Pass/Fail Criteria" on page 3-36. The file can be stored on floppy or internally using MMEMory:LOAD GPIB command.

Syntax

```
:MMEMory:LOAD:LLINe 1|2, <file_name>, [INTernal|FLOPpy]
```

Related Key

Load Limit Line 1
Load Limit Line 2

:MMEMory:LOAD:TRACe

Loads the specified trace from mass storage.

The :LOAD command transfers data from mass memory storage to the instruments internal memory.

Syntax

```
:MMEMory:LOAD:TRACe TRA|TRB|TRC|TRD|TRE|TRF,
  <file_name>, [INTernal|FLOPpy]
```

Related Key

(Recall Menu) Trace

:MMEMory:STORe:TRACe

Stores the specified trace to mass storage.

The :STORe command transfers data from the instruments internal storage to the mass memory device.

Syntax

```
:MMEMory:STORe:TRACe TRA|TRB|TRC|TRD|TRE|TRF,
  <file_name>, INTernal|FLOPpy
```

Related Key

(Save Menu) Trace Only

OUTPut Subsystem Commands

The output subsystem controls the characteristics of the source's output port. The commands in this subsystem have the following command hierarchy:

:OUTPut	:STATe

:OUTPut:STATe

Sets or returns the current state of the laser source.

Syntax

```
:OUTPut:STATe OFF|ON|0|1
:OUTPut:STATe?
```

Related Key Laser State On/Off

SENSe Subsystem Commands

The SENSe subsystem deals with controls that directly affect device-specific settings and not those related to the signal-oriented characteristics.

The commands in this subsystem have the following command hierarchy:

:SENSe

```
:NORMalize
    :ENABle 1|2; OFF|ON|0|1
    :WAVelength:STARt [NM|PM]
    :WAVelength:STOP [NM|PM]
:POWer[:DC]:RANGe:LOWer
    :AUTO OFF|ON|0|1
    :OFF|ON|0|1
:SWEep
    :POINts <numeric_value>
    :SPEed [PRECison|NORMal|FAST|0|1|2]
[:WAVelength]
    :CENTer <numeric_value> [M|UM|NM|PM|A|HZ|KHZ|MHZ|GHZ|THZ]
        :STEP
            :AUTO OFF|ON|0|1
            [:INCRement] <numeric_value> [M|UM|NM|A]
    :LIMit
    :OFFSet <numeric_value> [M|UM|NM|A]
    :SPAN <numeric_value> [M|UM|NM|A]
        :FULL
    :SRANge
        :LOWer <numeric_value> [M|UM|NM|PM|A|HZ|KHZ|MHZ|GHZ|THZ]
        :UPPer <numeric_value> [M|UM|NM|PM|A|HZ|KHZ|MHZ|GHZ|THZ]
    :STARt <numeric_value> [M|UM|NM|PM|A|HZ|KHZ|MHZ|GHZ|THZ]
    :STOP <numeric_value> [M|UM|NM|PM|A|HZ|KHZ|MHZ|GHZ|THZ]
:NORMalize
```

[:SENSe]:NORMalize:ENABle 1|2, OFF|ON|0|1

Turns on normalization for a channel.

Syntax

```
[:SENSe]:NORMalize:ENABle 1|2; OFF|ON|0|1
```

Related Commands

```
[:SENSe]:NORMalize:ENABle? 1|2
```

Related Key

Apply Input 1 Normalization
Apply Input 2 Normalization

[:SENSe]:NORMalize:WAVelength:STARt

Sets the start normalization wavelength.

Syntax

```
[:SENSe]:NORMalize:WAVelength:STARt <numeric_value>
[NM|PM]
```

Related Commands

```
:INITialize:NORMalize 1|2
[:SENSe]:NORMalize:WAVelength:STOP [NM|PM]
```

Related Key

(Normalize Setup) Start Wavelength

[:SENSe]:NORMalize:WAVelength:STOP

Sets the stop normalization wavelength.

Syntax

```
[:SENSe]:NORMalize:WAVelength:STOP [NM|PM]
```

Related Commands

```
:INITialize:NORMalize 1|2
[:SENSe]:NORMalize:WAVelength:STARt [NM|PM]
```

Related Key

(Normalize Setup) Stop Wavelength

[:SENSe]:POWer[:DC]:RANGe:AUTO ON|OFF|0|1

Disables or enables autoranging.

Syntax

```
[:SENSe]:POWer[:DC]:RANGe:AUTO ON|OFF|0|1
[:SENSe]:POWer[:DC]:RANGe?
```

Related Key Sensitivity Auto Man

[:SENSe]:POWer[:DC]:RANGe:LOWer

Sets the measurement range in dB.

Syntax

```
[:SENSe]:POWer[:DC]:RANGe:LOWer
[:SENSe]:POWer[:DC]:RANGe:LOWer?
```

Related Key Sensitivity Auto Man

[:SENSe]:POWer[:DC]:RANGe:LOWer:AUTO ON|OFF|0|1

Turns autosensitivity on or off. Units are in dB.

The sensitivity setting requests the lowest amplitude signal that can be measured relative to the power at “top of screen”. It is defined as the signal that is six times the RMS noise. The range of settings is between –94 dB and 300 dB. An error will be reported for values outside of this range and the sensitivity will round to the nearest valid sensitivity.

Auto sensitivity uses one of six transimpedance amplifier stages. The stage is selected based on the amplitude value at the top of the display. At least 40 dB of dynamic range can be seen with one stage.

Manual sensitivity allows you to specify the lowest signal you want to measure relative to the minimum value of the normalization. If the trace has not been normalized, then sensitivity is relative to the current output power of the TLS. Manual sensitivity can be adjusted from 0 to – 95 dB. Up to three of the six transimpedance amplifies stages are used depending the selected sensitivity level and the amplitude of the trace response. The WDCA takes one sweep for each transimpedance amplifier stage. Therefore, if three sweep stages are used, the actual sweep time may be three times the indicated sweep time (ST) as shown at the bottom of the display.

Syntax

```
[:SENSe]:POWer[:DC]:RANGe:LOWer:AUTO ON|OFF|0|1
[:SENSe]:POWer[:DC]:RANGe:LOWer?
```

Related Key Sensitivity Auto Man

[:SENSe]:SWEep:POINts

Sets the number of the data points acquired during a sweep. This command is used in conjunction with the TRACe:POINts command when downloading a trace. The minimum number of data points is three and the maximum is 9601.

Syntax

```
[:SENSe]:SWEep:POINts <numeric_value>
[:SENSe]:SWEep:POINts?
```

Related Key Sweep Points

[:SENSe]:SWEep:SPEed PRECision|NORMal|FAST|0|1|2

Sets the sweep speed of the tunable laser source. The actual sweep speed of the TLS will depend on the sensitivity setting.

Precision mode offers the most accurate sweep setting to use if you are concerned about wavelength and amplitude accuracy for final filter test applications. However, the increased sweep accuracy also increases the sweep time.

Normal mode provides excellent amplitude and wavelength accuracy. At a 2 nm span, two sweeps per second can be achieved.

Fast mode allows wider spans with faster sweep speed. In this mode you can measure filters with multiple changes or see if the rejection band holds over the wide ranges. Fast sweep mode may not be appropriate to use if the filter has very sharp edges.

Syntax

```
[:SENSe]:SWEep:SPEed PRECision|NORMal|FAST|0|1|2
```

Related Key

Precision Sweep Mode
Normal Sweep Mode
Fast Sweep Mode

[:SENSe][:WAVelength]:CENTer

Specifies the center wavelength.

The start and stop wavelength and, if necessary, the span are adjusted so that:

$$Center = Start + \left(\frac{Span}{2}\right) \quad \text{and} \quad Center = Stop - \left(\frac{Span}{2}\right)$$

Syntax

```
[:SENSe][:WAVelength]:CENTer <numeric_value>
  [M|UM|NM|PM|A|HZ|KHZ|MHZ|GHZ|THZ]
[:SENSe][:WAVelength]:CENTer?
```

Related Key

Center WL

[:SENSe][:WAVelength]:CENTer:STEP:AUTO

When on, the step size is automatic. When off, the step size is fixed. The value is set by the [SENSe:][WAVelength:]CENTer:STEP[:INCRement] command.

Syntax

```
[:SENSe][:WAVelength]:CENTer:STEP:AUTO OFF|ON|0|1
[:SENSe][:WAVelength]:CENTer:STEP:AUTO?
```

[:SENSe][:WAVelength]:CENTer:STEP[:INCRement]

Specifies the center wavelength step size.

Syntax

```
[:SENSe][:WAVelength]:CENTer:STEP[:INCRement]
  <numeric_value>[M|UM|NM|A]
[:SENSe][:WAVelength]:CENTer:STEP[:INCRement]?
```

Related Key (Wavelength Setup) Wavelength Step Size

[:SENSe][:WAVelength]:OFFSet

Specifies the wavelength offset. This is the offset between the measured wavelength and the displayed wavelength.

Syntax

```
[:SENSe][:WAVelength]:OFFSet <numeric_value>[M|UM|NM|A]
[:SENSe][:WAVelength]:OFFSet?
```

Related Key (Wavelength Setup) Wavelength Offset

[:SENSe][:WAVelength]:SPAN

Specifies the wavelength span.

The start and stop wavelength and, if necessary, the center wavelength are adjusted so that:

$$Span = 2(Center - Start) \quad \text{and} \quad Span = 2(Stop - Center)$$

The minimum value for Wavelength Span is 0.2 nm.

Syntax

```
[:SENSe][:WAVelength]:SPAN <numeric_value>[M|UM|NM|A]
[:SENSe][:WAVelength]:SPAN?
```

Related Key Span

[:SENSe][:WAVelength]:SPAN:FULL

Sets the wavelength span of the WDCA to full span.

Syntax

```
[:SENSe][:WAVelength]:SPAN:FULL
```

[:SENSe][:WAVelength]:STARt

Specifies the start wavelength.

The center wavelength and span are adjusted so that:

$$Start = Center - \left(\frac{Span}{2}\right) \quad \text{and} \quad Stop = Center + \left(\frac{Span}{2}\right)$$

If the instrument is in zero span, this command sets the center wavelength to the value specified.

Syntax

```
[:SENSe][:WAVelength]:STARt <numeric_value>
  [M|UM|NM|PM|A|HZ|KHZ|MHZ|GHZ|THZ]
[:SENSe][:WAVelength]:STARt?
```

Related Key Start WL

[:SENSe][:WAVelength]:STOP

Specifies the stop wavelength.

The center wavelength and span are adjusted so that:

$$Start = Center - \left(\frac{Span}{2}\right) \qquad \text{and} \qquad Stop = Center + \left(\frac{Span}{2}\right)$$

If the instrument is in zero span, this command sets the center wavelength to the value specified.

Syntax

```
[:SENSe][:WAVelength]:STOP <numeric_value>
  [M|PM|NM|UM|A|HZ|KHZ|MHZ|GHZ|THZ]
[:SENSe][:WAVelength]:STOP?
```

Related Key Stop WL

STATus Subsystem Commands

This subsystem controls the SCPI-defined status-reporting structures. These structures provide registers that you can use to determine if certain events have occurred.

The commands in this subsystem have the following command hierarchy:

:STATus

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

:STATus:OPERation:CONDition?

Queries the contents of the operation condition register.

Use this command to read the value of the OPERation Status register.

Syntax

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

Example

```
OUTPUT 720;":STATUS:OPERATION:CONDITION?"
```

Query Response

Returns a value from 0 to 32767

:STATus:OPERation:ENABle

Sets the contents of the operation enable register.

The enable mask selects which conditions in the event register cause the summary bit in the status byte to be set. If a bit in the enable mask is set true and the corresponding event occurs, the summary bit (bit 7 for the operation status) in the status byte will be set.

Syntax

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

Example

```
OUTPUT 720;":STATUS:OPERation:ENABLE 1024"
```

Query Response

When queried, the largest value that can be returned is 65535. This is because the most-significant register bit cannot be set true.

:STATus:OPERation[:EVENt]?

Queries the contents of the operation event register. This query reads the contents of the register and then clears it. The response will be a number from 0 to 32767 indicating which bits are set. Reading the register clears the register.

Syntax

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

Example

```
OUTPUT 720;":STATUS:OPERATION:EVENT?"
```

:STATus:OPERation:NTRansition

Selects bits in the event register which can be set by negative transitions of the corresponding bits in the condition register.

Changes in the state of a condition register bit causes the associated OPERation Status or QUEStionable Status register bit to be set. This command allows you to select a negative bit transition to trigger an event to be recognized. A negative transition is defined to occur whenever the selected bit changes states from a 1 to a 0. You can enter any value from 0 to 65535.

When queried, the largest value that can be returned is 32767. This is because the most-significant register bit cannot be set true.

Syntax

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

Example

```
OUTPUT 720;":STATUS:OPER:NTRansition 16"
```

:STATus:OPERation:PTRansition

Selects bits in the event register which can be set by positive transitions of the corresponding bits in the condition register.

Changes in the state of a condition register bit causes the associated OPERation Status or QUEStionable Status event register bit to be set. This command allows you to select a positive bit transition to trigger an event to be recognized. A positive transition is defined to occur whenever the selected bit changes states from a 0 to a 1. You can enter any value from 0 to 65535.

When queried, the largest value that can be returned is 32767. This is because the most-significant register bit cannot be set true.

Syntax

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

Example

```
OUTPUT 720;":STATUS:OPER:PTRansition 16"
```

:STATus:PRESet

Clears the event registers and sets all bits in the enable registers.

The “STATus:PRESet command is defined by SCPI to affect the enable register. If you want to clear all event registers and queues, use the *CLS command.

Table 6-4. Preset Values

Status Node	Preset Value
Operation enable register	0
Questionable enable register	0
PTRansition filters	32767
NTRansition filters	0

Syntax

```
:STATus:PRESet
```

Example

```
OUTPUT 720;":STATUS:PRESET"
```

Related Commands

*CLS

:STATus:QUEStionable:CONDition?

Queries the contents of the questionable condition register.

Use this command to read the value of the QUEStionable Status register. Refer to “Monitoring the Instrument” on page 5-20 for further information.

Syntax

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

Example

```
OUTPUT 720;":STATUS:QUEStionable:CONDITION?"
```

Query Response

Returns a value from 0 to 32767

:STATus:QUEStionable:ENABle

Sets or queries the contents of the questionable enable register.

The enable mask selects which conditions in the event register cause the summary bit in the status byte to be set. If a bit in the enable mask is set true and the corresponding event occurs, the summary bit (bit 3 for the questionable status) in the status byte will be set.

Syntax

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

Example

```
OUTPUT 720;":STATUS:QUESTIONABLE:ENABLE 1024"
```

Query Response

When queried, the largest value that can be returned is 65535. This is because the most-significant register bit cannot be set true.

:STATus:QUEStionable[:EVENt]?

Queries the contents of the questionable event register and then clears it.

The response will be a number from 0 to 32767 indicating which bits are set. Reading the register clears the register.

Syntax

```
:STATus:QUEStionable:EVENt?
```

Example

```
OUTPUT 720;":STATUS:QUEStionable:EVENT?"
```

SYSTem Subsystem Commands

The SYStem subsystem collects the functions that are not related to instrument performance. Examples include functions for performing general housekeeping and functions related to setting global configurations, such as TIME or DATE.

The commands in this subsystem have the following command hierarchy:

:SYSTem

```
:COMMunicate:GPIB[:SELF]:BUFFer OFF|ON|0|1
:DATE?
:ERRor:[:NEXT]?
:HELP:HEADers
:PON[:TYPE] PRESet|LAST
:PRESet
:TIME?
:TZONe:NAME?
:VERSion?
```

:SYSTem:COMMunicate:GPIB[:SELF]:BUFFer

Controls the internal GPIB input buffer. The buffer should be left on for normal operation. Default setting is on.

Syntax

```
:SYSTem:COMMunicate:GPIB[:SELF]:BUFFer OFF|ON|0|1
:SYSTem:COMMunicate:GPIB[:SELF]:BUFFer?
```

:SYSTem:DATE

Queries the date of the real-time clock of the WDCA.

Syntax

```
SYSTem:DATE?
```

Related Commands

SYSTem:DISPlay OFF|ON

:SYSTem:ERRor?

Queries the earliest entry in the error queue, then deletes it. The *CLS command clears the entire error queue.

The WDCA has a 30 entry error queue. The queue is a first-in, first-out buffer. Repeatedly sending the query :SYSTEM:ERROR? returns the error numbers and descriptions in the order in which they occur until the queue is empty. Any further queries returns `+0, “No errors”` until another error occurs.

For a complete list of error messages, refer to “Error Messages” on page 7-11.

Syntax

```
:SYSTem:ERRor?
```

Example

```
DIM Error$[250]
OUTPUT 723;”:SYSTEM:ERROR?”
ENTER 723;Error$
PRINT Error$
```

Query Response

`<value>, <string>`

<value> is an integer. *<string>* is the text of the error message.

The following is an example of a response:

`-113,”Undefined header”`

Related Key

Show Critical Errors
Show HW Errors
Show Warnings
Show Notices

:SYSTem:HELP:HEADers?

Returns a list of all commands and queries implemented by the instrument.

The returned ASCII string of commands is in the IEEE 488.2 arbitrary-block data format. The first line indicates the total number of bytes returned to the computer. That is, the # character is followed by one digit which indicates how many of the following digits convey the byte count. The next digits give the actual byte count. For example, in the listing below, 4387 bytes are indicated in the file.

Each command in the listing is separated by a linefeed character.

Syntax

```
:SYSTem:HELP:HEADers?
```

Example

The following is an example of the first few lines and last few lines returned in the string. The term *nquery* indicates that a command cannot be sent as a query. The term *qonly* indicates that a command can only be sent as a query.

```
#44387
:ABORt/nquery/
.
.
.
*IDN?/qonly/
*OPC
*RCL/nquery/
*RST/nquery/
*SAV/nquery/
*SRE
*STB?/qonly/
*TRG/nquery/
*TST?/qonly/
*WAI/nquery/
```

:SYSTem:PRESet

Performs the equivalent of pressing the front-panel PRESET key.

The instrument state is set according to the settings shown in the following table: refer to “Default values” on page 2-44.

Syntax

```
:SYSTem:PRESet
```

Related Key

Preset

:SYSTem:TIME

Queries the time of the real-time clock of the WDCA.

Syntax

```
:SYSTem:TIME?
```

:SYSTem:TZONe:NAME

Returns the time zone used by the real-time clock of the WDCA.

The time zone must be one of the following:

HST10	Hawaii Standard	GMT0bst	Greenwich Mean/British Summer
AST10adt	Aleutian Standard/Daylight	WAT1	Algeria, West Central Africa
YST9ydt	Yukon Standard/Daylight	MET1metdst	Middle European/Daylight
PST8PDT	Pacific Standard/Daylight	EET2	Turkey, Finland, Romania, Greece
MST7	Mountain Standard only	CAT2	Egypt, Sudan, Zaire, Central Africa
MST7MDT	Mountain Standard/Daylight	SAST2sadt	Republic of South Africa
CST6CDT	Central Standard/Daylight	WST3	Western Russia (Moscow)
EST5	Eastern Standard only	EAT3	Eastern Africa, Kenya, Ethiopia
EST5EDT	Eastern Standard/Daylight	WAT3	Moscow, Saudi Arabia, Syria
AST4ADT	Atlantic Standard/Daylight	PST5	Pakistan
NST330NDT	Newfoundland Standard/Daylight	IST530	India
SAT5	Peru, Ecuador, Columbia	TST7	Thailand
SAT430	Venezuela, Guyana, Surinam	SST7	Singapore
SAT4	Western Brazil, Bolivia, Chile	EAT8	Phillipines, Hong Kong, China, Taiwan
SAT3	Argentina, Eastern Brazil	WST8	Western Australia
IST1	Iceland	JST9	Japan, Korea
WAT0	NW Africa, Morocco, Mauritania	CST930cdt	Central Australia Standard/Daylight
WET0WETDST	Western Europe/Daylight	EST10edt	Eastern Australia Standard/Daylight
PWT0pst	Portuguese Winter/Summer	NZST12nzdt	New Zealand Standard/Daylight

Syntax

```
:SYSTem:TZONe:NAME?
```

:SYSTem:VERSion?

Queries the SCPI version. The SCPI version used in the Agilent 86082A is 1997.0.

Syntax

```
:SYSTem:VERSion?
```

TRACe Subsystem Commands

A TRACe or a DATA area is a named entity stored in instrument memory.

The commands in this subsystem have the following command hierarchy:

:TRACe

```
[:DATA]
    :X
        :STARt? <trace_name>
        :STOP? <trace_name>
        :TIMe:SSTop <trace_name>,<numeric_value>,<numeric_value>
        :TIMestamp<trace_name>,year,month,date, hour,min,sec
        :TYPE? <trace_name>
        [:WAVelength]:SSTop <trace_name>,<numeric_value>,<numeric_value>
    [:Y] <trace_name>
        :[:POWER]/nquery
        :RATio <trace_name>,<data_block>|<numeric_value>{,<numeric_value>}
:EXCHange <trace_1> | <trace_2>
:FEED
    :CONTrol <trace_name>, ALWays | NEVer
    :INPut <trace_name>, <numeric_value>
:POINts <trace_name>,<numeric_value>
```

:TRACe[:DATA]:X:STARt?

Returns the start value for the X-axis data for the trace.

The X-axis data will be evenly spaced points from STARt to STOP. The number of points is determined by the TRACe:POINts setting.

Syntax

```
:TRACe[:DATA]:X:STARt? TRA|TRB|TRC|TRD|TRE|TRF
```

:TRACe[:DATA]:X:STOP?

Returns the stop value for the X-axis data for the trace.

The X-axis data will be evenly spaced points from STARt to STOP. The number of points is determined by the TRACe:POINts setting.

Syntax

```
:TRACe[:DATA]:X:STOP? TRA|TRB|TRC|TRD|TRE|TRF
```

:TRACe[:DATA]:X:TIME:SSTop

This command sets the start and stop values for the X-axis data for the trace and sets the X-axis type to TIME.

The first <numeric_value> corresponds to the start, and the second corresponds to the stop. If the stop value is greater than the start value, a “Data out of range” error will be generated. The X-axis data will be evenly spaced points from start to stop. The number of points is determined by the TRACe:POINts setting. If the trace has an expression defined, this expression will be cleared when changing the X-axis start/stop.

Changing the X-axis data in a trace used in an expression (CALCulate:MATH:EXPRession) by another trace may cause an error in the expression if the X-axis data in the operands of the expression no longer match.

Syntax

```
:TRACe[:DATA]:X:TIME:SSTop TRA|TRB|TRC|TRD|TRE|TRF
  <numeric_value>,<numeric_value>[S|MS|US]
```

:TRACe[:DATA]:X:TYPE?

This query reads the X-axis type for the trace.

The X-axis will be WAV for a trace acquired in a normal span, or TIME for a trace acquired in zero span.

The trace names defined for the instrument are: TRA, TRB, TRC, TRD, TRE, and TRF. Specifying any other will generate an “Illegal parameter value” error.

Syntax

```
:TRACe[:DATA]:X:TYPE? TRA|TRB|TRC|TRD|TRE|TRF
```

:TRACe[:DATA]:X[:WAVelength]:SSTop

Sets the start and stop values for the X-axis data for the trace.

The first <numeric_value> corresponds to the start, and the second corresponds to the stop. If the stop value is a shorter wavelength than the start value, a “Data out of range” error will be generated. The X-axis data will be evenly spaced points from start to stop. The number of points is determined by the TRACe:POINts setting. If the trace has an expression defined, this expression will be cleared when the X-axis start/stop is changed.

Changing the X-axis data in a trace used in an expression (CALCulate:MATH:EXPRession) by another trace may cause an error in the expression if the X-axis data in the operands of the expression no longer match.

Syntax

```
:TRACe[:DATA]:X[:WAVelength]:SSTop
  TRA|TRB|TRC|TRD|TRE|TRF,
  <numeric_value>[M|UM|NM|PM|A|HZ|KHZ|MHZ|GHZ|THZ],
  <numeric_value>[M|UM|NM|PM|A|HZ|KHZ|MHZ|GHZ|THZ]
```

:TRACe[:DATA][:Y]?

Returns the Y-axis data points for the trace.

The units are determined by the definition of the trace. The trace data format used in the command is determined by the FORMat subsystem.

Syntax

```
:TRACe[:DATA][:Y]? TRA|TRB|TRC|TRD|TRE|TRF
  NORM1|NORM2
```

:TRACe[:DATA][:Y][:POWer]

Sets the normalization trace and trace data.

The units are determined by the definition of the trace. The trace data format used in the command is determined by the FORMat subsystem.

Syntax

```
:TRACe[:DATA][:Y][:POWer] TRA|TRB|TRC|TRD|TRE|TRF
  NORM1|NORM2
```

:TRACe[:DATA][:Y]:RATio

Sets the Y-axis data points for the trace.

The number of Y-axis data points is determined by the TRACe:POINts setting. If a single numeric value is given, all the Y-axis data points will be set to that value. If more than one value is sent, the trace length will be set to the number of values sent.

This command should be used when the trace data represents a power ratio (unitless number). The trace data format to be used with this command is determined by the FORMat subsystem.

Syntax

```
:TRACe[:DATA][:Y]:RATio TRA|TRB|TRC|TRD|TRE|TRF,
  <data_block>|<numeric_value> {,<numeric_value>}
```

:TRACe:EXCHange

Exchanges both the X-axis and Y-axis data of the two traces.

The only trace pairs that can be exchanged are TRA with any trace, and TRB with TRC. Specifying any other trace will generate an “Illegal parameter value” error.

The TRACe:FEED:CONTrol of the two traces is set to NEVer before the data is exchanged. Both X-axis and Y-axis data will be exchanged. After the data is exchanged, the TRACe:FEED:CONTrol of the two traces is left at NEVer.

Changing the X-axis data in a trace used in an expression (CALCulate:MATH:EXPRession) may cause an error in the expression if the X-axis data in the operands of the expression no longer match.

Syntax

```
:TRACe:EXCHange TRA, TRB|TRC|TRD|TRE|TRF
T:RACe:EXCHange TRB, TRC
```

Related Key Exchange Menu

:TRACe:FEED:CONTrol

Controls how often the specified trace accepts new data.

Setting the TRACe:FEED:CONTrol command to ALWays will allow the trace to always accept new data whenever data is available from the FEED. This is equivalent to turning on the trace update from the front panel.

Setting the TRACe:FEED:CONTrol command to NEVer will cause no new data to be fed into the trace. This is equivalent to turning off the trace update from the front panel.

When switching from NEVer to ALWays, all the valid data from the data source is immediately copied into the trace. If, for example:

- the instrument is in single sweep mode.
- TRA has SENSe:DATA as the FEED and NEVer as the FEED:CONTrol.
- SENSe:DATA contains valid measurement data

Setting the TRACe:FEED:CONTrol command from NEVer to ALWays for TRA will immediately copy the SENSe:DATA into trace A. If the instrument was in continuous sweep mode, and a sweep was in progress, setting the CONTrol command from NEVer to ALWays would immediately copy all the valid SENSe:DATA for the partial sweep.

Syntax

```
:TRACe:FEED:CONTrol TRA|TRB|TRC|TRD|TRE|TRF,
  ALWays|NEVer
:TRACe:FEED:CONTrol? TRA|TRB|TRC|TRD|TRE|TRF
```

Related Key Update A...F On|Off

:TRACe:FEED:INPut

Attaches the given trace to the specified input.

Syntax

```
:TRACe:FEED:INPut TRA|TRB|TRC|TRD|TRE|TRF, 1|2
:TRACe:FEED:INPut? TRA|TRB|TRC|TRD|TRE|TRF
```

:TRACe:TIMestamp

Sets the time stamp for the specified trace.

Syntax

```
:TRACe:TIMestamp<trace name>,year,month,date,hour,min,sec
:TRACe:TIMestamp? <trace name>
```

UNIT Subsystem Commands

Default values are defined, where applicable, for each SCPI command. The UNIT subsystem provides a mechanism to change the default values. The units selected apply to the designated command parameters for both command and response.

The commands in this subsystem have the following command hierarchy:

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

:UNIT:POWer

Selects the units in dBm or mW for the measured value.

Syntax

```
:UNIT:POWer DBM|W
```

Related Key (Amplitude Setup) Amplitude Units

:UNIT:RATio

Specifies units for the input and output of values that represent power ratios.

Specifies units for the input and output of values that represent power ratios. These commands are:

```
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth
  |BANDwidth:NDB
:TRACe:DATA:Y?
```

This command also sets UNIT:POWer to the corresponding setting.

Syntax

```
:UNIT:RATio DB|LINear|AUTO
:UNIT:RATio?
```

Agilent 86140B Series Command Compatibility to Agilent 86082A Series

Remote Command	86140B Series Compatibility	86082A Series Compatibility
*CLS/nquery/	Yes	Yes
*ESE	Yes	Yes
*ESR?/qonly/	Yes	Yes
*IDN?/qonly/	Yes	Yes
*OPC	Yes	Yes
*OPT?/qonly	Yes	Yes
*RCL/nquery/	Yes	Yes
*RST/nquery/	Yes	Yes
*SAV/nquery/	Yes	Yes
*SRE	Yes	Yes
*STB?/qonly/	Yes	Yes
*TST?/qonly/	Yes	Yes
*WAI/nquery/	Yes	Yes
:CALCulate[1|2|3|4|5|6]:AVERage:CLEar/nquery/	Yes	Yes
:CALCulate[1|2|3|4|5|6]:CENTermass:STATe/nquery	Yes	Yes
:CALCulate[1|2|3|4|5|6]:AVERage:COUNt	Yes	Yes
:CALCulate[1|2|3|4|5|6]:AVERage[:STATe]	Yes	Yes
::CALCulate[1|2|3|4|5|6]:LLINe [1|2]:DISPlay {ON[1|OFF|0}	No	Yes
:CALCulate[1|2|3|4|5|6]:LLINe [1|2]:FAIL?	No	Yes
:CALCulate[1|2|3|4|5|6]:LLINe [1|2]:STATe[?] {ON[1|OFF|0}	No	Yes
:CALCulate[1|2|3|4|5|6]:Marker[1|2|3|4]1|2|3|4:AOFF/nquery/	Yes	Yes
:CALCulate[1|2|3|4|5|6]:Marker[1|2|3|4]1|2|3|4:FUNCtion:BWIDth|BANDwidth:INTerpolate	Yes	Yes
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth|BANDwidth:NDB	Yes	Yes
:CALCulate[1|2|3|4|5|6]:Marker[1|2|3|4]:FUNCtion:BWIDth|BANDwidth:READout	Yes	Yes
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth|BANDwidth:RESult?/qonly/	Yes	Yes

Remote Command	86140B Series Compatibility	86082A Series Compatibility
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth|BANDwidth[:TYPE] PASSBand|NOTCh	No	Yes
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth|BANDwidth:X:CENTer?/qonly/	Yes	Yes
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth|BANDwidth:X:LEFT?/qonly/	Yes	Yes
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth|BANDwidth:X:REFerence	No	Yes
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth|BANDwidth:X:RIGHt?/qonly/	Yes	Yes
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:BWIDth|BANDwidth:Y:REFerence	No	Yes
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:DELTa:RESet/nquery/	Yes	Yes
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:DELTa:X:OFFSet:FREQuency/nquery/	Yes	Yes
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:DELTa:X:OFFSet:TIME/nquery/	Yes	Yes
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:DELTa:X:OFFSet?/qonly/	Yes	Yes
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:DELTa:X:OFFSet[:WAVelength]/nquery/	Yes	Yes
:CALCulate[1|2|3|4|5|6]:Marker[1|2|3|4]:FUNCtion:DELTa:X:READout	Yes	Yes
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:DELTa:X:REFerence?/qonly/	Yes	Yes
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:DELTa:Y:OFFSet?/qonly/	Yes	Yes
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:DELTa:Y:REFerence?/qonly/	Yes	Yes
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:DELTa[:STATe]	Yes	Yes
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:NOISe:BANDwidth	Yes	No
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:NOISe:RESult?/qonly/	Yes	No
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:NOISe[:STATe]	Yes	No
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:FUNCtion:PRESet/nquery/	Yes	Yes
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:INTerpolate	Yes	Yes
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MAXimum/nquery/	Yes	Yes
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MAXimum:LEFT/nquery/	Yes	Yes
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:MAXimum:NEXT/nquery/	Yes	Yes
:CALCulate[1|2|3|4|5|6]:Marker[1|2|3|4]:MAXimum:RIGHt/nquery/	Yes	Yes
:CALCulate[1|2|3|4|5|6]:Marker[1|2|3|4]:MINimum:LEFT/nquery/	Yes	Yes
:CALCulate[1|2|3|4|5|6]:Marker[1|2|3|4]:MINimum:NEXT/nquery/	Yes	Yes
:CALCulate[1|2|3|4|5|6]:Marker[1|2|3|4]:MINimum/nquery/	Yes	Yes
:CALCulate[1|2|3|4|5|6]:Marker[1|2|3|4]:MINimum:RIGHt/nquery/	Yes	Yes
:CALCulate[1|2|3|4|5|6]:Marker[1|2|3|4]:PEXCursion[:PEAK]	Yes	Yes
:CALCulate[1|2|3|4|5|6]:Marker[1|2|3|4]:PEXCursion:PIT	Yes	Yes
:CALCulate[1|2|3|4|5|6]:Marker[1|2|3|4]:SCENter/nquery/	Yes	Yes
:CALCulate[1|2|3|4|5|6]:Marker[1|2|3|4]:SRANge:LOWer:FREQuency/nquery/	Yes	Yes
:CALCulate[1|2|3|4|5|6]:Marker[1|2|3|4]:SRANge:LOWer:TIME/nquery/	Yes	Yes
:CALCulate[1|2|3|4|5|6]:Marker[1|2|3|4]:SRANge:LOWer?/qonly/	Yes	Yes

Remote Command	86140B Series Compatibility	86082A Series Compatibility
:CALCulate[1|2|3|4|5|6]:Marker[1|2|3|4]:SRANge:LOWer[:WAVelength]/nquery/	Yes	Yes
:CALCulate[1|2|3|4|5|6]:Marker[1|2|3|4]:SRANge[:STATe]	Yes	No
:CALCulate[1|2|3|4|5|6]:Marker[1|2|3|4]:SRANge:UPPer:FREQuency/nquery/	Yes	Yes
:CALCulate[1|2|3|4|5|6]:Marker[1|2|3|4]:SRANge:UPPer:TIME/nquery/	Yes	Yes
:CALCulate[1|2|3|4|5|6]:Marker[1|2|3|4]:SRANge:UPPer?/qonly/	Yes	Yes
:CALCulate[1|2|3|4|5|6]:Marker[1|2|3|4]:SRANge:UPPer[:WAVelength]/nquery/	Yes	Yes
:CALCulate[1|2|3|4|5|6]:Marker[1|2|3|4]:SRLevel/nquery/	Yes	Yes
:CALCulate[1|2|3|4|5|6]:Marker[1|2|3|4][:STATe]	Yes	Yes
:CALCulate[1|2|3|4|5|6]:Marker[1|2|3|4]:TRACe	Yes	Yes
:CALCulate[1|2|3|4|5|6]:Marker[1|2|3|4]:X:FREQuency/nquery/	Yes	Yes
:CALCulate[1|2|3|4|5|6]:Marker[1|2|3|4]:X:READout	Yes	Yes
:CALCulate[1|2|3|4|5|6]:Marker[1|2|3|4]:X:TIME/nquery/	Yes	Yes
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:X?/qonly/	Yes	Yes
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:X[:WAVelength]/nquery/	Yes	Yes
:CALCulate[1|2|3|4|5|6]:MARKer[1|2|3|4]:Y?/qonly/	Yes	Yes
:CALCulate[1|2|3|4|5|6]:MATH[:EXPRession][:DEFine]	Yes	Yes
:CALCulate[1|2|3|4|5|6]:MATH:STATe	Yes	Yes
:CALCulate[1|2|3|4|5|6]:MAXimum:CLEar/nquery/	Yes	Yes
:CALCulate[1|2|3|4|5|6]:MAXimum[:STATe]	Yes	Yes
:CALCulate[1|2|3|4|5|6]:MEAN[:DATA]?/qonly/	Yes	Yes
:CALCulate[1|2|3|4|5|6]:MEAN:RANGe:LOWer:FREQuency/nquery/	Yes	Yes
:CALCulate[1|2|3|4|5|6]:MEAN:RANGe:LOWer:TIME/nquery/	Yes	Yes
:CALCulate[1|2|3|4|5|6]:MEAN:RANGe:LOWer?/qonly/	Yes	Yes
:CALCulate[1|2|3|4|5|6]:MEAN:RANGe:LOWer[:WAVelength]/nquery/	Yes	Yes
:CALCulate[1|2|3|4|5|6]:MEAN:RANGe[:STATe]	Yes	Yes
:CALCulate[1|2|3|4|5|6]:MEAN:RANGe:UPPer:FREQuency/nquery/	Yes	Yes
:CALCulate[1|2|3|4|5|6]:MEAN:RANGe:UPPer:TIME/nquery/	Yes	Yes
:CALCulate[1|2|3|4|5|6]:MEAN:RANGe:UPPer?/qonly/	Yes	Yes
:CALCulate[1|2|3|4|5|6]:MEAN:RANGe:UPPer[:WAVelength]/nquery/	Yes	Yes
:CALCulate[1|2|3|4|5|6]:MEAN:STATe	Yes	Yes
:CALCulate[1|2|3|4|5|6]:MINimum:CLEar/nquery/	Yes	Yes
:CALCulate[1|2|3|4|5|6]:MINimum[:STATe]	Yes	Yes
:CALCulate[1|2|3|4|5|6]:OFFSet	Yes	Yes
:CALCulate[1|2|3|4|5|6]:PATH[:NAME]?	Yes	No
:CALCulate[1|2|3|4|5|6]:PATH:RESult?	Yes	No
:CALCulate[1|2|3|4|5|6]:THReshold	Yes	Yes

Remote Command	86140B Series Compatibility	86082A Series Compatibility
:CALCulate[1|2|3|4|5|6]:THReshold:STATe	Yes	Yes
:CALCulate[1|2|3|4|5|6]:TPOWer[:DATA]?/qonly/	Yes	No
:CALCulate[1|2|3|4|5|6]:TPOWer:IRANge:LOWer	Yes	No
:CALCulate[1|2|3|4|5|6]:TPOWer:IRANge[:STATe]	Yes	No
:CALCulate[1|2|3|4|5|6]:TPOWer:IRANge:UPPer	Yes	No
:CALCulate[1|2|3|4|5|6]:TPOWer:STATe	Yes	No
:CALibration:ALIGn[:AUTO]/nquery/	Yes	No
:CALibration:ALIGn[:AUTO]:MARKer{?:?}/nquery/	Yes	No
:CALibration:ALIGn:EXTernal/nquery/	Yes	No
:CALibration:ALIGn:PRESet/nquery/	Yes	No
:CALibration:DATE?/qonly/	Yes	Yes
:CALibration:POWer:DATE?/qonly/	Yes	No
:CALibration:POWer/nquery/	Yes	No
:CALibration:POWer:STATe	Yes	No
:CALibration:POWer:VALue	Yes	No
:CALibration:POWer:WAVelength	Yes	No
:CALibration:PRESet/nquery/	Yes	No
:CALibration:STATe	Yes	No
:CALibration:WAVelength:DATE?/qonly/	Yes	Yes
:CALibration:WAVelength[:EXTernal]:MULTIpoint:MARKer[1|2|3|4]/nquery/	Yes	No
:CALibration:WAVelength[:EXTernal]:MULTIpoint/nquery/	Yes	No
:CALibration:WAVelength[:EXTernal][:NORMal]:MARKer[1|2|3|4]/nquery/	Yes	No
:CALibration:WAVelength[:EXTernal][:NORMal]/nquery/	Yes	No
:CALibration:WAVelength[:EXTernal]:VALue	Yes	No
:CALibration:WAVelength:EWC:FUNCtion	Yes	No
:CALibration:WAVelength:EWC:RANGe	Yes	No
:CALibration:WAVelength:EWC:STATe	Yes	No
:CALibration:WAVelength:INTernal:MULTIpoint/nquery/	Yes	No
:CALibration:WAVelength:INTernal:NORMal/nquery/	Yes	Yes
:CALibration:WAVelength:INTernal/nquery/	No	Yes
:CALibration:WAVelength:MODE	Yes	No
:CALibration:WAVelength/nquery/	Yes	No
:CALibration:WAVelength:STATe	Yes	Yes
:CALibration:WAVelength:USER:DATA	Yes	No
:CALibration:ZERO[:AUTO]	Yes	Yes
:DISPlay[:WINDow[1]]:ANNotation[:ALL]	Yes	Yes

Remote Command	86140B Series Compatibility	86082A Series Compatibility
:DISPlay:[WINDow[1]] OFF|ON|0|1	Yes	Yes
:DISPlay[:WINDow[1]]:POPup[1|2|3|4][:ALL]/nquery/	Yes	Yes
:DISPlay[:WINDow[1]]:TEXT:CLEar/nquery/	Yes	Yes
:DISPlay[:WINDow[1]]:TEXT:DATA	Yes	Yes
:DISPlay[:WINDow[1]]:TRACe:ALL[:SCALe][:AUTO]:MARKer	Yes	No
:DISPlay[:WINDow[1]]:TRACe:ALL[:SCALe][:AUTO]/nquery/	Yes	No
:DISPlay[:WINDow[1]]:TRACe:ALL[:SCALe][:AUTO]:OPTimize	Yes	No
:DISPlay[:WINDow[1]]:TRACe:GRATicule:GRID[:STATe]	Yes	Yes
:DISPlay[:WINDow[1]]:TRACe[:STATe]	Yes	Yes
:DISPlay[:WINDow[1]]:TRACe:X[:SCALe]:AUTO:SPAN	Yes	No
:DISPlay[:WINDow[1]]:TRACe:X[:SCALe]:AUTO:SPAN:AUTO	Yes	No
:DISPlay[:WINDow[1]]:TRACe:Y[:SCALe]:AUTO:PDIVision	Yes	No
:DISPlay[:WINDow[1]]:TRACe:Y[:SCALe]	Yes	Yes
:DISPlay[:WINDow[1]]:TRACe:Y1|Y2[:SCALe]:AUTO:PDIVision:AUTO	Yes	No
:DISPlay[:WINDow[1]]:TRACe:Y1|Y2[:SCALe]:LINear	Yes	Yes
:DISPlay[:WINDow[1]]:TRACe:Y1|Y2[:SCALe]:PDIVision	Yes	Yes
:DISPlay[:WINDow[1]]:TRACe:Y1|Y2[:SCALe]:RLEVel	Yes	Yes
:DISPlay[:WINDow[1]]:TRACe:Y1|Y2[:SCALe]:RPOSition	Yes	Yes
:FORMat[:DATA]	Yes	Yes
:HCOPy:DATA?/qonly/	Yes	Yes
:HCOPy:DESTination	Yes	Yes
:HCOPy:DEVice:LANGuage	Yes	Yes
:HCOPy[:IMMediate]/nquery/	Yes	Yes
:INITiate:CONTinuous	Yes	Yes
:INITiate[:IMMediate]/nquery/	Yes	Yes
:INITiate:NORMalize 1|2	No	Yes
:INSTrument:CATalog:FULL?/qonly/	Yes	No
:INSTrument:CATalog?/qonly/	Yes	No
:INSTrument:NSELect	Yes	No
:INSTrument[:SELect]	Yes	No
:MEMory:STATe[:EXTended]?/qonly/	Yes	Yes
:MMEMory:CATalog?/qonly/	Yes	Yes
:MMEMory:DATA	Yes	Yes
:MMEMory:DELete/nquery/	Yes	Yes
:MMEMory:INITialize/nquery/	Yes	Yes
:MMEMory:LOAD:LLINe 1|2, <file_name>, [INTernal|FLOPpy]	No	Yes

Remote Command	86140B Series Compatibility	86082A Series Compatibility
:MMEMory:LOAD:TRACe/nquery/	Yes	Yes
:MMEMory:STORe:TRACe/nquery/	Yes	Yes
:OUTPut:STATe OFF|ON|0|1	No	Yes
[:SENSe]:BANDwidth[:RESolution]	Yes	No
[:SENSe]:BANDwidth[:RESolution]:AUTO	Yes	No
[:SENSe]:BANDwidth[:RESolution]:RATio	Yes	No
[:SENSe]:BANDwidth:VIDeo	Yes	No
[:SENSe]:BANDwidth:VIDeo:AUTO	Yes	No
[:SENSe]:CHOP[:STATe]	Yes	No
[:SENSe]:CORRection:RVELocity:MEDium	Yes	Yes
[:SENSe]:GORDer[:AUTO]	Yes	No
[:SENSe]:NORMalize:ENABle 1|2|,OFF|ON|0|1	No	Yes
[:SENSe]:NORMalize:WAVelength:STARt <numeric_value>	No	Yes
[:SENSe]:NORMalize:WAVelength:STOP <numeric_value>	No	Yes
[:SENSe]:POWer[:DC]:RANGe:AUTO	Yes	Yes
[:SENSe]:POWer[:DC]:RANGe:LOCK	Yes	No
[:SENSe]:POWer[:DC]:RANGe:LOWer	Yes	Yes
[:SENSe]:POWer[:DC]:RANGe:LOWer:AUTO OFF|ON|0|1	Yes	Yes
[:SENSe]:SWEep:POINts	Yes	Yes
[:SENSe]:SWEep:SPEed PRECision|NORMal|FAST|0|1|2	No	Yes
[:SENSe]:SWEep:TIME	Yes	Yes
[:SENSe]:SWEep:TIME:AUTO	Yes	No
[:SENSe][:WAVelength]:CENTer	Yes	Yes
[:SENSe][:WAVelength]:CENTer:STEP:AUTO	Yes	Yes
[:SENSe][:WAVelength]:CENTer:STEP[:INCRement]	Yes	Yes
[:SENSe][:WAVelength]:LIMIt	Yes	Yes
[:SENSe][:WAVelength]:OFFSet	Yes	Yes
[:SENSe][:WAVelength]:SPAN	Yes	Yes
[:SENSe][:WAVelength]:SPAN:FULL/nquery/	Yes	Yes
[:SENSe][:WAVelength]:SRANge:LOWer	Yes	No
[:SENSe][:WAVelength]:SRANge[:STATe]	Yes	No
[:SENSe][:WAVelength]:SRANge:UPPer	Yes	Yes
[:SENSe][:WAVelength]:STARt	Yes	Yes
[:SENSe][:WAVelength]:STOP	Yes	Yes
:OUTPut:STATE	No	Yes
:SOURce[n]:CATalog?/qonly/	Yes	No

Remote Command	86140B Series Compatibility	86082A Series Compatibility
:SOURce[n]:CURRent[:LEVel][:IMMediate][:AMPLitude]	Yes	No
:SOURce[n]:CURRent:LIMit[:AMPLitude]	Yes	No
:SOURce[n]:CURRent:PULSe:STATe	Yes	No
:SOURce[n]:PULSe:DCYCle	Yes	No
:SOURce[n]:PULSe:WIDTh	Yes	No
:SOURce[n]:STATe	Yes	No
:STATus:OPERation:CONDition?/qonly/	Yes	Yes
:STATus:OPERation:ENABle	Yes	Yes
:STATus:OPERation[:EVENt]?/qonly/	Yes	Yes
:STATus:OPERation:NTRansition	Yes	Yes
:STATus:OPERation:PTRansition	Yes	Yes
:STATus:PRESet/nquery/	Yes	Yes
:STATus:QUEStionable:CONDition?/qonly/	Yes	Yes
:STATus:QUEStionable:ENABle	Yes	Yes
:STATus:QUEStionable[:EVENt]?/qonly/	Yes	Yes
:SYSTem:COMMunicate:GPIB[:SELF]:BUFFer OFF|ON|0|1	Yes	Yes
:SYSTem:DATE	Yes	Yes
:SYSTem:DISPlay OFF|ON|0|1	Yes	Yes
:SYSTem:ERRor?/qonly/	Yes	Yes
:SYSTem:HELP:HEADers?/qonly/	Yes	Yes
:SYSTem:PON[:TYPE]	Yes	Yes
:SYSTem:PRESet/nquery/	Yes	Yes
:SYSTem:TIME	Yes	Yes
:SYSTem:TZONe:NAME	Yes	Yes
:SYSTem:VERSion?/qonly/	Yes	Yes
:TRACe[:DATA]:X:STARt?/qonly/	Yes	Yes
:TRACe[:DATA]:X:STOP?/qonly/	Yes	Yes
:TRACe[:DATA]:X:TIME:SSTop/nquery/	Yes	Yes
:TRACe[:DATA]:X:TYPE?/qonly/	Yes	Yes
:TRACe[:DATA]:X[:WAVelength]:SSTop/nquery/	Yes	Yes
:TRACe[:DATA][:Y]?/qonly/	Yes	Yes
:TRACe[:DATA][:Y][:POWer]/nquery/	No	Yes
:TRACe[:DATA][:Y]:RATio/nquery/	Yes	Yes
:TRACe:EXCHange/nquery/	Yes	Yes
:TRACe:FEED:CONTrol TRA|TRB|TRC|TRD|TRE|TRF, ALWays|NEVer	Yes	Yes
:TRACe:FEED:INPut TRA|TRB|TRC|TRD|TRE|TRF, 1|2	No	Yes

Remote Command	86140B Series Compatibility	86082A Series Compatibility
:TRACe:POINts TRA|TRB|TRC|TRD|TRE|TRF|Normal, <numeric_value>	No	Yes
:TRACe:TIMestamp<trace name>,year,month,date,hour,min,sec	No	Yes
:TRACe:TIMestamp? <numeric_value>	No	Yes
:TRIGger[:SEQuence]:DELay	Yes	No
:TRIGger[:SEQuence]:OUTPut	Yes	No
:TRIGger[:SEQuence]:OUTPut:PULSe:DCYCle	Yes	No
:TRIGger[:SEQuence]:OUTPut:PULSe:WIDTh	Yes	No
:TRIGger[:SEQuence]:SLOPe	Yes	No
:TRIGger[:SEQuence]:SOURce	Yes	No
:UNIT:POWer	Yes	Yes
:UNIT:RATio	Yes	Yes

7

Status Listings

Overview

This document describes the status listings of the Agilent 86082A WDCAWDCA. Status conditions for the optical spectWDCArum analyzer are grouped into categories.

Error Indicates the instrument is malfunctioning. Measurement accuracy is probably affected. Errors can be caused by either a hardware or a firmware problem. The instrument requires repair at a Agilent Technologies service center.

Status error Indicates an internal hardware function is unavailable or not operating within specifications. This is usually a temporary problem, such as a temperature control loop being unsettled. Status errors cannot be cleared by the user. When the condition causing the error is corrected the error will go away.

Warning A warning is displayed when the WDCA cannot satisfy a request from the user. Parameter entries that are out of range, illegal or unrecognized remote commands or missing hardware options can cause a warning to be displayed. A warning can be displayed if a requested measurement, for example AutoMeasure, cannot be performed due to a missing input signal.

UNCAL This message alerts the user the requested instrument setup can cause an invalid measurement. For example, manually setting the sweep time to a value that is too fast for other instrument settings.

Notice A user alert indicating something in the state of the instrument has changed. For example, if the user specifies a particular value for the current duty cycle, a notice will be displayed when the WDCA modifies the pulse width.

The following table lists the error numbers and their definitions.

Number Range	Definition
–1 to –999	Standard SCPI errors
1000 to 2999	analyzer notices
3000 to 4999	Application specific notices
5000 to 7999	analyzer warnings
8000 to 9999	Application specific warnings
10000 to 11999	analyzer status errors
12000 to 13999	Application specific status errors
20000 to 21999	analyzer errors
22000 to 23999	Application specific errors
30000 to 32767	Firmware errors

Error Reporting Behavior

Errors are displayed in an on-screen dialog box. To continue operation, the user must acknowledge the error by pressing a button.

Status errors are displayed with a descriptive line in the lower-left corner of the graticule. Immediately press the System key below the display, and then press the *Help* softkey to the right of the display to display more information. A softkey menu is displayed that includes, *Show Critical Errors*, *Show HW Errors*, *Show Warnings*, and *Show Notices*. If there are any errors, warnings or notices, one or more of these keys are enabled. Press the key to display a more complete description of the condition causing the message.

Warnings and notices are displayed in a status line at the bottom of the screen. They are cleared when any front-panel key, other than the System key, is pressed, or they will clear automatically after a few seconds. To view warnings and notices, press the front-panel System key, the *Help* softkey and then the *Show Warnings* or *Show Notices* softkey. UNCAL is displayed as red text in the lower-right corner of the graticule. It is only displayed when a condition exists that can lead to an inaccurate measurement. There are cases when valid measurements can be made while an UNCAL condition exists, but the operator should use caution.

SCPI-Defined Errors

These error messages and descriptions were copied from the *SCPI 1997 Volume 2: Command reference*. The sentences enclosed in brackets "[ ]" are copied from the error descriptions in the SCPI reference. References are also made to IEEE 488.2 sections for further clarification of events.

Not all the available SCPI error messages were utilized in the WDCA. Only those pertinent to the instrument were implemented.

Note that some of the error messages do not contain descriptions or help messages. These error numbers are reserved for future expansion. This document is subject to further change and development.

Standards related information

Further information on the Standard Commands for Programmable Instruments (SCPI) standard is available from the SCPI consortium.

Contact:

Fred Bode, Executive Director
SCPI Consortium
8380 Hercules Drive, Suite P3
La Mesa, CA 91942
Phone: (619) 697-8790
FAX: (619) 697-5955
CompuServe Number: 76516,254
Fbode@vxinl.com

For more detailed information on the IEEE 488.2 Standard, order a reference copy.

Contact:

The Institute of Electrical and Electronics Engineers, Inc.
345 East 47th Street
New York, New York 10017-2394
Phone: (800) 678-IEEE (US) 8 a.m. – 4:30 p.m. (EST)
(908) 981-1393 (International)
Fax: (908) 981-9667

Standard SCPI errors (–1 to –999)

All positive numbers are instrument-dependent. All negative numbers are reserved by the SCPI standard with certain standard error/event codes described in this document. The value, zero, is also reserved to indicate that no error or event has occurred.

Error/event numbers

The system-defined error/event numbers are chosen on an enumerated (“1 of N”) basis. The SCPI-defined error/event numbers and the <error/event description> portions of the full queue item are listed here. The first error/event described in each class (for example, –100, –200, –300, –400) is a “generic” error. In selecting the proper Error/event number to report, more specific error/event codes are preferred, and the generic error/event is used only if the others are inappropriate.

Command errors

An <error/event number> in the range [–199, –100] indicates that an IEEE 488.2 syntax error has been detected by the instrument’s parser. The occurrence of any error in this class shall cause the command error bit (bit 5) in the event status register (IEEE 488.2, section 11.5.1) to be set.

One of the following events has occurred:

- An IEEE 488.2 syntax error has been detected by the parser. That is, a controller-to-device message was received which is in violation of the IEEE 488.2 standard. Possible violations include a data element which violates the device listening formats or whose type is unacceptable to the device.

- An unrecognized header was received. Unrecognized headers include incorrect device-specific headers and incorrect or unimplemented IEEE 488.2 common commands.
- A Group Execute Trigger (GET) was entered into the input buffer inside of an IEEE 488.2 <PROGRAM MESSAGE>.

Events that generate command errors shall not generate execution errors, device-specific errors, or query errors; see the other error definitions in this chapter.

Table 7-1. Command Errors (1 of 4)

Error Number	Error Description [description/explanation/examples]

–101	desc = “Invalid character”
	help = ““
	[A syntactic element contains a character which is invalid for that type; for example, a header containing an ampersand, SETUP&. This error might be used in place of errors –114, –121, –141, and perhaps some others.]

–102	desc = “Syntax error”
	help = ““
	[An unrecognized command or data type was encountered for example, a sting was received when the instrument does not accept strings.]

–103	desc = “Invalid separator”
	help = ““
	[The parser was expecting a separator and encountered an illegal character; for example, the semicolon was omitted after a program message unit, *EMC 1:CH1:VOLTS 5.]

–104	desc = “Wrong Parameter Type”
	help = ““
	[The parser recognized a data element different than one allowed; for example, numeric or string data was expected but block data was encountered.]

–105	desc = “GET not allowed”
	help = ““
	[A Group Execute Trigger was received within a program message (see IEEE 488.2, 7.7).]

Table 7-1. Command Errors (2 of 4)

Error Number	Error Description [description/explanation/examples]

–108	desc = “Parameter not allowed”
	help = ““
	[More parameters were received than expected for the header; for example, the *EMC common command only accepts one parameter, so receiving *EMC 0,1 is not allowed.]

–109	desc = “Missing parameter”
	help = ““
	[Fewer parameters were received than required for the header; for example, the *EMC common command requires one parameter, so receiving *EMC is not allowed.]

–112	desc = “Program mnemonic too long” help = ““ [The header contains more that twelve characters (see IEEE 488.2, 7.6.1.4.1).]

–113	desc = “Undefined header”
	help = ““
	[The header is syntactically correct, but it is undefined for this specific device; for example, *XYZ is not defined for any device.]

–120	desc = “Numeric Data Error”
	help = “
	[This error, as well as errors –121 through –129, are generated when parsing a data element which appears to be numeric, including the non-decimal numeric types. This particular error message should be used if the device cannot detect a more specific error.]

–121	desc = “Invalid character in number” help = ““ [An invalid character for the data type being parsed was encountered; for example, an alpha in a decimal numeric or a “9” in octal data.]

–123	desc = “Exponent too large”
	help = ““
	[The magnitude of the exponent was larger than 32000 (see IEEE 488.2, 7.7.2.4.1).]

–124	desc = “Too many digits”
	help = ““
	[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).]

Table 7-1. Command Errors (3 of 4)

Error Number	Error Description [description/explanation/examples]
–128	desc = “Numeric data not allowed” help = ““ [A legal numeric data element was received, but the device does not accept one in this position for the header.]

–131	desc = “Invalid suffix”
	help = ““
	The suffix does not follow the syntax described in IEEE 488.2, 7.7.3.2, or the suffix is inappropriate for this device.]

–134	desc = “Suffix too long”
	help = ““
	[The suffix contained more than 12 characters (see IEEE 488.2, 7.7.3.4).]

–138	desc = “Suffix not allowed”
	help = ““
	[A suffix was encountered after a numeric element which does not allow suffixes.]

–148	desc = “Character data not allowed” help = ““ [A legal character data element was encountered where prohibited by the device.]

–150	desc = “String data error”
	help = ““
	[This error, as well as errors –151 through –159, are generated when parsing a string data element. This particular error message should be used if the device cannot detect a more specific error.]

–151	desc = “Invalid string data”
	help = ““
	A string data element was expected, but was invalid for some reason (see IEEE 488.2, 7.7.5.2); for example, an END message was received before the terminal quote character.]

–158	desc = “String data not allowed” help = ““ [A string data element was encountered but was not allowed by the device at this point in parsing.]

Table 7-1. Command Errors (4 of 4)

Error Number	Error Description [description/explanation/examples]

–161	desc = “Invalid block data”
	help = ““
	[A block data element was expected, but was invalid for some reason (see IEEE 488.2, 7.7.6.2); for example, an END message was received before the length was satisfied.]

–168	desc = “Block data not allowed” help = ““ [A legal block data element was encountered but was not allowed by the device at this point in parsing.]

–170	desc = “Expression error”
	help = ““
	[This error, as well as errors –171 through –179, are generated when parsing an expression data element. This particular error message should be used if the device cannot detect a more specific error.]

–171	desc = “Invalid expression”
	help = ““
	[The expression data element was invalid (see IEEE 488.2, 7.7.7.2); for example, unmatched parentheses or an illegal character.]

–178	desc = “Expression data not allowed” help = ““ [A legal expression data was encountered but was not allowed by the device at this point in parsing.]
–181	desc = “Invalid outside macro definition” help = ““ [Indicates that a macro parameter placeholder ($<number) was encountered outside of a macro definition.]
–183	desc = “Invalid inside macro definition” help = ““ [Indicates that the program message unit sequence, sent with a *DDT or *DMC command, is syntactically invalid (see IEEE 488.2, 10.7.6.3).]

Execution errors

An <error/event number> in the range [–299, –200] indicates that an error has been detected by the instrument's execution control block. The occurrence of any error in this class shall cause the execution error bit (bit 4) in the event status register (IEEE 488.2, section 11.5.1) to be set.

One of the following events has occurred:

- A <PROGRAM DATA> element following a header was evaluated by the device as outside of its legal input range or is otherwise inconsistent with the device's capabilities.
- A valid program message could not be properly executed due to some device condition.

Execution errors shall be reported by the device after rounding and expression evaluation operations have taken place. Rounding a numeric data element, for example, shall not be reported as an execution error. Events that generate execution errors shall not generate Command Errors, device-specific errors, or Query Errors; see the other error definitions in this section.

Table 7-2. Execution Errors (1 of 3)

Error Number	Error Description [description/explanation/examples]

–200	desc = "Execution error"
	help = ""
	[This is the generic syntax error 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.]

–213	desc = "INIT:IMM command ignored" help = "An initiate immediate remote command was received. The instrument was in continuous sweep mode when the command was received. The remote command is ignored unless the instrument is in single sweep mode. To select single sweep mode use the INIT:CONT OFF command."
–221	desc = "Settings Conflict Error" help = "A request was made but the instrument settings resulting from the request are in conflict with each other." [Indicates that a legal program data element was parsed but could not be executed due to the current device state (see IEEE 488.2, 6.4.5.3 and 11.5.1.1.5.)]

Table 7-2. Execution Errors (2 of 3)

Error Number	Error Description [description/explanation/examples]
–222	desc = “Data out of range” help = “A numeric value was entered which is outside the legal range of values for the parameter. The name of the parameter is listed at the end of the error message.” [Indicates that a legal program data element was parsed but could not be executed because the interpreted value was outside the legal range as defined by the device (see IEEE 488.2, 11.5.1.1.5.)]
–222	desc = “Span out of range” help = “A numeric value was entered which is outside the legal range of values for the span setting. This occurs if the value is too large, less than zero, or between zero and 0.2nm. Zero span may not be entered by using the DOWN arrow key from 0.2nm; a numeric value of zero must be entered instead.” [Indicates that a legal program data element was parsed but could not be executed because the interpreted value was outside the legal range as defined by the device (see IEEE 488.2, 11.5.1.1.5.)]

–223	desc = “Too much data”
	help = ““
	[Indicates that 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.]

–224	desc = “Illegal parameter value”
	help = ““
	[Used where exact value, from a list of possibles, was expected.]

–257	desc = “File name error”
	help = ““
	[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. The definition of what constitutes a file name error is device-specific.]

–272	desc = “Macro Exec Error”
	help = ““
	[Indicates that a syntactically legal macro program data sequence could not be executed due to some error in the macro definition (see IEEE 488.2, 10.7.6.3.)]

Table 7-2. Execution Errors (3 of 3)

Error Number	Error Description [description/explanation/examples]

–273	desc = “Illegal macro label”
	help = ““
	[Indicates that 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

–276	desc = “Macro recursion error”
	help = ““
	[Indicates that a syntactically legal macro program data sequence could not be executed because the device found it to be recursive (see IEEE 488.2, 10.7.6.6).]

–277	desc = “Macro redefinition not allowed” help = ““ [Indicates that a syntactically legal macro label in the *DMC command could not be executed because the macro label was already defined (see IEEE 488.2, 10.7.6.4).]
–278	desc = “Macro header not found” help = ““ [Indicates that a syntactically legal macro label in the *GMC? query could not be executed because the header was not previously defined.]

Device-specific errors

An <error/event number> in the range [–399, –300] or [1, 32767] indicates the instrument has detected an error which is not a command error, a query error, or an execution error; some device operations did not properly complete, possibly due to an abnormal hardware or firmware condition. These codes are also used for self-test response errors. The occurrence of any error in this class should cause the device-specific error bit (bit 3) in the event status register (IEEE 488.2, section 11.5.1) to be set. The meaning of positive error codes is device-dependent and may be enumerated or bit mapped; the <error message> string for positive error codes is not defined by SCPI and available to the device designer. Note that the string is not optional; if the designer does not wish to implement a string for a particular error, the null string should be sent (for example, 42,””). The occurrence of any error in this class should cause the device-specific error bit (bit 3) in the event status register (IEEE 488.2, section 11.5.1) to be set. Events that generate device-specific errors shall not generate command errors, execution errors, or query errors; see the other error definitions in this section.

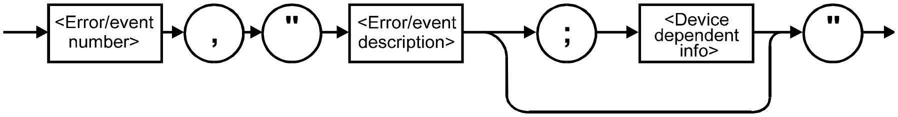


Table 7-3. Device-Specific Errors

Error Number	Error Description [description/explanation/examples]

–310	desc = “System error”
	help = ““
	[Indicates that some error, termed “system error” by the device, has occurred. This code is device-dependent.]

–321	desc = “Out of memory”
	help = ““
	[An internal operation needed more memory than was available.]

Query errors

An <error/event number> in the range [–499, –400] indicates that the output queue control of the instrument has detected a problem with the message exchange protocol described in IEEE 488.2, chapter 6. The occurrence of any error in this class shall cause the query error bit (bit 2) in the event status register (IEEE 488.2, section 11.5.1) to be set. These errors correspond to message exchange protocol errors described in IEEE 488.2, section 6.5.

One of the following is true:

- An attempt is being made to read data from the output queue when no output is either present or pending.
- Data in the output queue has been lost.

Events that generate query errors will not generate command errors, execution errors, or device-specific errors; see the other error definitions in this section.

Table 7-4. Query Errors (1 of 2)

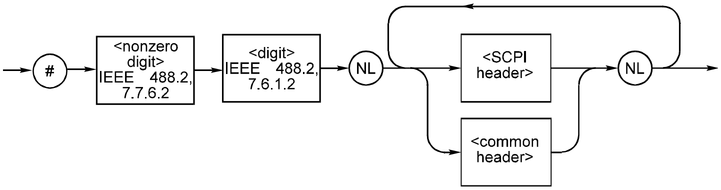
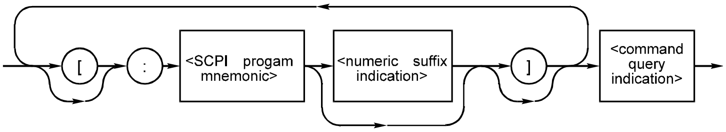
Error Number	Error Description [description/explanation/examples]
–400	Query error [This is the generic query error for devices that cannot detect more specific errors. This code indicates only that a Query Error as defined in IEEE 488.2, 11.5.1.1.7 and 6.3 has occurred.]  # <nonzero digit> IEEE 488.2, 7.7.6.2 <digit> IEEE 488.2, 7.6.1.2 NL <SCPI header> <common header> NL
–410	Query INTERRUPTED [Indicates that a condition causing an INTERRUPTED Query error occurred (see IEEE 488.2, 6.3.2.3); for example, a query followed by DAB or GET before a response was completely sent.]

Table 7-4. Query Errors (2 of 2)

Error Number	Error Description [description/explanation/examples]
	
–420	Query UNTERMINATED [Indicates that a condition causing an UNTERMINATED Query error occurred (see IEEE 488.2, 6.3.2.2); for example, the device was addressed to talk and an incomplete program message was received.]
–430	Query DEADLOCKED [Indicates that a condition causing an DEADLOCKED Query error occurred (see IEEE 488.2, 6.3.1.7); for example, both input buffer and output buffer are full and the device cannot continue.]
–440	Query UNTERMINATED after indefinite response [Indicates that a query was received in the same program message after an query requesting an indefinite response was executed (see IEEE 488.2, 6.5.7.5).]

Amplifier, Source Test, and Normalization Errors

Table 7-5. Amplifier, Source Test, Normalization Errors

Error Number	Error Description [description/explanation/examples]
–900	desc = “Amplifier unknown application error” help = ““ [Indicates that some unknown error occurred in the amplifier application. Retry again or restart the amplifier application.]

-901	desc = “Source not characterized”
	help = ““
	[Indicates that the source needs to be characterized before taking the amplifier sweep. Perform a source sweep to characterize the optical source signals.]

–902	desc = “Error accessing express object” help = ““ [Indicates that an internal error occurred while accessing the data objects.]
–903	desc = “Source Test not currently selected” help = ““ [Indicates that the source application related SCPI command is called without turning the source application on. Please turn ON the source application and retry the command.]
–904	desc = “Source Test option not supported” help = ““ [Indicates that all the legacy HP7145X source application commands are not supported. Please check the manuals and send the appropriate available source application legacy commands.]

–905	desc = “Normalization not valid”
	help = ““
	[Indicates that the normalization data is not valid. Trying to perform a measurement sweep without having valid normalization data causes this error. Also check the normalization data valid period.]

Analyzer Notices

System control-related error messages or warnings

The analyzer system changed a setting and generated a warning that the operation was performed.

Table 7-6. System Control Errors or Warnings

Error Number	Error Description [description/explanation/examples]
1000	desc = “Sensitivity forced to Auto” help = “Sensitivity has been forced to the Auto setting because another instrument setting was made which does not permit sensitivity to be in the Manual setting.”
1001	desc = “Old Mezzanine board in system” help = “This analyzer has an old mezzanine board installed. It is not necessary to upgrade the mezzanine board but the software will not be able to correctly determine which light source options are available.”
1002	desc = “No Delta Marker Amplitude for“ help = “The delta marker amplitude display has been turned off due to a units mismatch. The amplitude units of the reference marker are different than those of the delta marker. Values in differing units cannot be compared.”
1003	desc = “Trace length increased” help = “Trace length has been increased because the wavelength limit function was turned off. The minimum trace length is larger when the wavelength limit function is off and trace length was below the new minimum value.”
1004	desc = “Incomplete measurement file“ help = “An attempt was made to recall an incomplete measurement file. This may be caused by recalling a measurement file that was saved using an older version of firmware. All compatible measurement information was recalled.
2999	desc = “The Notices list has overflowed” help = “The Notices list has overflowed. The last entries received have been deleted.”

Analyzer Warnings

Table 7-7. Analyzer Warnings (1 of 12)

Error Number	Error Description [description/explanation/examples]
5000	desc = “AutoMeasure cannot find an input signal” help = “The auto-measure procedure cannot find a usable input signal. Make sure you have a signal connected to the optical input. Auto-measure will not work with very small input signals. They must be measured manually.”
5001	desc = “AutoAlign cannot find an input signal” help = “The auto-align procedure cannot find a usable input signal. Make sure you have a signal connected to the optical input.”
5002	desc = “Cal aborted: there is no active marker” help = “A wavelength calibration using the wavelength of the active marker was requested. The calibration cannot be made because all markers are turned off. Place a marker on the calibration signal and try again.”
5003	desc = “Cal aborted: trace resolution is inadequate” help = “A wavelength calibration using the wavelength of the active marker was requested. The calibration cannot be done because the resolution of the marker’s trace is inadequate. Trace resolution is defined as wavelength span divided by (trace points – 1) and must be less than 10pm. This problem can be corrected by one or more of the following steps: 1. Reduce wavelength span 2. Increase number of trace points”
5004	desc = “Cal aborted: wavelength correction too large” help = “A wavelength calibration was requested. The calibration was aborted since the correction needed is larger than 2nm. Make sure you have the correct index of refraction set. If the wavelength error is still this large the analyzer may need to be serviced. The wavelength calibration function is not intended to apply large arbitrary offsets. Use the wavelength offset function for this purpose.”

Table 7-7. Analyzer Warnings (2 of 12)

Error Number	Error Description [description/explanation/examples]
5005	desc = “Cal aborted: amplitude correction too large” help = “An amplitude calibration was requested. The calibration was aborted since the correction needed is more than +3dB or less than –10dB. Make sure you have done an Auto-Align prior to calibration. If the amplitude error is still this large the analyzer may need to be serviced. The amplitude calibration function is not intended to apply large arbitrary offsets. Use the amplitude offset function for this purpose.”
5006	desc = “Cal aborted: cannot find an input signal” help = “The calibration procedure cannot find an input signal close enough to the wavelength and/or amplitude specified. Make sure the wavelength and/or amplitude specified for calibration are correct and verify that an input signal of the correct wavelength and/or amplitude is connected to the optical input.”
5007	desc = “Calibration aborted: signal disappeared” help = “The calibration procedure found a signal which subsequently disappeared. Make sure the calibration signal is connected to the optical input and try again.”
5008	desc = “Calibration was not successful” help = “The requested calibration procedure did not complete successfully. Verify the instrument setup and the presence of a valid calibration signal. This problem can sometimes be caused by removing or changing the amplitude of the calibration signal during the calibration procedure. In rare cases this error can occur after the analyzer has received a severe mechanical shock. Try running AutoAlign”
5009	desc = “I/O error” help = “An error occurred while processing an input/output request”
5010	desc = “The floppy disk is full” help = “There is not enough free space left on the floppy disk to contain the new file(s). Either make room on the existing floppy by deleting unwanted files or try a different floppy disk.”
5011	desc = “There is no disk in the floppy disk drive” help = “An operation was requested which uses the floppy disk. There is no disk detected in the floppy disk drive. If there is a disk in the drive it may be of a format which cannot be read. This problem can be fixed by inserting a disk in the floppy disk drive or replacing the current disk with one which is properly formatted.”

Table 7-7. Analyzer Warnings (3 of 12)

Error Number	Error Description [description/explanation/examples]
5012	desc = “The floppy disk is not formatted” help = “An operation was requested which uses the floppy disk. The disk currently in the drive does not appear to be formatted. Use the format utility to format the disk or replace it with a formatted disk.”
5013	desc = “The floppy disk is write protected” help = “An operation was requested which writes to the floppy disk. The disk currently in the drive is write-protected. Turn off write protection on the current floppy disk or replace it with another disk.”
5014	desc = “The internal memory is full” help = “An operation was requested which uses internal memory. There is not enough free space left in internal memory for this operation. You must make room by deleting one or more files.”
5015	desc = “Up/down keys will not alter the span when it is zero” help = “The up/down step keys may not be used to alter the span setting when it is zero. A non-zero numeric value must be entered for span in order to leave zero span mode.”
5016	desc = “Incompatible measurement file” help = “An attempt was made to recall a measurement file which is incompatible with the current version of software.”
5017	desc = “The requested file does not exist” help = “An attempt was made to access a file which does not exist. Check the spelling of the file name against the listing of available files.”
5020–5030	desc = “I/O Error” help = “An unexpected error occurred during the I/O operation. Please try the operation again. If the operation involves the floppy disk drive, try a different floppy disk. If the error persists, please make a note of the error number and contact the nearest Agilent Technologies Instrument support center for assistance.
5031	desc = “Could not initialize floppy” help = “A request was made to initialize a floppy disk. The operation did not succeed. Check to see that there is a floppy disk inserted in the drive and that the disk is not write protected. This can also be caused by a defective floppy disk.”

Table 7-7. Analyzer Warnings (4 of 12)

Error Number	Error Description [description/explanation/examples]
5032–5043	desc = “I/O Error” help = “An unexpected error occurred during the I/O operation. Please try the operation again. If the operation involves the floppy disk drive, try a different floppy disk. If the error persists, please make a note of the error number and contact the nearest Agilent Technologies Instrument support center for assistance.
5044	desc = “Please cycle power to synchronize system time” help = “The system time clock has been set backwards. Due to internal system requirements it is necessary to cycle power before continuing. Please turn power off and back on again. This must be done to ensure proper system operation.”
5045	desc = “Sweep limit markers too close together” help = “The sweep limit function is on and the line markers are too close together. The analyzer cannot limit the sweep range to such a small value. The current sweep may be using a wider range than requested. Move the line markers farther apart or turn off sweep limit to resolve this problem.”
5046	desc = “AutoAlign cannot find input signal at marker” help = “The auto-align procedure could not find a usable input signal. If there is an active marker auto-align will attempt to align at the marker wavelength without searching for a signal peak. This error can also occur if the input signal amplitude is very low or if the analyzer has received a large mechanical shock or if the input signal is removed after the auto-align procedure has begun. Be sure a signal is connected to the optical input. If there is an active marker make sure it is placed on the input signal or turn off all markers.”
5047	desc = “Signal disappeared during AutoAlign” help = “The auto-align procedure did not complete. The input signal disappeared during the auto-align procedure. Make sure the input signal is connected to the optical input and try again.”
5048	desc = “Wrong trace X axis units for peak to center” help = “The peak-to-center function was requested. This function will only execute if the active trace X axis has units of meters (um nm pm). Change the active trace to one with an X axis in meters or re-measure the active trace in a non-zero span.”

Table 7-7. Analyzer Warnings (5 of 12)

Error Number	Error Description [description/explanation/examples]
5049	desc = “Wrong marker X axis units for active trace” help = “The active marker cannot be placed on the active trace because the desired X axis units do not match the X axis units of the active trace. This is normally caused by recalling a wavelength trace in zero span or a zero span trace in a wavelength span. It can also be caused by loading a trace with conflicting units by remote control.”
5050	desc = “Trace not displayed: wrong X axis units” help = “View has been turned on for this trace but it cannot be displayed. The X axis units in the trace do not match the X axis units being displayed. This is normally caused by trying to view a wavelength trace in zero span or a zero span trace in a wavelength span. It can also be caused by loading a trace with conflicting units by remote control.”
5051	desc = “AutoMeasure cannot find input signal at marker” help = “The auto-measure function was requested. The auto-measure at marker option was enabled but the active marker was not placed on a valid signal. A valid signal was found but the active marker is too far from that signal. To correct this problem place the marker on the desired signal turn off all markers or disable the auto-measure at marker function.”
5052	desc = “Sweep time auto is not allowed in zero span” help = “The sweep time function was requested to switch from manual to auto while in zero span. Automatic sweep time coupling cannot be done in zero span. The desired sweep time must be set explicitly.”
5053	desc = “Noise marker not allowed.” help = “A request was made to enable the noise marker readout. The readout cannot be enabled for one of the following reasons. Noise markers cannot be used on traces which have X axis units other than meters. This typically occurs in zero span where the X axis units are seconds. Noise markers cannot be used on traces which have Y axis units other than power. This typically happens when trace math is on and the trace Y axis represents a ratio.”
5054	desc = “Bandwidth markers are not allowed in zero span” help = “A request was made to enable the bandwidth marker readout while the analyzer was in zero span. Bandwidth markers are not permitted when span is set to zero. If a bandwidth marker readout is required set span to a value greater than zero.”
5055	desc = “Firmware Upgrade was not successful” help = “A firmware upgrade operation was requested. The firmware upgrade operation cannot be done at this time due to an internal software problem.

Table 7-7. Analyzer Warnings (6 of 12)

Error Number	Error Description [description/explanation/examples]
5056	desc = "Trajectory align cannot find input signal" help = "The trajectory align procedure cannot find a usable input signal. Make sure you have a signal connected to the optical input."
5057	desc = "Invalid settings for trajectory align" help = "The external trajectory align function was requested. It cannot be executed because the start and/or stop wavelength settings are invalid. Start and stop wavelengths must be between 600nm and 1700nm. In addition the difference between start and stop wavelengths (span) must be greater than 25nm. To correct this problem adjust the start and/or stop wavelength settings to be valid."
5058	desc = "Out of memory" help = "The analyzer has run out of execution memory. An internal function was aborted due to lack of execution memory. The correction for this problem is to cycle power."
5059	desc = "Trajectory align: marginal input signal" help = "The trajectory align procedure was aborted due to an input signal with marginal amplitude. The input signal was large enough when the function began it's operation but later became too small. This usually happens when the signal is on the edge of being too small. To correct this problem increase the input signal level. If this is not possible then a slight increase in signal level can sometimes be achieved by cleaning optical fiber connectors."
5060	desc = "Trajectory align failed" help = "The trajectory align procedure failed. The trajectory adjustments computed were invalid. This is usually occurs after the analyzer has received a large mechanical shock. To correct this problem try the trajectory align procedure again. If the error persists, contact the Agilent support center nearest your location."
5061	desc = "Invalid marker trace" help = "The marker could not be positioned. The marker is on a trace which does not contain any valid data. This is sometimes caused by a trace math result which is invalid."
5062	desc = "ADC Triggered Sweep Too Fast" help = "A sweep was taken with one of the ADC trigger modes enabled. The sweep rate was too fast to allow trace data to be acquired for every wavelength. To resolve this problem increase sweep time or increase the frequency the external trigger input signal."

Table 7-7. Analyzer Warnings (7 of 12)

Error Number	Error Description [description/explanation/examples]
5063	desc = “Key disabled during applications” help = “The key you pressed is not active while an application is running. Exiting the application should re-enable the key.”
5064	desc = “Invalid measurement file” help = “An attempt was made to recall a measurement file which is contains invalid or corrupt data.”
5065	desc = “Error detected in DSP sub-system” help = “The Digital Signal Processor has reported an unexpected error. Please record the hexadecimal number listed with the error and cycle power. If the error persists, contact the Agilent support center nearest your location.”
5066	desc = “Error in Enhanced Wavelength Calibration” help = “Enhanced Wavelength Calibration failed. Check OSA system. IF the error persists, contact the Agilent support center nearest your location.”
5067	desc = “OSNR marker not allowed” help = “A request was made to enable the OSNR marker readout. The readout cannot be enabled for one of the following reasons. OSNR markers cannot be used on traces which have X axis units other than meters. This typically occurs in zero span where the X axis units are seconds. OSNR markers cannot be used on traces which have Y axis units other than power. This typically happens when trace math is non and the trace Y axis represents a ratio.”
5068	desc = “Wrong trace X axis units for pit to center” help = “The pit-to-center function was requested. This function will only execute if the active trace X axis has units of meters (µm, nm, pm). Change the active trace to one with an X axis in meters or re-measure the active trace in a non-zero span.”
5069	desc = “No application found - Key disabled” help = “Currently no application loaded in the system.”
5070	desc = “Invalid Start & Stop wavelengths” help = “Invalid start and stop wavelengths. Please check start and stop wavelengths in Normalization setup screen.
5071	desc = “Invalid Normalization Step Size” help = “Normalization step size greater than span. Please check start, stop, and step size wavelengths in Normalization setup screen.”

Table 7-7. Analyzer Warnings (8 of 12)

Error Number	Error Description [description/explanation/examples]
6700	desc = “Math expression input parameter undefined.” help = “A math expression could not be evaluated because one or more input arguments are undefined. Please check the spelling of all input arguments.”
6701	desc = “Math expression input parameter has error.” help = “A math expression could not be evaluated because an error is present on one or more of the input arguments. The error must be cleared before the expression can be evaluated.”
6702	desc = “Math expression parameter has zero length.” help = “A math expression could not be evaluated because one or more input arguments has a zero length. This is often caused when the subset of a trace contains no points such as when advanced line marker functions are turned on and the line markers are too close together. It can also be caused when peak/pit searches find no peaks or pits.”
6720	desc = “Math expression input parameter has error.” help = “A math expression could not be evaluated because of improper input arguments. This could be due to one or more input arguments having the wrong type or size or because one or more inputs are not defined.”
6721	desc = “Internal error in marker search” help = “A marker search function has failed due to an internal software problem. Please try the operation again. If the error persists, contact the Agilent support center nearest your location.”
6722	desc = “Math expression input cannot be boolean.” help = “A math expression could not be evaluated because one or more of the input arguments is boolean. This function does not allow boolean input arguments.”
6723	desc = “Trace X axis values do not match.” help = “A math expression could not be evaluated because the input arguments have differing X axis values. This function requires that all input arguments have identical X axis values. All trace inputs must be taken with identical start/stop settings.”
6724	desc = “Error in source trace” help = “The source trace has an error. The error may possibly be that the source trace contains no points. Please try the operation again. If the error persists, contact the Agilent support center nearest your location.”

Table 7-7. Analyzer Warnings (9 of 12)

Error Number	Error Description [description/explanation/examples]
6725	desc = “Trace lengths do not match.” help = “A math expression could not be evaluated because the inputs have differing sizes (trace lengths). All inputs to this function must be of the same size.”
6726	desc = “Trace lengths do not match.” help = “A math expression could not be evaluated because the inputs have differing sizes (trace lengths). All inputs to this function must be of the same size.”
6727	desc = “Y axis counts do not match.” help = “A math expression could not be evaluated because the inputs have differing numbers of Y axis data. All input arguments to this function must have the same number of Y axis data points.”
6728	desc = “Incorrect number of inputs for math expression” help = “A math expression could not be evaluated because the number of input arguments to the function is incorrect.”
6729	desc = “Math expression expects units of dBm.” help = “A math expression could not be evaluated because the input argument does not have the required Y axis units of dBm.”
6730	desc = “Math expression expects units of watts.” help = “A math expression could not be evaluated because the input argument does not have the required Y axis units of watts.”
6731	desc = “Illegal combination of trace Y axis units” help = “A math expression could not be evaluated because the combination of Y axis units in the input arguments do not make sense. For example it is valid to divide watts by watts or to divide watts by a unitless value but it is invalid to divide a unitless value by watts.”
6732	desc = “Illegal combination of trace X axis units” help = “A math expression could not be evaluated because the input arguments have differing X axis units. All input arguments must have identical X axis units. This commonly happens when trace math is attempted between a zero span trace (time units) and a non-zero span trace (wavelength units).”

Table 7-7. Analyzer Warnings (10 of 12)

Error Number	Error Description [description/explanation/examples]
6733	desc = “Invalid combination of Y axis units” help = “A math expression could not be evaluated because of an invalid combination of Y axis units. The math operation being performed only allows one of the arguments to have units. All other arguments must be unitless. The math operator is shown in parenthesis in the short message above. For example multiplying a trace in watts (dBm) by another trace in watts (dBm) is not permitted.”
6734	desc = “Too many inputs have Y axis units” help = “A math expression could not be evaluated because of an invalid combination of inputs. Only one input argument is allowed to have non-blank Y axis units.”
6735	desc = “Trace has too many Y axes” help = “A math expression could not be evaluated because of an invalid input. One of the arguments contains multiple sets of Y data. Only one set of Y data is allowed for this math function.”
6736	desc = “Math expression expects boolean inputs.” help = “A math expression could not be evaluated because of an invalid input. The math function requires all of it’s arguments to be boolean. At least one of the arguments is not boolean.”
6737	desc = “Invalid math constant” help = “During evaluation of a math expression an invalid constant was encountered. The constant has undefined or default settings for it’s X and/or Y values.”
6738	desc = “Out of memory” help = “A trace or math operation was requested. There is insufficient memory available to perform the operation. Try reducing trace length or cycle power.”
6739	desc = “Too many peaks or pits” help = “A marker search operation was requested. There are too many valid pits or peaks in the current trace. Reduce the number of valid pits or peaks by increasing the peak or pit excursion setting.”
6741	desc = “Constant with multiple Y values” help = “During evaluation of a math expression a constant was encountered which contains more than one Y value. Constants are only permitted to have one Y value.”

Table 7-7. Analyzer Warnings (11 of 12)

Error Number	Error Description [description/explanation/examples]
6742	desc = “Requested amplitude not found” help = “A math expression to search for a specific amplitude in a trace did not succeed. There are no trace points with the desired amplitude.”
6744	desc = “Excursion should be in dB” help = “The units for the excursion are not in dB.”
6745	desc = “Log of a negative number is not allowed.” help = “During evaluation of a math expression the logarithm of a negative number was encountered.”
6746	desc = “Math expression contains a circular reference.” help = “A new math expression was entered. The expression was rejected because it would create a circular reference. For example if trace math for trace C is set to (A-B) then setting trace math for trace B to (C+D) would create a circular reference.”
6747	desc = “Next peak not found” help = “A next-peak search was requested. There are no more peaks on the current marker’s trace in the requested direction. To find additional peaks reduce the marker peak excursion setting or adjust sensitivity.”
6748	desc = “Next pit not found” help = “A next-pit search was requested. There are no more pits on the current marker’s trace in the requested direction. To find additional pits reduce the marker pit excursion setting or adjust sensitivity.”
6749	desc = “Peak not found” help = “A peak search was requested. There are no valid peaks on the current marker’s trace. Try decreasing the marker peak excursion setting or adjust the sensitivity setting.”
6750	desc = “Pit not found” help = “A pit search was requested. There are no valid pits on the current marker’s trace. Try decreasing the marker pit excursion setting or adjust the sensitivity setting.”
6751	desc = “Trace has no centroid” help = “The trace has no centroid. This may occur because the sum of trace points’ amplitudes is zero or there a no trace points.”
6752	desc = “The reference point is outside trace bounds” help = “The reference point’s wavelength is either too small or too large. The reference point will be clipped to a trace endpoint.”

Table 7-7. Analyzer Warnings (12 of 12)

Error Number	Error Description [description/explanation/examples]
6753	desc = “The window specified is invalid.” help = “The window specified is invalid. The window should be a constant.”
6754	desc = “Search failed: no data” help = “A marker search operation was requested. After clipping the trace data to screen limits and line marker limits (if enabled) there was no data to search. This can be caused when all trace data points are beyond current X axis screen limits or when there are no trace data points between the line markers. Trace data values which exceed the current Y axis screen limits will not cause this problem. Adjust the current X axis screen limits and/or move the line markers to include at least one trace data point.”
6755	desc = “Attempt to divide by 0” help = “At least one point was attempted to be divided by 0. The result for these divisions has been set to not-a-number. Please realize that further calculations with not-a-number values are undefined.”
6756	desc = “Syntax error; bad token: “ help = “The grammar expression entered cannot be parsed. Please check the string entered. The bad token attempts to indicate where the error occurred. The end of line indicates that the analyzer expected more information. Please consult the manual for additional help.”
7998	desc = “Unknown error detected” help = “An unlisted error was reported by the instrument software. If this error persists contact Agilent Technologies for assistance.”
7999	desc = “The warning list has overflowed” help = “The warning list has overflowed. The last entries received have been deleted.”

Application-Specific Warnings

Table 7-8. Application-Specific Warnings (1 of 6)

Error Number	Error Description [description/explanation/examples]
8001	desc = “Incorrect application type is listed in spec file.” help = “The application expects the first non-comment line of the specification file to contain the APPLICATION keyword followed by the application type. Either the APPLICATION keyword was missing or the wrong application type was specified for this particular application.”
8002	desc = “A keyword cannot be used as a variable name.” help = “The application has a list of keywords reserved for specification file use. These reserved words cannot be used as a name for one of the measurement parameters.”
8003	desc = “The variable name has already been used.” help = “The application allows new names to be assigned to measurement parameters. Once assigned these names can only be used as inputs to other measurements. Only the first ten characters are significant. The line number indicates where the problem was found. The word in the parentheses is the variable name causing the problem.”
8004	desc = “Wrong number of input parameters.” help = “The wrong number of inputs were found for a specification file command. The line number listed is where the problem was found in the specification file. The first number in parentheses is the number of inputs found and the second number is the number of inputs required.”
8005	desc = “An input parameter has the wrong units.” help = “One of the input parameters to the specification file has the wrong units specified. The line number listed is where the problem was found in the specification file. The first units listed in the parentheses was the units used by the parameter and the second units is the units required by the parameter. An empty pair of quotes ‘’ indicates unitless.”

Table 7-8. Application-Specific Warnings (2 of 6)

Error Number	Error Description [description/explanation/examples]
8006	desc = “The specification file cannot be imported.” help = “An error occurred while trying to import the specification file. Refer to the previous warnings for more information on specific errors in the specification file.”
8007	desc = “The specification file has a syntax error.” help = “One of the specification file lines has a syntax problem. The problem occurred on the line number listed. The string in the parentheses caused the problem.”
8008	desc = “A variable was used where a constant is expected.” help = “A variable name was used as a parameter where only a constant value or the DEFAULT keyword is allowed. The line number indicates where the problem was found. The variable name in the parentheses caused the problem.”
8009	desc = “The variable name has not been defined.” help = “An undefined variable name was used as a parameter for a specification. All variables used as parameters must be defined in a specification statement preceding the variable’s use. The line number indicates where the problem was found. The variable name in the parentheses is the undefined variable.”
8010	desc = “A variable can be used only once per statement.” help = “A variable name was used as multiple parameters for the same specification statement. The input parameter was ignored because using multiple references is not allowed. The line number indicates where the problem was found. The variable name in the parentheses is the variable causing the problem.”
8011	desc = “A keyword was used where a variable is expected.” help = “The application specification file has a list of reserved words which are used to specify the measurements to be made. These keywords cannot be used as input parameters for specification statements. The input parameter was ignored for this case. The line number indicates where the problem was found. The word in the parentheses is the reserved word causing the problem.”
8012	desc = “A label is required to identify the spec file.” help = “The label string is used to identify the specification file. It is required after specifying the application type. An empty string (pair of double quotes) is acceptable input.”
8013	desc = “The specification file cannot be exported.” help = “An error occurred while trying to export the specification file.”

Table 7-8. Application-Specific Warnings (3 of 6)

Error Number	Error Description [description/explanation/examples]
8014	desc = “Print statement ignored: no path is specified” help = “The PRINT statement needs to be after a PATH statement to indicate which PATH data is to be printed. The PRINT statement at the line number specified came before any PATH statement. This PRINT statement will not trigger any printouts.”
8015	desc = “DEFAULT cannot be used for a parameter.” help = “A parameter cannot be specified as DEFAULT. The parameter needs to be either a previously defined variable name or an actual value. The bad parameter was found on the line number specified. The number in the parentheses is the number of the parameter causing the problem.”
8016	desc = “A duplicate command is in the spec file.” help = “Certain commands should be used only once in the specification file. The duplicate command was found on the line number specified. The command in the parentheses is the duplicate.”
8017	desc = “A required command is missing from the spec file.” help = “Certain commands must be used in the specification file. The missing command is listed in the parentheses. This command should be used only once in the file.”
8018	desc = “The normalization interval limited to maximum.” help = “The normalization interval has a maximum value of 24 and a minimum value of 0.1 hours. The interval specified in the specification file is outside this range. The specification file can still be used but the calibration interval will be 24 hours.”
8019	desc = “The spec file could not be loaded.” help = “The application found a problem with the specification file. Possible causes can include the following: the file is incompatible with the application the file is from a previous version of the application the file was corrupted. The file with the problem is specified within the parentheses. Try re-importing the specification file.”
8020	desc = “The default spec file was loaded.” help = “The default specification file for the application was loaded. This is done when the application is first started or if a problem occurred when trying to load another specification file. Refer to the previous warnings for information on any problems loading another file.”

Table 7-8. Application-Specific Warnings (4 of 6)

Error Number	Error Description [description/explanation/examples]
8021	desc = “The spec file could not be found.” help = “The application tried to load a specification file which could not be found in the internal memory. This may have occurred if the last file used by the application was deleted. The filename is specified within the parentheses.”
8022	desc = “A closing quote is missing.” help = “The specification file contained a line with an opening quote but no closing quote could be found.”
8023	desc = “A default path name was used.” help = “The imported specification file was given a default path name. All specification files must have at least one PATH statement before any of the measurement statements. A default path name was used either because some measurement statements were used before a PATH statement or there was no PATH statement at all in the file.”
8024	desc = “A variable is expected.” help = “The statement expected a variable as an input parameter. The line number indicates which statement did not have the proper input parameter type.”
8025	desc = “The path name has already been used.” help = “The application allows new names to be assigned to measurement paths. Once assigned these names can no longer be used by other measurement paths. Only the first thirty-two characters are significant. The line number indicates where the problem was found. The word in the parentheses is the path name causing the problem.”
8026	desc = “A path name was used where a variable is expected.” help = “The input parameters for measurement statements can take variables as inputs. These variables are the results of other measurement statements. A path name is the name of a measurement path not the result of a measurement and cannot be used as an input parameter. The input parameter was ignored for this case. The line number indicates where the problem was found. The word in the parentheses is the path name causing the problem.”
8027	desc = “A parameter value was outside the legal range.” help = “A parameter value was used which was outside the allowable range. The first number in the parentheses is the number of the parameter causing the problem. The other two numbers are the minimum and maximum values allowed. These values are in base units (e.g meters watts).”

Table 7-8. Application-Specific Warnings (5 of 6)

Error Number	Error Description [description/explanation/examples]
8028	desc = “A minimum non-zero span is required.” help = “The application cannot run in a zero span setting. The start and stop wavelength must be separated by a minimum span. The line number indicates the line in the specification file which had a start and stop wavelength pair that did not meet the minimum span requirement.”
8029	desc = “The requested function is not yet implemented.” help = “The statement within the specification file is not implemented in this version of the application. The word in parentheses has been reserved for future use. The statement at the specified line number needs to be removed before the specification file can be imported.”
8030	desc = “A SETUP statement is needed before SWEEP.” help = “The wavelength limits of the SWEEP statement are checked against the wavelength limits in the SETUP statement. This check requires that the SETUP statement occur before the SWEEP statement. The line number indicates which SWEEP statement is not preceded by the SETUP statement.”
8031	desc = “The SWEEP limits are outside the SETUP range.” help = “The start and stop values for the SWEEP statement must be within the start and stop values of the SETUP statement. The line number indicates which SWEEP statement has values out of range. The number in the parentheses indicates which parameter is out of range.”
8032	desc = “The units of the inputs are not compatible.” help = “The units of the inputs to a math operation need to be compatible with the operation desired. Addition and subtraction need to have matching units. Multiplication needs to have a unitless parameter. Division needs a unitless denominator or the units for the numerator and denominator need to match. The line number indicates which statement had the mismatched input units.”
8033	desc = “The reference point is outside the SETUP range.” help = “The reference point value must be within the start and stop values of the SETUP statement. The line number indicates which statement had the value out of range.”
8034	desc = “The values are associated with different axes.” help = “The variables used in a basic math operation like ADD or DIV must be associated with the same data axis (e.g. wavelength or amplitude). Mixing the axes values within a single math operation is not allowed. The line number indicates which statement had the incompatible variables.”

Table 7-8. Application-Specific Warnings (6 of 6)

Error Number	Error Description [description/explanation/examples]
8035	desc = “The search limits are outside the SETUP range.” help = “The search limits for statements like PEAK or CENTER_OF_MASS must be within the start and stop values of the SETUP statement. The line number indicates which statement has values out of range. The number in the parentheses indicates which parameter is out of range.”
8036	desc = “The specification units do not match.” help = “The units for the minimum and maximum specification values need to match. The line number indicates which statement has the mismatching units.”
8037	desc = “The specification units do not match the input units.” help = “The units for the minimum and maximum specification values need to match the units for the input parameters. The line number indicates which statement has the mismatching units.”

Analyzer Status Errors

Table 7-9. Analyzer Status Errors

Error Number	Error Description [description/explanation/examples]
10000	desc = “Sweep Uncalibrated” help = “The current setting of sweep time may be too fast. This could result in an invalid measurement. In certain cases it is possible to sweep faster than the coupled sweep time without compromising measurement accuracy. Please refer to the users manual and relevant application notes for more information.”
11998	desc = “Too Many Hardware Status Errors” help = “The Hardware Status Error list has overflowed. Additional more recent items have been deleted.”

Analyzer Errors

Table 7-10. Analyzer Errors (1 of 2)

Error Number	Error Description [description/explanation/examples]
20001	desc = “Error detected in ADC sub-system” help = “An error has been detected in the Analog-to-Digital converter subsystem. Please record the hexadecimal number listed with the error and cycle power. If the error persists, contact the nearest Agilent Technologies Instrument support center for assistance.
20002	desc = “Error detected in slit positioning system” help = “An error has been detected in the motor which controls the resolution bandwidth slit wheel. Please record the hexadecimal number listed with the error and cycle power. If the error persists, contact the nearest Agilent Technologies Instrument support center for assistance.
20003	desc = “Error detected in grating positioning system” help = “An error has been detected in the motor which controls the diffraction grating. Please record the hexadecimal number listed with the error and cycle power. If the error persists, contact the nearest Agilent Technologies Instrument support center for assistance.
20005	desc = “Trans-Impedance calibration data is invalid” help = “Factory calibration data for the trans-impedance amplifier is invalid. Please record the hexadecimal number listed with the error and cycle power. If the error persists, contact the nearest Agilent Technologies Instrument support center for assistance.
20007	desc = “Sweep Timeout” help = “A sweep was started but did not finish in the expected amount of time. The trace data acquired during this sweep may not be valid. Try taking another sweep. If the error persists, contact the nearest Agilent Technologies Instrument support center for assistance.

Table 7-10. Analyzer Errors (2 of 2)

Error Number	Error Description [description/explanation/examples]
20100	Desc = “Communication problem with the TLS“ help = “The receiver is unable to communicate with the Agilent 8164A Tunable Laser Source. Check the TLS power and that the GPIB cable is connected between the TLS and the auxiliary GPIB port on the receiver. If this error persists, contact the nearest Agilent Service Center for assistance.”
21999	desc = “The Error list has overflowed” help = “The Error list has overflowed. The last entries received have been deleted.”

Firmware Errors

Table 7-11. Firmware Errors

Error Number	Error Description [description/explanation/examples]
30000	desc = “Internal Communications Error” help = “An internal software error has occurred involving communications between different software processes. Please record this error including the extra text and cycle power. If the error persists, contact the nearest Agilent Technologies Instrument support center for assistance.
30001	desc = “Auto-Measure Software Error” help = “An error has been detected in the auto-measure software. There is an internal problem with the software. Please make a note of the text in parentheses at the end of the error message and cycle power. If the error persists, contact the nearest Agilent Technologies Instrument support center for assistance.

8

Maintenance

Troubleshooting

Start-up Process

Running the power on self-test cycle will often provide valuable error information to help isolate problems with the TLS and WDCA. Cycling the power on both instruments will force a startup process on the system. The TLS has a longer startup cycle and will cause the WDCA to wait until the TLS is active.

The WDCA begins it’s startup process by executing a routine to test for the presence of the TLS on the system GPIB. Once the TLS is identified and its firmware version verified as required for operation within the WDCA system, the TLS is triggered and the trigger signal observed. If the trigger is detected, the TLS is set to sweep and, again, the trigger signal is detected. This sequence of operations verifies the GPIB and trigger conditions as well as operation between the two system components. If the TLS needs a re-align procedure, there will be a prompt message displayed on the TLS front panel.

Once the TLS is recognized by the WDCA and the electrical cable connections verified, the WDCA will begin its boot process. The system will initialize with the default settings, see “Default values” on page 2-44. You can then initiate a single sweep to see if the two instruments will communicate with each other.

Error Messages

The TLS performs an extensive self-test during startup, measuring power and wavelength to verify proper operation. Any errors will be reported to the WDCA and a message will appear on the display directing you to the errors on the display of the TLS.

The instruments will report errors individually during the self test. A failure in the laser will usually generate an error message that will be reported by the WDCA. Failures in the WDCA will be reported in the usual error message display area on the lower left of the display and in the error queue located under the system/help menus. The error queue contains messages that are briefly flashed on the WDCA display. The queue will store the messages until the queue is cleared by the clear queue key or by reading all the errors remotely. “Help” on page 2-21 offers more information on how to access or clear system errors.

Chapter 7, “Status Listings”, gives detailed information on all the messages and errors that may be reported.

Isolating Instrument Problems

Try to determine which of the two system components, the TLS or the WDCA, is causing the problem. The TLS and the WDCA are calibrated independently with no requirement to be tested together as a matched pair. The only requirement is for the TLS to have the necessary level of firmware to be identified by and function with the WDCA. If possible, substitute replacement instruments and repeat the measurement.

- Be certain you are not observing a problem caused by dirty connectors. Thoroughly clean all connections and repeat the measurement. For more information see “Cleaning Connections for Accurate Measurements” on page 8-11.
- Verify the connections between the instruments. There must be an optical cable connection between the TLS output and the WDCA input, a GPIB cable connecting the TLS to the auxiliary GPIB port on the WDCA, and a BNC cable connecting the TLS output trigger and the WDCA trigger input. Refer to “Front Panel Configuration” on page 1-34 and to “Rear Panel Configuration” on page 1-35 for more information.

Amplitude related errors

You can isolate amplitude related errors by connecting a fiber optic cable between the TLS output and the WDCA input 1. Use an FC-PC (WDCA) and an FC/APC (TLS) single mode optical fiber.

- In continuous sweep mode the output from the laser should be a stable -10 dBm signal. A missing or erratic signal would indicate a problem with the fiber or either system component.
- Perform a normalization with the fiber optic cable in place. A flat line should appear at the reference level. With the scale per division changed to 0.1dB, you should observe a flat line with no variation.
- Connect the TLS output to a power meter and observe the power level. If the power remains constant, this indicates the WDCA is causing the error.

Wavelength related errors

You can isolate wavelength related errors by performing a wavelength calibration using the internal calibration cell.

1. Connect a fiber optic cable between Output 2 (high power laser output) of the TLS and the source input of the WDCA.
2. Select System > Calibration and perform the wavelength calibration.
3. If the calibration results are off by more than 20 pm, an error will be generated. This error would indicate that the TLS has shifted wavelength and is causing the error.

Changing the Printer Paper

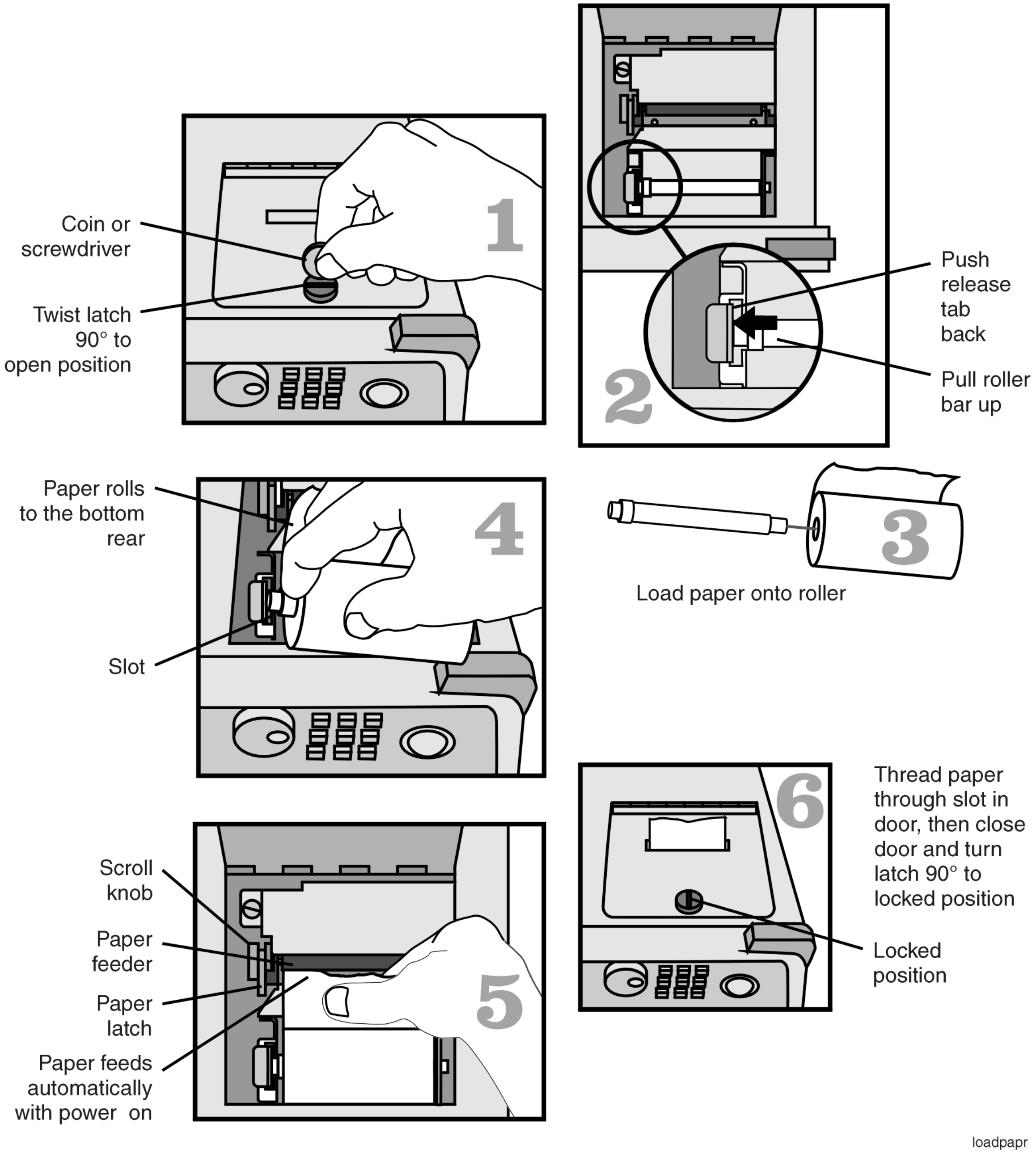
CAUTION Avoid dropping the coin or screwdriver, used to open the printer door, into the printer assembly.

CAUTION Always use Agilent brand paper to ensure quality printing and long printer life. Order paper as Agilent part number 9270-1370.

CAUTION Never use the printer without printer paper as this can damage the printer head.

- If the instrument is on, the paper feeder automatically scrolls whenever paper is inserted into the feeder.
- Lift the paper latch to correct paper alignment problems. The scroll knob allows you to manually advance the paper.

CAUTION To avoid damage caused by losing parts and tools inside the instrument, always turn the instrument off *before* installing the printer paper.


1
Coin or screwdriver
Twist latch 90° to open position
2
Push release tab back
Pull roller bar up
3
Load paper onto roller
4
Paper rolls to the bottom rear
Slot
5
Scroll knob
Paper feeder
Paper latch
Paper feeds automatically with power on
6
Thread paper through slot in door, then close door and turn latch 90° to locked position
Locked position
loadpapr

Printer Head Cleaning Procedure

Lint from normal use of the printer may eventually collect on the printer head and degrade print quality. Use the procedure provided in this section to clean the printer head.

WARNING **This servicing procedure is for use by qualified personnel only. To avoid electrical shock, do not perform this procedure unless you are qualified to do so.**

CAUTION Electrostatic discharge (ESD) can damage or destroy the printer. Therefore, this procedure must always be performed at a static-safe work station.

Static-safe Work Station

Figure 8-1 on page 8-8 shows an example of a static-safe work station using conductive table and floor mats and wrist and heel straps. To ensure user safety, the static-safe accessories must provide at least 1 MΩ of isolation from ground. Refer to Table 8-1 for information on ordering static-safe accessories.

Table 8-1. Static-Safe Accessories

Agilent Part Number	Description
9300-0797	3M static control mat 0.6 m × 1.2 m (2 ft× 4 ft) and 4.6 cm (15 ft) ground wire. (The wrist-strap and wrist-strap cord are not included. They must be ordered separately.)
9300-0980	Wrist-strap cord 1.5 m (5 ft).

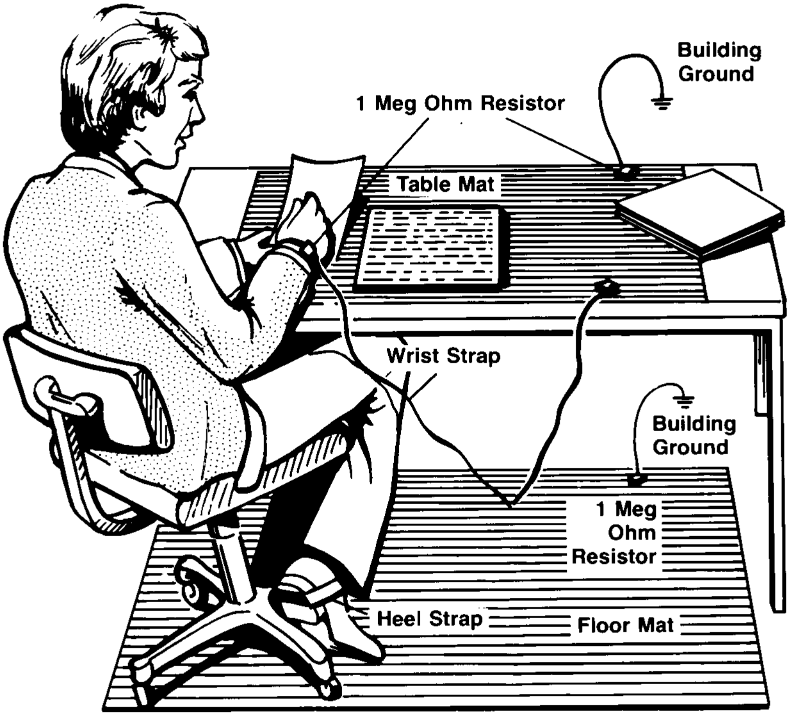


Figure 8-1. Example of a static-safe workstation

To clean the printer head

Table 8-2. Printer Accessories

Agilent Part Number	Description
9270-1605	Printer Paper

1 Turn off the analyzer, and remove the line power cord.

2 Place the instrument at a static-safe work station as described in the introduction to this procedure.

3 Use a coin or screwdriver to open the printer door that is located on the top of the instrument.

CAUTION Avoid dropping the coin or screwdriver, used to open the printer door, into the printer assembly.

4 Lift up the paper latch as shown in the following diagram, and remove the paper.

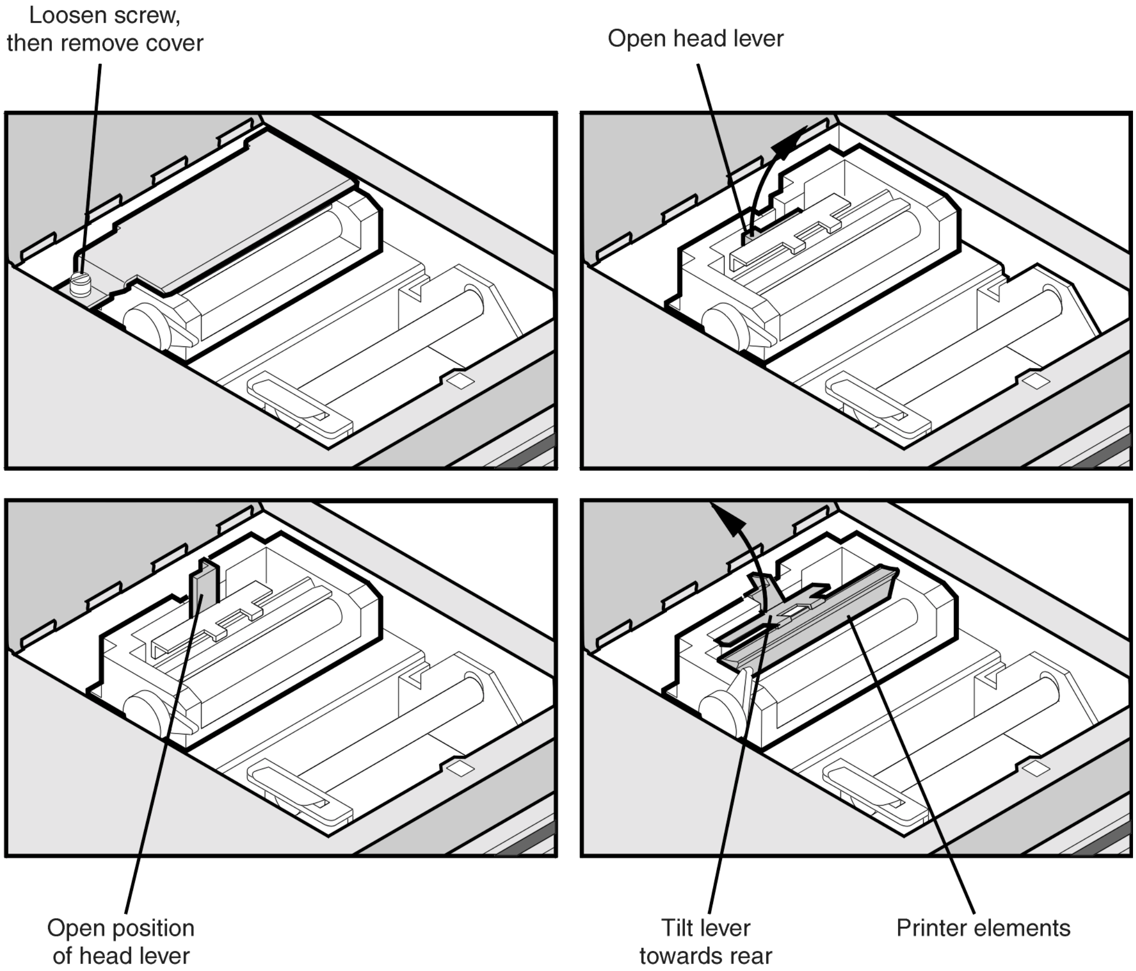


Figure 8-2. Cleaning the Printer Head

5 Unscrew the retaining screw that secures the sheet-metal cover that protects the printer head from electrostatic discharge. Slide the sheet-metal cover

towards the retaining screw and then lift it straight up to remove.

6 Lift the printer head lever to the vertical position. Then, tilt the lever towards the instrument’s rear panel to rotate the printer head up.

7 Clean the printer head using a cotton swab and isopropyl alcohol.

CAUTION Use of other cleaning materials or fluids may damage the printer.

8 After the printer head has thoroughly dried, use the printer head lever to return the printer head to its original position.

9 Replace and secure the sheet-metal cover for the printer head.

10 Replace the printer paper, and close the printer access door.

Cleaning Connections for Accurate Measurements

Today, advances in measurement capabilities make connectors and connection techniques more important than ever. Damage to the connectors on calibration and verification devices, test ports, cables, and other devices can degrade measurement accuracy and damage instruments. Replacing a damaged connector can cost thousands of dollars, not to mention lost time! This expense can be avoided by observing the simple precautions presented in this book. This book also contains a brief list of tips for caring for electrical connectors.

Choosing the Right Connector

A critical but often overlooked factor in making a good lightwave measurement is the selection of the fiber-optic connector. The differences in connector types are mainly in the mechanical assembly that holds the ferrule in position against another identical ferrule. Connectors also vary in the polish, curve, and concentricity of the core within the cladding. Mating one style of cable to another requires an adapter. Agilent Technologies offers adapters for most instruments to allow testing with many different cables. Figure 8-3 on page 8-12 shows the basic components of a typical connectors.

The system tolerance for reflection and insertion loss must be known when selecting a connector from the wide variety of currently available connectors. Some items to consider when selecting a connector are:

- How much insertion loss can be allowed?
- Will the connector need to make multiple connections? Some connectors are better than others, and some are very poor for making repeated connections.
- What is the reflection tolerance? Can the system take reflection degradation?

- Is an instrument-grade connector with a precision core alignment required?
- Is repeatability tolerance for reflection and loss important? Do your specifications take repeatability uncertainty into account?
- Will a connector degrade the return loss too much, or will a fusion splice be required? For example, many DFB lasers cannot operate with reflections from connectors. Often as much as 90 dB isolation is needed.

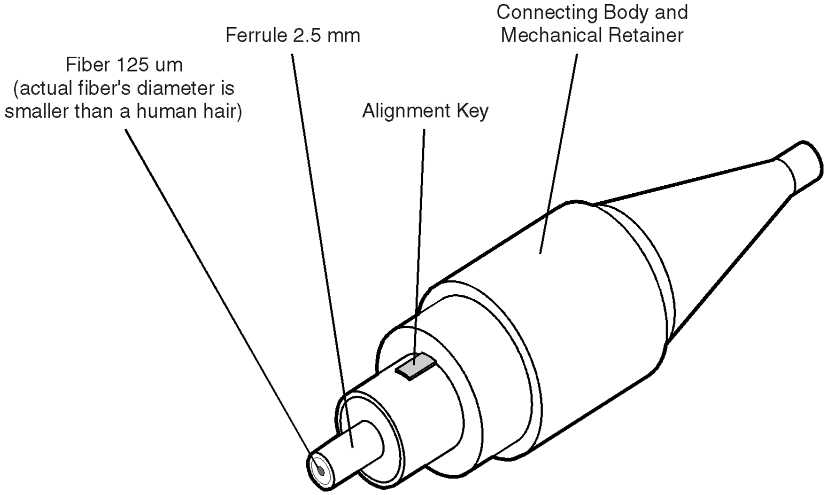

Connecting Body and Mechanical Retainer
Ferrule 2.5 mm
Fiber 125 um (actual fiber's diameter is smaller than a human hair)
Alignment Key

Figure 8-3. Basic components of a connector.

Over the last few years, the FC/PC style connector has emerged as the most popular connector for fiber-optic applications. While not the highest performing connector, it represents a good compromise between performance, reliability, and cost. If properly maintained and cleaned, this connector can withstand many repeated connections.

However, many instrument specifications require tighter tolerances than most connectors, including the FC/PC style, can deliver. These instruments cannot tolerate connectors with the large non-concentricities of the fiber common with ceramic style ferrules. When tighter alignment is required, Agilent Technologies instruments typically use a connector such as the Diamond HMS-10, which has concentric tolerances within a few tenths of a micron. Agilent Technologies then uses a special universal adapter, which allows other cable types to mate with this precision connector. See Figure 8-4.

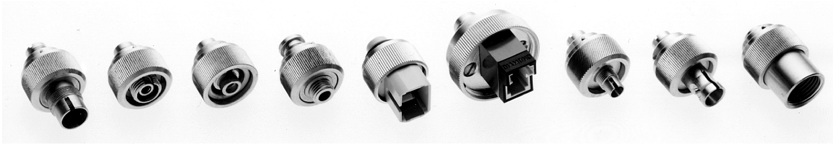


Figure 8-4. Universal adapters to Diamond HMS-10.

The HMS-10 encases the fiber within a soft nickel silver (Cu/Ni/Zn) center which is surrounded by a tough tungsten carbide casing, as shown in Figure 8-5.

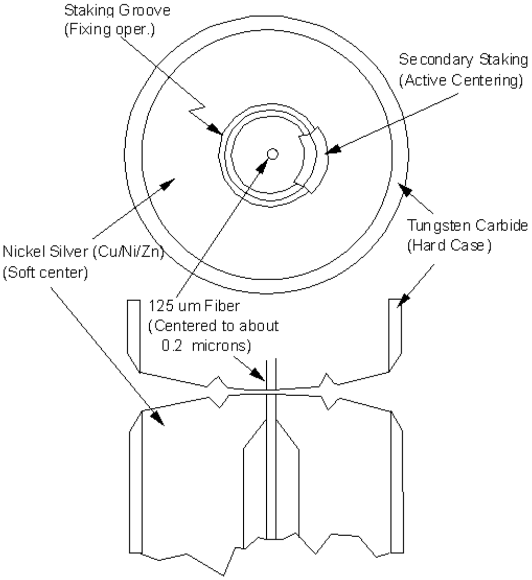


Figure 8-5. Cross-section of the Diamond HMS-10 connector.

The nickel silver allows an active centering process that permits the glass fiber to be moved to the desired position. This process first stakes the soft nickel silver to fix the fiber in a near-center location, then uses a post-active staking to shift the fiber into the desired position within 0.2 µm. This process, plus the keyed axis, allows very precise core-to-core alignments. This connector is found on most Agilent Technologies lightwave instruments.

The soft core, while allowing precise centering, is also the chief liability of the connector. The soft material is easily damaged. Care must be taken to minimize excessive scratching and wear. While minor wear is not a problem if the glass face is not affected, scratches or grit can cause the glass fiber to move out of alignment. Also, if unkeyed connectors are used, the nickel silver can be pushed onto the glass surface. Scratches, fiber movement, or glass contamination will cause loss of signal and increased reflections, resulting in poor return loss.

Inspecting Connectors

Because fiber-optic connectors are susceptible to damage that is not immediately obvious to the naked eye, poor measurements result without the user being aware. Microscopic examination and return loss measurements are the best way to ensure good measurements. Good cleaning practices can help ensure that optimum connector performance is maintained. With glass-to-glass interfaces, any degradation of a ferrule or the end of the fiber, any stray particles, or finger oil can have a significant effect on connector performance. Where many repeat connections are required, use of a connector saver or patch cable is recommended.

Figure 8-6 shows the end of a clean fiber-optic cable. The dark circle in the center of the micrograph is the fiber's 125 μm core and cladding which carries the light. The surrounding area is the soft nickel-silver ferrule. Figure 8-7 shows a dirty fiber end from neglect or perhaps improper cleaning. Material is smeared and ground into the end of the fiber causing light scattering and poor reflection. Not only is the precision polish lost, but this action can grind off the glass face and destroy the connector.

Figure 8-8 shows physical damage to the glass fiber end caused by either repeated connections made without removing loose particles or using improper cleaning tools. When severe, the damage of one connector end can be transferred to another good connector endface that comes in contact with the damaged one. Periodic checks of fiber ends, and replacing connecting cables after many connections is a wise practice.

The cure for these problems is disciplined connector care as described in the following list and in “Cleaning Connectors” on page 8-18.

Use the following guidelines to achieve the best possible performance when making measurements on a fiber-optic system:

- Never use metal or sharp objects to clean a connector and never scrape the connector.
- Avoid matching gel and oils.

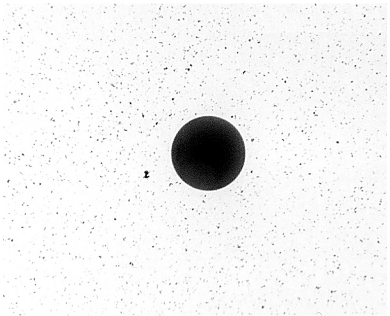


Figure 8-6. Clean, problem-free fiber end and ferrule.

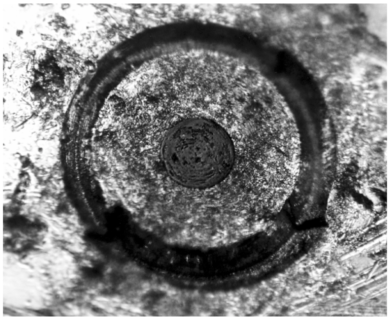


Figure 8-7. Dirty fiber end and ferrule from poor cleaning.

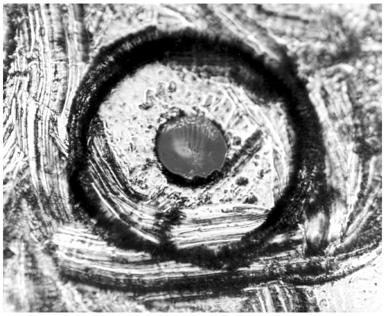


Figure 8-8. Damage from improper cleaning.

While these often work well on first insertion, they are great dirt magnets. The oil or gel grabs and holds grit that is then ground into the end of the fiber. Also, some early gels were designed for use with the FC, non-contacting connectors, using small glass spheres. When used with contacting connectors, these glass balls can scratch and pit the fiber. If an index matching gel or oil must be used, apply it to a freshly cleaned connector, make the measurement, and then immediately clean it off. Never use a gel for longer-term connections and never use it to improve a damaged connector. The gel can mask the extent of damage and continued use of a damaged fiber can transfer damage to the instrument.

- When inserting a fiber-optic cable into a connector, gently insert it in as straight a line as possible. Tipping and inserting at an angle can scrape material off the inside of the connector or even break the inside sleeve of connectors made with ceramic material.
- When inserting a fiber-optic connector into a connector, make sure that the fiber end does not touch the outside of the mating connector or adapter.
- Avoid over tightening connections.

  Unlike common electrical connections, tighter is *not* better. The purpose of the connector is to bring two fiber ends together. Once they touch, tightening only causes a greater force to be applied to the delicate fibers. With connectors that have a convex fiber end, the end can be pushed off-axis resulting in misalignment and excessive return loss. Many measurements are actually improved by backing off the connector pressure. Also, if a piece of grit does happen to get by the cleaning procedure, the tighter connection is more likely to damage the glass. Tighten the connectors just until the two fibers touch.

- Keep connectors covered when not in use.
- Use fusion splices on the more permanent critical nodes. Choose the best connector possible. Replace connecting cables regularly. Frequently measure the return loss of the connector to check for degradation, and clean every connector, every time.

All connectors should be treated like the high-quality lens of a good camera. The weak link in instrument and system reliability is often the inappropriate use and care of the connector. Because current connectors are so easy to use, there tends to be reduced vigilance in connector care and cleaning. It takes only one missed cleaning for a piece of grit to permanently damage the glass and ruin the connector.

Measuring insertion loss and return loss

Consistent measurements with your lightwave equipment are a good indication that you have good connections. Since return loss and insertion loss are key factors in determining optical connector performance they can be used to determine connector degradation. A smooth, polished fiber end should produce a good return-loss measurement. The quality of the polish establishes the difference between the “PC” (physical contact) and the “Super PC” connectors. Most connectors today are physical contact which make glass-to-glass connections, therefore it is critical that the area around the glass core be clean and free of scratches. Although the major area of a connector, excluding the glass, may show scratches and wear, if the glass has maintained its polished smoothness, the connector can still provide a good low level return loss connection.

If you test your cables and accessories for insertion loss and return loss upon receipt, and retain the measured data for comparison, you will be able to tell in the future if any degradation has occurred. Typical values are less than 0.5 dB of loss, and sometimes as little as 0.1 dB of loss with high performance connectors. Return loss is a measure of reflection: the less reflection the better (the larger the return loss, the smaller the reflection). The best physically contacting connectors have return losses better than 50 dB, although 30 to 40 dB is more common.

Visual inspection of fiber ends

Visual inspection of fiber ends can be helpful. Contamination or imperfections on the cable end face can be detected as well as cracks or chips in the fiber itself. Use a microscope (100X to 200X magnification) to inspect the entire end face for contamination, raised metal, or dents in the metal as well as any other imperfections. Inspect the fiber for cracks and chips. Visible imperfections not touching the fiber core may not affect performance (unless the imperfections keep the fibers from contacting).

WARNING

Always remove both ends of fiber-optic cables from any instrument, system, or device before visually inspecting the fiber ends. Disable all optical sources before disconnecting fiber-optic cables. Failure to do so may result in permanent injury to your eyes.

Cleaning Connectors

The procedures in this section provide the proper steps for cleaning fiber-optic cables and Agilent Technologies universal adapters. The initial cleaning, using the alcohol as a solvent, gently removes any grit and oil. If a caked-on layer of material is still present, (this can happen if the beryllium-copper sides of the ferrule retainer get scraped and deposited on the end of the fiber during insertion of the cable), a second cleaning should be performed. It is not uncommon for a cable or connector to require more than one cleaning.

CAUTION

Agilent Technologies strongly recommends that index matching compounds *not* be applied to their instruments and accessories. Some compounds, such as gels, may be difficult to remove and can contain damaging particulates. If you think the use of such compounds is necessary, refer to the compound manufacturer for information on application and cleaning procedures.

Table 8-3. Cleaning Accessories

Item	Agilent Part Number
Cotton swabs	8520-0023
Small foam swabs	9300-1223

Table 8-4. Dust Caps Provided with Lightwave Instruments

Item	Agilent Part Number
Laser shutter cap	08145-64521
FC/PC dust cap	08154-44102
Biconic dust cap	08154-44105
ST dust cover	1401-0291

To clean a non-lensed connector

CAUTION Do not use any type of foam swab to clean optical fiber ends. Foam swabs can leave filmy deposits on fiber ends that can degrade performance.

1 Apply pure isopropyl alcohol to a clean lint-free cotton swab or lens paper.

Cotton swabs can be used as long as no cotton fibers remain on the fiber end after cleaning.

2 Clean the ferrules and other parts of the connector while avoiding the end of the fiber.

3 Apply isopropyl alcohol to a new clean lint-free cotton swab or lens paper.

4 Clean the fiber end with the swab or lens paper.

Do *not* scrub during this initial cleaning because grit can be caught in the swab and become a gouging element.

5 Immediately dry the fiber end with a clean, dry, lint-free cotton swab or lens paper.

6 Blow across the connector end face from a distance of 6 to 8 inches using filtered, dry, compressed air. Aim the compressed air at a shallow angle to the fiber end face.

Nitrogen gas or compressed dust remover can also be used.

CAUTION Do not shake, tip, or invert compressed air canisters, because this releases particles in the can into the air. Refer to instructions provided on the compressed air canister.

7 As soon as the connector is dry, connect or cover it for later use.

If the performance, after the initial cleaning, seems poor try cleaning the connector again. Often a second cleaning will restore proper performance. The second cleaning should be more arduous with a scrubbing action.

To clean an adapter

The fiber-optic input and output connectors on many Agilent Technologies instruments employ a universal adapter such as those shown in the following picture. These adapters allow you to connect the instrument to different types of fiber-optic cables.

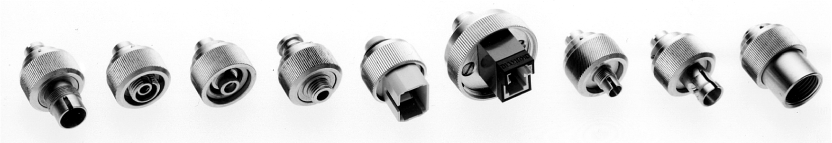


Figure 8-9. Universal adapters.

1 Apply isopropyl alcohol to a clean foam swab.

Cotton swabs can be used as long as no cotton fibers remain after cleaning. The foam swabs listed in this section's introduction are small enough to fit into adapters.

Although foam swabs can leave filmy deposits, these deposits are very thin, and the risk of other contamination buildup on the inside of adapters greatly outweighs the risk of contamination by foam swabs.

2 Clean the adapter with the foam swab.

3 Dry the inside of the adapter with a clean, dry, foam swab.

4 Blow through the adapter using filtered, dry, compressed air.

Nitrogen gas or compressed dust remover can also be used. Do not shake, tip, or invert compressed air canisters, because this releases particles in the can into the air. Refer to instructions provided on the compressed air canister.

Care of Connector Savers

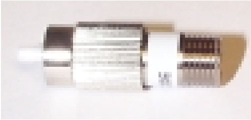
The connector saver is used to protect the input connector of the analyzer from damage. It functions as a standoff between the front panel input connector and the input fiber. If the fibers are not thoroughly cleaned, repeated connections can result in a scratched, chipped, or dirty input connector. Contamination, or damage to the connector, will cause loss of signal and increased reflections resulting in poor return loss and measurement error. With glass-to-glass interfaces, any degradation of a ferrule or fiber endface, any stray particles, or finger oil on the endface can have a significant effect on connector performance. Many times an instrument must be serviced to replace a damaged connector when thousands of dollars and lost time could have been avoided if better care were given to the fiber optic connector.

The connector saver is intended to serve as a replaceable interface. When cleaned and attached to the front panel, it should remain in place and successive input fiber changes should be attached to the connector saver. If the connector saver is damaged, you can easily replace it. Connector saver damage can be detected by examining the connector saver with a fiber scope, a reflectometer, or substituting a new connector saver (and observing a difference in the measurement).

The connector saver should be cleaned before any connection. While the connector saver is provided to protect the front panel input connector from dirty fibers, the fiber and connector saver must be cleaned. Any dirt between fiber connections causes high insertion loss and increased reflections and can also damage the fibers themselves. The advantage to using the connector saver is that if it is damaged, it is much less expensive and faster to replace than the front panel input connector.

Cleaning Connector Savers

The two ends of the connector saver should be cleaned differently. The ferrule fiber end of the connector saver can be cleaned in the same manner as a fiber patch cord, or a cable, using the method described below. The recessed fiber end of the input connector saver presents a different cleaning challenge. Because of the limited access within the recessed end, it is not recommended that alcohol be used in the cleaning process. If not quickly wiped away with another cotton swab, or blown dry with compressed air, alcohol can leave a residue which interferes with measurements. We have found that in the recessed end of the connector saver, it is difficult to prevent this alcohol residue buildup.



To clean the ferrule end of the connector saver

1 Apply isopropyl alcohol to a clean cotton swab.

Cotton swabs can be used making sure no cotton fibers remain after cleaning. Although foam swabs can leave filmy deposits, these deposits are very thin. The risk of other contamination buildup on the connection surface greatly outweighs the risk of contamination by foam swabs.

2 Clean the adapter with a wet swab.

3 Clean the adapter with a clean, dry swab.

4 Blow across the adapter using filtered, dry, compressed air.

CAUTION Do not shake, tip, or invert compressed air canisters. This releases particles from the can into the air flow. Refer to the instructions on the compressed air canister.

To clean the recessed end of the connector saver

- To clean the recessed end of the connector saver, it is recommended that a wrapped tip swab or stick cleaner be used. Berkshire's LT670183 wrapped tip cotton swab, or Cletop stick cleaners have proven to be an effective cleaning solution. To clean the recessed end of the connector saver, insert the stick type cleaner, and rotate it server times the same direction. Each swab should only be used once. While this may not remove all of the contaminants, it will displace them away from the center of the fiber (away from the core).

Returning the Instrument for Service

Agilent Technologies aims to maximize the value you receive, while minimizing your risk and problems. We strive to ensure that you get the test and measurement capabilities you paid for and obtain the support you need. Our extensive support resources and services can help you choose the right Agilent products for your applications and apply them successfully. Every instrument and system we sell has a global warranty. Support is normally available for at least five years beyond the production life of the product. Two concepts underlie Agilent's overall support policy: "Our Promise" and "Your Advantage".

Our Promise means your Agilent test and measurement equipment will meet its advertised performance and functionality. When you are choosing new equipment, we will help you with product information, including realistic performance specifications and practical recommendations from experienced test engineers. When you use Agilent equipment, we can verify that it works properly, help with product operation, and provide basic measurement assistance for the use of specified capabilities, at no extra cost upon request. 9Often, many self-help tools are available.

Your Advantage means that Agilent offers a wide range of additional expert test and measurement services, which you can purchase according to your unique technical business needs. Solve problems efficiently and gain a competitive edge by contracting with us for calibration, extra-cost upgrades, out-of-warranty repairs, and on-site education and training, as well as design, system integration, project management, and other professional engineering services. Experienced Agilent engineers and technicians worldwide can help you maximize your productivity, optimize the return on investment of your Agilent instruments and systems, and obtain dependable measurement accuracy for the life of those products.

You can get assistance with your test and measurement needs by internet, phone, or fax.

Online assistance: www.agilent.com/find/assist

United States

(tel) 1 800 452 4844

Latin America

(tel) (305) 269 7500

(fax) (305) 269 7599

Canada

(tel) 1 877 894 4414

(fax) (905) 206 4120

Australia

(tel) 1 800 629 485

(fax) (61 3) 9210 5947

Europe

(tel) (31 20) 547 2323

(fax) (31 20) 547 2390

New Zealand

(tel) 0 800 738 378

(fax) 64 4 495 8950

Japan

(tel) (81) 426 56 7832

(fax) (81) 426 56 7840

Asia Pacific

(tel) (852) 3197 7777
(fax) (852) 2506 9284

Preparing the Instrument for Shipping

Use the troubleshooting steps in the previous section to determine whether the WDCA, the TLS, or both need to be returned for repair. If system errors and troubleshooting indicate that only one component is faulty, return only that instrument. If you are unable to determine the cause, return both instruments.

If possible, provide a DUT which can demonstrate the system error. Duplicating the exact measurement environment is essential to troubleshooting difficult problems. Any extra equipment and accessories will be returned with the instrument.

See the list of service center numbers to call for repair on page 8-24.

1. Write a complete description of the failure and attach it to the instrument. Include any specific performance details related to the problem. The following information should be returned with the instrument.

   - Type of service required.
   - Date instrument was returned for repair.
   - Description of the problem:
     - Whether problem is constant or intermittent.
     - Whether instrument is temperature-sensitive.
     - Whether instrument is vibration-sensitive.
     - Instrument settings required to reproduce the problem. If you are returning the system for repair, save the instrument state where the problem is observed to the WDCA internal drive. Provide the state filename along with the error information.
     - Performance data.
     - Relevant error messages. Provide a hard copy printout if possible. Have these errors and error numbers available when you call the service center and make certain they are entered on the repair request.
   - Company name and return address.
   - Name and phone number of technical contact person.
   - Model number of returned instrument.
   - Full serial number of returned instrument.
   - List of any accessories returned with instrument.

2 Call the Agilent service center nearest you to have a repair number assigned to the instrument or system. You will receive instructions on shipping and an estimate on repair time. It the system is within the warranty time period, Agilent will pay the repair costs. If not, the approximate repair costs will be quoted. When the instrument arrives at the service center, you will be contacted with the exact repair costs.

3 Cover all front or rear-panel connectors that were originally covered when you first received the instrument.

CAUTION Cover electrical connectors to protect sensitive components from electrostatic damage. Cover optical connectors to protect them from damage due to physical contact or dust.

CAUTION Instrument damage can result from using packaging materials other than the original materials. Never use styrene pellets as packaging material. They do not adequately cushion the instrument or prevent it from shifting in the carton. They may also cause instrument damage by generating static electricity.

4 Pack the instrument in the original shipping containers. Original materials are available through any Agilent Technologies office. Or, use the following guidelines:

- Wrap the instrument in antistatic plastic to reduce the possibility of damage caused by electrostatic discharge.
- For instruments weighing less than 54 kg (120 lb), use a double-walled, corrugated cardboard carton of 159 kg (350 lb) test strength.
- The carton must be large enough to allow approximately 7 cm (3 inches) on all sides of the instrument for packing material, and strong enough to accommodate the weight of the instrument.
- Surround the equipment with approximately 7 cm (3 inches) of packing material, to protect the instrument and prevent it from moving in the carton. If packing foam is not available, the best alternative is S.D-240 Air Cap™ from Sealed Air Corporation (Commerce, California 90001). Air Cap looks like a plastic sheet filled with air bubbles. Use the pink (antistatic) Air Cap™ to reduce static electricity. Wrapping the instrument several times in this material will protect the instrument and prevent it from moving in the carton.

5 Seal the carton with strong nylon adhesive tape.

6 Mark the carton “FRAGILE, HANDLE WITH CARE”.

7 Retain copies of all shipping papers.

Agilent Technologies Service Offices

Before returning an instrument for service, call the Agilent Technologies Instrument Support Center at (800) 403-0801, visit the Test and Measurement Web Site at http://www.agilent.com/find/assist, or call one of the numbers listed below.

Agilent Technologies Service Numbers

Austria	01/25125-7171
Belgium	32-2-778.37.71
Brazil	(11) 7297-8600
China	86 10 6261 3819
Denmark	45 99 12 88
Finland	358-10-855-2360
France	01.69.82.66.66
Germany	0180/524-6330
India	080-34 35788
Italy	+39 02 9212 2701
Ireland	01 615 8222
Japan	(81)-426-56-7832
Korea	82/2-3770-0419
Mexico	(5) 258-4826
Netherlands	020-547 6463
Norway	22 73 57 59
Russia	+7-095-797-3930
Spain	(34/91) 631 1213
Sweden	08-5064 8700
Switzerland	(01) 735 7200
United Kingdom	01 344 366666
United States and Canada	(800) 403-0801

9

Specifications and Regulatory Information

Specifications and Regulatory Information

This chapter contains specifications and characteristics for the Agilent 86082A analyzers. For specifications specific to the 81640 and 81680 tunable laser sources refer to the *Tunable Laser Modules User's Guide*.

Definition of Terms

Characteristics and specifications

The distinction between specifications and *characteristics* is described as follows:

- **Specifications** describe warranted performance.
- **Typical** is nonwarranted information, but indicates tested performance which most units will achieve.
- **Characteristics** provide useful, but nonwarranted and often untested information about the functions and performance of the instrument. *Characteristics are printed in italics.*

Normalization

Normalization removes amplitude variations caused by your test setup and provides a known reference for comparison. It is very important to normalize your test setup in order to make accurate measurements. Normalization assures that measurement results reflect only the characteristics of your device. Because the source power varies with the wavelength span, normalization also improves the amplitude accuracy of the WDCA. This is accomplished by adjusting the source power based on the specified normalization start and stop wavelengths. This power will remain unchanged until another normalization is performed over a different wavelength span.

NOTE

Normalization is required for each receiver input and is valid only for the sweep mode used during the normalization process.

Wavelength

- **Wavelength Calibration** uses the WDCA's built-in absorption cell to calibrate wavelength accuracy. You can request a wavelength calibration at any time. A wavelength calibration will automatically be performed each time the TLS goes through its Lambda Zero cycle.
- **Wavelength Resolution** is the resolution of the marker readout on a narrow wavelength sweep. **Note:** For broader sweeps, the WDCA reduces the visible resolution. The GPIB marker query retains full resolution.
- **Absolute Accuracy** refers to the wavelength accuracy after the user has performed the user normalization and user wavelength calibration using the

procedure in “Performing a User Wavelength Calibration” on page 3-17.

- **Relative accuracy** is the difference between the wavelengths measured by two different markers on the same trace.
- **Tuning Repeatability** refers to the wavelength accuracy of returning to a wavelength after having tuned to a different wavelength.
- **Reproducibility** refers to the amount of wavelength drift which can occur over the specified time while the WDCA sweeps across a DUT of known wavelength.

Amplitude

- **Total Amplitude Accuracy** is the RSS (root sum of the square) of the total amplitude accuracy, amplitude stability, polarization dependent loss, and scale fidelity.
- **Scale Fidelity** refers to the potential errors in amplitude readout at amplitudes other than at the normalization point. This specification is sometimes called linearity.
- **Polarization Dependence**, sometimes termed PDL for polarization dependent loss, refers to the amplitude change that can be seen by varying the polarization of the light entering the analyzer. This is not to be confused with amplitude variations caused by the varying distribution of energy between the different modes in fiber that is multi mode at the wavelength of interest.

Minimum DUT Bandwidth

The WDCA has finite receiver bandwidth. If the DUT response changes too fast as a function of wavelength, the detectors will lag, introducing an error. The instrument is sensitive to both slew and bandwidth. The bandwidth is provided as a guide to what the instrument can perform.

Sensitivity

Sensitivity is the smallest amplitude that can be detected, relative to the amplitude at the top graticule of the display. Increasing sensitivity reduces trace noise and also increases the sweep time. Leaving sensitivity in Auto mode allows the instrument to set the maximum sensitivity while maintaining the fastest sweep time.

Dynamic range

Dynamic Range takes into account interactions between the DUT, laser line width, and the receiver detection bandwidth. These characteristics limit how finely you can resolve trace features, particularly if they are close together and at different amplitudes.

Receiver Noise Level

Receiver Noise Level is the smallest optical power that can be detected. This value is provided so that you can determine how much fixture insertion loss can be tolerated. For example, if a 40 dB integrating sphere is used at a wavelength where the laser power is –12 dBm, the WDCA will receive only – 52 dBm maximum. This leaves only a 33 dB range to measure the response of the DUT.

Sweep speed

- **Precision** mode offers the most accurate sweep setting to use if you are concerned about wavelength and amplitude accuracy for final filter test applications, but is the slowest.
- **Normal** mode provides excellent amplitude and wavelength accuracy. The speed of two sweeps per second over a 2 nm span can be achieved. Normal mode is the best general purpose mode.
- **Fast** mode allows wider spans with faster sweep speed. In this mode you can measure filters with multiple channels or see if the rejection band holds over wide ranges. Fast sweep mode may not be appropriate to use if the filter has very sharp edges. The fast sweeping speed may not accurately display the signal.

Acc Unspec (Accuracy Unspecified)

As you increase sensitivity, the instrument has to sweep slower in order to reduce the noise. The WDCA has to slow the laser source down in sync with the receiver. The laser has a limited number of sweep rates available, and only three are calibrated. At high sensitivity, the instrument has to either sweep too fast or go to an uncalibrated sweep rate. Either can occur depending on sweep settings. When either occurs, the Acc Unspec notation appears in the lower right corner of the display and these specifications do not apply.

Trigger Cable

The laser source triggers the receiver via the rear panel, external trigger, BNC cable connection. The source receiver synchronization is critical to the wavelength accuracy. These specifications apply only for a one meter or less BNC cable from the laser to the receiver.

Specifications

Agilent Technologies warrants that the following specifications will be met under the following instrument operating conditions:

Unless otherwise indicated,

- the specified temperature range is 10° to 30° C
- only over the 1510 to 1580 nm wavelength range
- device and fixtures present at least a 30 dB return loss to the laser source
- normalization is repeated when the temperature changes 4° C since the last normalization or at least once every 24 hours and normalization is performed at the same sweep mode as the measurement
- wavelength calibration is repeated when the temperature changes 4° C from the last calibration or at least once every 24 hours
- laser sweeps at 5 nm/s (Precision Sweep mode)
- laser warm up of 90 minutes prior to performing measurements
- Acc Unspec notation not present on display
- specifications assume sufficient sensitivity used to eliminate noise

Specification Discussion

Trace Amplitude Accuracy is the observed trace accuracy after performing normalization with a thru cable. The instrument is most accurate at the top of the scale, so the specification is split. Amplitude accuracy does not include the effects of the connector repeatability. A typical connector insertion loss, while normally small, can change as much as 0.25 dB from connection to connection. The insertion loss can be reduced by normalizing with two patch cords connected to a barrel and then inserting the DUT without disconnecting the patch cords from the instrument. A splice at the DUT connection provides the best accuracy. The total amplitude accuracy specification includes the RSS (root sum of the square) combination of:

- polarization dependent loss (PDL) which assumes polarization changes between normalization and insertion of the DUT

- amplitude stability of the system
- scale fidelity

If the DUT and associated cables are polarization maintaining, temperature is tightly controlled, and the DUT is inserted without removing connectors, the amplitude accuracy should be the same as the amplitude stability value.

Angled Connector Specifications

Wavelength and amplitude accuracy specifications require an angled connector from the source output to the normalized receiver input ports.

Angled contact connectors help you to control return loss. With angled fiber endfaces, reflected light tends to reflect into the cladding, reducing the amount of light that reflects back to the source.

NOTE If the contact connector on your TLS is angled, you can only use cables with angled connectors with your TLS. The receiver input has flat connectors and the TLS output has angled connects. Do not connect an angled patch cord to the flat receiver input connector or a flat patch cord to the angled TLS connector output.

The angled connector symbol is typically colored green. You should connect straight contact fiber end connectors with neutral sleeves to straight contact connectors and connect angled contact fiber end connectors with green sleeves to angled contact connectors.

Wavelength Specifications

Agilent Technologies warrants that the following specifications will be met under the following instrument operating conditions:

Unless otherwise indicated,

- the specified temperature range is 10° to 30° C
- only over the 1510 to 1580 nm wavelength range
- device and fixtures present at least a 30 dB return loss to the laser source
- normalization is repeated when the temperature changes 4° C since the last normalization or at least once every 24 hours and normalization is performed at the same sweep mode as the measurement
- wavelength calibration is repeated when the temperature changes 4° C from the last calibration or at least once every 24 hours
- laser warm up of 90 minutes prior to performing measurements
- sweep span is less than 50 nm
- Acc Unspec notation not present on display
- specifications assume sufficient sensitivity used to eliminate noise

Table 9-1. Wavelength Specifications

	Precision Sweep Mode	Normal Sweep Mode	Fast Sweep Mode
Wavelength Range			
C Band[a]	1515 nm to 1575 nm *1480 nm to 1575 nm (typical)*		
L Band[b]	1520 nm to 1620 nm		
Absolute Wavelength Accuracy			
C Band[a]	±7 pm ±*5 pm (typical)*	±7 pm ±*5.1 pm (typical)*	±9 pm ±*7 pm (typical)*
L Band[b]	±9 pm ±*7 pm (typical)*	±9 pm ±*7 pm (typical)*	±11.1 pm ±*9.1 pm (typical)*
Relative Accuracy			
≥1 nm delta	±2 pm	± 2 pm	±3 pm
≤20 nm delta	±4 pm	± 4 pm	±5 pm
>20 nm delta	absolute wavelength accuracy applies	absolute wavelength accuracy applies	absolute wavelength accuracy applies
Wavelength Resolution *(characteristic)*	± *1.0 pm*		
Wavelength Reproducibility *(characteristic)*	± *1.0 pm*		

a. Option 111
b. Option 112

Amplitude Specifications

Agilent Technologies warrants that the following specifications will be met under the following instrument operating conditions:

Unless otherwise indicated,

- the specified temperature range is 10° to 30° C
- only over the 1510 to 1580 nm wavelength range
- device and fixtures present at least a 30 dB return loss to the laser source
- normalization is repeated when the temperature changes 4° C since the last normalization or at least once every 24 hours and normalization is performed at the same sweep mode as the measurement
- wavelength calibration is repeated when the temperature changes 4° C from the last calibration or at least once every 24 hours
- laser warm up of 90 minutes prior to performing measurements
- sweep span is less than 20 nm
- point spacing is less than 0.01 nm
- Acc Unspec notation not present on display
- specifications assume sufficient sensitivity used to eliminate noise

Table 9-2. Amplitude Specifications[a]

	Precision Sweep Mode	Normal Sweep Mode	Fast Sweep Mode
Minimum DUT Bandwidth	2 pm	4 pm	14 pm
Total Accuracy[b,c]			
<10 dB insertion loss[d]	±0.05 dB	±0.07 dB	±0.09 dB
Full Range	±0.09 dB	±0.10 dB	±0.11 dB
Amplitude Stability[d]			
Sweep to sweep	±0.015 dB	±0.02 dB	±0.04 dB
Continuous over ±4° C temperature swing	±0.04 dB	±0.06 dB	±0.08 dB
Polarization Dependent Loss (PDL)	±0.025 dB		
Scale Fidelity			
<10 dB insertion loss[d]	±0.02 dB		
Auto Sensitivity	±0.05 dB		
Full Range	±0.07 dB		
Dynamic Range @ 7 pm offset			
Manual Sensitivity	65 dB		
Auto Sensitivity On	30 dB		
Return Loss	30 dB *35 dB (typical)*		
Receiver Noise Level	–85 dBm		
Max Safe Power			
Total Safe Power	+10 dBm		

[a] Not including connector repeatability.

[b] RSS of PDL, stability, and scale fidelity.

[c] Assumes sufficient sensitivity used.

[d] Insertion loss from the DUT, cabling, connectors, couplers, fixtures, and so on.

System and Other Specifications

Table 9-3. System and Other Specifications

	Precision Sweep Mode	Normal Sweep Mode	Fast Sweep Mode
Sweep Rate	5 nm/sec.	10 nm/sec.	40 nm/sec.
Trace Update Rate[a] **(Auto Sensitivity)**	1 trace/sec. over 2 nm	2 traces/sec. over 2 nm	2 traces/sec. over 8 nm
Max Source Output Power			
C-Band Range	≥ –6 dBm		
L-Band Range	≥ –7 dBm		

[a] 1001 data points (standard trace).

General Instrument Specifications

Table 9-4. General Instrument Specifications

Dimensions	
Receiver	222 mm H x 425 mm W x 427 mm D (8.8” x 16.8” x 16.8”)
Source	145 mm H x 426 mm W x 545 mm D (5.7” x 16.8” x 21.5”)
Weight	
Receiver	11.8 kg (26 lbs)
Source	16.4 kg (36 lbs)
Temperature Ranges	
Specified Temperature	+10 °C to +30 °C within ±4 °C of last normalization and wavelength calibration
Power Requirements	
Receiver	
Voltage and Frequency	90 Vac to 260 Vac, 44 Hz to 444 Hz
Max. Power Consumption	230 W
Source	
Voltage and Frequency	100 Vrms to 240 Vrms ±10%
Max. Power Consumption	280 VA
Computer Interfacing	
Remote Control Compatibility	IEEE-488.1, IEEE-488.2 (100%)
Remote Control Interfaces	GPIB, Parallel Printer Port, External VGA Monitor, Keyboard (PS/2) and Mouse
Floppy Disk	3.5 inch 1.44 MB, MS-DOS®
Data Export	Spreadsheet and Word Processor Compatible (CSV ASCII)
Graphics Export	CGM, PCL
Instrument Drivers	Universal Instrument Drivers (PNP), Compatible with Agilent VEE, Labview®, Visual Basic and C++

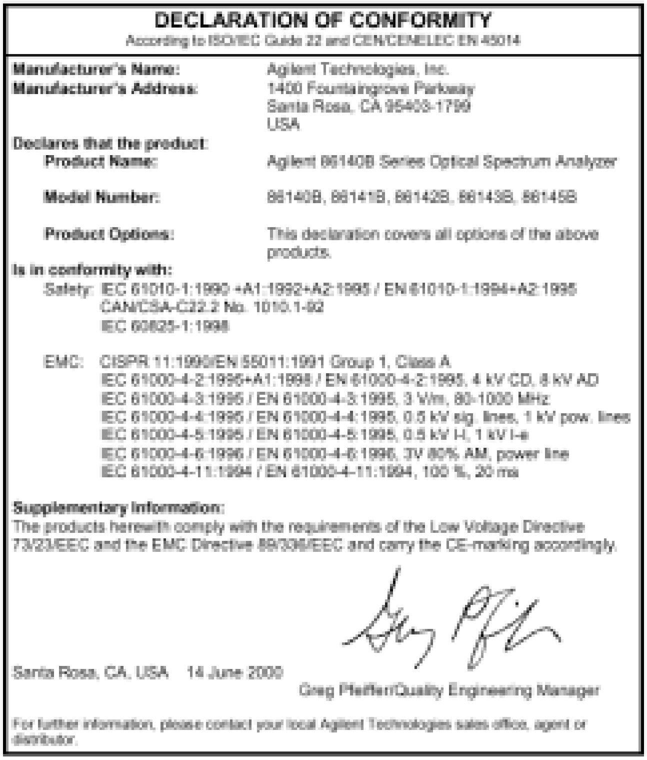
Regulatory Information

- This product complies with 21 CFR 1040.10 and 1040.11.

Notice for Germany: Noise Declaration

Acoustic Noise Emission	Geraeuschemission
LpA < 70 dB	LpA < 70 dB
Operator position	am Arbeitsplatz
Normal position	normaler Betrieb
per ISO 7779	nach DIN 45635 t.19

Declaration of Conformity



DECLARATION OF CONFORMITY

According to ISO/IEC Guide 22 and CEN/CENELEC EN 45014

Manufacturer's Name: Agilent Technologies, Inc.

Manufacturer's Address:
1400 Fountaingrove Parkway
Santa Rosa, CA 95403-1799
USA

Declares that the product:

Product Name: Agilent 86140B Series Optical Spectrum Analyzer

Model Number: 86140B, 86141B, 86142B, 86143B, 86145B

Product Options: This declaration covers all options of the above products.

Is in conformity with:

Safety: IEC 61010-1:1990 +A1:1992+A2:1995 / EN 61010-1:1994+A2:1995
CAN/CSA-C22.2 No. 1010.1-92
IEC 60825-1:1998

EMC: CISPR 11:1990/EN 55011:1991 Group 1, Class A
IEC 61000-4-2:1995+A1:1998 / EN 61000-4-2:1995, 4 kV CD, 8 kV AD
IEC 61000-4-3:1995 / EN 61000-4-3:1995, 3 V/m, 80-1000 MHz
IEC 61000-4-4:1995 / EN 61000-4-4:1995, 0.5 kV sig. lines, 1 kV pow. lines
IEC 61000-4-5:1995 / EN 61000-4-5:1995, 0.5 kV l-l, 1 kV l-e
IEC 61000-4-6:1996 / EN 61000-4-6:1996, 3V 80% AM, power line
IEC 61000-4-11:1994 / EN 61000-4-11:1994, 100 %, 20 ms

Supplementary Information:
The products herewith comply with the requirements of the Low Voltage Directive 73/23/EEC and the EMC Directive 89/336/EEC and carry the CE-marking accordingly.

Santa Rosa, CA, USA 14 June 2000

Greg Pfeiffer/Quality Engineering Manager

For further information, please contact your local Agilent Technologies sales office, agent or distributor.

Index