











LabVIEW COM Demo

User Manual

1	Overview	3
2	Related Documents	3
3	LabVIEW COM Demo	4
3.1	VI Hierarchy	4
3.2	VI's	4
3.3	Add or Edit Wavelengths.vi 	5
3.3.1	Event Table	5
3.3.2	List of Controls	5
3.3.3	List of Indicators	5
3.4	Channel States.ctl 	5
3.5	Channel Template.vit 	6
3.5.1	States	6
3.5.2	Events	6
3.5.3	List of Controls	7
3.5.4	List of Indicators	8
3.6	Filter out non-numeric characters.vi 	8
3.6.1	List of Controls	8
3.6.2	List of Indicators	8
3.7	Get Handles for Scan Devices.vi 	8
3.7.1	List of Controls	8
3.7.2	List of Indicators	9
3.8	Get values from Variant.vi 	9
3.8.1	List of Controls	9
3.8.2	List of Indicators	9
3.9	LV Demo Main.vi 	9
3.9.1	Event States	10
3.9.2	List of Controls	11
3.9.3	List of Indicators	11
3.10	Set Handles on Open.vi 	12
3.10.1	List of Controls	12
3.11	Stop All Channels Streaming.vi 	12
3.11.1	List of Controls	12
3.11.2	List of Indicators	12
3.12	Time to Day-Hr-Min-Sec-msec.vi 	13
3.12.1	List of Controls	13
3.12.2	List of Indicators	13
4	Using OphirLMMeasurement in Your LabVIEW VI	14
5	Simple Measurement.vi	17
6	Revision History	18

1 Overview

This document describes the demo LabVIEW VI's provided that drive Ophir devices using the OphirLMMeasurement COM object.

The LabVIEW Demo program demonstrates communication with Ophir devices in the LabVIEW environment.

The goal of our VI package is to provide the following:

- A fully functional demo application
- A block diagram structure that is easily understood that will be the end-user's basis for customization and adaptation.

Since it is a fully functional application, the LabVIEW programmer can concentrate on his application without getting bogged down with Ophir's device communication details.

This library was developed in LabVIEW 8.6.1 and has been tested in LabVIEW 2009 and LabVIEW 2010 as well.

2 Related Documents

- 1) **OphirLMMeasurement COM Object.doc**. Description of the OphirLMMeasurement COM object that Ophir provides for the client application to communicate with and control Ophir equipment

3.3 Add or Edit Wavelengths.vi

The Add or Edit Wavelength.vi displays a dialog box with a list of wavelengths which the user can change or add to. The VI consists of an Event Loop State machine with the following events:

3.3.1 Event Table

Event	Description
Panel Close	Called when the window is closed (this state stops the running of this VI)
New Value	Called when the New Value control changes. Note that the new value is passed to the Filter out non-numeric characters.vi to remove any non-numeric values.
Wavelengths	Called when the user selects a new wavelength from the Wavelengths list. The newly selected wavelength is then filled to the New Value control.
Add	Adds the value in New Value to the list of wavelengths and sends to device for that channel.
Replace	Replaces the value in New Value with the one selected in Wavelengths list and sends that new value to the device for the selected channel.

3.3.2 List of Controls

	Control	Description
*	Dev Refnum In	Reference used to communicate (via OphirLMMeasurement COM object) with the device
*	Handle	Input of handle from calling VI
*	Ch	Input of channel from calling VI
*	Wavelen In	List (array of string) of current wavelengths
	Wavelengths	Editable list of current wavelengths
	New Value	String of new value to add or replace to device
	Add	Add button
	Replace	Replace button

3.3.3 List of Indicators

	Indicators	Description
*	Error Out	Passes out any errors which occur in the VI
*	Wavelen out	List of current wavelengths.

3.4 Channel States.cti

Although listed in the VI hierarchy, this is not really a VI. It is a type definition used by the VI and it defines the states of the Channel Template state machine (type definitions allow controls, indicators and constants to be used through a program and have the benefit that when the type definition is changed, every instance that is based on the type definition automatically updates to have the new values).



3.5 Channel Template.vit

This VI is called and loaded into memory 4 times – one time for each channel (each time creating a clone of the original template). The Channel Template VI is a state machine (a While loop with a Case in it).

After the state machine is run there is a Case which controls measurements. If the Measure? shift register of the state machine is True, the VI reads the current value from the device for the selected channel. Time is translated from a value of milliseconds to the format of Day/Hr:Min:Sec.msec and, if not equal to zero, is displayed in the Ch1 Time/Status cluster.

3.5.1 States

State	Description
Init	Initializes the background color of the VI (depending on the input channel number), sets the local Ch variable and sets the title Ch string
Poll	Consists of an Event Loop (see below)
Set Handle	Called by the Main VI whenever a device's handle changes (or when a device changes). <ul style="list-style-type: none"> - Sets the local Handle control, & reads the device's head type & sensor name to set the ChType indicator. Also checks to see if device's channel has an active sensor (IsSensorExists). If not, this state disables the entire Channel VI if the Channel is not the first channel. - Inverts the GetAll cluster to force an update of all values. - Disables Edit Wavelengths and Delete Wavelength buttons if wavelengths are not modifiable (GetWavelengthsExtra.modifiable = False)
Measure	Called by the Main VI whenever measurement changes. Sets the Measure? Shift register which is used to control the measuring case (see below)

3.5.2 Events

The Poll state has an Event Case to handle user input. The Event case has the following 20 events:

Event	Description
Timeout	Does nothing, passing control back to the Main VI after 1 millisecond
GetRanges	Polls the device for the specified channel to get the ranges and fills the ranges in the Ranges control.
Ranges	Sets the range for the current device/channel
GetWavelength	Polls the device for the specified channel to get the wavelengths and fills the wavelengths in the Wavelengths control.
Wavelengths	Sets the wavelength for the current device/channel
Edit Wavelengths	Calls Add or Edit Wavelengths.vi
Del Wavelen	Deletes the selected wavelength and forces a query of current wavelengths
GetDiffuser	Queries device for current channel's list of diffusers and sets Diffusers control.
GetMode	Queries device for current channel's list of modes and sets Modes control.
GetPulseLen	Queries device for current channel's list of pulse lengths and sets PulseLen control.
GetThreshold	Queries device for current channel's list of thresholds and sets Thresholds control.
GetFilters	Queries device for current channel's list of Filters and sets Filters control.
GetTrig	Queries device for current channel's list of triggers and sets Triggers control.
Diffusers	Sets device's diffuser value for current channel
Modes	Sets device's mode value for current channel
PulseLen	Sets device's pulse length value for current channel
Thresholds	Sets device's threshold value for current channel
Filters	Sets device's filter value for current channel
Triggers	Sets device's trigger value for current channel
Save Settings	Sends command to save current settings to device for current channel

3.5.3 List of Controls

	Control	Description
R	State	Determines which state to perform (Init, Poll, Set Handle or Measure)
	Ch	Local variable holding the channel of this iteration of the Channel Template VI
	Ch 1 Cluster	See below
	Is Pulsar	Boolean to indicate if the device is a Pulsar (set in the Get Handle state)
	Get All cluster	Local variable used to force any one of 9 states: Get Range, Get Wavelengths, Get Diffusers, Get Modes, Get Thresholds, Get Filters, Get Triggers
	Handle	Local variable holding the handle for this channel
*	Handle In	Input of handle from Main VI
*	Channel In	Input of channel from Main VI
*	Dev Refnum in	Reference used to communicate (via OphirLMMMeasurement COM Object) with the device
*	Set Measure	Boolean to set whether or not VI should measure data

3.5.4 List of Indicators

* Indicators	Description
Error Out	Passes out any errors which occur in the VI
Ch	Displays channel number

3.6 Filter out non-numeric characters.vi



This VI is called by [Add or Edit Wavelengths.vi](#) to filter out any non-numeric characters the user may have accidentally typed. It has one string input and one string output (filtered version of string input).

3.6.1 List of Controls

* Control	Description
String in	Pre-filtered wavelength string

3.6.2 List of Indicators

* Indicators	Description
String out	Filtered wavelength string

3.7 Get Handles for Scan Devices.vi



When the USB is scanned new devices may appear, existing devices may disappear, or no change may occur. Since the program associates a handle with each Ophir device (Serial Number), some administrative work must be done to keep handles associated with their devices. This VI does this administration. If the new number of devices is the same as the old number of devices, the VI simply passes the existing handles back out. However, if the number of devices is not the same, then it searches the old device names for each new device. If it finds the new device in the list of old devices, then it pulls that device's handle from the list of previous handles. Otherwise it assigns a value of 0.

3.7.1 List of Controls

Control	Description
* New Device Name	List of new device names
* All Handles In	List of Handles
* Old Device Names	List of old device names

3.7.2 List of Indicators

	Indicators	Description
*	# of Handles	Number of handles of current configuration
*	All Handles Out	List of Handles (1 for each device).

3.8 Get values from Variant.vi



This VI converts the input Variant parameter into an array of strings. It also takes an input Value and returns this same value in Value Out if there is at least one element in the list of strings converted from the variant. If not, however, the Value out is set to zero. In addition, if the converted string array is empty, the Strings out is set to have 1 element which reads "N/A" and the Disabled out value is set to 2 (= disabled), otherwise it is set to 1 (= enabled).

3.8.1 List of Controls

	Control	Description
*	Variant In	Variant input control
*	Value In	I32 Index to string array input

3.8.2 List of Indicators

	Indicators	Description
*	Value Out	Equal to Value In unless Strings Out is empty (then = 0)
*	Strings out	List output contained in Variant In (if Variant In was empty, this array has 1 element = N/A)
*	Disabled out	= 0 if Variant in contained valid list, otherwise =2 (disabled)

3.9 LV Demo Main.vi



LV Demo Main.vi is the top level VI of this program. It has the main task of initializing the system and handling global user actions as well as polling the cloned channel VI's. This VI initializes the 4 cloned channel windows and loads them into their respective subpanels. In addition, during initialization, the software opens a reference to OphirLMMMeasurement.

The main VI handles global user requests as well as loading into memory and calling a "cloned" sub-VI for each of the 4 channels (not all devices use all 4 channels).

The "cloned" channel sub-VIs are loaded into memory from a VI template and each is loaded into its own subpanel (this is how LabVIEW allows one VI to embed another VI into its window). Thus, each channel's code is identical (and any changes made to the template are automatically implemented for all channels).

3.9.1 Event States

After initialization, the software runs its main loop which consists of an Event Case with the following 22 states:

State	Description
Timeout	Used to poll each Channel's cloned VI to handle that channel's actions
Stop/Panel close	Called when Stop button selected or Panel closed. Closes all open handles of all devices and closes references to all cloned Channel Vis
Get Lib Ver	Polls OphirLMMeasurement COM Object to return the library version
Get Driver Ver	Polls OphirLMMeasurement COM Object to return the driver version
ScanUSB	Scans the USB bus to get a list of all connected devices; loads device names into the Devices control and initializes all handles to 0
Open	Opens the currently selected device, stores its handle in All Handles array and sends the handle to all Channel Vis.
Close	Closes the selected device
Reset All	Closes all devices and turns measuring off (in case it was on) for each channel. Also forces a rescan of the USB port (using the rescan again control)
Devices	When the user selects a different device from the Devices control, this state sends the new device's handle to each Channel VI
Update Info	Updates the Device Name
Reset Device	Turns off measuring for each Channel and resets all devices.
ConfigCh/Stream Modes...	This state is necessary to handle user input to controls which don't directly do anything (e.g., entering a new value which is only sent when a button is pressed). LabVIEW requires a state for each control changed so the change is updated for the next iteration of the loop.
Set Config	Sets the Channel, Stream Mode and Stream Mode Value for the current device
Start Stream	Starts streaming for all channels of the current device. Tells Channel Vis to start measuring.
Stop Stream	Stops streaming for all channels of the current device. Tells Channel Vis to stop measuring.
StopAll	Stops all streaming
Get Modes	Gets External Trigger Modes from device and displays in externTrigModes
GetWindow	Gets External Trigger Window time from device and displays in extTrig Window
Modify	Set external trigger window time
Write	Writes the value in the Writestr control to the device. If AutoRead = True, then it also reads from the device and displays the result in ReadStr
Read	Reads from the device and displays the result in ReadStr
AutoRead	Doesn't do anything- Event loops need to handle changes of items changed even if they don't need to do anything with that change or the change won't be used on the next iteration of the loop

After any state is called, if an error occurs it is displayed. When the user activates any button and there is no error, the last error (if any) is cleared.

3.9.2 List of Controls

As the top VI in the hierarchy none of the Controls are VI Inputs

Control	Description
All Handles	Holds the handle for each device opened by the software
Gen'l Ops	Cluster holding the controls: - Scan USB [Boolean], - Device list [array I32 of device names], - Open [Boolean], - Close [Boolean] and - Reset All [Boolean]
Dev Controls	Cluster containing the controls: - Update Info [Boolean] - Reset Device [Boolean] - Configure Channel [I32] - Stream Modes [I32] - Value [I32] - Set Configure [Boolean] - Start Stream [Boolean] - Stop Stream [Boolean] - Stop All [Boolean] - Get Modes [Boolean] - extTrigModes [I32] - Get Window [Boolean] - ExtTrig Window [I32] - Modify [Boolean]
Read/Write	Cluster containing the following controls: - Automatically Read after Write [Boolean] - Write [Boolean] - Read [Boolean] - WriteStr [string]
Scan Again	Boolean used artificially force the program to rescan the USB port for devices.

3.9.3 List of Indicators

As the top VI in the hierarchy none of the Indicators are VI Outputs

Indicator	Description
Library Version	String indicator displays the Library Version
Driver Version	String indicator displays the Driver version
Device Name	String indicator displays the device name, ROM version & serial number
Handle	Displays the current device's handle
Readstr	Displays the text read from the device

3.10 Set Handles on Open.vi



If there is no error on opening a device (that is, if the device is not already open), this VI performs the following tasks:

- Closes the previous handle (if that handle was not equal to 0)
- Sends the new handle to all 4 Channel Template Vis
- Replaces the new handle value of this device in the array of Handles
- Adds the Handle to the list of Devices and Handles on the Main VI's window

3.10.1 List of Controls

	Control	Description
*	All Handles Ref	Reference to the All Handles array of the calling VI
*	hDevice	Handle of the selected device.
*	Selected Device	Index of the selected device in the list of devices.
*	Error In Cluster	Error cluster used to determine whether or not the code in this VI should be run
*	Devices List Box Ref	Reference to the Devices List Box of the calling VI
*	Ch Ref Array	Array of references to Channel Template windows
*	LMMMeasurementLib.ICoL MMeasurement	Reference to call COM functions

3.11 Stop All Channels Streaming.vi



This VI calls the OphirLMMMeasurement method **StopStream** for each channel of the currently selected device and tells each [Channel Template.vit](#) to stop measurements

3.11.1 List of Controls

	Indicators	Description
*	LMMMeasurementLib.ICoL MMeasurement	Reference to call COM functions
*	Handle	Handle of the selected device.
*	Template Refs	Array of references to Channel Template windows

3.11.2 List of Indicators

	Indicators	Description
*	Error Out	LabVIEW Error Cluster

3.12 Time to Day-Hr-Min-Sec-msec.vi

This VI converts the input Time in milliseconds to a string with the time formatted as Day/Hr:Min:Sec.msec.

3.12.1 List of Controls

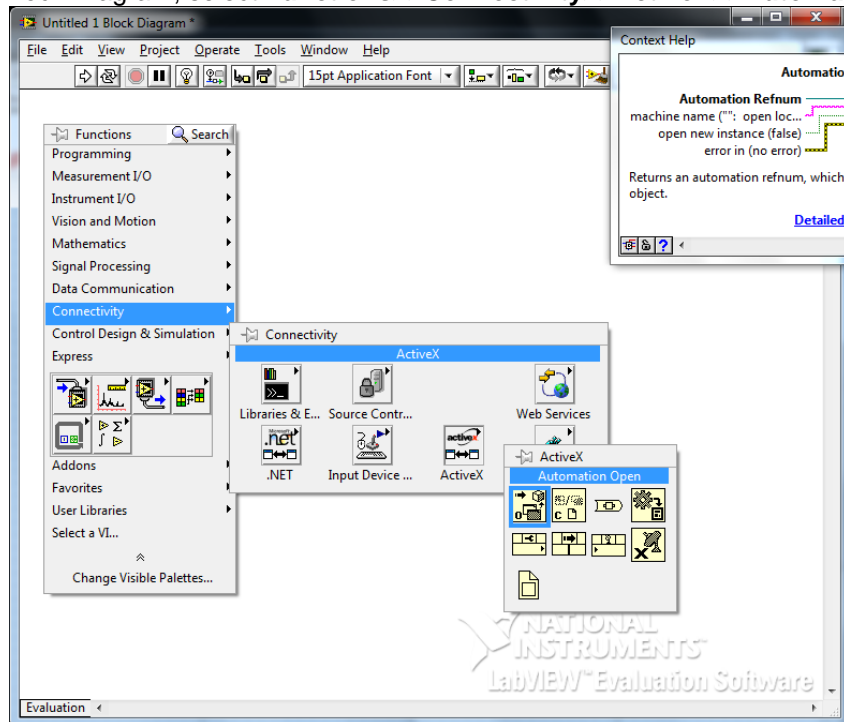
	Control	Description
*	Time in millsec	Input time in milliseconds

3.12.2 List of Indicators

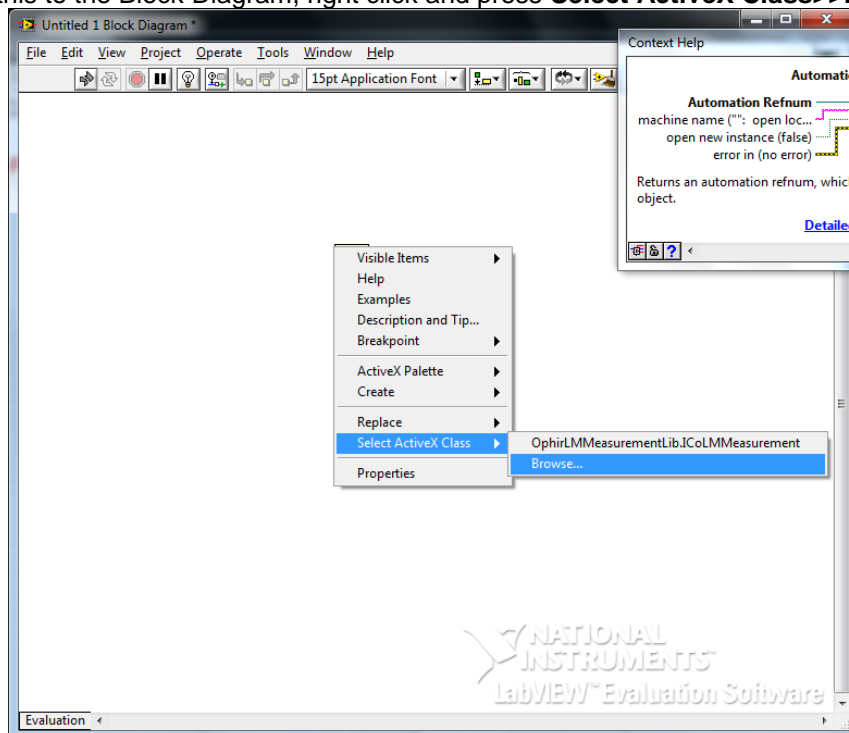
	Indicators	Description
*	Time Day/Hr:Min:Sec.msec	String output of time formatted as Day/Hr:Min:Sec.msec

4 Using OphirLMMeasurement in Your LabVIEW VI

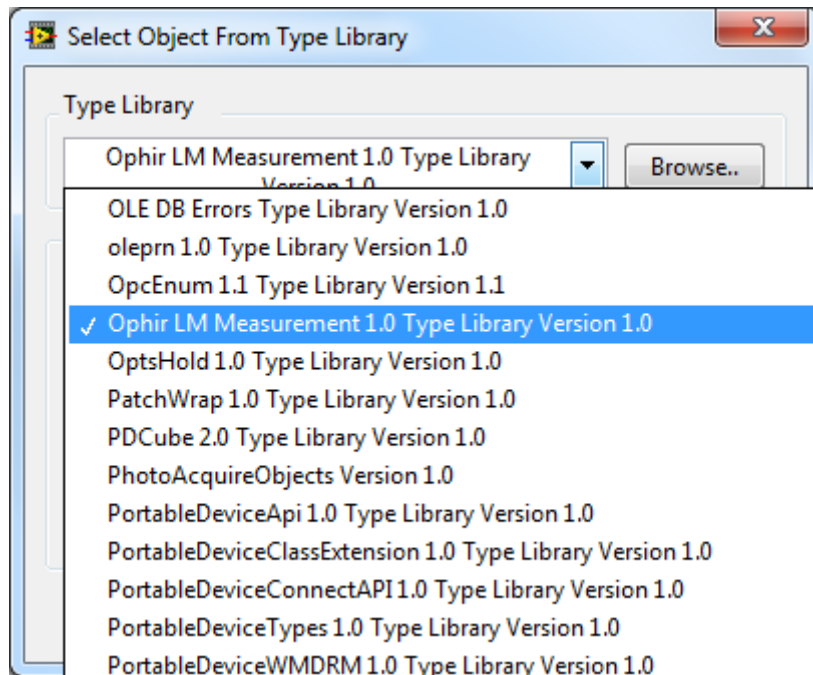
1. In the Block Diagram, select **Functions>>Connectivity>>ActiveX>>Automation Open**



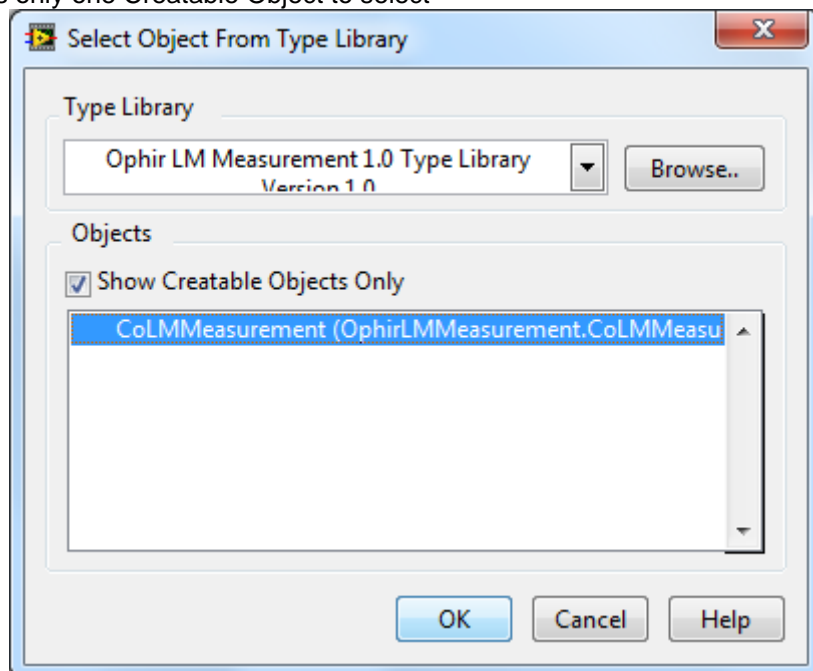
2. Drag this to the Block Diagram, right click and press **Select ActiveX Class>>Browse**



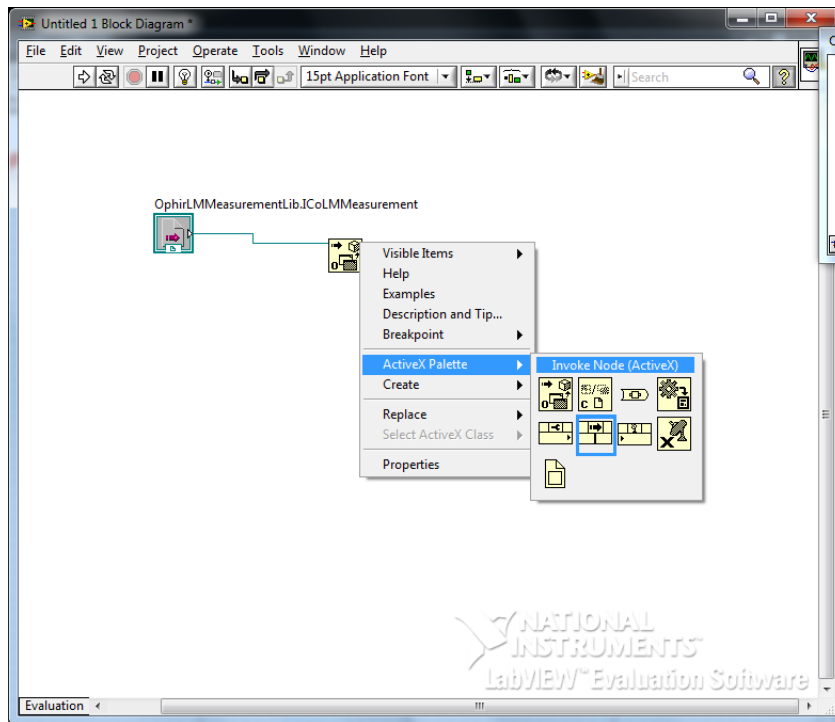
3. This opens the **Select Object From Type Library** dialog box. Select OphirLMMeasurement from the drop down list



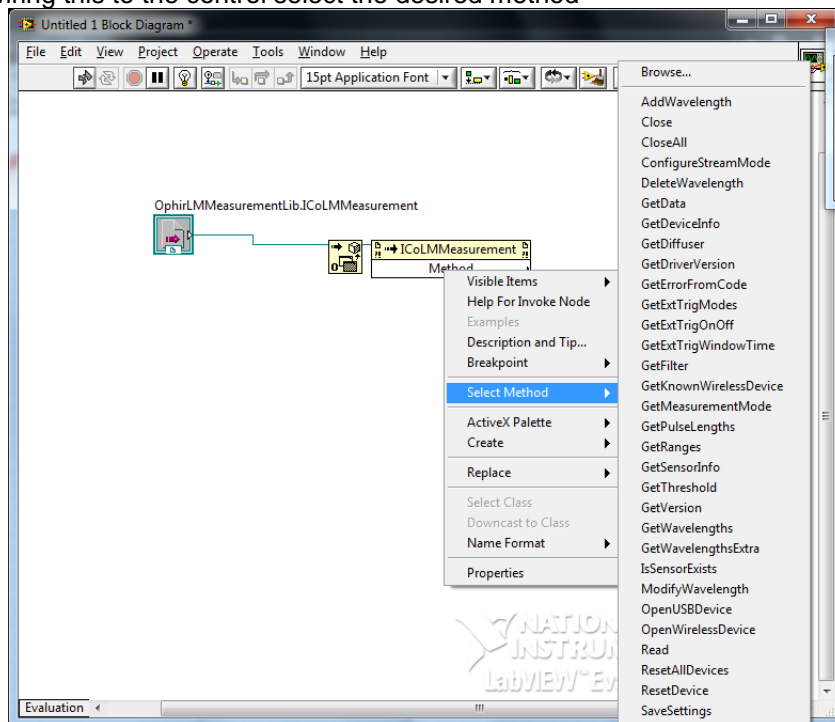
4. There is only one Creatable Object to select



5. To select a method, right click and select **ActiveX Palette>>Invoke Node (ActiveX)**

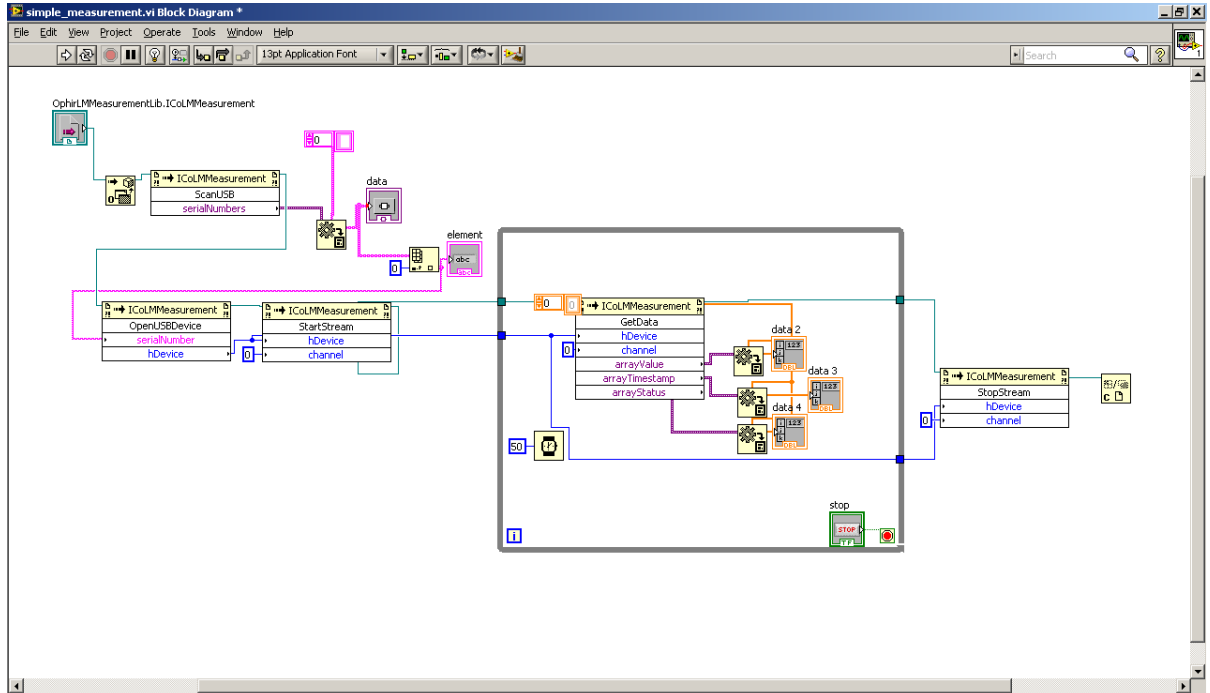


6. After wiring this to the control select the desired method



5 Simple Measurement.vi

At Ophir, we found that for some LabVIEW customers, the LabVIEW COM Demo package is too much. These customers want a simple example that does the least necessary to get data from the Ophir device into LabVIEW. We therefore include in the LabVIEW package simple_measurement.vi. This VI scans the USB for a device, opens it, and begins measuring. This is enough to get started.



6 Revision History

- | | | |
|-----|------------|---|
| 01. | 10-Nov-10. | Initial Revision. |
| 02. | 30-Apr-14. | Added Section Simple Measurement.vi |