User Manual

FieldMaxII-P™

Laser Energy Meter

COHERENT

User Manual
FieldMaxII-P
Laser Energy Meter

COHERENT
7470 SW Bridgeport Rd.
Portland, OR 97224

Every effort has been made to ensure that the data given in this document is accurate. The information, figures, tables, specifications and schematics contained herein are subject to change without notice. Coherent makes no warranty or representation, either expressed or implied with respect to this document. In no event will Coherent be liable for any direct, indirect, special, incidental or consequential damages resulting from any defects in its documentation.

Technical Support

In the U.S.:

Should you experience difficulties with your product, or need technical information, please visit our website: www.coherent.com. You can obtain additional support by either telephoning our Technical Support Hotline at 1.800.343.4912, or e-mailing our Support Team at support.instruments@coherent.com. Telephone coverage is available Monday through Friday (except U.S. holidays).

If you call outside our office hours, your call will be taken by our answering system and will be returned when the office reopens.

If there are technical difficulties with your product that cannot be resolved by support mechanisms outlined above, please e-mail or telephone Coherent Technical Support with a

description of the problem and the corrective steps attempted. When communicating with our Technical Support Department, via the web or telephone, the model and serial number of the product will be required by the Support Engineer responding to your request.

Outside the U.S.:

If you are located outside the U.S., visit our website for technical assistance, or telephone our local Service Representative. Representative phone numbers and addresses can be found on the Coherent website, www.coherent.com.

Coherent provides web and telephone technical assistance as a service to its customers and assumes no liability thereby for any injury or damage that may occur contemporaneous with such services. These support services do not, under any circumstances, affect the terms of any warranty agreement between Coherent and the buyer. Operating a Coherent product with any of its interlocks defeated is always at the operator's risk.

TABLE OF CONTENTS

List of Tables

List of Figures

Preface

This manual contains user information for the FieldMaxII-P™ laser energy meter.

Software Installation

For complete software installation instructions, refer to the *FieldMaxII Software Installation Quick Start Guide* (1096359) that shipped with your product.

If that document is unavailable, insert the FieldMaxII-P CD into your CD-ROM drive. If Autorun is enabled on your system, installation will start automatically; otherwise, select Run from the Start menu and then type ***D*:\Setup.exe** (substitute the appropriate letter of your CD-ROM drive for ***D***).

U.S. Export Control Laws Compliance

It is the policy of Coherent to comply strictly with U.S. export control laws.

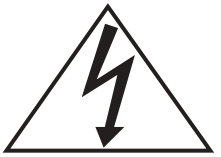
Export and re-export of lasers manufactured by Coherent are subject to U.S. Export Administration Regulations, which are administered by the Commerce Department. In addition, shipments of certain components are regulated by the State Department under the International Traffic in Arms Regulations.

The applicable restrictions vary depending on the specific product involved and its destination. In some cases, U.S. law requires that U.S. Government approval be obtained prior to resale, export or re-export of certain articles. When there is uncertainty about the obligations imposed by U.S. law, clarification should be obtained from Coherent or an appropriate U.S. Government agency.

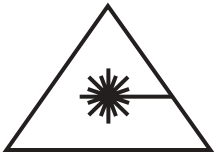
Publication Updates

To view information that may have been added or changed since this publication went to print, connect to www.Coherent.com.

Symbols Used in This Document



This symbol is intended to alert the operator to the presence of dangerous voltages associated with the product that may be of sufficient magnitude to constitute a risk of electrical shock.



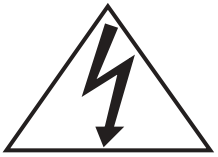
This symbol is intended to alert the operator to the danger of exposure to hazardous visible and invisible radiation.

This symbol is intended to alert the operator to the presence of important operating and maintenance instructions.

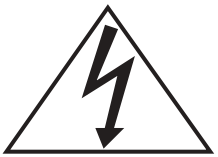
SAFETY

Carefully review the following safety information to avoid personal injury and to prevent damage to this meter or any sensor connected to it. Except for replaceable batteries (discussed under "Battery Replacement" on page 15), there are no user-serviceable parts in the FieldMaxII-P meter. For service information, refer to "Obtaining Service" on page 73.

Use only the power cord specified for the meter. The grounding conductor of the cord must be connected to earth ground.



Do not operate the meter if its panels are removed or any of the interior circuitry is exposed.



Do not operate the meter in wet or damp conditions, or in an explosive atmosphere.

Operate the meter only within the specified voltage range.

Do not apply a voltage outside the specified range of the input connections.

Do not operate the meter if there are suspected failures. Refer damaged units to qualified Coherent service personnel.

Declaration of Conformity

3123451 Revision AB

Declaration of Conformity

We

Coherent, Inc.
7470 SW Bridgeport Road
Portland, Oregon, USA 97224

declare under sole responsibility that the

FieldMax II

meets the intent of Directive 89/336/EEC for Electromagnetic Compatibility. Compliance was demonstrated per testing to EN61326 Electromagnetic Compatibility Product Family Standard for Measurement, Control and Laboratory Equipment to include the following test specifications as of June 2005:

EN55011 Class A Radiated Emissions
EN55011 Class A Conducted Emissions
EN61000-3-2 Powerline Harmonics – Performance Criteria A
EN61000-3-3 Powerline Voltage Fluctuation and Flicker – Performance Criteria A
EN61000-4-2 Electrostatic Discharge – Performance Criteria B
Unit may respond to an ESD event but will return to normal without user intervention.

EN61000-4-3 Radiated Immunity – Performance Criteria A
EN61000-4-4 Electrical Fast Transient Immunity - Performance Criteria A
EN61000-4-5 Electrical Slow Transient Immunity- Performance Criteria A
EN61000-4-6 Conducted RF Immunity - Performance Criteria A
EN61000-4-11 Power Line Dropout - Performance Criteria A

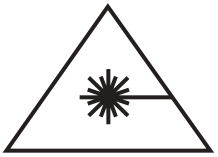
Director of Engineering Date: 3/9/05

Site Manager Date: 8-9-05

Coherent, Inc. Page 1 of 1

QUICK START

This section presents a "mini-tutorial" that explains how to connect a sensor to your FieldMaxII-P meter and begin taking measurements within minutes. For in-depth information about the procedure introduced in this section, refer to "Operation" on page 17.



Follow all laser safety procedures. The laser must be blocked or switched OFF before beginning the procedure described in this section.

Power to the FieldMaxII-P instrument must be OFF before beginning the procedure described in this section.

Do not exceed the energy density limits of the sensor.

Measuring Energy With a Pyroelectric Sensor

The following figure outlines how to set up a pyroelectric sensor to take an energy measurement.

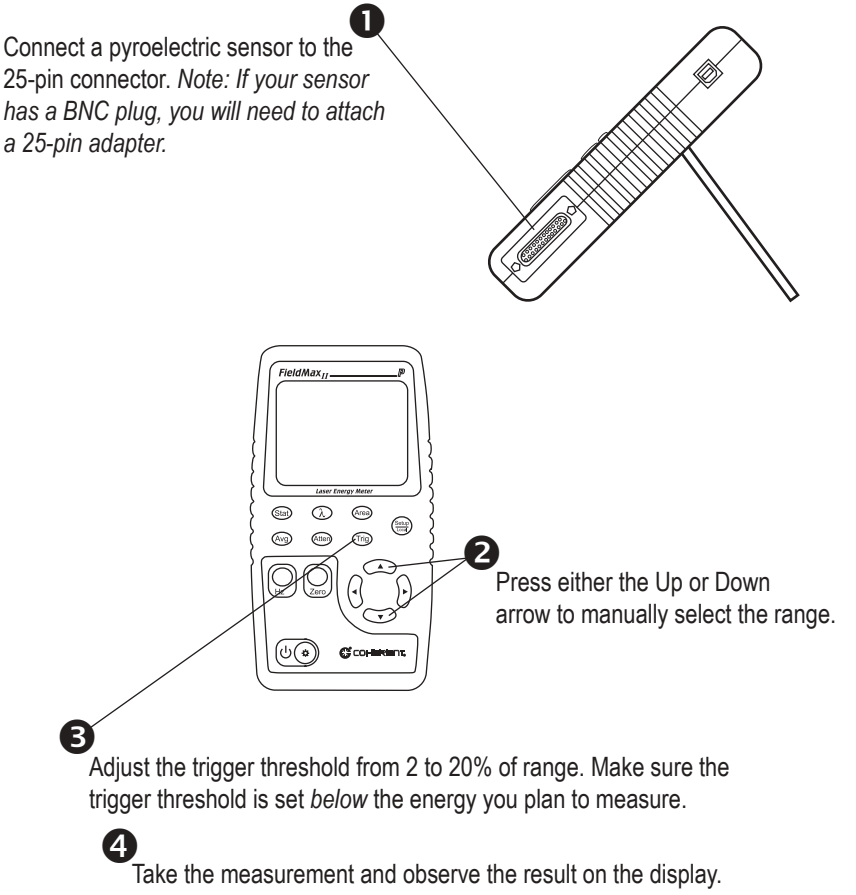


Figure 1. Measuring Energy With a Pyroelectric Sensor

DESCRIPTION

Thank you for purchasing the FieldMaxII-P™—a versatile, easy-to-use digital energy meter designed for field service and production applications.

This section discusses the following topics:

- Front panel (page 9)
- Right side panel (page 13)
- Left side panel (page 14)

There are also instructions on how to replace the alkaline batteries (page 15) and a brief overview of the AC adapter (page 16).

Here is a list of specific features included in your FieldMaxII-P meter:

- 73 x 58 mm backlit LCD display
- Fast and effective laser tuning mode
- Works with pyroelectric sensors
- Measures energy up to 300 pps
- Intuitive soft key-driven user interface
- USB 1.1
- Portable AC/DC operation
- Compact, rugged enclosure with stand

The versatile FieldMaxII-P measures:

- Energy: J, J/cm^2
- Frequency: Hz
- Full statistics: max, min, mean, and standard deviation

Front Panel

The front panel (shown in Figure 2) includes a liquid crystal display (LCD) and buttons that are used to enter parameters, select modes, and change ranges.

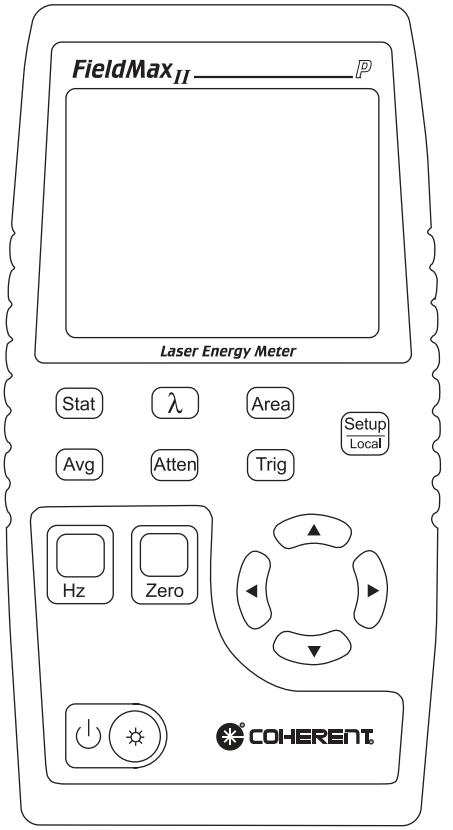


Figure 2. Front Panel

Buttons

The following buttons are on the front panel of the FieldMaxII-P:

- Setup/Local—starts or ends a parameter edit cycle. This button is also used to cancel the front panel lockout when Remote Control is active.
- Stat—statistics processing parameter
- Wave (λ)—wavelength compensation parameter
- Area—area correction parameter
- Avg—display smoothing parameter
- Atten—attenuation correction parameter
- Trig—trigger level parameter
- Hz—Rep Rate Display mode
- Zero—start batch. When Statistics mode is not active, this button can also be used to restart a batch.
-  —Power Switch/Backlight Toggle button
- Up Arrow (▲)—field adjust or range select
- Down Arrow (▼)—field adjust or range select
- Left Arrow (◀)—field select
- Right Arrow (▶)—field select

For detailed information about each of these buttons, refer to “Operation” on page 17.

Display

The LCD display provides visual measurement information. Figure 3 shows all the possible segments that may appear on the display.

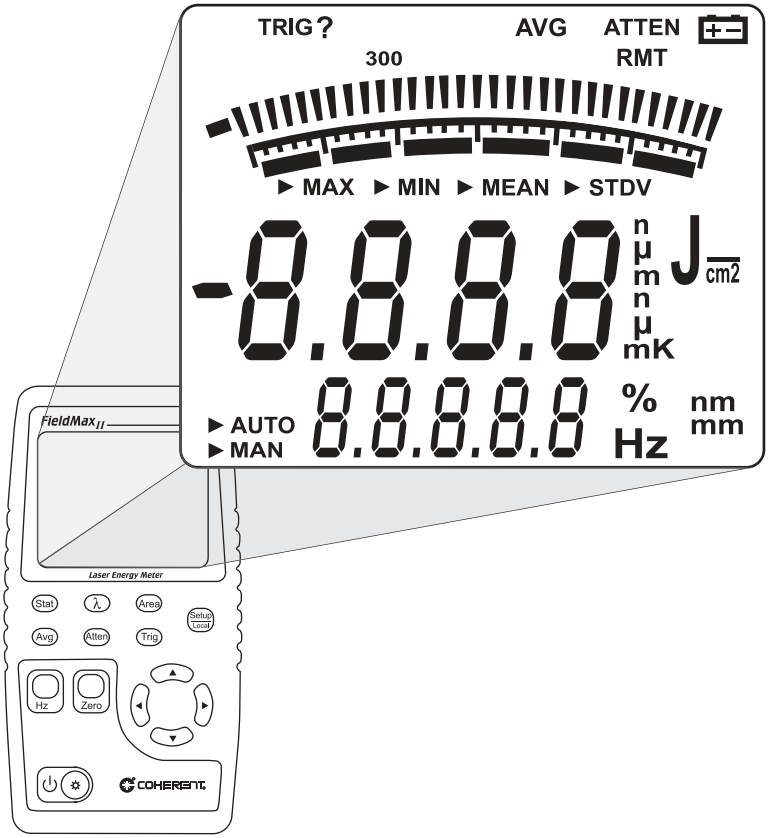


Figure 3. LCD Display

The type of sensor being used and the individually-selected settings determine what type of information will actually appear on the display

Information that appears on the display is divided into the groups described in the following list (Figure 3 on page 11 shows the general location of each group).

- Annunciators: TRIG, Range Hint, AVG, ATTEN, RMT, and Battery
- Digital tuning feature
- Tuning meter scale
- Statistical parameters: MAX, MIN, MEAN, and STDV
- Numeric measurement value—large numeric characters
- Measurement units and engineering prefixes
- Numeric data entry, batch count, parameter settings, and Hertz measurement values
- Data entry units, Hertz units, and current parameter units

For detailed information about these settings, refer to "Button Functions" on page 31.

Right Side Panel

The right side panel contains the USB and Sensor connectors (refer to Figure 4).

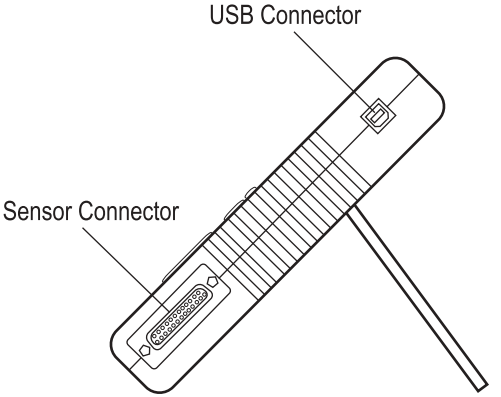


Figure 4. Right Side Panel

USB Connector

Attaching the cable—shipped with the meter—to this standard USB connector allows communication between FieldMaxII-P and a computer with a USB interface.

Sensor Connector

Use this connection to attach a DB-25 SmartProbe connector or adapter.

Left Side Panel

The left side panel contains the Analog Out and Power Jack connectors (refer to Figure 5).

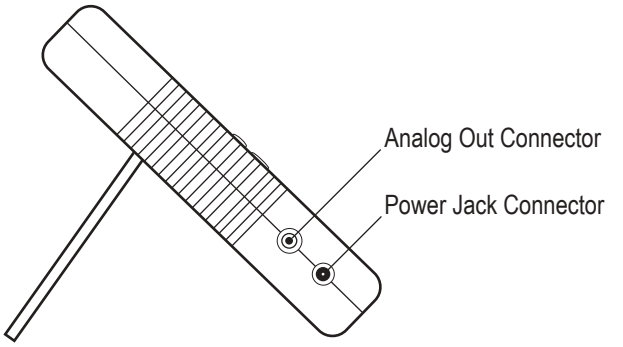


Figure 5. Left Side Panel

Analog Out Connector

When power is on, the Analog Out connector outputs a voltage proportional to the current laser measurement. The output voltage is zero (0) volts when the measured energy is zero (0) or less. The output voltage is the full-scale output voltage when the measured energy is full-scale or overranged. The full-scale output voltage (1V, 2V, or 5V) is selected via the host interface. Factory default full-scale output voltage is 2V.

Power Jack Connector

Connect the supplied power cord to this jack.

Battery Replacement

FieldMaxII-P uses six 1.5V alkaline batteries, or a 90-to-260 VAC, 50/60 Hz AC adapter (refer to "AC Adapter" on page 16 for more information). Figure 6 illustrates how to replace the batteries.

An optional rechargeable battery pack is also available for FieldMaxII-P. Visit our website (www.Coherent.com), or contact a Coherent representative for more information (see Table 3, "Coherent Service Centers," on page 74 for contact information).

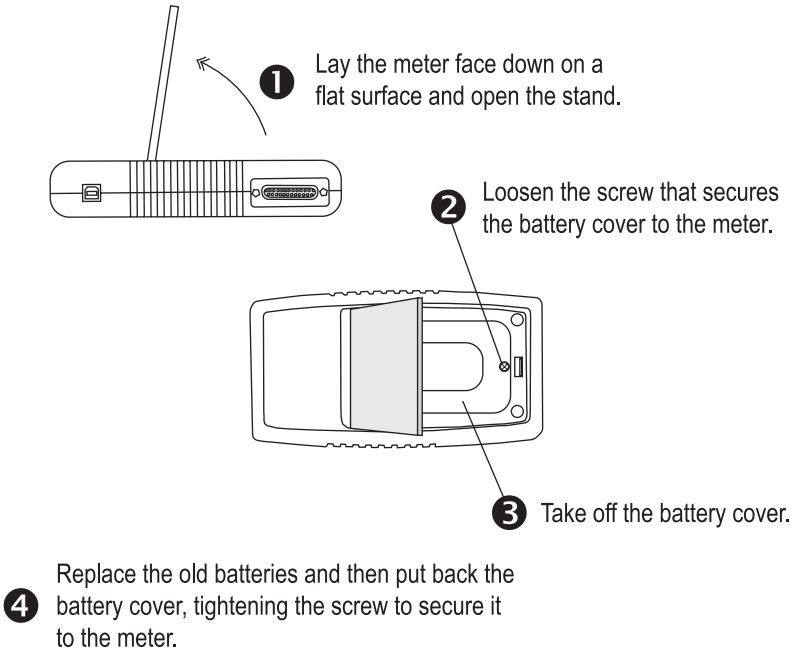


Figure 6. Battery Replacement

AC Adapter

Using an AC adapter prolongs battery life. FieldMaxII-P automatically senses when an adapter is used.

When batteries rather than an adapter, are used, the Battery annunciator flashes if the battery charge is low.

While in battery operation, if a sensor is not connected to the meter, power is automatically turned off after ten minutes.

OPERATION

This section discusses the following topics:

Tuning Mode

As shown in Figure 7, tuning is visually displayed on the LCD using tuning needles and zone indicator bars.

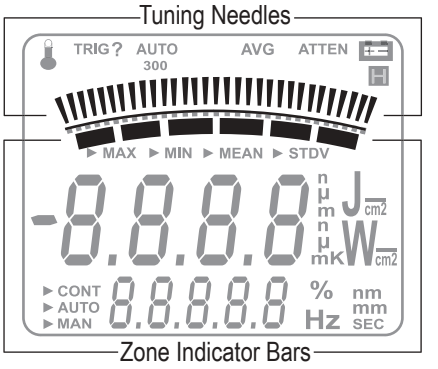


Figure 7. Location of Tuning Needles and Zone Indicator Bars

Tuning Needles

Tuning needles—which divide a given tuning zone into thirty "increments"—are used to peak a laser output. As the top or bottom of a zone is reached, the tuning needles automatically move to the center of the next zone (see Figure 8). Zone indicator bars let you know when this happens (refer to "Zone Indicator Bars" on page 20 for more information).

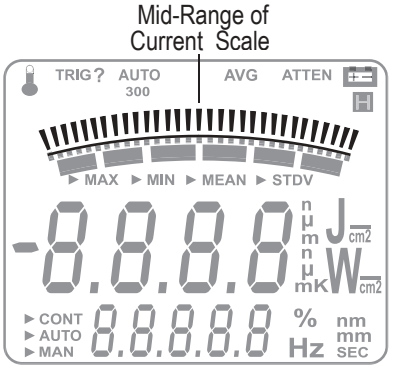


Figure 8. Current Scale Mid-Range

Zone Indicator Bars

Zone indicator bars are a series of six segments, as shown in Figure 9:

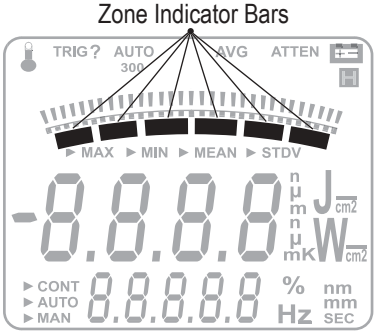


Figure 9. Zone Indicator Bars

These bars act as visual indicators while the tuning needles automatically move through zones, and also provide a relative indication of where the measurement falls within the active range.

Zone indicator bars always appear in pairs, with each overlapping zone representing 1/3 of full scale. Figure 10 shows how the five zone indicator bars correlate to full scale measurement on the tuning meter scale.

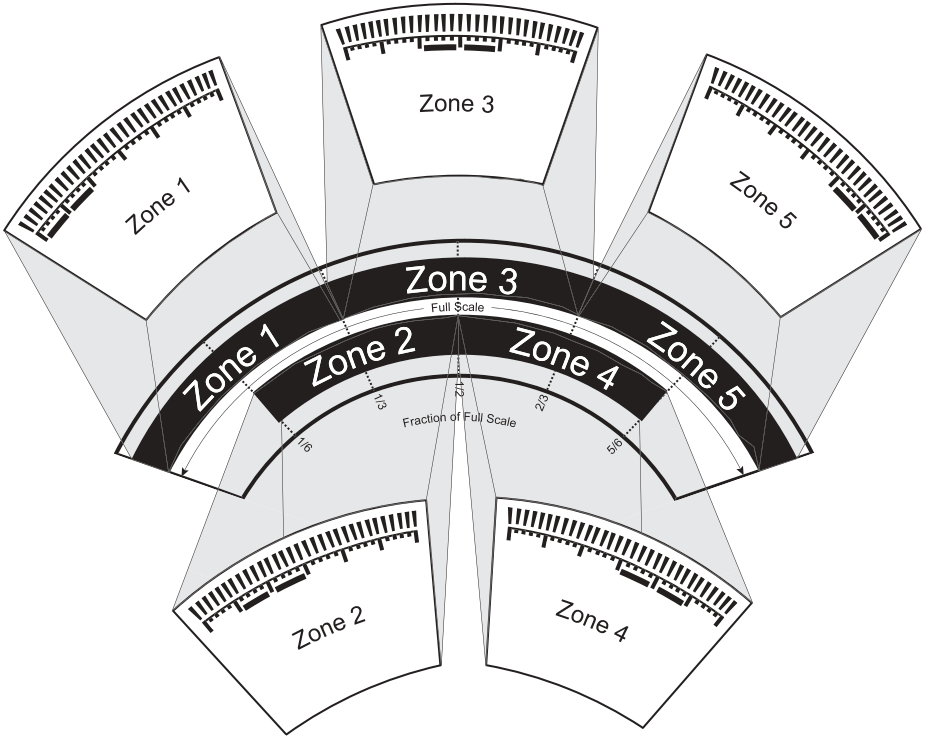


Figure 10. Comparison of Zone Indicator Bars to Full Scale Measurement

Figure 11 shows an example of how the zone indicator bars overlap on a 30-watt scale:

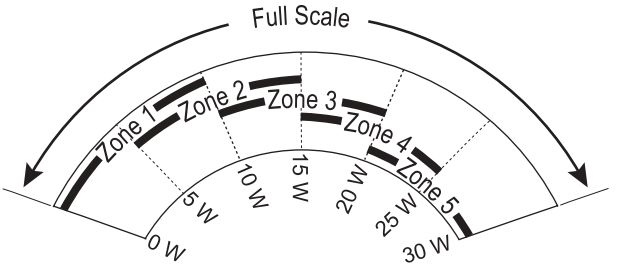


Figure 11. Tuning Mode Example - Full Scale

Annunciators

Annunciators refers to the icon-type symbols that appear on the LCD (Figure 3 on page 11 shows all the annunciators on the FieldMaxII-P meter). The update rate for all annunciators is 3 times per second.

TRIG

Whenever a trigger is detected, "TRIG" displays in the TRIG annunciator position. "TRIG?" displays in the absence of a trigger.

Range Hint

The Range Hint annunciator—towards the top of the LCD, just above the tuning needles—displays "3," "30," or "300." These numbers represent the full-scale range currently selected by the user. Range Hint is discussed in more detail under "Measurement Display and Range Selection" on page 33.

AVG

When AVG (display smoothing) is active, display values are averaged by samples. “AVG” displays when averaging is active. Nothing displays in this position if averaging is not active.

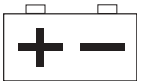
ATTEN

The ATTEN annunciator indicates if attenuation correction is applied to the measurement value. “ATTEN” displays whenever attenuation is active.

RMT

The RMT annunciator indicates that the FieldMaxII meter is currently in Remote Control mode. Remote Control is discussed under “Setup/Local” on page 26.

Battery



The Battery annunciator flashes whenever the batteries need to be replaced.

The Battery annunciator only appears when the meter is running on battery power, not the AC adapter.

Fault Displays

FieldMaxII-P is capable of detecting internal and user-induced faults. When a fault is detected, the letters "Er"—followed by a numeric fault code (see Table 1)—appear on the display. You can dismiss a fault code by pressing any button, or by correcting the cause of the fault.

Table 1. Fault Codes

ERROR CONDITION	FAULT CODE
Unrecognized sensor	1
Sensor communication failure	2
Sensor error	3
Sensor error	4
Sensor/firmware version mismatch (sensor format version exceeds capability of the instrument firmware—firmware upgrade needed)	5
Sample rate fault	6
Hardware fault (detectable hardware error)	20
Data overflow (result of an arithmetic operation that is greater than can be held in the allocated storage)	41
Wrong type of sensor is attached to the instrument	42

As an example, “Er 4” appears on the display if there is a sensor error. You can dismiss the fault by removing the sensor from the meter, or by pressing any button.

Attaching an unrecognized sensor to the meter creates a special fault condition. This condition is characterized by displaying a sensor fault (1 through 5).

Invalid and Not Available Data Conditions

The update rate for invalid or not available data conditions is three times per second.

Invalid Data

Invalid data is obtained whenever the meter over-ranges. If “OL” appears, start a new batch by pressing the Zero button. All data used to generate a batch result must be valid.

Not Available Data

Measurement data may be unavailable at certain times during meter operation. When data is not available, a series of dashes appears in the measurement area of the display. The following conditions will generate unavailable data:

- Meter is powered on and no pulse triggers are detected.
- Statistics mode is entered and batch data has not been compiled.

Setup Parameters

This section explains how to select and set user-definable parameters.

Setup/Local

The Setup/Local button serves several purposes:

1. If an edit cycle is not in progress, pressing the Setup/Local button initiates an edit cycle. If the next button pressed represents an edit parameter (STAT, WAVE, AREA, ATTEN, TRIG, or AVG button), Edit mode is entered and the parameter for the applicable button may be edited using the edit buttons. Additional presses of the Setup/Local button—without first pressing an edit parameter button—are ignored.
2. If a parameter has not been selected, pressing the Setup/Local button cancels an edit cycle.
3. If Edit mode is active and a parameter has been selected, pressing the Setup/Local button commits the adjusted edit parameter to the instrument.

4. Setup/Local is used to cancel the front panel lockout when Remote Control is active.

Remote Control is active when the instrument is connected to a host computer via a USB connection and is communicating with a host application program. While the instrument is in Remote Control mode, all front panel buttons—except Power/Backlight and Setup/Local—are disabled. Pressing the Setup/Local button while in Remote Control mode cancels Remote Control and returns the instrument to Local (all instrument functions available) mode.

The Setup/Local button is also used to edit user-defined parameters. Pressing this button initiates a parameter edit cycle.

A *parameter edit cycle* consists of:

- Pressing the Setup/Local button to begin the cycle.
- Pressing the parameter button (Stat, Wave (λ), Area, Avg, Atten, or Trig) that needs to be edited. If an edit cycle has been initiated, successive presses of the same edit parameter button will be ignored.
- Using the arrow buttons to select the appropriate field and adjust the data value.
- Pressing the Setup/Local button a second time to end the cycle and commit the new data value.

Stat

Pressing the Stat button will:

- Enter Statistics mode—if Statistics mode is not active and the button is pressed for less than two seconds.
- Exit Statistics mode.
- Enter Edit mode and select the statistics parameters to be edited (Batch Size and Restart mode) after edit cycle initiation (if Setup/Local is pressed beforehand). Batch size is 2 to 99,999 pulses.
- View the statistics parameters if the button is pressed for two seconds or more.

Wave

λ You can configure FieldMaxII-P to automatically account for any difference between the laser wavelength and the calibration wavelength.

After pressing Setup/Local, the Wave button is used to enter Edit mode and set the wavelength. If an edit cycle has not been initiated, pressing the Wave button will display the wavelength value. The available wavelength range is 1.00 to 99,999 nm. The actual range is sensor-dependent.

If Wavelength compensation information is not programmed into the sensor, you will not be able to change the wavelength data value.

Area

This mode allows the measurement of laser energy in terms of fluence.

The parameter is entered as a diameter and assumes a circular beam or aperture. The range for Area mode is 0.01 to 999.99 mm.

Pressing the Area button will:

- Toggle the state of Area Correction mode, if the button is pressed for less than one second. Enter Edit mode and select the beam diameter, if Setup/ Local is pressed beforehand.
- View the beam diameter, if the button is pressed for one second or more.

Avg

Average mode enables display smoothing, which suppresses variations in the display reading that can make it difficult to read.

Display values are averaged by pulses.

The FieldMaxII-P uses a boxcar averaging method, with a boxcar length of 32 pulses.

Pressing the Avg button toggles the state of Average mode.

Atten

Attenuation mode allows you to get true measurements using an attenuator that has a known attenuation factor. When Attenuation Correction mode is enabled, the measured value is adjusted to indicate the measurement at the attenuator and not the sensor. The range for this mode is 0.01 to 999.99.

Here's an example of how to determine the attenuation correction factor that needs to be set in the FieldMaxII-P meter: If a 1 J laser beam is focused through an attenuator that has an attenuation factor of 50%, then, to get a true laser measurement value, the correction factor in the FieldMaxII-P instrument needs to be set to *2*. In other words, since only half the energy of the beam is transmitted through the attenuator, the measured result must be doubled to obtain a true laser measurement.

The Atten button is used to initiate several activities:

- Toggle the state of Attenuation Correction mode, if the button is pressed for less than one second.
- Enter Edit mode and select the attenuation factor, if Setup/Local is pressed beforehand.
- View the attenuation factor, if the button is pressed for one second or more.

Trig

After pressing the Setup/Local button, the Trig button is used to enter Edit mode and select the trigger level parameter that will be edited. If an edit cycle has not been initiated, pressing the Trig button will display the trigger level parameter. Trigger has a range of 2 to 20% of full scale. Refer to "Internal Triggering Mode" on page 35 for details of the trigger function

Button Functions

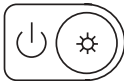
Hz

Pressing the Hz button toggles the Pulse Frequency Display mode. If Hz mode is on, the pulse frequency displays in the parameter edit region of the display. If Hz mode is off, the pulse frequency does not display. If the pulse frequency is >300 Hz, the displays shows a series of three dashes, followed by the letters, “Hz.”

Zero

The Zero button is used to manually start a batch while in Statistics mode.

Power Switch and Backlight Toggle Button



The combination Power Switch and Backlight Toggle button serves the dual purpose of turning power on/off to the meter, and toggling the backlight.

- When the meter is off, the power-on state is activated by pressing the button for one second.
- Pressing the button for one second while in the power-on state turns the meter off.
- When the meter is in the power-on state, the backlight state is toggled by pressing the button for less than one second.
- The backlight is always off when power is first applied to the meter.

Up and Down Arrows

These buttons serve a dual purpose. When Edit mode is active, the buttons are used to adjust the currently-selected edit field. When Edit mode is not active, the buttons are used to select the measurement range.

Left and Right Arrows

This button pair has a dual purpose. When the Edit mode is active, the buttons allow you to select the edit field of the currently-selected edit parameter. When the Edit mode is not active and Statistics mode is active, the buttons are used to select the statistical parameter of interest (MAX, MIN, MEAN). These buttons are nonfunctional when Edit mode and Statistics mode are not active.

Statistics Mode

FieldMaxII-P can be configured to display statistical data instead of instantaneous measurements. Statistical data is generated on a pulse-by-pulse basis.

For more information on selecting parameters while in Statistics mode, refer to "Left and Right Arrows," above.

Invalid Data

A statistical batch requires valid data for every data point in the batch. If a batch collection of data is in process and invalid data is measured, the batch is considered contaminated and the batch immediately ends with no batch result computed.The error that caused the contaminated batch is displayed.

Measurement Display and Range Selection

The display update rate for numeric measurement is three times per second.

Measurement range is selected in decade steps. Range selection (shown in Table 2 on page 34) is dependent on the sensor characteristics, as well as user-determined measurement settings.

FieldMaxII-P uses the *"3's" Rule*—a display formatting rule in which the display value is not allowed to exceed 3, 30, or 300, depending on where the decimal point falls, with the decimal point located in a fixed position, as determined by the current range. Typically if a reading exceeds the "3's" limit, the instrument is over-ranged.

Over-ranging refers to a meter setup condition in which the sensor output signal is greater than the maximum allowable level for the selected range. An “OL” (over-load) appearing on the display signifies an over-range condition. Over-ranging generates invalid data.Table 2 indicates the display format for different full-scale range settings.

Table 2. Full Scale Range Settings

FULL SCALE MEASUREMENT	DISPLAY FORMAT
3 nJ	X.YYY nJ
30 nJ	XX.YY nJ
300 nJ	XXX.Y nJ
3 µJ	X.YYY µJ
30 µJ	XX.YY µJ
300 µJ	XXX.Y µJ
3 mJ	X.YYY mJ
30 mJ	XX.YY mJ
300 mJ	XXX.Y mJ
3 J	X.YYY J
30 J	XX.YY J
300 J	XXX.Y J

When area correction is enabled in Joules mode, a "/cm^2" will be appended to the units.

The Range Hint annunciator displays the full-scale range value with the engineering prefix and units omitted. For example, when the range is 30 J, the Range Hint annunciator displays "30." Note that it also displays "30" when the selected range is 30 mJ, 30 µJ, or 30 nJ. For more information about the Range Hint annunciator, refer to "Range Hint" on page 22.

Internal Triggering Mode

For greatest accuracy and repeatability, FieldMaxII-P must trigger reliably for each laser pulse. *Internal triggering* refers to extracting an artificial trigger from the incoming signal.

To successfully extract an internally-generated trigger, set the meter range so that the incoming signal is at least 5% of the full scale. Set the trigger level at least 2% above the noise level and at least 2% below the peak height.

For example, with a peak height of 300 mJ, set the meter to a range of 3 J. A peak will occur at 10% of full scale, well above the 5% level. The trigger level should not be set higher than 8%. If the noise level is 5% of full scale (approximately 150 mJ), the trigger level should not be set lower than 7%.

In the example shown in Figure 12, the internal trigger threshold has been set to 8% (shown as a dashed line). Pulse A will definitely not generate a reliable trigger. Pulse B may generate a trigger, but not reliably. Pulses C and D will definitely generate reliable triggers.

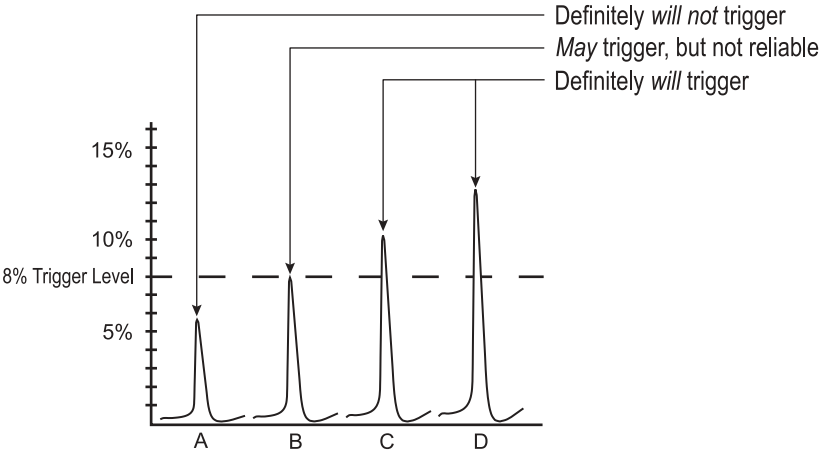


Figure 12. Internal Trigger Threshold

A full-scale signal on one range may not trigger on the next higher range unless the trigger level is set to less than 10%. For example, a near full-scale signal of 280 mJ on the 300 mJ range is less than 10% of full scale on the next higher range (3 J), and would therefore not trigger the reading. To obtain reliable triggering in this instance, adjust the trigger level to less than 8%.

The trigger is synchronous with the leading edge of the pulse, but the actual peak is determined algorithmically by sampling the input signal near the trigger. From the trigger point forward, the algorithm searches for peaks and from the trigger point back, it searches for a baseline.

Hz Display Mode

If desired, pulse frequencies may be displayed when a pyroelectric sensor is attached to the meter. For more information, refer to “Hz” on page 31.

Zeroing

The Zero button is used to manually re-zero the statistical batch.

SPECIAL TOPICS

This section discusses the following topics:

Trigger States and the Trigger Annunciator

Two segments are dedicated to the indication of the trigger state of the meter. These are the TRIG and the ? segment that follows it.

There are two possible trigger states: “triggered” and “wait for trigger.” The triggered state indicates that the meter is capturing laser pulses. The wait for trigger state indicates an absence of pulse triggers. When a pulse is captured, the trigger state becomes active and the wait for trigger state becomes inactive. If no pulses are captured within 1.67 seconds since the last pulse was captured, the wait for trigger state becomes active and the triggered state becomes inactive. The TRIG annunciator reads “TRIG” in the triggered state and “TRIG?” in the wait for trigger state.

Digital Tuning Feature Use in Statistics Mode

The tuning needles and zone indicator bars are not present in Statistics mode.

ActiveX Installation

Included with the software is an ActiveX server DLL that enables a programmer to quickly and efficiently communicate with FieldMaxII. This ActiveX server DLL—and its accompanying tutorial—are shipped with every FieldMaxII-P. View the Readme file on the accompanying CD for ActiveX installation instructions.

USB Driver Installation

When first connecting the meter to a PC with the USB cable, you will be prompted through an installation process. USB drivers will be automatically installed onto your computer. Insert the CD into your CD drive when prompted.

HOST INTERFACE

This section presents the following topics:

- Introduction (this page)
- ActiveX usage guidelines (this page)

Introduction

The interface is implemented as an ActiveX DLL server called **FieldMax2DLLServer.dll**. The DLL is COM-compliant and serves to transport data between the FieldMaxII and a host application program. This DLL has no GUI elements.

ActiveX Usage Guidelines

The following discussion explains how to use the FieldMaxII ActiveX DLL server in an application program. The DLL server package—provided by Coherent—defines all ActiveX interfaces and object classes. The user uses the functionality provided by the ActiveX package to communicate with the FieldMaxII meter.

Usage Overview

To reliably communicate with the meter, the program must provide infrastructure via the ActiveX server. A number of object classes and interfaces that will aid in setting up the infrastructure are provided.

cFM2Listener class and IFM2Listener interface—An object class that dispatches meter USB connect/disconnect and data event notifications to the application code. Notifications are permitted by means of a user-implemented callback object that is passed to this object. It implements the IFM2Listener interface. One cFM2Listener object is needed to manage all meters with which the program will communicate.

cFM2ScanUSBForChange class—An object class that is the engine for capturing meter USB connect and disconnect status changes. The cFM2Listener object is passed to this object to provide a means of sending USB connection status changes to the application code via the listener. One cFM2ScanUSBForChange object is needed to manage all meters with which the program will communicate.

cFM2ScanForData class—An object class that is the engine for capturing meter data transmissions. One cFM2ScanForData object is needed to manage all meters with which the program will communicate.

cFM2Notify class—A callback object that implements the IFM2DeviceEvents interface. The application code is responsible for providing the code and instantiating this object (that is, implementing the class and instantiating the class object). All status change events and data transmissions are channeled through this object to the application code. One cFM2Notify object is needed for all meters with which the program will communicate.

cFM2Device class—An object class that abstracts configuration and status access to a meter. Configuration is accessed by calling any of the numerous cFM2Device.<parameter>Command() methods. Status

is accessed by reading any of the numerous cFM2Device.<parameter> properties. It implements the IFM2Device interface. The cFM2Listener object creates a cFM2Device object with every meter USB connect event.

cFM2Devices class—An object class that manages a collection of cFM2Device objects for application code indexing. The collection is managed internally within the cFM2Listener object. cFM2Device objects are added to the collection with every meter USB connect event. Items in the collection are searchable using the device index key. cFM2Device objects are removed from the collection with every meter USB disconnect event. The collection is passed to the cFM2Notify object with every meter USB connect or disconnect event.

Coding Essentials

Setup

First, supply the callback object by writing code for the cFM2Notify class that implements IFM2DeviceEvents interface.

In the application code:

1. Create a cFM2Listener object and obtain an IFM2Listener interface to it.
2. Create the cFM2ScanUSBForChange and cFM2ScanForData objects.
3. Create the cFM2Notify callback object.

4. Pass the callback object to the cFM2Listener object.
5. Start the cFM2ScanUSBForChange object status change engine.

The following Visual Basic 6.0 code shows the steps:

```
'Global declarations:

'This is the private instance of the cFM2Listener class
Dim FieldMax2Listener As cFM2Listener

'This points to the cFM2Listener object's IFM2Listener interface
Dim ThisListener As IFM2Listener

'Timer objects
Dim ScanUSBForChange As cFM2ScanUSBForChange
Dim ScanForData As cFM2ScanForData

'Callback sink (receives event notifications)
Dim NotifyMe As cFM2Notify

'Executable code:

'Create a cFM2Listener object and obtain an IFM2Listener interface
'to it
Set FieldMax2Listener = New cFM2Listener
Set ThisListener = FieldMax2Listener

'Create the cFM2ScanUSBForChange and cFM2ScanForData objects
Set ScanUSBForChange = New cFM2ScanUSBForChange
Set ScanForData = New cFM2ScanForData

'Create the cFM2Notify callback object
Set NotifyMe = New cFM2Notify

'Pass the callback object to the cFM2Listener object
FieldMax2Listener.DeviceEvents = NotifyMe

'Start the cFM2ScanUSBForChange object status change engine
ScanUSBForChange.CheckTimer ThisListener
```

The Callback Object

The callback object code—supplied by the application code—must completely conform to the cFM2Notify class definition and must implement the IFM2DeviceEvents interface. All status change events and data transmissions are channeled through this object to the application code. The code within this object will steer data and events as required by the designer.

The following Visual Basic 6.0 code provides the minimal framework in the implementation of cFM2Notify for handling callbacks:

```
Implements IFM2DeviceEvents

Private m_CallbackEvent As String
Private m_CallbackMessage As String
Private m_DeviceIndex As Integer
Private m_SerialNumber As String
Private m_ZeroDeviceTimeoutCounter As Integer

'Methods

Private Sub IFM2DeviceEvents_DisplayErrorToClient()
      'Steer the m_CallbackEvent string to the
      'appropriate application code error handler
End Sub

Private Sub IFM2DeviceEvents_NotifyData( _
        ByVal CallbackData As IFM2DeviceEvents)
      'Steer the CallbackData object to the appropriate
      'application data event handler
End Sub

Private Sub IFM2DeviceEvents_NotifyDeviceStatus( _
        ByVal CallbackData As IFM2DeviceEvents, _
        ByVal DevicesList As cFM2Devices)
      frmMain.NotifyDeviceStatus CallbackData, DevicesList
      'Steer the CallbackData and DevicesList objects to the
      'appropriate application status event handler
End Sub
```

```
Private Sub IFM2DeviceEvents_DisplayZeroDeviceProgressToClient()
      'Steer the m_CallbackMessage string and
      'm_ZeroDeviceTimeoutCounter integer to the appropriate
      'application code error handler
End Sub

'Read/write properties

Private Property Let IFM2DeviceEvents_CallbackEvent( _
       ByVal RHS As String)
      m_CallbackEvent = RHS
End Property

Private Property Get IFM2DeviceEvents_CallbackEvent() As String
      IFM2DeviceEvents_CallbackEvent = m_CallbackEvent
End Property

Private Property Let IFM2DeviceEvents_CallbackMessage( _
       ByVal RHS As String)
      m_CallbackMessage = RHS
End Property

Private Property Get IFM2DeviceEvents_CallbackMessage() As String
      IFM2DeviceEvents_CallbackMessage = m_CallbackMessage
End Property

Private Property Let IFM2DeviceEvents_DeviceIndex( _
       ByVal RHS As Integer)
      m_DeviceIndex = RHS
End Property

Private Property Get IFM2DeviceEvents_DeviceIndex() As Integer
      IFM2DeviceEvents_DeviceIndex = m_DeviceIndex
End Property

Private Property Let IFM2DeviceEvents_SerialNumber( _
       ByVal RHS As String)
      m_SerialNumber = RHS
End Property

Private Property Get IFM2DeviceEvents_SerialNumber() As String
      IFM2DeviceEvents_SerialNumber = m_SerialNumber
End Property

'Write only properties
```

```
Private Property Let IFM2DeviceEvents_ZeroDeviceTimeoutCounter( _
        ByVal RHS As Integer)
      m_ZeroDeviceTimeoutCounter = RHS
End Property
```

Status Change Notifications

Status change notifications are dispatched to the application code through the IFM2DeviceEvents_NotifyDeviceStatus callback method. The CallbackData and DevicesList objects are passed to the application. Status change event types are steered by testing the CallbackData.CallbackEvent string. The CallbackEvent string will be one of the following:

- "MeterAdded"
- "MeterRemoved"

The DevicesList is updated with each call to this callback method. A reference to the cFM2Device object from which this callback comes is obtained by indexing into the DevicesList collection through the DevicesList.Item search method using the string version of CallbackData.DeviceIndex as the search key.

The data check timer must be started for the new devices list when a meter is added.

The following Visual Basic 6.0 code shows how to obtain reference to the cFM2Device object and start the data check timer when a meter is added:

```
'Declaration:

Dim ThisDevice As IFM2Device

'Executable code:

'Get the current device from the collection
Set ThisDevice = m_DevicesList.Item(CallbackData.DeviceIndex)
```

```
'Start the data check timer
ScanForData.CheckTimer m_DevicesList
```

The device and device collection object may be persisted to extend the lifetime of these objects beyond the scope of the callback.

When a meter is removed, the DevicesList will be current but the CallbackData.DeviceIndex search key will not be valid since the device is gone.

Data Transmission Notifications

Data transmission notifications are dispatched to the application code through the IFM2DeviceEvents_ NotifyData callback method. The CallbackData object is passed to the application. Data transmission types are steered by testing the CallbackData.CallbackEvent string. The CallbackEvent string will be one of the following:

- "Fault"
- "ProbeRemoved"
- "ProbeAdded"
- "PowerOn"
- "PowerOff"
- "MeasurementData"
- "PacketIsOverrange"
- "OverTemperature"
- "MeasurementDataLost"

A reference to the cFM2Device object from which this callback comes is obtained by indexing into the DevicesList collection (persisted through the Notify-

DeviceStatus callback method) through the DevicesList.Item search method using the string version of CallbackData.DeviceIndex as the search key.

The following Visual Basic 6.0 code shows how to obtain reference to the cFM2Device object and start the data check timer when a data transmission is notified:

```
'Declaration:

Dim ThisDevice As IFM2Device

'Executable code:

'Get the current device from the collection
Set ThisDevice = m_DevicesList.Item(CallbackData.DeviceIndex)

'Handle the data
Select Case CallbackData.CallbackEvent
      Case "Fault"
        'Fault event is enumerated in ThisDevice.LastFault

      Case "ProbeRemoved"
        'Probe was removed

      Case "ProbeAdded"
        'Probe was added, probe type is ThisDevice.ProbeType

      Case "PowerOn"
        'Meter power is on, all properties are current and accurate

      Case "PowerOff"
        'Meter power is off, only the ThisDevice.PowerState is
        'current and accurate

      Case "MeasurementData"
        'Measurement data event, data is ThisDevice.LastData

      Case "PacketIsOverrange"
        'Measurement data event, data is over-ranged

      Case "OverTemperature"
        'Probe is in an over-temperature state

      Case "MeasurementDataLost"
        'Data transmission gap due to buffer overflow
```

```
      Case Else
        'Unknown event
End Select
```

A **Fault** event indicates a meter malfunction. The fault code may be queried by reading the cFM2Device.LastFault property. The fault code may be dismissed by calling the cFM2Device.DismissFault() method.

A **Probe Removed** event indicates that a probe was removed from the meter. Data transmission will cease when the probe is removed. If the meter USB disconnect event occurs when the probe is connected, a Probe Removed event will not happen before the Meter Removed event.

A **Probe Added** event indicates that a probe was added to the meter. Data transmission can happen only after a probe is attached.

A **Power On** event indicates that meter power has been turned on.

A **Power Off** event indicates that meter power has been turned off.

A **Measurement Data** event indicates a valid measurement reading is available The measurement reading is in two parts: a value (Watts or Joules) and a period. The value may be queried by reading the cFM2Device.LastData property. The period may be queried by reading the cFM2Device.LastPeriod property. The period is expressed in units of 8 µsec intervals when a pyroelectric probe is attached. When a thermopile or optical probe is attached and an overtemperature condition exists, the period is *1*; otherwise, it is *0*.

An **Overrange** packet event indicates that the current reading is overranged.

An **Overtemperature** event indicates that the probe is in an overtemperature condition. If the probe is in an overtemperature condition, one Overtemperature event will accompany every measurement data event.

A **Measurement Data Lost** event indicates that the meter communication system has dropped data and is not processing data fast enough. Data synchronization is lost and the system automatically attempts to resynchronize the data stream. A 1-2 second interruption in the data stream will occur during resynchronization attempts.

Error Callback

There is a variety of reasons for the root cause of errors. When the DLL server detects an error, the IFM2DeviceEvents_DisplayErrorToClient () callback method is called. The callback method may find the string description of the error in the IFM2DeviceEvents_ CallbackMessage property.

Zero Status Callback

Probe zeroing monopolizes the meter and can last up to 60 seconds. Zero status events are dispatched to the application code using the IFM2DeviceEvents_ DisplayZeroDeviceProgressToClient() callback method. Zero status events are dispatched—once per second while zeroing is in progress—to permit the application code to monitor progress. The cFM2Notify callback object CallbackMessage and ZeroDeviceTimeout-Counter properties are used to monitor zero progress. The CallbackMessage string is a text description of the callback event. The ZeroDeviceTimeoutCounter integer

is a countdown value that starts at 60 and decrements by one for each event. The terminal value for ZeroDeviceTimeoutCounter will be *0* or *-1* when zero completes. *0* indicates a successful completion. *-1* indicates a failure.

Teardown

Devices connected to the application must be closed. The cFM2Notify object passes a reference to the cFM2Devices collection with every callback to the application code via the IFM2DeviceEvents_NotifyDeviceStatus callback method. All connections must be closed by calling the IFM2Device_CloseAllUSBDeviceDrivers() using the IFM2Device_DeviceHandle as the method argument.

The following Visual Basic 6.0 code shows how to close the connections:

```
'Global declarations:

'Devices collection returned from the callback object
Dim DevicesList As cFM2Devices

'Executable code:

If Not (DevicesList Is Nothing) Then
      If Not (DevicesList.Count = 0) Then
        Dim device As IFM2Device

        For Each device In DevicesList
            device.CloseAllUSBDeviceDrivers device.DeviceHandle
        Next
      End If
End If
```

The remaining teardown is simply the process of undoing the setup.

In the application code:

1. Stop all engine timers.
2. Destroy the cFM2Notify callback object.
3. Destroy the cFM2ScanUSBForChange and cFM2ScanForData objects.
4. Destroy the cFM2Listener object and the reference to the IFM2Listener interface to it.

The following Visual Basic 6.0 code shows the steps:

```
'Stop all engine timers
ScanUSBForChange.StopTimer
ScanForData.StopTimer

'Destroy the cFM2Notify callback object
Set NotifyMe = Nothing

'Destroy the cFM2ScanUSBForChange and cFM2ScanForData objects
Set ScanForData = Nothing
Set ScanUSBForChange = Nothing

'Destroy the cFM2Listener object and
'the reference to the IFM2Listener interface to it
Set FieldMax2Listener = Nothing
Set ThisListener = Nothing
```

Teardown is an important cleanup function. Failing to properly cleanup can result in memory leaks, a hung application, and other unexpected behavior.

Type Library Details

The type information in this section was obtained using OLEView. Additional explanations accompany each entry. The type library entries that serve internal processing purposes only are not explained.

Interface IFM2Listener

method GetUSBDeviceStatusChange();
(serves internal processing purposes only)

propget DeviceIndex([out, retval] short);*
(serves internal processing purposes only)

propget SerialNumber([out, retval] BSTR);*
Read only property to identify the serial number that pertains to a meter USB connect/disconnect event.

propput DeviceHandle([in] long);
propget DeviceHandle([out, retval] long);*
(serves internal processing purposes only)

Interface cFM2Listener

propget DeviceIsPresent([out, retval] VARIANT_ BOOL);*
Read only property that reflects the state of meter added or removed. It is true if the meter is added (connected) and false if the meter is removed (disconnected).

*propget DevicesList([out, retval] cFM2Devices**);*
Read only property, a collection type that holds the list of all connected meters.

propput DeviceEvents([in] IFM2DeviceEvents rhs);*
Write only property used to pass the callback object to the listener.

Interface IFM2DeviceEvents

method DisplayErrorToClient();
The callback method that dispatches errors. Errors are described in text with the CallbackMessage property.

method NotifyData([in] IFM2DeviceEvents Callback Data);*
The data event callback method. Notification details are found in the CallbackData properties passed into the method.

method NotifyDeviceStatus([in] IFM2DeviceEvents CallbackData, [in] cFM2Devices* DevicesList);*
The meter USB connect/disconnect event callback method. Notification details are found in the CallbackData properties passed into the method. The device list, which is updated with every callback, contains the current list of connected meters.

method DisplayZeroDeviceProgressToClient();
The zero progress callback method. Zero status events are dispatched once per second while zeroing is in progress to permit the application code to monitor progress. The CallbackMessage string is a text description of the callback event. The ZeroDeviceTimeoutCounter integer is a countdown value that starts at 60 and decrements by one for each event. When zero completes, the terminal value for ZeroDeviceTimeoutCounter will be *0* or *-1*. *0* indicates a successful completion. *-1* indicates a failure.

propput CallbackEvent([in] BSTR);
propget CallbackEvent([out, retval] BSTR);*
The string property that identifies the event type.

propput CallbackMessage([in] BSTR);
propget CallbackMessage([out, retval] BSTR);*
The string property that identifies additional event information.

propput DeviceIndex([in] short);
propget DeviceIndex([out, retval] short);*
The device index property that is used to index into the DevicesList collection to locate device objects.

propput SerialNumber([in] BSTR);
propget SerialNumber([out, retval] BSTR);*
The property to identify the serial number that pertains to an event.

propput ZeroDeviceTimeoutCounter([in] short rhs);
A write only property used to enable the application to store the zero countdown timer.

Interface cFM2ScanUSBFor-Change

*method CheckTimer([in, out] IFM2Listener** This-Listener);*
The method to start the engine for detecting meter USB connect and disconnect events.

method StopTimer();
The method to stop the engine for detecting meter USB connect and disconnect events.

Interface cFM2ScanForData

*method CheckTimer([in, out] cFM2Devices** DevicesList);*
The method to start the engine for detecting data events.

method StopTimer();
The method to stop the engine for detecting data events.

Interface IFM2Device

method CloseAllUSBDeviceDrivers([in] long DeviceHandle);
The method call to close a connection. Use the DeviceHandle property as the argument.

method DismissFault();
The method used to dismiss any active fault. Query the LastFault property to obtain the fault code before it is dismissed.

method GetUSBDeviceData([in] long DeviceHandle);
(serves internal processing purposes only)

method RestartBatch();
The method to restart a statistics batch.

method ZeroDevice();
The method to start a zero procedure.

propput DeviceEvents([in] IFM2DeviceEvents rhs);*
The write only property which holds a reference to the callback object so that data events can be dispatched to the application.

method AnalogOutFullscaleVoltageCommand([in] short CommandOption);
The method to set the analog out full scale voltage. Valid ComandOption values follow the Visual Basic 6.0 enumeration of:

```
Public Enum AnalogOutFullscaleVoltage
      aofv1V = 1
      aofv2V = 2
      aofv5V = 3
End Enum
```

method AreaCorrectionDiameterCommand([in] double CommandOption);
The method to set the area correction aperture diameter in millimeters. Valid CommandOption values range from *0.01* to *999.99*.

method AreaCorrectionEnabledCommand([in] VARIANT_BOOL CommandOption);
The method to enable or disable area correction mode. Valid CommandOption values are *0* for disabled and non-zero for enabled.

method AttenuationCorrectionFactorCommand([in] double CommandOption);
The method to set the attenuation correction factor. Valid CommandOption values range from *0.01* to *999.99*.

method AttenuationCorrectionModeEnabledCommand ([in] VARIANT_BOOL CommandOption);
The method to enable or disable attenuation correction mode. Valid CommandOption values are *0* for disabled and non-zero for enabled.

method AutoRangingEnabledCommand([in] VARIANT_BOOL CommandOption);
The method to enable or disable auto ranging mode. Valid CommandOption values are *0* for disabled and non-zero for enabled.

method AverageModeEnabledCommand([in] VARIANT_BOOL CommandOption);
The method to enable or disable display smoothing mode. Valid CommandOption values are *0* for disabled and non-zero for enabled.

method AverageWindowSizePulsesCommand([in] short CommandOption);
The method to set the display smoothing window size in pulses. This parameter has effect only when a pyroelectric probe is attached to a TOP meter. Valid CommandOption values range from *2* to *1000*.

method AverageWindowSizeSecondsCommand([in] short CommandOption);
The method to set the display smoothing window size in seconds. This parameter has effect only when a thermopile or optical probe is attached to a TOP meter. Valid CommandOption values range from *1* to *60*.

method BacklightCommand([in] VARIANT_BOOL CommandOption);
The method to turn on or off the backlight. Valid CommandOption values are *0* for off and non-zero for on.

method BatchRestartModeCommand([in] short CommandOption);
The method to set the batch restart mode. Valid Comand Option values follow the Visual Basic 6.0 enumeration of:

```
Public Enum BatchRestartMode
      brmManual = 1
      brmAuto = 2
End Enum
```

method BatchSizePulsesCommand([in] long Command Option);
The method to set the statistics batch size in pulses. This parameter has effect only when a pyroelectric probe is attached. Valid CommandOption values range from *2* to *99999*.

method BatchSizeSecondsCommand([in] long CommandOption);
The method to set the statistics batch size in seconds. This parameter has effect only when a thermopile or optical probe is attached. Valid CommandOption values range from *1* to *99999*.

method HertzModeCommand([in] VARIANT_BOOL CommandOption);
The method to enable or disable frequency display mode. This parameter has effect only when a pyroelectric probe is attached. Valid CommandOption values are *0* for disabled and *non-zero* for enabled.

method HoldModeCommand([in] VARIANT_BOOL CommandOption);
The method to enable or disable display hold mode. This parameter has effect only with a TO meter. Valid CommandOption values are *0* for disabled and *non-zero* for enabled.

method MeasurementModeCommand([in] short CommandOption);
The method to set the measurement mode (Watts or Joules). Valid ComandOption values follow the Visual Basic 6.0 enumeration of:

```
Public Enum MeasurementMode
      mmJoules = 1
      mmWatts = 2
End Enum
```

method PowerStateCommand([in] VARIANT_BOOL CommandOption);
The method to turn on or off the meter. Valid CommandOption values are *0* for off and *non-zero* for on.

All device properties except PowerState are not valid (hence not meaningful) when meter power is off.

---

method RangeCommand([in] double CommandOption);
The method to set the meter fullscale range. Valid CommandOption values must not exceed the MaxRange property. The range is expressed as the expected fullscale value that is expected to be measured. The meter automatically selects the next higher available range to accommodate the requested fullscale range.

method SpeedupAnalogOutputCommand([in] VARIANT_BOOL CommandOption);
The method to enable or disable application of thermopile speedup to the analog output channel. Valid CommandOption values are 0 for disabled and non-zero for enabled.

method SpeedupDigitalDisplayCommand([in] VARIANT_BOOL CommandOption);
The method to enable or disable application of thermopile speedup to the digital display. Valid CommandOption values are *0* for disabled and *non-zero* for enabled.

method SpeedupHostDataCommand([in] VARIANT_BOOL CommandOption);
The method to enable or disable application of thermopile speedup to the host data output channel. Valid CommandOption values are 0 for disabled and non-zero for enabled.

method SpeedupMeterCommand([in] VARIANT_BOOL CommandOption);
The method to enable or disable application of thermopile speedup to the simulated analog meter. Valid CommandOption values are 0 for disabled and non-zero for enabled.

method StatisticsModeCommand([in] short Command Option);
The method to set the statistics mode. Valid CommandOption values follow the Visual Basic 6.0 enumeration of:

```
Public Enum StatisticsMode
      smOff = 1
      smMax = 2
      smMin = 3
      smMean = 4
      smStdv = 5
End Enum
```

method TriggerLevelCommand([in] short Command Option);
The method to set the pulse trigger level in percent. This parameter has effect only when a pyroelectric probe is attached. Valid CommandOption values range from *2* to *20*.

method WavelengthCommand([in] long Command Option);
The method to set the wavelength of operation for wavelength correction. Valid CommandOption values range from the MinRange property to the MaxRange property.

propget AnalogOutFullscaleVoltage([out, retval] short);*
The read only property indicating the analog out fullscale voltage.

propget AreaCorrectionDiameter([out, retval] double);*
The read only property indicating the area correction diameter.

propget AreaCorrectionEnabled([out, retval] VARIANT_BOOL);*
The read only property indicating the state of area correction mode.

propget AttenuationCorrectionFactor([out, retval] double);*
The read only property indicating the attenuation correction factor.

propget AttenuationCorrectionModeEnabled ([out, retval] VARIANT_BOOL);*
The read only property indicating the attenuation correction mode.

propget AutoRangingEnabled([out, retval] VARIANT_BOOL);*
The read only property indicating the auto ranging mode.

propget AverageModeEnabled([out, retval] VARIANT_BOOL);*
The read only property indicating display smoothing mode.

propget AverageWindowSizePulses([out, retval] short);*
The read only property indicating display smoothing window size in pulses. This property is meaningful only when a pyroelectric probe is attached to a TOP meter.

propget AverageWindowSizeSeconds([out, retval] short);*
The read only property indicating display smoothing window size in seconds. This property is meaningful only when a thermopile or optical probe is attached to a TOP meter.

propget Backlight([out, retval] VARIANT_BOOL);*
The read only property indicating the backlight state.

propget BatchRestartMode([out, retval] short);*
The read only property indicating batch restart mode.

propget BatchSizePulses([out, retval] long);*
The read only property indicating batch size in pulses. This property is meaningful only when a pyroelectric probe is attached to a TOP meter.

propget BatchSizeSeconds([out, retval] long);*
The read only property indicating batch size in seconds. This property is meaningful only when a thermopile or optical probe is attached to a TOP meter.

propget HertzMode([out, retval] VARIANT_BOOL);*
The read only property indicating the frequency display mode. This parameter has effect only when a pyroelectric probe is attached.

propget HoldMode([out, retval] VARIANT_BOOL);*
The read only property indicating hold mode state. This parameter has effect only with a TO meter.

propget LastData([out, retval] double);*
The read only property indicating the last measurement value. This property is read in response to a “MeasurementData” event.

propget LastFault([out, retval] short);*
The read only property indicating the fault code of the last fault event. This property is read in response to a “Fault” event. The last fault code may be cleared by calling the DismissFault() method.

propget LastPeriod([out, retval] double);*
The read only property indicating the last period value. This property is meaningful only when a pyroelectric probe is attached. Frequency in Hz is calculated as 125000/LastPeriod.

propget MaxRange([out, retval] double);*
The read only property indicating the maximum full scale range that the meter will accommodate. This property automatically varies as different probes are attached.

propget MaxWavelength([out, retval] long);*
The read only property indicating the maximum wavelength that the meter will accommodate. This property automatically varies as different probes are attached.

propget MeasurementMode([out, retval] short);*
The read only property indicating the measurement mode (Watts or Joules).

propget MeterType([out, retval] short);*
The read only property indicating the meter type (TOP, TO, or P). This property follows the Visual Basic 6.0 enumeration of:

```
Public Enum MeterType
      mtTOP = 1
      mtTO = 2
      mtP = 3
End Enum
```

propget MinRange([out, retval] double);*
The read only property indicating the minimum full scale range that the meter will accommodate. This property automatically varies as different probes are attached.

propget MinWavelength([out, retval] long);*
The read only property indicating the minimum wavelength that the meter will accommodate. This property automatically varies as different probes are attached.

propget PowerState([out, retval] VARIANT_BOOL);*
The read only property indicating meter power state.

propget ProbeType([out, retval] short);*
The read only property indicating the attached probe type. This property follows the Visual Basic 6.0 enumeration of:

```
Public Enum ProbeType
      ptNone = 0
      ptThermopile = 1
      ptPyroelectric = 2
      ptOptical = 3
End Enum
```

propget Range([out, retval] double);*
The read only property indicating the currently selected full scale range.

propget SpeedupAnalogOutput([out, retval] VARIANT_BOOL);*
The read only property indicating mode of application of thermopile speedup to the analog output channel (enabled or disabled).

propget SpeedupDigitalDisplay([out, retval] VARIANT_BOOL);*
The read only property indicating mode of application of thermopile speedup to the digital display (enabled or disabled).

propget SpeedupHostData([out, retval] VARIANT_BOOL);*
The read only property indicating mode of application of thermopile speedup to the host data output channel (enabled or disabled).

propget SpeedupMeter([out, retval] VARIANT_BOOL);*
The read only property indicating mode of application of thermopile speedup to the simulated analog meter (enabled or disabled).

propget StatisticsMode([out, retval] short);*
The read only property indicating the statistics mode.

propget TriggerLevel([out, retval] short);*
The read only property indicating pulse trigger level. This property is meaningful only when a pyroelectric probe is attached.

propget Wavelength([out, retval] long);*
The read only property indicating the wavelength of operation.

propput DeviceHandle([in] long);
propget DeviceHandle([out, retval] long);*
The property to be used as the argument to the Close-AllUSBDeviceDrivers() method.

propput DeviceIndex([in] short);
propget DeviceIndex([out, retval] short);*
The property to be used as the key to searches in the device list.

propput SerialNumber([in] BSTR);
propget SerialNumber([out, retval] BSTR);*
The property to be used to identify the meter serial number.

Interface cFM2Devices

method Item([in, out] BSTR IndexKey, [out, retval] IFM2Device**);*
The method used to search for device objects from the collection. The key is the string version of the device index (for example, if the device index is 3, IndexKey will be "3").

propget Count([out, retval] long);*
The read only property indicating the number of entries in the collection.

CALIBRATION AND WARRANTY

This section discusses the following topics:

Calibration

Coherent laser power and energy meters are precision instruments, capable of delivering very accurate measurements, as well as providing many years of useful service. To maintain this high level of performance, it is important to have your measurement system serviced and recalibrated once a year.

Coherent Calibration Facilities and Capabilities

As the largest laser manufacturer in the world, Coherent has been able to build state-of-the-art calibration facilities containing the widest possible range of laser types and technologies. This enables us to perform instrument and sensor calibration under virtually any combination of wavelength, power, and operating characteristics. Sensors are calibrated against NIST-traceable working standard sensors which are, in turn, calibrated against NIST-calibrated golden standard sensors. These working and golden standards are maintained with the utmost care, recalibrated annually, and verified even more regularly. We maintain multiple NIST-calibrated standards at many laser wavelengths to support the growing calibration needs of our customers. Optical calibration is a core competency at Coherent and we strive to continually improve our methods, precision, and repeatability. Additionally, most of the calibrations are performed with highly automated systems, thus reducing the possibility of human error to nearly zero. Strict quality inspections during many stages of calibration and testing assure a precise and accurate instrument that is NIST traceable and CE marked. The benefit to our customers is that instruments calibrated by Coherent will consistently perform as expected under their actual use conditions. We are a registered ISO 9001:2000 company, our products are NIST traceable, and our calibration labs are fully ANSI Z540 compliant.

In addition to the technological advantage, we also strive to deliver the best service in the industry, with a knowledgeable and responsive staff, and rapid turnaround.

Limited Warranty

Coherent, Inc. (the "Company") warrants its laser power and energy meters and sensors products ("Products") to the original purchaser (the "Customer") that the product is free from defects in materials and workmanship and complies with all specifications, active at the time of purchase, for a period of twelve (12) months.

Coherent, Inc. will, at its option, repair or replace any product or component found to be defective during the warranty period. This warranty applies only to the original purchaser and is not transferable.

Extended Lifetime Warranty

Coherent, Inc. (the "Company") offers original purchasers (the "Customer") purchasing laser power and energy meters and sensors products ("Products") an extended, lifetime warranty program, which includes all parts and labor. In order to qualify for this warranty, a Customer must return the Product to the Company for recalibration and recertification (traceable to NIST and MIL-STD-45662A) within one year from the date of purchase, and annually thereafter. The Company will recertify the Product, provide software upgrades, and perform any needed repairs, for a fixed service fee (as established by the Company from time to time and in effect at the time of service).

If the Product fails and is returned to the Company within one year following the date of recalibration service, the Company will, at its option, repair or replace the Product or any component found to be defective. This warranty applies only to the original purchaser and is not transferable.

If the Product is not returned for recalibration or service prior to the one-year anniversary, the lifetime warranty program expires. The lifetime warranty program may be reinstated, at Coherent's option, after completion of a fee-based product evaluation and repair, and subsequent recalibration and recertification service.

Warranty Limitations

The foregoing warranties shall not apply, and Coherent reserves the right to refuse warranty service, should malfunction or failure result from:

- Damage caused by improper installation, handling, or use.
- Laser damage (including sensor elements damaged beyond repair).
- Failure to follow recommended maintenance procedures.
- Unauthorized product modification or repair.
- Operation outside the environmental specifications of the product.

Coherent assumes no liability for Customer-supplied material returned with Products for warranty service or recalibration.

THIS WARRANTY IS EXCLUSIVE IN LIEU OF ALL OTHER WARRANTIES WHETHER WRITTEN, ORAL, OR IMPLIED. COHERENT SPECIFICALLY DISCLAIMS THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL

THE COMPANY BE LIABLE FOR ANY INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH ITS PRODUCTS.

Obtaining Service

In order to obtain service under this warranty, Customer must notify the Company of the defect before the expiration of the warranty period and make suitable arrangements for the performance of service. The Company shall, in its sole discretion, determine whether to perform warranty service at the Customer's facility, at the Company's facility or at an authorized repair station.

If Customer is directed by the Company to ship the product to the Company or a repair station, Customer shall package the product (to protect from damage during shipping) and ship it to the address specified by the Company, shipping prepaid. The customer shall pay the cost of shipping the Product back to the Customer in conjunction with annual recalibration and repair; the Company shall pay the cost of shipping the Product back to the Customer in conjunction with product failures within the first twelve months of time of sale or between annual recalibrations.

A Returned Material Authorization number (RMA) assigned by the Company must be included on the outside of all shipping packages and containers. Items returned without an RMA number are subject to return to the sender.

For the latest Customer Service information, refer to our website: www.coherent.com.

Detailed instructions on how to prepare a product for shipping are shown under “Product Shipping Instructions” on page 75.

Table 3. Coherent Service Centers

LOCATION	PHONE	FAX	E-MAIL
USA	1.800.343.4912	971.327.2777	info_service@coherent.com
Europe	+49 (6071) 9680	971.327.2777	info_service@coherent.com
International	971.327.2700	971.327.2777	info_service@coherent.com

Product Shipping Instructions

To prepare the product for shipping to Coherent:

1. Contact Coherent Customer Service (refer to Table 3 on page 74) for a Return Material Authorization number.
2. Attach a tag to the product that includes the name and address of the owner, the person to contact, the serial number, and the RMA number you received from Coherent Customer Service.
3. Wrap the product with polyethylene sheeting or equivalent material.
4. If the original packing material and carton are not available, obtain a corrugated cardboard shipping carton with inside dimensions that are at least 6 in (15 cm) taller, wider, and deeper than the product. The shipping carton must be constructed of cardboard with a minimum of 375 lb (170 kg) test strength. Cushion the instrument in the shipping carton with packing material or urethane foam on all sides between the carton and the product. Allow 3 in (7.5 cm) on all sides, top, and bottom.
5. Seat the shipping carton with shipping tape or an industrial stapler.
6. Ship the product to:

   Coherent, Inc.
   7470 SW Bridgeport Rd.
   Portland, OR 97224
   Attn: RMA # (*add the RMA number you received from Coherent Customer Service*)

APPENDIX A: SPECIFICATIONS

Table 4 lists specifications for the FieldMaxII-P.

Table 4. Specifications

PARAMETER	DESCRIPTION
ELECTRICAL/MECHANICAL	
Analog Output	1, 2, or 5V full-scale (user-adjustable) 100 ohm source impedance
Battery Operating Time[a] (approx) Continuous measurements in Pyroelectric mode Continuous measurements in Thermopile or Optical mode with passive sensor Shelf life when not used	(Six 1.5V AA alkaline batteries) 24 hr. 36 hr. 8 yr.
Calibration Accuracy	± 1%
Digital Output	USB 1.1
Digital Tuning Needle	100 mS (tau) 20 Hz (update rate)
Instrument Power	Six 1.5V AA alkaline batteries, or 90-to-260 VAC, 50/60 Hz AC
Pyroelectric Input (maximum voltage input)	18V
Internal Trigger	2 to 20% full scale (selectable)

Table 4. Specifications (Continued)

PARAMETER	DESCRIPTION
Linearity	± 1%
Measurement Resolution	± 0.1% of full scale
ENVIRONMENTAL	
Altitude	<4,500 m (operating) <12,000 m (storage)
Relative Humidity	<90% (5 to 40° C) (operating) <95% (0 to 70° C) (storage)
Temperature	5 to 40° C (operating) -20 to 70° C (storage)
RANGES	
Area Parameter (entered as a diameter)	0.01 to 999.99 mm
Attenuation (Attenuation parameter)	0.01 to 999.99
Batch Size (Statistics parameter)	2 to 99,999 pulses
Measurement Range (full scale, sensor-dependent)	3 nJ to 300 kJ
Rep Rate	± 1 Hz (accuracy) 300 Hz (maximum) 1 Hz (resolution)
Window Size (Avg parameter)	2 to 99,999 pulses

Table 4. Specifications (Continued)

PARAMETER	DESCRIPTION
PHYSICAL CHARACTERISTICS	
Dimensions (h x w x d) (approx)	8 in. (20 cm) 4 in. (10 cm) 1.5 in. (4 cm)
Display	58 x 73 mm fixed-segment LCD with backlight
Weight (approx, including batteries)	1.1 lb. (0.5 kg)
MISCELLANEOUS	
Regulations Met	CE

a. Coherent recommends a Panasonic AM-3PI AA battery.

INDEX

FieldMaxII-P User Manual™

Part No. 1086242, Rev. AB