



# LMMeasurement COM Object

## User Manual

# OphirLMMeasurement COM Object

1	Revision History .....	2
2	Overview.....	2
2.1	Introduction.....	2
2.2	USB Details of Devices .....	3
3	OphirLMMeasurement: Detailed Description .....	4
3.1	Registration .....	4
3.1.1	Additional Registrations.....	4
3.1.2	Troubleshooting Regsvr32 .....	5
3.1.3	Versions of OphirLMMeasurement .....	5
3.2	Creating an Object .....	6
3.3	Threading issues .....	6
3.4	Error Codes .....	6
3.5	Methods and Events.....	7
3.5.1	Device Communications.....	7
3.5.2	General Information and Diagnostics .....	7
3.5.3	Sensor Configuration.....	7
3.5.4	Measurement Delivery .....	7
3.5.5	Trigger Settings .....	7
3.5.6	Events.....	7
3.5.7	ASCII Commands.....	7
3.5.8	Alphabetic Listing .....	8
3.6	Detailed Explanations of Methods.....	9
3.6.1	Conventions.....	9
3.6.2	Device Communications.....	9
3.6.3	General Information and Diagnostics .....	12
3.6.4	Sensor Configuration.....	14
3.6.5	Measurement Delivery .....	26
3.6.6	Trigger Settings .....	28
3.6.7	Events.....	30
3.6.8	ASCII Commands.....	31
4	Data Streams.....	32
4.1	Measurement Streaming Model .....	32
4.2	Setup and Start .....	32
4.3	Data Collection .....	32
4.3.1	Retrieving Data.....	32
4.3.2	Interpreting Status Codes.....	33
4.4	Stopping and Closing .....	33
4.5	GetData Status Codes .....	34

# 1 Revision History

15	20-Apr-26	<ol style="list-style-type: none"><li>1. Support for X940</li><li>2. Rewrote Data Streams section</li><li>3. Fixed bad layout</li></ol>
13	31-Jul-22	<ol style="list-style-type: none"><li>4. Mention Visual C++ redistributable</li></ol>
12	05-Jan-21	<ol style="list-style-type: none"><li>1. Support for X938 (partial)</li><li>2. Support for VS 2019</li><li>3. Added GetData Status Codes: Temperature, Alert Hot, Pulse Width, and PfP Energy</li></ol>
11	25-Mar-20	<ol style="list-style-type: none"><li>1. Removed XP support</li><li>2. Added status to <a href="#">GetData</a> method for reporting pulse frequency</li><li>3. Added information about the new interface ICoLMMeasurement2</li><li>4. Added functions for calibration date</li><li>5. Added functions for pulsed power pulse width</li><li>6. Added functions for low frequency power pulse frequency</li></ol>
10	11-Apr-19	<ol style="list-style-type: none"><li>1. Added 844-PE-USB</li><li>2. Listed Python and Java as environments we supply a demo for</li><li>3. Added <a href="#">Creating an Object</a> and <a href="#">Threading issues</a> sections</li></ol>
09	22-Mar-17	<ol style="list-style-type: none"><li>1. Added C++ demo information</li></ol>
08	30-May-16	<ol style="list-style-type: none"><li>1. Added 841-PE-USB and 1919-R</li><li>2. Added status report to <a href="#">GetData</a> method for reporting Filter status for sensors that self-detect their Filter State</li><li>3. Remove component for 32-bit applications on 64-bit Windows</li></ol>
01	30-Jul-14	<ol style="list-style-type: none"><li>1. Initial Revision</li></ol>

## 2 Overview

### 2.1 Introduction

This document describes OphirLMMeasurement. This is the COM object provided by Newport that allows user applications to interface with the Newport 843-R-USB, 1919-R, 841-PE-USB, 844-PE-USB, X938, and X940.

Besides their use as standalone, fully featured laser power/energy meters, Newport devices can also be used through OphirLMMeasurement. This COM object allows system integrators to integrate the measurement expertise of Newport equipment with in-house analysis packages.

Demo client applications in VB.NET 2010/2019, VC# 2010/2019, VC++ 2010/2019, Python, Java, MatLab, and LabVIEW 8.6 are provided with the PMManager installation and can be found in C:\Program Files\Newport\PMManager 4.xx\Automation Examples\Com object.

In addition, OphirLMMeasurement has been tested with Delphi 5. In practice, OphirLMMeasurement can be used in any environment that allows interaction with COM automation servers (although it hasn't been tested with tools other than those mentioned).

Special note for C++ users: Due the intricacy involved in using COM objects in C++, the C++ demo includes a wrapper class which provides a standard C++ interface to the COM object. To use this wrapper class, add OphirLMMeasurement.h and OphirLMMeasurement.cpp to your project, and #include "OphirLMMeasurement.h". The file program.cpp is a demo of how to use the class. See the comments in OphirLMMeasurement.h for more details.

Note: We provide a COM object and not a .NET automation because many clients' applications do not use .NET and it is easy to use a COM object from .NET.

## **2.2 USB Details of Devices**

The following lists the differences between the various Newport devices vis-à-vis their USB communications. OphirLMMeasurement hides all this from the client application but the system integrator must be aware of this to ensure he sets up the system with the appropriate hardware support.

- 841-PE-USB One channel, Full Speed, Bus Powered, up to 5 interrupt IN endpoints
- 844-PE-USB One channel, Full Speed, Bus Powered, up to 5 interrupt IN endpoints
- 843-R-USB One channel, High Speed, Self powered, one interrupt IN endpoint
- 1919-R One channel, High Speed, Self powered, one interrupt IN endpoint
- X938 Up to 2 channels, High Speed, Self-powered, one interrupt IN endpoint
- X940 Up to 2 channels, High Speed, Self-powered, one interrupt IN endpoint

## 3 OphirLMMeasurement: Detailed Description

This section describes registration of OphirLMMeasurement as well as its methods, events, and error codes.

### 3.1 Registration

PManager PC software (3.10 and higher) registers OphirLMMeasurement as part of the installation process of the PManager application. Registration is also possible via the Menu of the PManager application.

#### 3.1.1 Additional Registrations

After installing the PManager PC application on one PC, it is possible to register the COM objects on additional PC's without installing the full PManager package. To do so, use the following steps

1. Install the Visual Studio 2019 Visual C++ redistributable, available at <https://docs.microsoft.com/en-us/cpp/windows/latest-supported-vc-redist>.
2. Copy the necessary files
  - For 32-bit applications (running on 32-bit or 64-bit Windows): From **C:\Program Files\Newport\PManager 3.xx\Automation Examples\Com object\Redistributable\x86** to the target PC
  - For 64-bit applications: From **C:\Program Files\Newport\PManager 3.xx\Automation Examples\Com object\Redistributable\x64** to the target PC.
3. Run **installophircom.bat** file on the target PC. This will install the USB drivers as well as register the OphirLMMeasurement COM object and all other components necessary for full functionality. The following is the contents of **installophircom.bat**
  - rem 1) install COM (this is Ophir specific)
  - regsvr32 OphirLMMeasurement.dll
  - rem 2) install Ophir device drivers (Ophir specific INF files)
  - swapinf StarlabDrivers
4. If there is a need to remove the software from the target PC and to clean its registry, run **RemoveOphirCOM.bat** that is provided. The following is the contents of **RemoveOphirCOM.bat**
  - rem 1) remove Ophir device drivers (Ophir specific INF files)
  - swapinf StarlabDriversRemove
  - rem 2) remove COM (this is Ophir specific)
  - regsvr32 /u OphirLMMeasurement.dll
5. The default COM object installation includes Ophir dialog windows informing the user to disconnect Ophir devices. If it is necessary to hide this message, then the install and uninstall batch files can be run silently. To do this, modify **installophircom.bat** and **RemoveOphirCOM.bat** as follows
  - **installophircom.bat**
    1. regsvr32 /s OphirLMMeasurement.dll
    2. swapinf StarlabDrivers --silent
  - **RemoveOphirCOM.bat**
    1. swapinf StarlabDriversRemove --silent
    2. regsvr32 /u /s OphirLMMeasurement.dll
  - Note: If you chose to install and remove silently, you must verify that the Ophir device has been disconnected before starting the process. Otherwise the installation will fail may leave unwanted information in the OC and you won't know about it

### 3.1.2 Troubleshooting Regsvr32

Microsoft reports that there are occasions where Regsvr32 doesn't execute properly even though it is being run as an Administrator (see <http://support.microsoft.com/kb/249873>). The way to overcome this is to re-run the Regsvr32 command from an elevated command prompt.

To open an elevated command prompt, follow these steps:

- **Windows 8.1 and Windows 8**

Swipe in from the right edge of the screen, and then tap **Search**. Or, if you are using a mouse, point to the lower-right corner of the screen, and then click **Search**. Type **Command Prompt** in the **Search** box, right-click **Command Prompt**, and then click **Run as administrator**. If you are prompted for an administrator password or for a confirmation, type the password, or click **Allow**.

- **Windows 7 and Windows Vista**

Click **Start**, type **Command Prompt** or **cmd** in the **Search** box, right-click **Command Prompt**, and then click **Run as administrator**. If you are prompted for an administrator password or for a confirmation, type the password, or click **Allow**.

- **Windows 10 and Windows 11**

Click the **Start** menu, type **Command Prompt** or **cmd** in the **Search** box, right-click **Command Prompt**, and then click **Run as administrator**. If you are prompted for an administrator password or for a confirmation, type the password, or click **Allow**.

### 3.1.3 Versions of OphirLMMeasurement

The OphirLMMeasurement component works in Windows Vista, 7, 8, 8.1, 10, and 11. We provide 2 variations.

- 32-bit component for 32-bit applications.
- 64-bit component for 64-bit applications.

## 3.2 Creating an Object

In order to use the library an object must be created. This object has methods which allow controlling the Newport measuring devices. To create the object, the ProgID `OphirLMMeasurement.CoLMMeasurement` or the GUID `{B180613C-E514-4739-AADC-CAD4493910D7}` can be used. How precisely to create a COM object depends on your development environment; see the provided samples for some possibilities.

Some environments require an interface name to be specified. For version 10.0 of the COM object (which comes with PMManager 3.50) the interface is `ICoLMMeasurement2`. The older interface `ICoLMMeasurement` will still work, but will not allow accessing the functions added in this version. Environments that do not require specifying the interface name will automatically use the newer interface.

## 3.3 Threading issues

The COM object provided is Apartment-threaded. This means (approximately) that it can be only used directly from the thread it was created in. The simplest way to use this library from multiple threads is to create a separate object in each thread. Since the state of the instruments is shared between objects, this has the same effect.

## 3.4 Error Codes

- 0x00000000 (S\_OK): No Error
- 0x80004001 (E\_NOTIMPL): Function Not Implemented
- 0x80070057 (E\_INVALIDARG): Invalid Argument
- 0x80004005 (E\_FAIL): Unspecified Failure
- 0x80040200: Device not opened
- 0x80040201: Device Already Opened
- 0x80040202: Drivers cannot be loaded
- 0x80040203: Load File Missing
- 0x80040300: Device Failed
- 0x80040301: Device Firmware is Incorrect
- 0x80040302: Sensor Failed
- 0x80040303: Sensor Firmware is Incorrect
- 0x80040304: Bad Device Handle
- 0x80040305: Bad Sensor Channel
- 0x80040306: This Sensor is Not Supported
- 0x80040307: Not Applicable in this Device
- 0x80040308: The Device is no longer Available
- 0x80040400: Save To Sensor Failed
- 0x80040401: Param Error
- 0x80040402: Failed to create Safe Array
- 0x80040403: Not Applicable in this Sensor
- 0x80040404: Value Out of Range
- 0x80040405: Command Failed
- 0x80040500: Stream Mode Not Started
- 0x80040501: A channel is in Stream Mode

The OphirLMMeasurement COM object follows the standard COM practice of returning an HRESULT from its methods. The HRESULT can be passed to [GetErrorFromCode](#) to get a descriptive string, or the standard COM `GetErrorInfo` function can be called (see documentation for your environment). Some client environments (such as VB6 and all .NET languages) do not return this HRESULT from the method call; instead they throw an exception when a method returns a failure HRESULT. This exception will normally contain within it the error number and the descriptive string.

## 3.5 Methods and Events

### 3.5.1 Device Communications

<a href="#">Close</a>	<a href="#">CloseAll</a>	<a href="#">GetKnownWirelessDevices</a>
<a href="#">OpenUSBDevice</a>	<a href="#">OpenWirelessDevice</a>	<a href="#">ResetAllDevices</a>
<a href="#">ResetDevice</a>	<a href="#">ScanUSB</a>	<a href="#">ScanWireless</a>

### 3.5.2 General Information and Diagnostics

<a href="#">GetDeviceCalibrationDueDate</a>	<a href="#">GetDeviceInfo</a>	<a href="#">GetDriverVersion</a>
<a href="#">GetErrorFromCode</a>	<a href="#">GetSensorCalibrationDueDate</a>	<a href="#">GetSensorInfo</a>
<a href="#">GetVersion</a>	<a href="#">IsSensorExists</a>	

### 3.5.3 Sensor Configuration

<a href="#">AddWavelength</a>	<a href="#">DeleteWavelength</a>	<a href="#">GetDiffuser</a>
<a href="#">GetFilter</a>	<a href="#">GetLowFreqPowerPulseFreq</a>	<a href="#">GetMeasurementMode</a>
<a href="#">GetPulsedPowerPulseWidth</a>	<a href="#">GetPulseLengths</a>	<a href="#">GetRanges</a>
<a href="#">GetThreshold</a>	<a href="#">GetWavelengths</a>	<a href="#">GetWavelengthsExtra</a>
<a href="#">ModifyWavelength</a>	<a href="#">SaveSettings</a>	<a href="#">SetDiffuser</a>
<a href="#">SetFilter</a>	<a href="#">SetLowFreqPowerPulseFreq</a>	<a href="#">SetMeasurementMode</a>
<a href="#">SetPulsedPowerPulseWidth</a>	<a href="#">SetPulseLength</a>	<a href="#">SetRange</a>
<a href="#">SetThreshold</a>	<a href="#">SetWavelength</a>	

### 3.5.4 Measurement Delivery

<a href="#">ConfigureStreamMode</a>	<a href="#">GetData</a>	<a href="#">StartStream</a>
<a href="#">StopAllStreams</a>	<a href="#">StopStream</a>	

### 3.5.5 Trigger Settings

<a href="#">GetExtTrigModes</a>	<a href="#">GetExtTrigOnOff</a>	<a href="#">GetExtTrigWindowTime</a>
<a href="#">SetExtTrigMode</a>	<a href="#">SetExtTrigOnOff</a>	<a href="#">SetExtTrigWindowTime</a>

### 3.5.6 Events

<a href="#">DataReady</a>	<a href="#">PlugAndPlay</a>	
---------------------------	-----------------------------	--

### 3.5.7 ASCII Commands

<a href="#">Read</a>	<a href="#">Write</a>	
----------------------	-----------------------	--

### 3.5.8 Alphabetic Listing

Method	<a href="#">Communications</a>	<a href="#">Diagnostics</a>	<a href="#">Configuration</a>	<a href="#">Measurement</a>	<a href="#">Trigger</a>	<a href="#">Legacy</a>
<a href="#">AddWavelength</a>			•			
<a href="#">Close</a>	•					
<a href="#">CloseAll</a>	•					
<a href="#">ConfigureStreamMode</a>				•		
<a href="#">DataReady</a> (event)						
<a href="#">DeleteWavelength</a>			•			
<a href="#">GetData</a>				•		
<a href="#">GetDeviceCalibrationDueDate</a>		•				
<a href="#">GetDeviceInfo</a>		•				
<a href="#">GetDiffuser</a>			•			
<a href="#">GetDriverVersion</a>		•				
<a href="#">GetErrorFromCode</a>		•				
<a href="#">GetExtTrigModes</a>					•	
<a href="#">GetExtTrigOnOff</a>					•	
<a href="#">GetExtTrigWindowTime</a>					•	
<a href="#">GetFilter</a>			•			
<a href="#">GetKnownWirelessDevices</a>	•					
<a href="#">GetLowFreqPowerPulseFreq</a>			•			
<a href="#">GetMeasurementMode</a>			•			
<a href="#">GetPulsedPowerPulseWidth</a>			•			
<a href="#">GetPulseLengths</a>			•			
<a href="#">GetRanges</a>			•			
<a href="#">GetSensorCalibrationDueDate</a>		•				
<a href="#">GetSensorInfo</a>		•				
<a href="#">GetThreshold</a>			•			
<a href="#">GetVersion</a>		•				
<a href="#">GetWavelengths</a>			•			
<a href="#">GetWavelengthsExtra</a>			•			
<a href="#">IsSensorExists</a>		•				
<a href="#">ModifyWavelength</a>			•			
<a href="#">OpenUSBDevice</a>	•					
<a href="#">OpenWirelessDevice</a>	•					
<a href="#">PlugAndPlay</a> (event)						
<a href="#">Read</a>						•
<a href="#">ResetAllDevices</a>	•					
<a href="#">ResetDevice</a>	•					
<a href="#">SaveSettings</a>			•			
<a href="#">ScanUSB</a>	•					
<a href="#">ScanWireless</a>	•					
<a href="#">SetDiffuser</a>			•			
<a href="#">SetExtTrigMode</a>					•	
<a href="#">SetExtTrigOnOff</a>					•	
<a href="#">SetExtTrigWindowTime</a>					•	
<a href="#">SetFilter</a>			•			
<a href="#">SetLowFreqPowerPulseFreq</a>			•			
<a href="#">SetMeasurementMode</a>						
<a href="#">SetPulsedPowerPulseWidth</a>			•			
<a href="#">SetPulseLength</a>			•			
<a href="#">SetRange</a>			•			
<a href="#">SetThreshold</a>			•			
<a href="#">SetWavelength</a>			•			
<a href="#">StartStream</a>				•		
<a href="#">StopAllStreams</a>				•		
<a href="#">StopStream</a>				•		
<a href="#">Write</a>						•

## 3.6 Detailed Explanations of Methods

### 3.6.1 Conventions

#### 3.6.1.1 Sensor Identification

The set of devices that OphirLMMeasurement supports includes multichannel devices which may have more than one sensor attached. In order to provide a common interface across all devices, OphirLMMeasurement uniquely identifies a sensor based on the device that it is attached to (hDevice) and the channel to which it is attached. For devices with only one channel <channel> should be 0. For a 2-channel device it can be 0 or 1, and for a 4-channel device 0, 1, 2, or 3.

#### 3.6.1.2 Strings

In order to guarantee compatibility with the various Development Environments, OphirLMMeasurement returns strings in the BSTR type. Arrays of strings are returned as VARIANT \*.

#### 3.6.1.3 Common Parameters

Parameter	Description
[in] LONG hDevice	A handle to an open device, received from <a href="#">OpenUSBDevice</a> .
[in] LONG channel	A channel number, 0 to 1, as applicable to the device.
[out] VARIANT* options	A list of values that a setting can take, in the form of an array of BSTR. If the setting is not applicable to the chosen sensor, the array will be empty.
[out] LONG* index	The 0-based index of the currently selected value of a setting. If the setting is not applicable to the chosen sensor, -1.
[in] LONG index	The 0-based index of a value of a setting. This must be in the range of the options valid for the setting, as returned in the <options> parameter of a Get... method.

### 3.6.2 Device Communications

These methods are responsible for managing the communications with Newport devices. This includes finding and registering the device(s) within OphirLMMeasurement, starting a communications session, and closing down communications with the device(s).

#### 3.6.2.1 Close

Name	<b>Close</b>
Parameters	<ul style="list-style-type: none"><li>[in] LONG hDevice</li></ul>
Use	This method ends the communication session with <hDevice>.
Return Codes	<ul style="list-style-type: none"><li>0x00000000 (S_OK): No Error</li><li>0x80040304: Bad Device Handle</li><li>0x80040501: A channel is in Stream Mode</li></ul>
See Also:	<a href="#">Device Communications</a> , <a href="#">Alphabetic Listing</a> , <a href="#">Common Parameters</a>

### 3.6.2.2 CloseAll

Name	<b>CloseAll</b>
Parameters	None
Use	Equivalent to calling <a href="#">Close</a> on all open devices.
Return Codes	<ul style="list-style-type: none"><li>• 0x00000000 (S_OK): No Error</li><li>• 0x80040501: A channel is in Stream Mode</li></ul>
See Also:	<a href="#">Device Communications</a> , <a href="#">Alphabetic Listing</a>

### 3.6.2.3 GetKnownWirelessDevices

Name	<b>GetKnownWirelessDevices</b>
Parameters	<ul style="list-style-type: none"><li>• [out] VARIANT* serialNumbers</li></ul>
Use	This method is an infrastructure stub for future support and at present only returns an error code
Return Codes	<ul style="list-style-type: none"><li>• 0x80004001 (E_NOTIMPL): Function Not Implemented</li></ul>
See Also:	<a href="#">Device Communications</a> , <a href="#">Alphabetic Listing</a>

### 3.6.2.4 OpenUSBDevice

Name	<b>OpenUSBDevice</b>
Parameters	<ul style="list-style-type: none"><li>• [in] BSTR serialNumber</li><li>• [out] LONG* hDevice</li></ul>
Use	Given <serialNumber> that was found with the ScanUSB method, will open a communication session with a device. <hDevice> is a handle which will be used for all communication with the device.
Return Codes	<ul style="list-style-type: none"><li>• 0x00000000 (S_OK): No Error</li><li>• 0x80070057 (E_INVALIDARG): Invalid Argument</li><li>• 0x80040201: Device Already Opened</li><li>• 0x80040300: Device Failed</li><li>• 0x80040301: Device Firmware is Incorrect</li><li>• 0x80040302: Sensor Failed</li><li>• 0x80040303: Sensor Firmware is Incorrect</li><li>• 0x80040306: This Sensor is Not Supported</li><li>• 0x80040308: The Device is no longer Available</li><li>• 0x80040405: Command Failed</li></ul>
See Also:	<a href="#">Device Communications</a> , <a href="#">Alphabetic Listing</a> , <a href="#">ScanUSB</a>

### 3.6.2.5 OpenWirelessDevice

Name	<b>OpenWirelessDevice</b>
Parameters	<ul style="list-style-type: none"><li>• [in] BSTR name</li><li>• [out] LONG* hDevice</li></ul>
Use	This method is for future support.
Return Codes	<ul style="list-style-type: none"><li>•</li></ul>
See Also:	<a href="#">Device Communications</a> , <a href="#">Alphabetic Listing</a>

### 3.6.2.6 ResetAllDevices

Name	<b>ResetAllDevices</b>
Parameters	None
Use	<p>Closes and restarts all open devices. After calling this function <a href="#">ScanUsb</a> should be called to redetect the devices.</p> <p>Note: This function was only needed for older instruments. All current instruments detect automatically when the sensor is changed.</p> <p>Note: This method will only function when no devices are streaming data (i.e. any channels which were put into Stream mode with the <a href="#">StartStream</a> method have been taken out of stream mode with <a href="#">StopStream</a> or <a href="#">StopAllStreams</a>).</p>
Return Codes	<ul style="list-style-type: none"> <li>• 0x00000000 (S_OK): No Error</li> <li>• 0x80040501: A channel is in Stream Mode</li> </ul>
See Also:	<a href="#">Device Communications</a> , <a href="#">Alphabetic Listing</a>

### 3.6.2.7 ResetDevice

Name	<b>ResetDevice</b>
Parameters	<ul style="list-style-type: none"> <li>• [in] LONG hDevice</li> </ul>
Use	<p>Restarts &lt;hDevice&gt;. Used after changing sensor attached to the device</p> <p>Note: This function was only needed for older instruments. All current instruments detect automatically when the sensor is changed.</p> <p>Note: This command method will only function when the device isn't streaming data (i.e. any channels which were put into Stream mode with the <a href="#">StartStream</a> method have been taken out of stream mode with <a href="#">StopStream</a> or <a href="#">StopAllStreams</a>).</p>
Return Codes	<ul style="list-style-type: none"> <li>• 0x00000000 (S_OK): No Error</li> <li>• 0x80040200: Device not opened</li> <li>• 0x80040304: Bad Device Handle</li> <li>• 0x80040308: The Device is no longer Available</li> <li>• 0x80040405: Command Failed</li> <li>• 0x80040501: A channel is in Stream Mode</li> </ul>
See Also:	<a href="#">Device Communications</a> , <a href="#">Alphabetic Listing</a> , <a href="#">Common Parameters</a>

### 3.6.2.8 ScanUSB

Name	<b>ScanUSB</b>
Parameters	<ul style="list-style-type: none"> <li>• [out] VARIANT* serialNumbers</li> </ul>
Use	<p>Scans the USB for all Newport devices that are presently attached to the PC.</p> <p>&lt;serialNumbers&gt; will contain an array of BSTR.</p>
Return Codes	<ul style="list-style-type: none"> <li>• 0x00000000 (S_OK): No Error</li> <li>• 0x80040202: Drivers cannot be loaded</li> <li>• 0x80040203: Load File Missing</li> <li>• 0x80040402: Failed to create Safe Array</li> <li>• 0x80040501: A channel is in Stream Mode</li> </ul>
See Also:	<a href="#">Device Communications</a> , <a href="#">Alphabetic Listing</a> , <a href="#">OpenUSBDevice</a>

### 3.6.2.9 ScanWireless

Name	<b>ScanWireless</b>
Parameters	<ul style="list-style-type: none"> <li>[out] VARIANT* names</li> </ul>
Use	This method is for future support.
Return Codes	<ul style="list-style-type: none"> <li></li> </ul>
See Also:	<a href="#">Device Communications</a> , <a href="#">Alphabetic Listing</a>

## 3.6.3 General Information and Diagnostics

These methods provide general purpose information about the devices, sensors, and software that are presently attached and participating in the present communication session

### 3.6.3.1 GetDeviceInfo

Name	<b>GetDeviceInfo</b>
Parameters	<ul style="list-style-type: none"> <li>[in] LONG hDevice</li> <li>[out] BSTR* deviceName</li> <li>[out] BSTR* romVersion</li> <li>[out] BSTR* serialNumber</li> </ul>
Use	Gets name, ROM version and serial number of the device.
Return Codes	<ul style="list-style-type: none"> <li>0x00000000 (S_OK): No Error</li> <li>0x80040200: Device not opened</li> <li>0x80040304: Bad Device Handle</li> <li>0x80040308: The Device is no longer Available</li> </ul>
See Also:	<a href="#">General Information and Diagnostics</a> , <a href="#">Alphabetic Listing</a> , <a href="#">Common Parameters</a>

### 3.6.3.2 GetDeviceCalibrationDueDate

Name	<b>GetDeviceCalibrationDueDate</b>
Parameters	<ul style="list-style-type: none"> <li>[in] LONG hDevice</li> <li>[out] DATE* dueDate</li> </ul>
Use	Get the date this instrument is due to be calibrated.  Note: Some instruments do not yet report the calibration due date and will return error 0x80040307.
Return Codes	<ul style="list-style-type: none"> <li>0x00000000 (S_OK): No Error</li> <li>0x80040200: Device not opened</li> <li>0x80040304: Bad Device Handle</li> <li>0x80040307: Not Applicable in this Device</li> <li>0x80040308: The Device is no longer Available</li> </ul>
See Also:	<a href="#">General Information and Diagnostics</a> , <a href="#">Alphabetic Listing</a> , <a href="#">Common Parameters</a> , <a href="#">GetSensorCalibrationDueDate</a>

### 3.6.3.3 GetDriverVersion

Name	<b>GetDriverVersion</b>
Parameters	<ul style="list-style-type: none"> <li>[out] BSTR* info</li> </ul>
Use	Get version of drivers that OphirLMMeasurement uses.
Return Codes	<ul style="list-style-type: none"> <li>0x00000000 (S_OK): No Error</li> <li>0x80004005 (E_FAIL): Unspecified Failure</li> </ul>
See Also:	<a href="#">General Information and Diagnostics</a> , <a href="#">Alphabetic Listing</a>

### 3.6.3.4 GetErrorFromCode

Name	<b>GetErrorFromCode</b>
Parameters	<ul style="list-style-type: none"> <li>[in] LONG errorCode</li> <li>[out] BSTR* info</li> </ul>
Use	Given an error code, will pass back a string explanation of the error.
Return Codes	<ul style="list-style-type: none"> <li>0x00000000 (S_OK): No Error</li> </ul>
See Also:	<a href="#">General Information and Diagnostics</a> , <a href="#">Alphabetic Listing</a>

### 3.6.3.5 GetSensorCalibrationDueDate

Name	<b>GetSensorCalibrationDueDate</b>
Parameters	<ul style="list-style-type: none"> <li>[in] LONG hDevice</li> <li>[in] LONG channel</li> <li>[out] DATE* dueDate</li> </ul>
Use	Get the date this sensor is due to be calibrated.  Note: Some instruments do not yet report the calibration due date and will return error 0x80040307.
Return Codes	<ul style="list-style-type: none"> <li>0x00000000 (S_OK): No Error</li> <li>0x80040200: Device not opened</li> <li>0x80040304: Bad Device Handle</li> <li>0x80040305: Bad Sensor Channel</li> <li>0x80040307: Not Applicable in this Device</li> <li>0x80040308: The Device is no longer Available</li> </ul>
See Also:	<a href="#">General Information and Diagnostics</a> , <a href="#">Alphabetic Listing</a> , <a href="#">Common Parameters</a> , <a href="#">GetDeviceCalibrationDueDate</a>

### 3.6.3.6 GetSensorInfo

Name	<b>GetSensorInfo</b>
Parameters	<ul style="list-style-type: none"> <li>[in] LONG hDevice</li> <li>[in] LONG channel</li> <li>[out] BSTR* serialNumber</li> <li>[out] BSTR* sensorType</li> <li>[out] BSTR* sensorName</li> </ul>
Use	Gets serial number, type (thermopile, photodiode, pyroelectric), and name of sensor.
Return Codes	<ul style="list-style-type: none"> <li>0x00000000 (S_OK): No Error</li> <li>0x80040200: Device not opened</li> <li>0x80040304: Bad Device Handle</li> <li>0x80040305: Bad Sensor Channel</li> <li>0x80040308: The Device is no longer Available</li> </ul>
See Also:	<a href="#">General Information and Diagnostics</a> , <a href="#">Alphabetic Listing</a> , <a href="#">Common Parameters</a>

### 3.6.3.7 GetVersion

Name	<b>GetVersion</b>
Parameters	<ul style="list-style-type: none"> <li>[out] LONG* version</li> </ul>
Use	Get version of the OphirLMMMeasurement COM object.
Return Codes	<ul style="list-style-type: none"> <li>0x00000000 (S_OK): No Error</li> </ul>
See Also:	<a href="#">General Information and Diagnostics</a> , <a href="#">Alphabetic Listing</a>

### 3.6.3.8 IsSensorExists

Name	<b>IsSensorExists</b>
Parameters	<ul style="list-style-type: none"> <li>• [in] LONG hDevice</li> <li>• [in] LONG channel</li> <li>• [out] VARIANT_BOOL* exists</li> </ul>
Use	Check if a sensor is attached to <hDevice> <channel>.
Return Codes	<ul style="list-style-type: none"> <li>• 0x00000000 (S_OK): No Error</li> <li>• 0x80040200: Device not opened</li> <li>• 0x80040304: Bad Device Handle</li> <li>• 0x80040308: The Device is no longer Available</li> </ul>
See Also:	<a href="#">General Information and Diagnostics</a> , <a href="#">Alphabetic Listing</a> , <a href="#">Common Parameters</a>

## 3.6.4 Sensor Configuration

These methods are used to configure the measurement parameters of the sensor. Some of these methods apply to all types of sensors (e.g. Range methods) and some are sensor-type specific (e.g. Filter methods apply to Photodiode sensors only, Diffuser methods are only for Pyroelectric and PD Energy sensors, etc.)

### 3.6.4.1 AddWavelength

Name	<b>AddWavelength</b>
Parameters	<ul style="list-style-type: none"> <li>• [in] LONG hDevice</li> <li>• [in] LONG channel</li> <li>• [in] LONG wavelength</li> </ul>
Use	<p>Add &lt;wavelength&gt; to list of favorite wavelengths of the selected sensor.</p> <p>Wavelength: Must be between the lower and upper wavelength limits as returned by the <a href="#">GetWavelengthsExtra</a> method.</p> <p>Note: This method is only applicable for sensors with a continuous spectrum. The set of lasers cannot be altered in sensors with a discrete set of lasers settings.</p>
Return Codes	<ul style="list-style-type: none"> <li>• 0x00000000 (S_OK): No Error</li> <li>• 0x80040200: Device not opened</li> <li>• 0x80040304: Bad Device Handle</li> <li>• 0x80040305: Bad Sensor Channel</li> <li>• 0x80040308: The Device is no longer Available</li> <li>• 0x80040403: Not Applicable in this Sensor</li> <li>• 0x80040404: Wavelength Out of Range</li> <li>• 0x80040405: Command Failed</li> <li>• 0x80040501: A channel is in Stream Mode</li> </ul>
See Also:	<a href="#">Sensor Configuration</a> , <a href="#">Alphabetic Listing</a> , <a href="#">Common Parameters</a> , <a href="#">DeleteWavelength</a> , <a href="#">GetWavelengths</a> , <a href="#">GetWavelengthsExtra</a> , <a href="#">ModifyWavelength</a> , <a href="#">SetWavelength</a>

### 3.6.4.2 DeleteWavelength

Name	<b>DeleteWavelength</b>
Parameters	<ul style="list-style-type: none"> <li>• [in] LONG hDevice</li> <li>• [in] LONG channel</li> <li>• [in] LONG index</li> </ul>
Use	<p>Delete wavelength of selected sensor specified in &lt;index&gt;.</p> <p>Note: If the currently selected wavelength is deleted, a different wavelength is automatically chosen. If there is only one wavelength left it cannot be deleted.</p> <p>Note: This method is only applicable for sensors with a continuous spectrum. The set of lasers cannot be altered in sensors with a discrete set of lasers settings</p>
Return Codes	<ul style="list-style-type: none"> <li>• 0x00000000 (S_OK): No Error</li> <li>• 0x80040200: Device not opened</li> <li>• 0x80040304: Bad Device Handle</li> <li>• 0x80040305: Bad Sensor Channel</li> <li>• 0x80040308: The Device is no longer Available</li> <li>• 0x80040401: Param Error</li> <li>• 0x80040403: Not Applicable in this Sensor</li> <li>• 0x80040405: Command Failed</li> <li>• 0x80040501: A channel is in Stream Mode</li> </ul>
See Also:	<a href="#">Sensor Configuration</a> , <a href="#">Alphabetic Listing</a> , <a href="#">Common Parameters</a> , <a href="#">AddWavelength</a> , <a href="#">GetWavelengths</a> , <a href="#">GetWavelengthsExtra</a> , <a href="#">ModifyWavelength</a> , <a href="#">SetWavelength</a>

### 3.6.4.3 GetDiffuser

Name	<b>GetDiffuser</b>
Parameters	<ul style="list-style-type: none"> <li>• [in] LONG hDevice</li> <li>• [in] LONG channel</li> <li>• [out] LONG* index</li> <li>• [out] VARIANT* options</li> </ul>
Use	<p>Get the diffuser state of the selected sensor and what the diffuser options are.</p> <p>For sensors that are not equipped with an adjustable diffuser, &lt;index&gt; will always return 0. This method applies to Pyroelectric and PD Energy sensors only.</p>
Return Codes	<ul style="list-style-type: none"> <li>• 0x00000000 (S_OK): No Error</li> <li>• 0x80040200: Device not opened</li> <li>• 0x80040304: Bad Device Handle</li> <li>• 0x80040305: Bad Sensor Channel</li> <li>• 0x80040308: The Device is no longer Available</li> <li>• 0x80040402: Failed to create Safe Array</li> </ul>
See Also:	<a href="#">Sensor Configuration</a> , <a href="#">Alphabetic Listing</a> , <a href="#">Common Parameters</a> , <a href="#">SetDiffuser</a>

### 3.6.4.4 GetFilter

Name	<b>GetFilter</b>
Parameters	<ul style="list-style-type: none"> <li>• [in] LONG hDevice</li> <li>• [in] LONG channel</li> <li>• [out] LONG* index</li> <li>• [out] VARIANT* options</li> </ul>
Use	<p>Get the filter state of the selected sensor and what the filter options are.</p> <p>For sensors that are not equipped with an adjustable filter, &lt;index&gt; will always return 0. This method applies to Photodiode sensors only.</p>
Return Codes	<ul style="list-style-type: none"> <li>• 0x00000000 (S_OK): No Error</li> <li>• 0x80040200: Device not opened</li> <li>• 0x80040304: Bad Device Handle</li> <li>• 0x80040305: Bad Sensor Channel</li> <li>• 0x80040308: The Device is no longer Available</li> <li>• 0x80040402: Failed to create Safe Array</li> </ul>
See Also:	<a href="#">Sensor Configuration</a> , <a href="#">Alphabetic Listing</a> , <a href="#">Common Parameters</a> , <a href="#">SetFilter</a>

### 3.6.4.5 GetLowFreqPowerPulseFreq

Name	<b>GetLowFreqPowerPulseFreq</b>
Parameters	<ul style="list-style-type: none"> <li>• [in] LONG hDevice</li> <li>• [in] LONG channel</li> <li>• [out] DOUBLE* value</li> <li>• [out] DOUBLE* min</li> <li>• [out] DOUBLE* max</li> </ul>
Use	<p>Get the current pulse frequency for low frequency power measurement mode, and the maximum and minimum supported values.</p> <p>This method applies to Photodiode sensors only, and is only supported on some instruments.</p>
Return Codes	<ul style="list-style-type: none"> <li>• 0x00000000 (S_OK): No Error</li> <li>• 0x80040200: Device not opened</li> <li>• 0x80040304: Bad Device Handle</li> <li>• 0x80040305: Bad Sensor Channel</li> <li>• 0x80040308: The Device is no longer Available</li> <li>• 0x80040403: Not Applicable in this Sensor</li> </ul>
See Also:	<a href="#">Sensor Configuration</a> , <a href="#">Alphabetic Listing</a> , <a href="#">Common Parameters</a> , <a href="#">SetLowFreqPowerPulseFreq</a>

### 3.6.4.6 GetMeasurementMode

Name	<b>GetMeasurementMode</b>
Parameters	<ul style="list-style-type: none"> <li>• [in] LONG hDevice</li> <li>• [in] LONG channel</li> <li>• [out] LONG* index</li> <li>• [out] VARIANT* options</li> </ul>
Use	<p>Get the measurement mode of the selected sensor and what the measurement modes are.</p> <p>This method applies to all types of sensors.</p>
Return Codes	<ul style="list-style-type: none"> <li>• 0x00000000 (S_OK): No Error</li> <li>• 0x80040200: Device not opened</li> <li>• 0x80040304: Bad Device Handle</li> <li>• 0x80040305: Bad Sensor Channel</li> <li>• 0x80040308: The Device is no longer Available</li> <li>• 0x80040402: Failed to create Safe Array</li> </ul>
See Also:	<a href="#">Sensor Configuration</a> , <a href="#">Alphabetic Listing</a> , <a href="#">Common Parameters</a> , <a href="#">SetMeasurementMode</a>

### 3.6.4.7 GetPulsedPowerPulseWidth

Name	<b>GetPulsedPowerPulseWidth</b>
Parameters	<ul style="list-style-type: none"> <li>• [in] LONG hDevice</li> <li>• [in] LONG channel</li> <li>• [out] LONG* value</li> <li>• [out] LONG* min</li> <li>• [out] LONG* max</li> </ul>
Use	<p>Get the current pulse width for pulsed power measurement mode, and the maximum and minimum supported values, in milliseconds.</p> <p>This method applies to Thermopile sensors only, and is only supported on some instruments.</p>
Return Codes	<ul style="list-style-type: none"> <li>• 0x00000000 (S_OK): No Error</li> <li>• 0x80040200: Device not opened</li> <li>• 0x80040304: Bad Device Handle</li> <li>• 0x80040305: Bad Sensor Channel</li> <li>• 0x80040308: The Device is no longer Available</li> <li>• 0x80040403: Not Applicable in this Sensor</li> </ul>
See Also:	<a href="#">Sensor Configuration</a> , <a href="#">Alphabetic Listing</a> , <a href="#">Common Parameters</a> , <a href="#">SetPulsedPowerPulseWidth</a>

### 3.6.4.8 GetPulseLengths

Name	<b>GetPulseLengths</b>
Parameters	<ul style="list-style-type: none"><li>• [in] LONG hDevice</li><li>• [in] LONG channel</li><li>• [out] LONG* index</li><li>• [out] VARIANT* options</li></ul>
Use	<p>Get the selected pulse length of the sensor in selected channel and what the pulse length options are.</p> <p>This method applies to Pyroelectric and PD Energy sensors only.</p> <p>The strings returned are the maximum pulse length (in time) for each of the various pulse length settings.</p>
Return Codes	<ul style="list-style-type: none"><li>• 0x00000000 (S_OK): No Error</li><li>• 0x80040200: Device not opened</li><li>• 0x80040304: Bad Device Handle</li><li>• 0x80040305: Bad Sensor Channel</li><li>• 0x80040308: The Device is no longer Available</li><li>• 0x80040402: Failed to create Safe Array</li></ul>
See Also:	<a href="#">Sensor Configuration</a> , <a href="#">Alphabetic Listing</a> , <a href="#">Common Parameters</a> , <a href="#">SetPulseLength</a>

### 3.6.4.9 GetRanges

Name	<b>GetRanges</b>
Parameters	<ul style="list-style-type: none"><li>• [in] LONG hDevice</li><li>• [in] LONG channel</li><li>• [out] LONG* index</li><li>• [out] VARIANT* options</li></ul>
Use	<p>Get the measurement range of the selected sensor and the list of all available measurement ranges.</p> <p>This method applies to all types of sensors.</p>
Return Codes	<ul style="list-style-type: none"><li>• 0x00000000 (S_OK): No Error</li><li>• 0x80040200: Device not opened</li><li>• 0x80040304: Bad Device Handle</li><li>• 0x80040305: Bad Sensor Channel</li><li>• 0x80040308: The Device is no longer Available</li><li>• 0x80040402: Failed to create Safe Array</li></ul>
See Also:	<a href="#">Sensor Configuration</a> , <a href="#">Alphabetic Listing</a> , <a href="#">Common Parameters</a> , <a href="#">SetRange</a>

### 3.6.4.10 GetThreshold

Name	<b>GetThreshold</b>
Parameters	<ul style="list-style-type: none"> <li>• [in] LONG hDevice</li> <li>• [in] LONG channel</li> <li>• [out] LONG* index</li> <li>• [out] VARIANT* options</li> </ul>
Use	<p>Get the selected threshold setting of the sensor in selected channel and what the threshold options are.</p> <p>This method applies to Thermopile sensors measuring energy and to some Pyroelectric sensors as well.</p> <p>For Thermopile sensors, the set of available thresholds will generally be “LOW MEDIUM HIGH”.</p> <p>For Pyroelectric sensors with adjustable thresholds, there is a threshold setting per pulse length.</p>
Return Codes	<ul style="list-style-type: none"> <li>• 0x00000000 (S_OK): No Error</li> <li>• 0x80040200: Device not opened</li> <li>• 0x80040304: Bad Device Handle</li> <li>• 0x80040305: Bad Sensor Channel</li> <li>• 0x80040308: The Device is no longer Available</li> <li>• 0x80040402: Failed to create Safe Array</li> </ul>
See Also:	<a href="#">Sensor Configuration</a> , <a href="#">Alphabetic Listing</a> , <a href="#">Common Parameters</a> , <a href="#">SetThreshold</a>

### 3.6.4.11 GetWavelengths

Name	<b>GetWavelengths</b>
Parameters	<ul style="list-style-type: none"> <li>• [in] LONG hDevice</li> <li>• [in] LONG channel</li> <li>• [out] LONG* index</li> <li>• [out] VARIANT* options</li> </ul>
Use	<p>Gets the selected wavelength of selected sensor and list of all available wavelengths.</p> <p>This method applies to all types of sensors. Response is the same format for sensors with discrete and continuous spectrums. For additional spectrum information use <a href="#">GetWavelengthsExtra</a>.</p> <p>For sensors with a continuous spectrum, these strings will be numeric, for sensors with discrete spectrums, these strings may contain non-numeric characters as well (e.g. VIS or NIR).</p>
Return Codes	<ul style="list-style-type: none"> <li>• 0x00000000 (S_OK): No Error</li> <li>• 0x80040200: Device not opened</li> <li>• 0x80040304: Bad Device Handle</li> <li>• 0x80040305: Bad Sensor Channel</li> <li>• 0x80040308: The Device is no longer Available</li> <li>• 0x80040402: Failed to create Safe Array</li> </ul>
See Also:	<a href="#">Sensor Configuration</a> , <a href="#">Alphabetic Listing</a> , <a href="#">Common Parameters</a> , <a href="#">AddWavelength</a> , <a href="#">DeleteWavelength</a> , <a href="#">GetWavelengthsExtra</a> , <a href="#">ModifyWavelength</a> , <a href="#">SetWavelength</a>

### 3.6.4.12 GetWavelengthsExtra

Name	<b>GetWavelengthsExtra</b>
Parameters	<ul style="list-style-type: none"> <li>• [in] LONG hDevice</li> <li>• [in] LONG channel</li> <li>• [out] VARIANT_BOOL*modifiable</li> <li>• [out] LONG* minWavelength</li> <li>• [out] LONG* maxWavelength</li> </ul>
Use	<p>Get additional wavelength information of selected channel.</p> <p>If spectrum is continuous, then</p> <ul style="list-style-type: none"> <li>• modifiable: True</li> <li>• minWavelength, maxWavelength: minimum and maximum limits for wavelength adjustment with the <a href="#">AddWavelength</a> and <a href="#">ModifyWavelength</a> methods</li> </ul> <p>If spectrum is discrete then</p> <ul style="list-style-type: none"> <li>• modifiable: False</li> <li>• minWavelength, maxWavelength: Not applicable</li> </ul>
Return Codes	<ul style="list-style-type: none"> <li>• 0x00000000 (S_OK): No Error</li> <li>• 0x80040200: Device not opened</li> <li>• 0x80040304: Bad Device Handle</li> <li>• 0x80040305: Bad Sensor Channel</li> <li>• 0x80040308: The Device is no longer Available</li> <li>• 0x80040403: Not Applicable in this Sensor</li> <li>• 0x80040501: A channel is in Stream Mode</li> </ul>
See Also:	<a href="#">Sensor Configuration</a> , <a href="#">Alphabetic Listing</a> , <a href="#">Common Parameters</a> , <a href="#">AddWavelength</a> , <a href="#">DeleteWavelength</a> , <a href="#">GetWavelengths</a> , <a href="#">ModifyWavelength</a> , <a href="#">SetWavelength</a>

### 3.6.4.13 ModifyWavelength

Name	<b>ModifyWavelength</b>
Parameters	<ul style="list-style-type: none"> <li>• [in] LONG hDevice</li> <li>• [in] LONG channel</li> <li>• [in] LONG index</li> <li>• [in] LONG wavelength</li> </ul>
Use	<p>Modify selected sensor's favorite wavelength stored at &lt;index&gt; to the value &lt;wavelength&gt;.</p> <p>Note: &lt;wavelength&gt; must be between the lower and upper wavelength limits as returned by the <a href="#">GetWavelengthsExtra</a> method.</p> <p>Note: This method is only applicable for sensors with a continuous spectrum. The set of lasers cannot be altered in sensors with a discrete set of lasers settings.</p>
Return Codes	<ul style="list-style-type: none"> <li>• 0x00000000 (S_OK): No Error</li> <li>• 0x80040200: Device not opened</li> <li>• 0x80040304: Bad Device Handle</li> <li>• 0x80040305: Bad Sensor Channel</li> <li>• 0x80040308: The Device is no longer Available</li> <li>• 0x80040401: Param Error</li> <li>• 0x80040403: Not Applicable in this Sensor</li> <li>• 0x80040404: Wavelength Out of Range</li> <li>• 0x80040405: Command Failed</li> <li>• 0x80040501: A channel is in Stream Mode</li> </ul>
See Also:	<a href="#">Sensor Configuration</a> , <a href="#">Alphabetic Listing</a> , <a href="#">Common Parameters</a> , <a href="#">AddWavelength</a> , <a href="#">DeleteWavelength</a> , <a href="#">GetWavelengths</a> , <a href="#">GetWavelengthsExtra</a> , <a href="#">SetWavelength</a>

### 3.6.4.14 SaveSettings

Name	<b>SaveSettings</b>
Parameters	<ul style="list-style-type: none"> <li>• [in] LONG hDevice</li> <li>• [in] LONG channel</li> </ul>
Use	Save all configuration parameters of selected sensor.
Return Codes	<ul style="list-style-type: none"> <li>• 0x00000000 (S_OK): No Error</li> <li>• 0x80040200: Device not opened</li> <li>• 0x80040304: Bad Device Handle</li> <li>• 0x80040305: Bad Sensor Channel</li> <li>• 0x80040308: The Device is no longer Available</li> <li>• 0x80040400: Save To Sensor Failed</li> <li>• 0x80040501: A channel is in Stream Mode</li> </ul>
See Also:	<a href="#">Sensor Configuration</a> , <a href="#">Alphabetic Listing</a> , <a href="#">Common Parameters</a>

### 3.6.4.15 SetDiffuser

Name	<b>SetDiffuser</b>
Parameters	<ul style="list-style-type: none"> <li>• [in] LONG hDevice</li> <li>• [in] LONG channel</li> <li>• [in] LONG index</li> </ul>
Use	<p>Set diffuser state of selected sensor to &lt;index&gt;</p> <p>This method applies to Pyroelectric and PD Energy sensors only.</p>
Return Codes	<ul style="list-style-type: none"> <li>• 0x00000000 (S_OK): No Error</li> <li>• 0x80040200: Device not opened</li> <li>• 0x80040304: Bad Device Handle</li> <li>• 0x80040305: Bad Sensor Channel</li> <li>• 0x80040308: The Device is no longer Available</li> <li>• 0x80040401: Param Error</li> <li>• 0x80040403: Not Applicable in this Sensor</li> <li>• 0x80040405: Command Failed</li> <li>• 0x80040501: A channel is in Stream Mode</li> </ul>
See Also:	<a href="#">Sensor Configuration</a> , <a href="#">Alphabetic Listing</a> , <a href="#">Common Parameters</a> , <a href="#">GetDiffuser</a>

### 3.6.4.16 SetFilter

Name	<b>SetFilter</b>
Parameters	<ul style="list-style-type: none"> <li>• [in] LONG hDevice</li> <li>• [in] LONG channel</li> <li>• [in] LONG index</li> </ul>
Use	<p>Set filter state of selected sensor to &lt;index&gt;.</p> <p>This method applies to Photodiode sensors only.</p>
Return Codes	<ul style="list-style-type: none"> <li>• 0x00000000 (S_OK): No Error</li> <li>• 0x80040200: Device not opened</li> <li>• 0x80040304: Bad Device Handle</li> <li>• 0x80040305: Bad Sensor Channel</li> <li>• 0x80040308: The Device is no longer Available</li> <li>• 0x80040401: Param Error</li> <li>• 0x80040403: Not Applicable in this Sensor</li> <li>• 0x80040405: Command Failed</li> <li>• 0x80040501: A channel is in Stream Mode</li> </ul>
See Also:	<a href="#">Sensor Configuration</a> , <a href="#">Alphabetic Listing</a> , <a href="#">Common Parameters</a> , <a href="#">GetFilter</a>

### 3.6.4.17 SetLowFreqPowerPulseFreq

Name	<b>SetLowFreqPowerPulseFreq</b>
Parameters	<ul style="list-style-type: none"> <li>• [in] LONG hDevice</li> <li>• [in] LONG channel</li> <li>• [in] DOUBLE value</li> </ul>
Use	<p>Set the pulse frequency for low frequency power measurement mode.</p> <p>This method applies to Photodiode sensors only, and is only supported on some instruments.</p>
Return Codes	<ul style="list-style-type: none"> <li>• 0x00000000 (S_OK): No Error</li> <li>• 0x80040200: Device not opened</li> <li>• 0x80040304: Bad Device Handle</li> <li>• 0x80040305: Bad Sensor Channel</li> <li>• 0x80040308: The Device is no longer Available</li> <li>• 0x80040403: Not Applicable in this Sensor</li> <li>• 0x80040404: Value Out of Range</li> <li>• 0x80040405: Command Failed</li> <li>• 0x80040501: A channel is in Stream Mode</li> </ul>
See Also:	<a href="#">Sensor Configuration</a> , <a href="#">Alphabetic Listing</a> , <a href="#">Common Parameters</a> , <a href="#">GetLowFreqPowerPulseFreq</a>

### 3.6.4.18 SetMeasurementMode

Name	<b>SetMeasurementMode</b>
Parameters	<ul style="list-style-type: none"> <li>• [in] LONG hDevice</li> <li>• [in] LONG channel</li> <li>• [in] LONG index</li> </ul>
Use	<p>Set measurement mode of selected sensor to &lt;index&gt;.</p> <p>This method applies to all types of sensors.</p>
Return Codes	<ul style="list-style-type: none"> <li>• 0x00000000 (S_OK): No Error</li> <li>• 0x80040200: Device not opened</li> <li>• 0x80040304: Bad Device Handle</li> <li>• 0x80040305: Bad Sensor Channel</li> <li>• 0x80040308: The Device is no longer Available</li> <li>• 0x80040401: Param Error</li> <li>• 0x80040403: Not Applicable in this Sensor</li> <li>• 0x80040405: Command Failed</li> <li>• 0x80040501: A channel is in Stream Mode</li> </ul>
See Also	<a href="#">Sensor Configuration</a> , <a href="#">Alphabetic Listing</a> , <a href="#">Common Parameters</a> , <a href="#">GetMeasurementMode</a>

### 3.6.4.19 SetPulsedPowerPulseWidth

Name	<b>SetPulseLength</b>
Parameters	<ul style="list-style-type: none"> <li>• [in] LONG hDevice</li> <li>• [in] LONG channel</li> <li>• [in] LONG value</li> </ul>
Use	<p>Get the pulse width for pulsed power measurement mode, in milliseconds.</p> <p>This method applies to Thermopile sensors only, and is only supported on some instruments.</p>
Return Codes	<ul style="list-style-type: none"> <li>• 0x00000000 (S_OK): No Error</li> <li>• 0x80040200: Device not opened</li> <li>• 0x80040304: Bad Device Handle</li> <li>• 0x80040305: Bad Sensor Channel</li> <li>• 0x80040308: The Device is no longer Available</li> <li>• 0x80040403: Not Applicable in this Sensor</li> <li>• 0x80040404: Value Out of Range</li> <li>• 0x80040405: Command Failed</li> <li>• 0x80040501: A channel is in Stream Mode</li> </ul>
See Also:	<a href="#">Sensor Configuration</a> , <a href="#">Alphabetic Listing</a> , <a href="#">Common Parameters</a> , <a href="#">GetPulsedPowerPulseWidth</a>

### 3.6.4.20 SetPulseLength

Name	<b>SetPulseLength</b>
Parameters	<ul style="list-style-type: none"> <li>• [in] LONG hDevice</li> <li>• [in] LONG channel</li> <li>• [in] LONG index</li> </ul>
Use	<p>Set pulse length setting of the selected sensor to &lt;index&gt;.</p> <p>This method applies to Pyroelectric and PD Energy sensors only.</p>
Return Codes	<ul style="list-style-type: none"> <li>• 0x00000000 (S_OK): No Error</li> <li>• 0x80040200: Device not opened</li> <li>• 0x80040304: Bad Device Handle</li> <li>• 0x80040305: Bad Sensor Channel</li> <li>• 0x80040308: The Device is no longer Available</li> <li>• 0x80040401: Param Error</li> <li>• 0x80040403: Not Applicable in this Sensor</li> <li>• 0x80040405: Command Failed</li> <li>• 0x80040501: A channel is in Stream Mode</li> </ul>
See Also:	<a href="#">Sensor Configuration</a> , <a href="#">Alphabetic Listing</a> , <a href="#">Common Parameters</a> , <a href="#">GetPulseLengths</a>

### 3.6.4.21 SetRange

Name	<b>SetRange</b>
Parameters	<ul style="list-style-type: none"> <li>• [in] LONG hDevice</li> <li>• [in] LONG channel</li> <li>• [in] LONG index</li> </ul>
Use	<p>Set measurement range of selected sensor to &lt;index&gt;.</p> <p>This method applies to all types of sensors.</p>
Return Codes	<ul style="list-style-type: none"> <li>• 0x00000000 (S_OK): No Error</li> <li>• 0x80040200: Device not opened</li> <li>• 0x80040304: Bad Device Handle</li> <li>• 0x80040305: Bad Sensor Channel</li> <li>• 0x80040308: The Device is no longer Available</li> <li>• 0x80040401: Param Error</li> <li>• 0x80040403: Not Applicable in this Sensor</li> <li>• 0x80040405: Command Failed</li> <li>• 0x80040501: A channel is in Stream Mode</li> </ul>
See Also:	<a href="#">Sensor Configuration</a> , <a href="#">Alphabetic Listing</a> , <a href="#">Common Parameters</a> , <a href="#">GetRanges</a>

### 3.6.4.22 SetThreshold

Name	<b>SetThreshold</b>
Parameters	<ul style="list-style-type: none"> <li>• [in] LONG hDevice</li> <li>• [in] LONG channel</li> <li>• [in] LONG index</li> </ul>
Use	<p>Set threshold of selected sensor to &lt;index&gt;.</p> <p>This method applies to Thermopile sensors measuring energy and to some Pyroelectric sensors as well.</p>
Return Codes	<ul style="list-style-type: none"> <li>• 0x00000000 (S_OK): No Error</li> <li>• 0x80040200: Device not opened</li> <li>• 0x80040304: Bad Device Handle</li> <li>• 0x80040305: Bad Sensor Channel</li> <li>• 0x80040308: The Device is no longer Available</li> <li>• 0x80040401: Param Error</li> <li>• 0x80040403: Not Applicable in this Sensor</li> <li>• 0x80040405: Command Failed</li> <li>• 0x80040501: A channel is in Stream Mode</li> </ul>
See Also:	<a href="#">Sensor Configuration</a> , <a href="#">Alphabetic Listing</a> , <a href="#">Common Parameters</a> , <a href="#">GetThreshold</a>

### 3.6.4.23 SetWavelength

Name	<b>SetWavelength</b>
Parameters	<ul style="list-style-type: none"> <li>• [in] LONG hDevice</li> <li>• [in] LONG channel</li> <li>• [in] LONG index</li> </ul>
Use	<p>Set wavelength of selected sensor to &lt;index&gt;.</p> <p>This method applies to all types of sensors.</p>
Return Codes	<ul style="list-style-type: none"> <li>• 0x00000000 (S_OK): No Error</li> <li>• 0x80040200: Device not opened</li> <li>• 0x80040304: Bad Device Handle</li> <li>• 0x80040305: Bad Sensor Channel</li> <li>• 0x80040308: The Device is no longer Available</li> <li>• 0x80040401: Param Error</li> <li>• 0x80040403: Not Applicable in this Sensor</li> <li>• 0x80040405: Command Failed</li> <li>• 0x80040501: A channel is in Stream Mode</li> </ul>
See Also:	<a href="#">Sensor Configuration</a> , <a href="#">Alphabetic Listing</a> , <a href="#">Common Parameters</a> , <a href="#">AddWavelength</a> , <a href="#">DeleteWavelength</a> , <a href="#">GetWavelengths</a> , <a href="#">GetWavelengthsExtra</a> , <a href="#">ModifyWavelength</a>

## 3.6.5 Measurement Delivery

See section [Data Streams](#) for detailed explanations of the various streams of data that come out of the devices and through OphirLMMeasurement, setting up the stream, starting and stopping it, and parsing the response.

### 3.6.5.1 ConfigureStreamMode

Name	<b>ConfigureStreamMode</b>
Parameters	<ul style="list-style-type: none"> <li>• [in] LONG hDevice</li> <li>• [in] LONG channel</li> <li>• [in] LONG mode</li> <li>• [in] LONG nValue</li> </ul>
Use	<p>Configure special measurement delivery options for the selected device and channel. There are currently no special options defined for any Newport instruments.</p>
Return Codes	<ul style="list-style-type: none"> <li>• 0x00000000 (S_OK): No Error</li> <li>• 0x80040200: Device not opened</li> <li>• 0x80040304: Bad Device Handle</li> <li>• 0x80040305: Bad Sensor Channel</li> <li>• 0x80040308: The Device is no longer Available</li> <li>• 0x80040403: Not Applicable in this Sensor</li> <li>• 0x80040405: Command Failed</li> <li>• 0x80040501: A channel is in Stream Mode</li> </ul>
See Also:	<a href="#">Measurement Delivery</a> , <a href="#">Alphabetic Listing</a> , <a href="#">Common Parameters</a>

### 3.6.5.2 GetData

Name	<b>GetData</b>
Parameters	<ul style="list-style-type: none"> <li>• [in] LONG hDevice</li> <li>• [in] LONG channel</li> <li>• [out] VARIANT* arrayValue // array of double</li> <li>• [out] VARIANT* arrayTimestamp // array of double</li> <li>• [out] VARIANT* arrayStatus // array of LONG</li> </ul>
Use	<p>This method retrieves the buffered data that has been received from the meter. It should be called after the <a href="#">DataReady</a> event has been received, or at appropriate intervals. The returned data is removed from the buffer. If called when no new data is available, the arrays will be empty.</p> <p>The three arrays are the same size, with corresponding elements containing the value, timestamp and status for a single item of data. Note that not every item of data represents a measurement, therefore it is essential to check the status to know what is being returned.</p> <p>See section <a href="#">Data Streams</a> for detailed explanations of setting up the stream, starting and stopping it, and interpreting the response.</p>
Return Codes	<ul style="list-style-type: none"> <li>• 0x00000000 (S_OK): No Error</li> <li>• 0x80040200: Device not opened</li> <li>• 0x80040304: Bad Device Handle</li> <li>• 0x80040305: Bad Sensor Channel</li> <li>• 0x80040308: The Device is no longer Available</li> <li>• 0x80040402: Failed to create Safe Array</li> <li>• 0x80040500: Stream Mode Not Started</li> </ul>
See Also:	<a href="#">Measurement Delivery</a> , <a href="#">Alphabetic Listing</a> , <a href="#">Common Parameters</a> , <a href="#">Data Streams</a> , <a href="#">GetData Status Codes</a>

### 3.6.5.3 StartStream

Name	<b>StartStream</b>
Parameters	<ul style="list-style-type: none"> <li>• [in] LONG hDevice</li> <li>• [in] LONG channel</li> </ul>
Use	<p>Instruct the selected sensor to begin data streaming measurements to the PC.</p> <p>When StartStream is called, the COM object flushes the buffer of previously gathered measurements.</p> <p>OphirLMMMeasurement will fire the <a href="#">DataReady</a> event when a new data packet has arrived and is ready for reading by the client application.</p> <p>See section <a href="#">Data Streams</a> for detailed explanations of setting up the stream, starting and stopping it, and parsing the response.</p>
Return Codes	<ul style="list-style-type: none"> <li>• 0x00000000 (S_OK): No Error</li> <li>• 0x80004005 (E_FAIL): Unspecified Failure</li> <li>• 0x80040200: Device not opened</li> <li>• 0x80040304: Bad Device Handle</li> <li>• 0x80040305: Bad Sensor Channel</li> <li>• 0x80040308: The Device is no longer Available</li> <li>• 0x80040403: Not Applicable in this Sensor</li> <li>• 0x80040405: Command Failed</li> <li>• 0x80040501: A channel is in Stream Mode</li> </ul>
See Also:	<a href="#">Measurement Delivery</a> , <a href="#">Alphabetic Listing</a> , <a href="#">Common Parameters</a> , <a href="#">Data Streams</a>

### 3.6.5.4 StopAllStreams

Name	<b>StopAllStreams</b>
Parameters	None
Use	Stops data streams of all sensors.  See section <a href="#">Data Streams</a> for detailed explanations of setting up the stream, starting and stopping it, and parsing the response.
Return Codes	<ul style="list-style-type: none"> <li>0x00000000 (S_OK): No Error</li> </ul>
See Also:	<a href="#">Measurement Delivery</a> , <a href="#">Alphabetic Listing</a> , <a href="#">Data Streams</a>

### 3.6.5.5 StopStream

Name	<b>StopStream</b>
Parameters	<ul style="list-style-type: none"> <li>[in] LONG hDevice</li> <li>[in] LONG channel</li> </ul>
Use	Stops data streaming from selected sensor.  See section <a href="#">Data Streams</a> for detailed explanations of setting up the stream, starting and stopping it, and parsing the response.
Return Codes	<ul style="list-style-type: none"> <li>0x00000000 (S_OK): No Error</li> <li>0x80040200: Device not opened</li> <li>0x80040304: Bad Device Handle</li> <li>0x80040305: Bad Sensor Channel</li> <li>0x80040308: The Device is no longer Available</li> <li>0x80040500: Stream Mode Not Started</li> </ul>
See Also:	<a href="#">Measurement Delivery</a> , <a href="#">Alphabetic Listing</a> , <a href="#">Common Parameters</a> , <a href="#">Data Streams</a>

## 3.6.6 Trigger Settings

These methods are not supported by Newport meters.

### 3.6.6.1 GetExtTrigModes

Name	<b>GetExtTrigModes</b>
Parameters	<ul style="list-style-type: none"> <li>[in] LONG hDevice</li> <li>[out] LONG* index</li> <li>[out] VARIANT* options</li> </ul>
Use	Gets the selected external trigger mode of the device and the set of available external trigger modes.
Return Codes	<ul style="list-style-type: none"> <li>0x00000000 (S_OK): No Error</li> <li>0x80040200: Device not opened</li> <li>0x80040304: Bad Device Handle</li> <li>0x80040307: Not Applicable in this Device</li> <li>0x80040308: The Device is no longer Available</li> <li>0x80040402: Failed to create Safe Array</li> </ul>
See Also:	<a href="#">Trigger Settings</a> , <a href="#">Alphabetic Listing</a> , <a href="#">Common Parameters</a> , <a href="#">External Trigger Modes</a>

### 3.6.6.2 GetExtTrigOnOff

Name	<b>GetExtTrigOnOff</b>
Parameters	<ul style="list-style-type: none"><li>• [in] LONG hDevice</li><li>• [in] LONG channel</li><li>• [out] LONG* index</li><li>• [out] VARIANT* options</li></ul>
Use	To get the external trigger state.
Return Codes	<ul style="list-style-type: none"><li>• 0x00000000 (S_OK): No Error</li><li>• 0x80040200: Device not opened</li><li>• 0x80040304: Bad Device Handle</li><li>• 0x80040305: Bad Sensor Channel</li><li>• 0x80040308: The Device is no longer Available</li><li>• 0x80040402: Failed to create Safe Array</li></ul>
See Also:	<a href="#">Trigger Settings</a> , <a href="#">Alphabetic Listing</a> , <a href="#">Common Parameters</a>

### 3.6.6.3 GetExtTrigWindowTime

Name	<b>GetExtTrigWindowTime</b>
Parameters	<ul style="list-style-type: none"><li>• [in] LONG hDevice</li><li>• [out] LONG* extTrigWindow</li></ul>
Use	Get the external trigger window time of <hDevice> in microseconds.
Return Codes	<ul style="list-style-type: none"><li>• 0x00000000 (S_OK): No Error</li><li>• 0x80040200: Device not opened</li><li>• 0x80040304: Bad Device Handle</li><li>• 0x80040307: Not Applicable in this Device</li><li>• 0x80040308: The Device is no longer Available</li></ul>
See Also:	<a href="#">Trigger Settings</a> , <a href="#">Alphabetic Listing</a> , <a href="#">Common Parameters</a> , <a href="#">External Trigger Window</a>

### 3.6.6.4 SetExtTrigMode

Name	<b>SetExtTrigMode</b>
Parameters	<ul style="list-style-type: none"><li>• [in] LONG hDevice</li><li>• [in] LONG index</li></ul>
Use	Set the external trigger mode of <hDevice>.
Return Codes	<ul style="list-style-type: none"><li>• 0x00000000 (S_OK): No Error</li><li>• 0x80040200: Device not opened</li><li>• 0x80040304: Bad Device Handle</li><li>• 0x80040307: Not Applicable in this Device</li><li>• 0x80040308: The Device is no longer Available</li><li>• 0x80040401: Param Error</li><li>• 0x80040405: Command Failed</li></ul>
See Also:	<a href="#">Trigger Settings</a> , <a href="#">Alphabetic Listing</a> , <a href="#">Common Parameters</a> , <a href="#">External Trigger Modes</a>

### 3.6.6.5 SetExtTrigOnOff

Name	<b>SetExtTrigOnOff</b>
Parameters	<ul style="list-style-type: none"> <li>• [in] LONG hDevice</li> <li>• [in] LONG channel</li> <li>• [in] LONG* index</li> </ul>
Use	Set the external trigger state.
Return Codes	<ul style="list-style-type: none"> <li>• 0x00000000 (S_OK): No Error</li> <li>• 0x80040200: Device not opened</li> <li>• 0x80040304: Bad Device Handle</li> <li>• 0x80040305: Bad Sensor Channel</li> <li>• 0x80040308: The Device is no longer Available</li> <li>• 0x80040401: Param Error</li> <li>• 0x80040403: Not Applicable in this Sensor</li> <li>• 0x80040405: Command Failed</li> <li>• 0x80040501: A channel is in Stream Mode</li> </ul>
See Also:	<a href="#">Trigger Settings</a> , <a href="#">Alphabetic Listing</a> , <a href="#">Common Parameters</a>

### 3.6.6.6 SetExtTrigWindowTime

Name	<b>SetExtTrigWindowTime</b>
Parameters	<ul style="list-style-type: none"> <li>• [in] LONG hDevice</li> <li>• [in] LONG extTrigWindowTime</li> </ul>
Use	Set the external trigger window time of <hDevice> in microseconds.
Return Codes	<ul style="list-style-type: none"> <li>• 0x00000000 (S_OK): No Error</li> <li>• 0x80040200: Device not opened</li> <li>• 0x80040304: Bad Device Handle</li> <li>• 0x80040307: Not Applicable in this Device</li> <li>• 0x80040308: The Device is no longer Available</li> <li>• 0x80040401: Param Error</li> <li>• 0x80040405: Command Failed</li> </ul>
See Also:	<a href="#">Trigger Settings</a> , <a href="#">Alphabetic Listing</a> , <a href="#">Common Parameters</a> , <a href="#">External Trigger Window</a>

## 3.6.7 Events

### 3.6.7.1 DataReady

Name	<b>DataReady</b>
Parameters	<ul style="list-style-type: none"> <li>• [in] LONG hDevice</li> <li>• [in] LONG channel</li> </ul>
Use	<p>This event is fired to inform the client application that the sensor at &lt;hDevice&gt; &lt;channel&gt; has delivered a new data payload.</p> <p>OphirLMMeasurement will not deliver the data to the client application until it calls the <a href="#">GetData</a> method.</p> <p>Note: Further DataReady events for the specified sensor will not be fired until OphirLMMeasurement's flag has been cleared by the client application calling <a href="#">GetData</a>.</p>
Return Codes	N/A
See Also:	<a href="#">Measurement Delivery</a> , <a href="#">Events</a> , <a href="#">Alphabetic Listing</a> , <a href="#">Common Parameters</a>

### 3.6.7.2 PlugAndPlay

Name	<b>PlugAndPlay</b>
Parameters	None
Use	Fired when a device that has been found by the <a href="#">ScanUSB</a> method has been disconnected from the system. The client application can rescan the USB to discover which device has been disconnected.
Return Codes	N/A
See Also:	<a href="#">Device Communications</a> , <a href="#">General Information and Diagnostics</a> , <a href="#">Events</a> , <a href="#">Alphabetic Listing</a>

### 3.6.8 ASCII Commands

We recommend using the OphirLMMMeasurement methods wherever possible. Features that do not have a direct method call can be accessed using the ASCII command methods.

The function sequence is always a call to Write followed by a call to Read, i.e. every Write must be followed by a Read, and every Read must be preceded by a Write. Failure to adhere to this sequence may cause communication trouble.

For a description of available commands and device compatibility, please see “**Newport User Commands.pdf**” included in the installation.

#### 3.6.8.1 Read

Name	<b>Read</b>
Parameters	<ul style="list-style-type: none"><li>• [in] LONG hDevice</li><li>• [out] BSTR* reply</li></ul>
Use	Read <reply> from <hDevice>. <reply> is in the form of an ASCII string.
Return Codes	<ul style="list-style-type: none"><li>• 0x00000000 (S_OK): No Error</li><li>• 0x80040200: Device not opened</li><li>• 0x80040304: Bad Device Handle</li><li>• 0x80040308: The Device is no longer Available</li><li>• 0x80040405: Command Failed</li></ul>
See Also:	<a href="#">ASCII Commands</a> , <a href="#">Alphabetic Listing</a> , <a href="#">Common Parameters</a> , <a href="#">Working with Legacy Methods</a>

#### 3.6.8.2 Write

Name	<b>Write</b>
Parameters	<ul style="list-style-type: none"><li>• [in] LONG hDevice</li><li>• [in] BSTR command</li></ul>
Use	Write <command> to <hDevice>. <command> is in the form of an ASCII string.
Return Codes	<ul style="list-style-type: none"><li>• 0x00000000 (S_OK): No Error</li><li>• 0x80040200: Device not opened</li><li>• 0x80040304: Bad Device Handle</li><li>• 0x80040308: The Device is no longer Available</li><li>• 0x80040405: Command Failed</li></ul>
See Also:	<a href="#">ASCII Commands</a> , <a href="#">Alphabetic Listing</a> , <a href="#">Common Parameters</a> , <a href="#">Working with Legacy Methods</a>

## 4 Data Streams

This section describes how measurement data is delivered from Newport instruments to a client application through the OphirLMMMeasurement COM object, and how a client application typically works with that data.

The methods that control data delivery are listed in [Measurement Delivery](#). This section focuses on how those methods are used together.

### 4.1 Measurement Streaming Model

Measurement data is delivered using a streaming model. Once streaming is started, the instrument continuously sends data to the COM object, which buffers it until the client application retrieves it.

A typical measurement session follows this sequence:

1. Open the device and configure the sensor.
2. Start streaming measurements.
3. Retrieve buffered data during or after the measurement session.
4. Stop streaming and close the device.

All sensor types and measurement modes use the same data retrieval mechanism and data format.

### 4.2 Setup and Start

Before starting a measurement stream, the client application must:

1. **Detect and open a device**  
Use [ScanUSB](#) to find connected instruments, then call [OpenUSBDevice](#) to open one of them and obtain a device handle (hDevice).
2. **Optionally configure the sensor**  
Sensor parameters such as measurement mode, range, wavelength, filter, diffuser, pulse length, or threshold may be configured before streaming begins. See [Sensor Configuration](#). The configuration methods generally come in pairs, a Get method which returns a list of options and the currently selected option, and a Set method which allows choosing a new option. Configuration methods cannot be called while a channel is streaming.
3. **Start streaming**  
Call [StartStream](#)(hDevice, channel) to begin measurement delivery.  
When streaming starts, any previously buffered data is discarded.

### 4.3 Data Collection

Once streaming has started, the COM object buffers incoming data from the instrument. How and when the data is retrieved is up to the client application.

Valid data collection strategies include:

- Responding to the [DataReady](#) event and retrieving data as it arrives.
- Periodically polling at a fixed interval.
- Allowing data to accumulate in the buffer and retrieving it once at the end of the measurement session.
- Clearing the buffer by calling [GetData](#) (and discarding the results), performing measurements, then calling [GetData](#) to retrieve the measurements.

#### 4.3.1 **Retrieving Data**

Measurement data is retrieved by calling [GetData](#)(hDevice, channel, arrayValue, arrayTimestamp, arrayStatus)

This method returns three parallel arrays of equal length:

- **Value** – the measurement value (e.g. Watts or Joules) as a double
- **Timestamp** – time of measurement, in milliseconds as a double
- **Status** – status code describing the measurement as a LONG

Each index across the arrays corresponds to one data item. After [GetData](#) returns, the retrieved items are removed from the internal buffer. If no new data is available, all arrays are empty.

### 4.3.2 Interpreting Status Codes

Not every data item represents a valid measurement. The status code must always be checked to determine the meaning of each returned value.

Status codes indicate conditions such as:

- Valid measurement
- Overrange or saturation
- Pulse acquisition states
- Beam tracking components
- Informational notifications (e.g. temperature or filter state changes)

The complete list is provided in [GetData Status Codes](#).

## 4.4 Stopping and Closing

When measurements are complete, stop streaming using [StopStream](#) or [StopAllStreams](#), then close the device using [Close](#). After a device is closed, no further configuration or measurement calls should be made for that handle.

## 4.5 GetData Status Codes

Each data item returned by [GetData](#) includes a status code describing its meaning. Client applications must interpret this code before using the measurement value. The following is the list of status codes that can be returned. Other codes that may be reported are reserved for internal use or future expansion.

Status Code	Meaning	When and Where
0	OK	Power/Energy/Exposure measurements for all sensors
1	Overrange	Power and Energy measurements for all sensors
2	Saturated	Power and Energy measurements for all sensors
3	Missing	Energy with Pyroelectric and PD Energy sensors on an X938 or X940 device
4	Reset state	Single shot energy or Pulsed Power with a Thermopile sensor
5	Waiting	Single shot energy or Pulsed Power with a Thermopile sensor
6	Summing	Single shot energy or Pulsed Power with a Thermopile sensor
7	Timeout	Single shot energy or Pulsed Power with a Thermopile sensor
8	Peak Over	Single shot energy or Pulsed Power with a Thermopile sensor
9	Energy Over	Single shot energy or Pulsed Power with a Thermopile sensor
0x010000	X measurement OK	Track measurements with a BeamTrack sensor
0x010001	X measurement Error	Track measurements with a BeamTrack sensor
0x020000	Y measurement OK	Track measurements with a BeamTrack sensor
0x020001	Y measurement Error	Track measurements with a BeamTrack sensor
0x030000	Size measurement OK	Track measurements with a BeamTrack sensor
0x030001	Size measurement Error	Track measurements with a BeamTrack sensor
0x030002	Size measurement Warning	Track measurements with a BeamTrack sensor
0x040001	Filter State Change	Photodiode sensors with built-in filter state detection reporting a change in filter state (in to out or vice-versa)
0x050000	Pulse frequency	Power and Energy with Pyroelectric and PD Energy sensors
0x100000	Temperature	
0x200000	Alert Hot	
0x300000	Pulse Width	
0x400000	PfP Energy	