

# How to Upgrade Your Application to Use OmniDriver 1.6

## Overview

This document provides contains information for upgrading your application to use OmniDriver 1.6. For more OmniDriver information, see the [OmniDriver Programming Guide](#).

## C/C++ Applications Using the C++ Interface to OmniDriver

### ► Procedure

1. Define an environment variable named JAVA\_HOME which points to the location where you installed the Java JDK. For example: C:\Program Files\Java\jdk1.6.0\_21. You should see a directory named **include** directly beneath the JAVA\_HOME directory.
2. You must restart your IDE after defining JAVA\_HOME for this new environment variable to be accessible.
3. You can then modify the properties of your Visual Studio project to use this environment variable to specify the location of required header files. For example: \$(JAVA\_HOME)\include.

When you install OmniDriver 1.6, a new environment variable will be defined: OMNIDRIVER\_HOME.

OMNIDRIVER\_HOME will be set to the value C:\Program Files\Ocean Optics\OmniDriverSPAM (or wherever you have installed the OmniDriver product). You can then modify the properties of your Visual Studio project to use this environment variable to specify the location of required header files. Example: \$(OMNIDRIVER\_HOME)\include.

To build your C++ application, you must define following header file paths:

- \$(OMNIDRIVER\_HOME)\include
- \$(JAVA\_HOME)\include
- \$(JAVA\_HOME)\include\win32

You will also need the following library path:

- \$(OMNIDRIVER\_HOME)\OOI\_HOME

You must also specify the following linker "Additional Dependencies"

- OmniDriver32.lib
- common32.lib
- optional: Spam32.lib

## Note for 64-bit platform

If you are building your application to run on a 64-bit platform, change all library references to their 64-bit form.

- OmniDriver64.lib
- common64.lib
- Spam64.lib

## Troubleshooting

**Symptom:** UnsatisfiedLinkError: NatUSB.dll (can't find dependent libraries)

**Solution:** Uninstall the old driver for your spectrometer(s). Then reinstall your spectrometer using the new WinUSB driver.

## C# Applications

If you attempt to rebuild your 1.5 application after installing 1.6, you may get the error "The referenced component 'OmniDriver' could not be found." or "The type or namespace 'OmniDriver' could not be found...". Use the following procedure to prevent these errors.

### ► Procedure

To convert the OmniDriver 1.5 COM application to use the OmniDriver 1.6 .NET object/assembly, use the following procedure:

1. Expand the References node in the Solution Explorer for your project.
2. Right-click on **OmniDriver** and choose **Remove**.
3. Right-click on **References** and choose **Add Reference**.
4. Click on the **Browse** tab.
5. Navigate to the OOI\_HOME directory. Highlight **NETOmniDriver.dll**, and click **OK**.
6. You may need to do the same for NETSpam.dll if your application uses functions from the SPAM library.
7. If you attempt to rebuild your application at this point, you will see errors like "The type of namespace name 'CCoWrapperClass' does not exist in the namespace 'OmniDriver (are you missing an assembly reference?)"  
To solve this... replace "CCoWrapperClass" with "CCoWrapper".

8. Remove calls to the wrapper.CreateWrapper() method. It is no longer necessary to call this method.

---

### Note

It is still necessary to call the `CreateObjectName()` methods for SPAM objects which take one or more arguments. For example, you should continue to call the method `CIELAB.CreateCIELAB(myCIEColorObject)` because it takes an argument. But you do *not* need to call `AdvancedColor.CreateAdvancedColor()` because it does not take an argument.

---

9. When you rebuild your application, you may see one of the following error messages: "Operator '=' cannot be applied to operands of type 'short' and 'bool', or "The best overloaded method match for SOME\_METHOD() has some invalid arguments". The solution is simply to replace the true/false symbolic values with 1 or 0, respectively.

## C++ Applications Using the COM Interface to OmniDriver

You must add a reference to the new .NET assembly.

1. Click on menu item **Project | References...**
2. Expand Common Properties and click on References.
3. Click the **Add Path** button.
4. Browse to the OOI\_HOME directory and add the **NETOmniDriver.dll**.

## Visual Basic COM Applications

This section contains the procedure to convert Visual Basic COM Applications to use the OmniDriver 1.6 .NET object/assembly.

If you attempt to rebuild your 1.5 application after installing OmniDriver 1.6, you may get the error "Cannot load type library for reference 'OmniDriver'..." or "The referenced component 'OmniDriver' could not be found." Use the following procedure to prevent these errors.

### ► Procedure

1. Remove the reference to the old OmniDriver COM object (if necessary):
  - a. In the Solution Explorer window, expand the References node. If the References node is not visible, try rebuilding your application (it will report the above-mentioned error). This will cause the References node to appear in the Solution Explorer window. If the References node does *not* appear, it means the IDE cannot find the old reference. In this case there is nothing that you need remove, so proceed to Step 2 below.
  - b. Click on the "OmniDriver" reference and choose "Remove".

## How to Upgrade Your Application to Use OmniDriver 1.6

---

2. Click on the menu item **Project | Add Reference**.
3. Click on the **Browse** tab of the **Add Reference** dialog box.
4. Navigate to the OOI\_HOME directory and highlight **NETOmniDriver.dll** (typically located in C:\Program Files\Ocean Optics\OmniDriver\OOI\_HOME). Click **OK**.

To declare an instance of the new OmniDriver .NET assembly:

```
Dim wrapper As New OmniDriver.CCoWrapper
```

Remove all calls to wrapper.CreateWrapper(): This method is no longer needed. However, you should continue to call the “Create” methods for SPAM objects when these “Create” methods accept one or more parameters.

## VB6 Applications

---

### Caution: VB6 Limitation

**VB6 applications are not supported on 64-bit operating systems. The problem arises because 64-bit operating systems require 64-bit drivers (which Ocean Optics provides). But, VB6 can only generate 32-bit executables, and you cannot access a 64-bit driver from a 32-bit application. Microsoft does not support mixing 32-bit and 64-bit modules in a single application.**

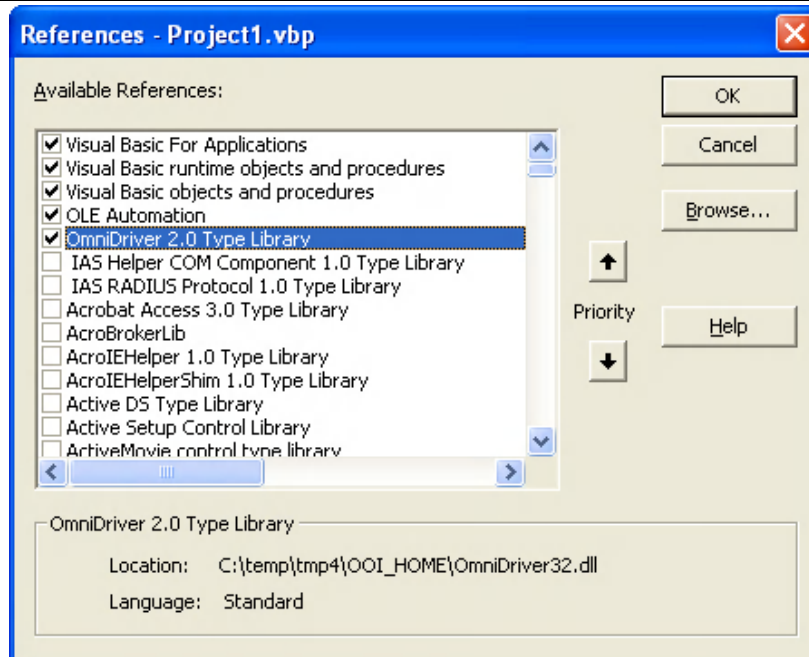
---

## VB6 Applications Using the COM Interface

If you try to make an EXE without changing the reference to the OmniDriver COM object, you will see the following error message:



This error message is associated with your declaration(s) of OmniDriver COM objects. Click **OK** and then the References dialog box will appear:



► **Procedure**

To solve this problem:

1. Uncheck the reference to the obsolete OmniDriver type library.
2. Select **Project | References**.
3. Click the **Browse** button.
4. Navigate to OOI\_HOME directory, highlight NETOmniDriver.tlb, and click **Open**.
5. Scroll down in the list of available references and check **OmniDriver .NET interface**.
6. If you are using SPAM functions, you will also need to add a reference to the new NETSpam.tlb type library following Steps 1-5.

Change Dim wrapper as New OmniDriver.CCoWrapper to Dim wrapper as new CCoWrapper.

Change Dim wrapperExtensions as OmniDriver.CCoWrapperExtensions to Dim wrapperExtensions as CCoWrapperExtensions.

Make similar changes to any other OmniDriver objects which you have declared in your application.

Remove all calls to wrapper.CreateWrapper(). This method is no longer needed. However, you should continue to call the “Create” methods for SPAM objects when these “Create” methods accept one or more parameters.

You may see the following error message:

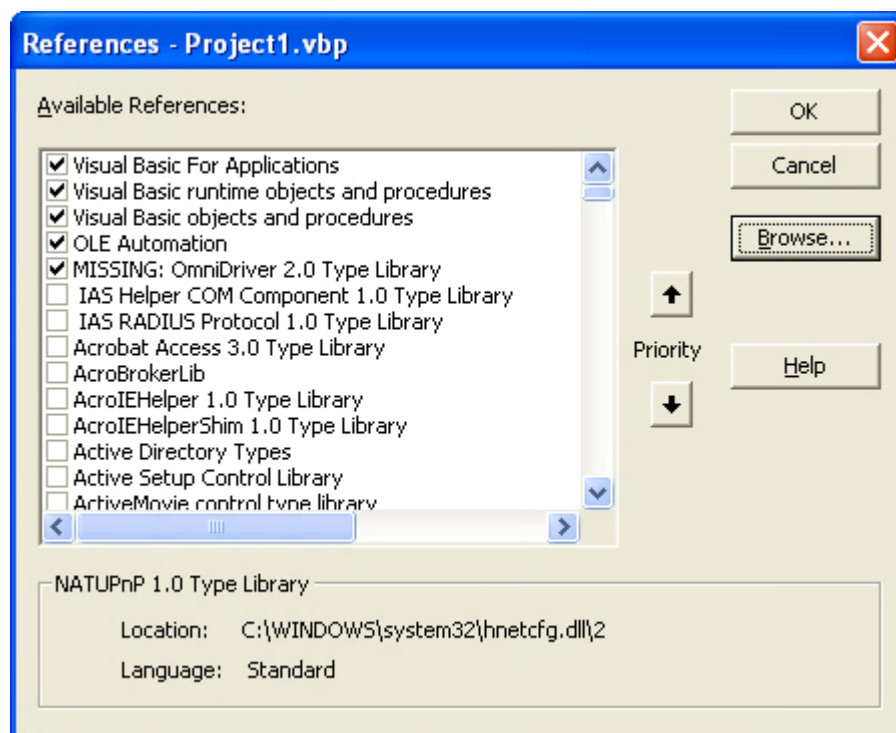


To solve this error, append “\_2” to the method name. Or, if the “\_2” was already present, remove it.

Alternatively, you can solve this error by adding the channelIndex argument following the spectrometerIndex argument. Be sure to set the channelIndex argument to zero since this was the previous default setting that your application expects.

## VB6 Applications Using the “C” Interface

If you try to make an EXE without changing the reference to the OmniDriver COM object, you will see the following screen:



### ► Procedure

To solve this problem:

1. Uncheck **MISSING: OmniDriver 2.0 Type Library** and click **OK**.
2. Select **Project | References**.
3. Click the **Browse** button.

4. Navigate to OOI\_HOME directory, highlight NETOmniDriver.tlb, and click OK
5. If you are using SPAM functions, you will also need to add a reference to the NETSpam.tlb type library.

Change Dim wrapper as New OmniDriver.CCoWrapper to Dim wrapper as new CCoWrapper.

Change Dim wrapperExtensions as OmniDriver.CCoWrapperExtensions to Dim wrapperExtensions as CCoWrapperExtensions.

Make similar changes to any other OmniDriver objects which you have declared in your application.

Remove all calls to wrapper.CreateWrapper(). This method is no longer needed.

However, you should continue to call the "Create" methods for SPAM objects when these "Create" methods accept one or more parameters.

