ASCOM VB LocalServer Development Framework Checklist

From the **ASCOM Platform 6.1SP1** and **ASCOM Platform Developer Components** documentation, there are two documented ways of developing an ASCOM Local Server:

A. ("Making a Local Server based Driver.pdf" recommendation)
- Start a Visual Studio solution with the local server template
- add 1 or more driver projects using the driver template
- develop and test the driver(s) as in-proc DLLs
- change driver(s) to be LocalServer served
- test drivers served by the local server

B. (LocalServer "ReadMe.htm" recommendation)
- Start a Visual Studio solution with the driver template
- optionally add more driver projects using the driver template
- develop and test the driver(s) as in-proc DLLs
- add a project with the local server template
- change driver(s) to be LocalServer served
- test drivers served by the local server

Type B is much more work to put the local server files in the correct namespace and other naming cleanup from the template wizard (not really recommended here) - it requires "Find In Files" and replacements to correct naming usage.

The following is a checklist and an annotated walk-through example of generating an ASCOM VB LocalServer for a fictitious company/product called Acme serving an ASCOM Focuser driver and a FilterWheel driver assumed to share a single serial port via a custom controller - therefore the need to use a LocalServer. The screen images items associated with each step are highlighted in **red**. The end result is a LocalServer skeleton framework with a tested "working", "non-functional" local server serving two ASCOM drivers.
- - "working" means the served drivers pass the Conformance Checker tool
- - "non-functional" means they control no actual hardware (yet)

Development Environment
- 64-bit Windows 7
- Visual Basic 2010 Express Edition
- ASCOM Platform 6.1SP1 installed
- ASCOM Platform Developer Components installed
- ASCOM Driver Conformance Checker installed
- ASCOM 6 Local Server Template (VB) installed (not part of Platform 6.1SP1)

Note: All projects in a multi-project solution in Express Editions of Visual Studio must use the same programming language.

Visual Basic 2010 Express ← Run as administrator !!!

Checklist **Image**

□ 0) Obtain and copy the "**ASCOM 6 Local Server Template (VB)**" file
"*ASCOM LocalServer Template VB.zip*" to the Visual Basic Project Templates folder:
C:\Users\username\Documents\Visual Studio 2010\Templates\ProjectTemplates\Visual Basic\ASCOM6

□ 1) New Project - **ASCOM 6 Local Server Template (VB)** NewProject.png
- Name: **Acme**

□ 2) File➢Save All – SaveProject.png
- Name: **Server**
- Location: **...\VB Projects\ASCOM**
- Solution Name: **Acme** ← **MAKE SURE**!!! because Visual Studio makes same as Name!!!
- [✓] Create directory for solution

□ 3) Set Server Properties➢Application➢Assembly name: **ASCOM.Acme.Server** ServerAssemblyName.png

□ 4) Add New Project for Focuser driver
- File➢Add➢New Project...
- **ASCOM Device Driver (VB)**
  - Name: **Focuser**

AddFocuser.png

□ 5) ASCOM Driver Project Wizard
- Device Class - **Focuser**
- Device Name/Model - **Acme**
- Create

WizardFocuser.png

□ 6) Set Focuser Properties➢Compile➢Advanced Compile Options...➢
Target framework (all configurations): **.NET Framework 3.5**
The driver must use the same .NET Framework as the server.

FocuserNET3.5.png

□ 7) Enable only 32-bit code generation for the **Release** configuration of the Focuser driver by modifying (outside of the Visual Basic IDE environment) the <PlatformTarget> tag in the Focuser driver's Focuser.vbproj file to be **x86** located under the following tag
<PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Release|AnyCPU' ">
(i.e. in Focuser.vbproj: <PlatformTarget>**x86**</PlatformTarget>)

□ 8) Enable XML documentation for the Focuser driver by modifying (outside of the Visual Basic IDE environment) the empty <DocumentationFile> tags in the Focuser driver's Focuser.vbproj file to be the project's name **Focuser.xml** located under the following tags:
<PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Debug|AnyCPU' ">
<PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Release|AnyCPU' ">
(i.e. in Focuser.vbproj: <DocumentationFile>**Focuser.xml**</DocumentationFile>)
(no "Generate XML documentation file" option in Visual Basic 2010 Express Edition's Properties➢Compile)

□ 9) Add New Project for FilterWheel driver
- File➢Add➢New Project...
- **ASCOM Device Driver (VB)**
  - Name: **FilterWheel**

AddFilterWheel.png

□ 10) ASCOM Driver Project Wizard
- Device Class - **FilterWheel**
- Device Name/Model - **Acme**
- Create

WizardFilterWheel.png

□ 11) Set FilterWheel Properties➢Compile➢Advanced Compile Options...➢
Target framework (all configurations): **.NET Framework 3.5**
The driver must use the same .NET Framework as the server.

FilterWheelNET3.5.png

□ 12) Enable only 32-bit code generation for the **Release** configuration of the Focuser driver by modifying (outside of the Visual Basic IDE environment) the <PlatformTarget> tag in the Focuser driver's Focuser.vbproj file to be **x86** located under the following tag
<PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Release|AnyCPU' ">
(i.e. in Focuser.vbproj: <PlatformTarget>**x86**</PlatformTarget>)

□ 13) Enable XML documentation for the Focuser driver by modifying (outside of the Visual Basic IDE environment) the empty <DocumentationFile> tags in the Focuser driver's Focuser.vbproj file to be the project's name **Focuser.xml** located under the following tags:
<PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Debug|AnyCPU' ">
<PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Release|AnyCPU' ">
(i.e. in Focuser.vbproj: <DocumentationFile>**Focuser.xml**</DocumentationFile>)
(no "Generate XML documentation file" option in Visual Basic 2010 Express Edition's Properties➢Compile)

□ 14) Build the solution

□ 15) Run the ASCOM **Conform** tool. If it is running in 64 bit mode, change it to run in 32 bit mode with:
Options➢Conformance Options➢General➢Conform Settings:
[✓] Run as 32bit on a 64bit OS
(this is needed because Visual Studio's *Register for COM interop* only registers the drivers as a 32bit COM driver, but not also as a 64bit COM driver on a 64-bit machine - as would be done by the Inno Setup installer)

Non64bitReg
Conform32Bits.png

□ 16) Using the ASCOM **Conform** tool, Options➢Check Focuser, Options➢Select Driver, select the *Acme Focuser*

ConformChooserFocuser.png

□ 17) Select ASCOM Focuser Chooser➢Properties... to get the DeviceName Setup dialog for the Focuser

ConformFocuserProperties.png

□ 18) Run the Check Conformance and verify that no errors, warnings or issues are found and the Focuser driver passes ASCOM validation!!

ConformFocuser.png

□ 19) Using the ASCOM **Conform** tool, Options➢Check Filter Wheel, Options➢Select Driver, select the *Acme FilterWheel*

ConformChooserFilterWheel.png

□ 20) Select ASCOM FilterWheel Chooser➢Properties... to get the DeviceName Setup dialog for the FilterWheel

ConformFilterWheelProperties.png

□ 21) Run the Check Conformance and verify that no errors, warnings or issues are found and the FilterWheel driver passes ASCOM validation!!

ConformFilterWheel.png

□ 22) Add New Project for application for testing the drivers
- File➢Add➢New Project...
- **ASCOM Driver Test Forms Application (VB)**
- Name: **TestDrivers**

AddTestDrivers.png

□ 23) ASCOM Driver Project Wizard
- Device Class - **Focuser**
- Device Name/Model - **Acme**
- Create

WizardFocuser.png

□ 24) If Option Strict is turned on, remove the following error:
On Form1.vb, line 43, add **.ToString** to the end of the line

OptionStrictError.png

□ 25) Right-Click Solution Explorer➢TestDrivers project➢Set as Startup Project to set the TestDrivers project as the startup project

□ 26) Build the solution

□ 27) Run the code (F5), click the test form's Choose button, select the *Acme Focuser*, select ASCOM Focuser Chooser➢Properties... to get the DeviceName Setup dialog for the Focuser, OK those dialogs and verify the ASCOM.Acme.Focuser is shown on the test form.

TestDriversResults.png

At this point, additional code can be added to the separate Focuser and FilterWheel drivers to independently control the Focuser and FilterWheel hardware and additional code and controls can be added to the TestDrivers project to exercise and debug the features of the in-proc DLL Focuser and FilterWheel drivers.

□ 28) Clean the solution with Build➢Clean Solution so that the driver will be automatically unregistered from COM and ASCOM
(if menu Build➢Clean Solution is not shown, use Tools➢Customize➢Commands➢Menu bar: Build➢Add Command...➢Categories: Build, Commands: Clean Solution, OK, Close to add that menu item)

At this point, the Acme Focuser and FilterWheel should no longer be available in **Conform**'s Select Driver.

Now, make the changes to incorporate the **LocalServer** functionality.

□ 29) In Focuser Properties➢Compile➢
- Set Configuration: **All Configurations**, Platform: **Active (Any CPU)**
- Set Output➢Output path: **..\Server\bin\Debug\**
- Save project

FocuserOutput.png

□ 30) Disable Register for COM interop for the Focuser driver by modifying (outside of the Visual Basic IDE environment) the <RegisterForComInterop> tag in the Focuser driver's Focuser.vbproj file to be **false** located under the following tag: (default was "true")
<PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Debug|AnyCPU' ">
(i.e. in Focuser.vbproj: <RegisterForComInterop>**false**</RegisterForComInterop>)

□ 31) Right-Click Focuser project➢Add Reference...➢Projects➢Server to add a reference to the Server project to the Focuser Project

□ 32) Add the following class to the Focuser project's Driver.vb file just before the Focuser class definition: FocuserDriverMods.png

```
    Friend Class FocuserLocalServerConstants
        Friend Const DRIVER_ID As String = "ASCOM.Acme.Focuser";
        Friend Const DRIVER_DESCRIPTION As String = "Acme Focuser";
    End Class
```

□ 33) Add the following attribute declarations to the Focuser project's Driver.vb Focuser class definition: FocuserDriverMods.png

```
  <ProgId(FocuserLocalServerConstants.DRIVER_ID)> _
  <ServedClassName(FocuserLocalServerConstants.DRIVER_DESCRIPTION)> _
```

□ 34) Change the Focuser project's Driver.vb Focuser class definition to inherit *ReferenceCountedObjectBase*: FocuserDriverMods.png

```
  Public Class Focuser
     Inherits ReferenceCountedObjectBase
```

□ 35) Change the Focuser project's Driver.vb *driverID* definition to: FocuserDriverMods.png

```
  Friend Shared driverID As String =
                      FocuserLocalServerConstants.DRIVER_ID;
```

□ 36) Change the Focuser project's Driver.vb *driverDescription* definition to: FocuserDriverMods.png

```
  Private Shared driverDescription As String =
                      FocuserLocalServerConstants.DRIVER_DESCRIPTION;
```

□ 37) Remove the Focuser project's Driver.vb ASCOM registration region code

□ 38) In FilterWheel Properties➢Compile➢ FilterWheelOutput.png
- Set Configuration: **All Configurations**, Platform: **Active (Any CPU)**
- Set Output➢Output path: **..\Server\bin\Debug\**
- Save project

□ 39) Disable Register for COM interop for the Focuser driver by modifying (outside of the Visual Basic IDE environment) the <RegisterForComInterop> tag in the Focuser driver's Focuser.vbproj file to be **false** located under the following tag: (default was "true")
<PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Debug|AnyCPU' ">
(i.e. in Focuser.vbproj: <RegisterForComInterop>**false**</RegisterForComInterop>)

□ 40) Right-Click FilterWheel Project➢Add Reference...➢Projects➢Server to add a reference to the Server project to the FilterWheel project

□ 41) Add the following class to the FilterWheel project's Driver.vb file just before the FilterWheel class definition: FilterWheelDriverMods.png

```
 Friend Class FilterWheelLocalServerConstants
     Friend Const DRIVER_ID As String = "ASCOM.Acme.FilterWheel";
     Friend Const DRIVER_DESCRIPTION As String = "Acme FilterWheel";
 End Class
```

□ 42) Add the following attribute declarations to the FilterWheel project's Driver.vb FilterWheel class definition: FilterWheelDriverMods.png

```
<ProgId(FilterWheelLocalServerConstants.DRIVER_ID)> _
<ServedClassName(FilterWheelLocalServerConstants.DRIVER_DESCRIPTION)>
_
```

□ 43) Change the FilterWheel project's Driver.vb FilterWheel class definition to inherit *ReferenceCountedObjectBase*: FilterWheelDriverMods.png

```
  Public Class FilterWheel
      Inherits ReferenceCountedObjectBase
```

□ 44) Change the FilterWheel project's Driver.vb *driverID* definition to: FilterWheelDriverMods.png

```
 Friend Shared driverID As String =
                  FilterWheelLocalServerConstants.DRIVER_ID;
```

□ 45) Change the FilterWheel project's Driver.vb *driverDescription* definition to: FilterWheelDriverMods.png

```
  Private Shared driverDescription As String =
                  FilterWheelLocalServerConstants.DRIVER_DESCRIPTION;
```

□ 46) Remove the FilterWheel project's Driver.vb ASCOM registration region code

□ 47) Right-Click Solution Explorer➢Server project➢Set as Startup Project to set the local server as the startup project — ServerStartup.png

□ 48) Build the solution

□ 49) Add Server Properties➢Debug➢Start Options➢Command line arguments: **/register** — ServerRegister.png

□ 50) Run the project (to have the local server register the drivers with COM and ASCOM)
(this registers the drivers for both 32 bit and 64 bit {on a 64-bit machine}, so the
ASCOM Conform tool can now be run as 64 bits without problems)

□ 51) Using the ASCOM Conform tool, Options➢Check Focuser, Options➢Select Driver, select the Acme Focuser

□ 52) Select ASCOM Focuser Chooser➢Properties... to get the DeviceName Setup dialog for the Focuser

□ 53) Run the Check Conformance and verify that no errors, warnings or issues are found and the ASCOM.Acme.Focuser driver passes ASCOM validation!!

□ 54) Using the ASCOM Conform tool, Options➢Check Filter Wheel, Options➢Select Driver, select the Acme FilterWheel

□ 55) Select ASCOM FilterWheel Chooser➢Properties... to get the DeviceName Setup dialog for the FilterWheel

□ 56) Run the Check Conformance and verify that no errors, warnings or issues are found and the ASCOM.Acme.FilterWheel driver passes ASCOM validation!!

□ 57) Change Server Properties➢Debug➢Start Options➢Command line arguments: **/unregister**

□ 58) Run the project (to have the local server unregister the drivers with COM and ASCOM)

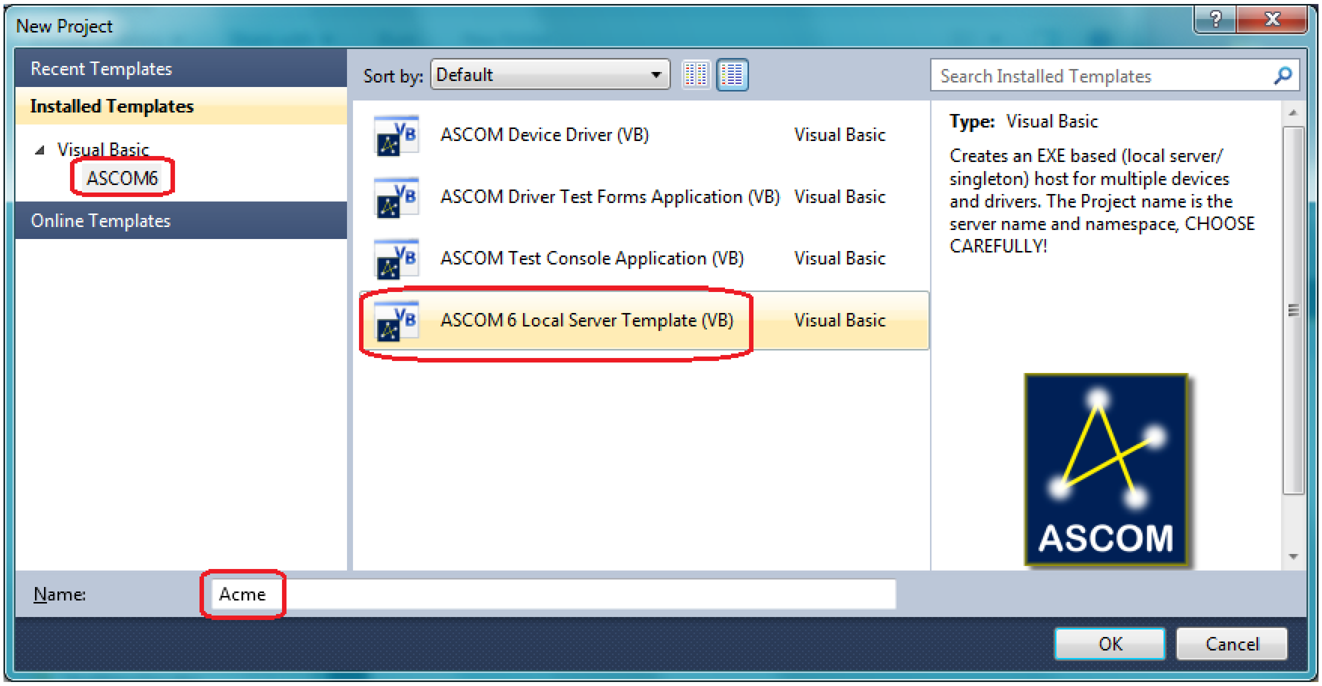
At this point, code can be changed in the Focuser and FilterWheel drivers to appropriately work with the common Focuser and FilterWheel hardware and additional code and controls can be added to the TestDrivers project to exercise and debug the features of the LocalServer-served Focuser and FilterWheel drivers.

When the ASCOM local server and drivers development is complete, the ASCOM Driver Install Script Generator an be used to generate an Inno Setup script to generate a Windows setup executable that can be used to distribute the server and drivers just developed.
Note: The server/driver solution should be closed in the IDE before running the Inno Setup compiler.

ASCOM VB LocalServer Development Framework Walk-Through

Visual Basic 2010 Express ← Run as administrator !!!

1) Start with New Project - **ASCOM 6 Local Server Template (VB)**
   - Name: **Acme**



New Project

Recent Templates

Installed Templates

Visual Basic

ASCOM6

Online Templates

Sort by: Default

Search Installed Templates

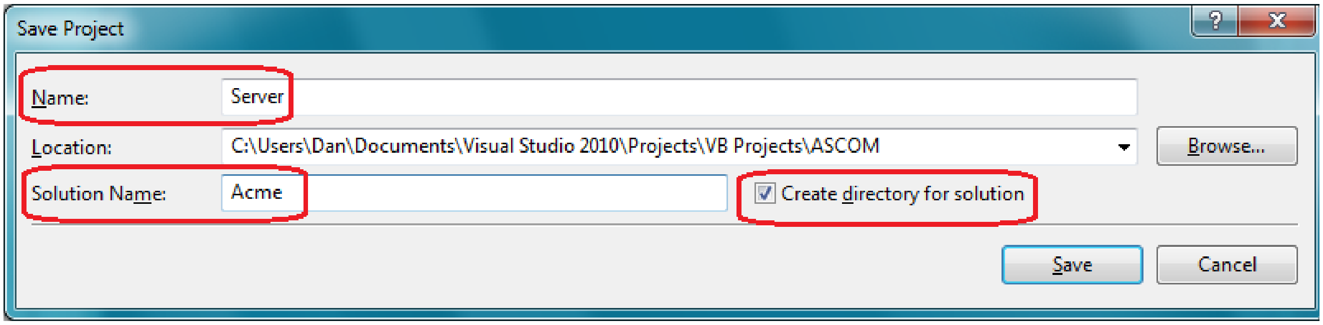
ASCOM Device Driver (VB)	Visual Basic
ASCOM Driver Test Forms Application (VB)	Visual Basic
ASCOM Test Console Application (VB)	Visual Basic
ASCOM 6 Local Server Template (VB)	Visual Basic

Type: Visual Basic

Creates an EXE based (local server/ singleton) host for multiple devices and drivers. The Project name is the server name and namespace, CHOOSE CAREFULLY!

Name: Acme

2) File➢Save All –
   - Name: **Server**
   - Location: **...\VB Projects\ASCOM**
   - Solution Name: **Acme** ← **MAKE SURE**!!! because Visual Studio makes same as Name!!!
   - [✓] Create directory for solution



Save Project

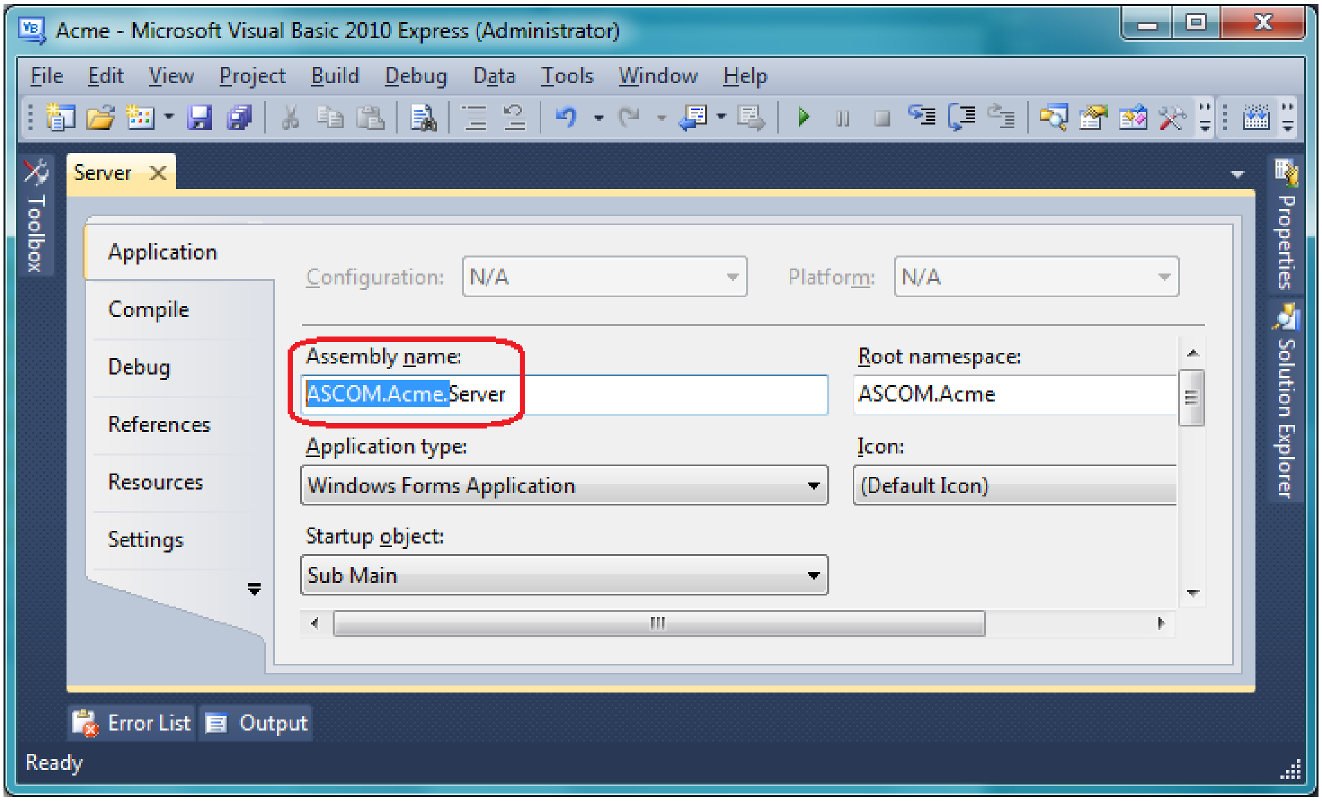
Name: Server

Location: C:\Users\Dan\Documents\Visual Studio 2010\Projects\VB Projects\ASCOM

Solution Name: Acme

☑ Create directory for solution

3) Set Server Properties➢Application➢Assembly name: **ASCOM.Acme.Server**
(The template wizard does not include full assembly name - done here for consistency.)


Acme - Microsoft Visual Basic 2010 Express (Administrator)
File Edit View Project Build Debug Data Tools Window Help
Toolbox
Server
Properties
Solution Explorer
Application
Compile
Debug
References
Resources
Settings
Error List
Output
Ready

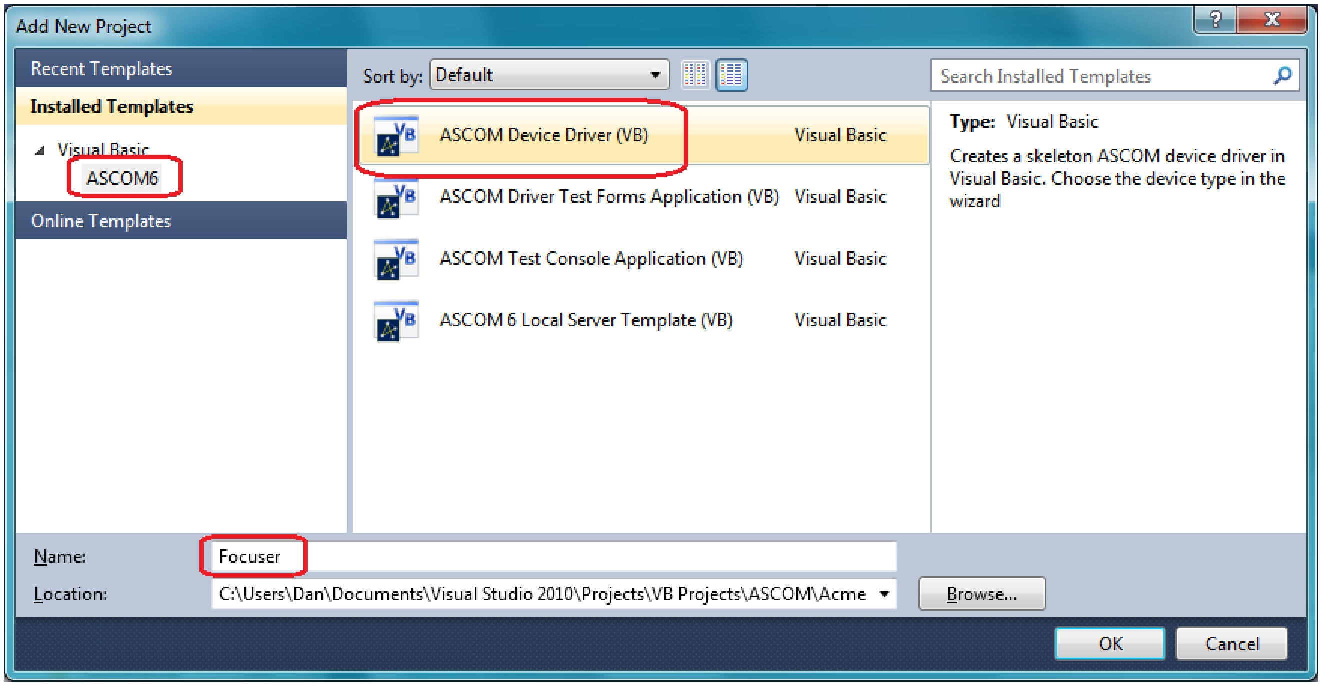
Configuration: N/A
Platform: N/A

Assembly name: ASCOM.Acme.Server
Root namespace: ASCOM.Acme

Application type: Windows Forms Application
Icon: (Default Icon)

Startup object: Sub Main

4) Add New Project for Focuser driver
- File➢Add➢New Project...
- **ASCOM Device Driver (VB)**
  - Name: **Focuser**



Sort by: Default

ASCOM Device Driver (VB)	Visual Basic
ASCOM Driver Test Forms Application (VB)	Visual Basic
ASCOM Test Console Application (VB)	Visual Basic
ASCOM 6 Local Server Template (VB)	Visual Basic

Type: Visual Basic

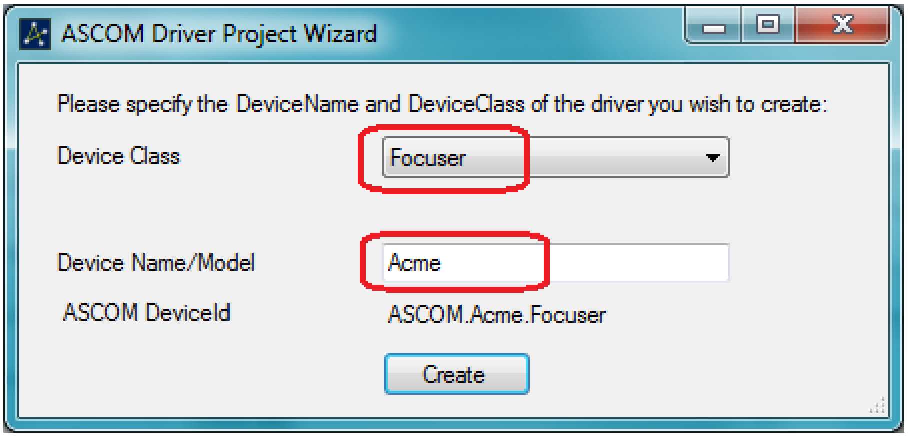
Creates a skeleton ASCOM device driver in Visual Basic. Choose the device type in the wizard

Name: Focuser

Location: C:\Users\Dan\Documents\Visual Studio 2010\Projects\VB Projects\ASCOM\Acme

5) ASCOM Driver Project Wizard

- Device Class - **Focuser**
- Device Name/Model - **Acme**
- Create



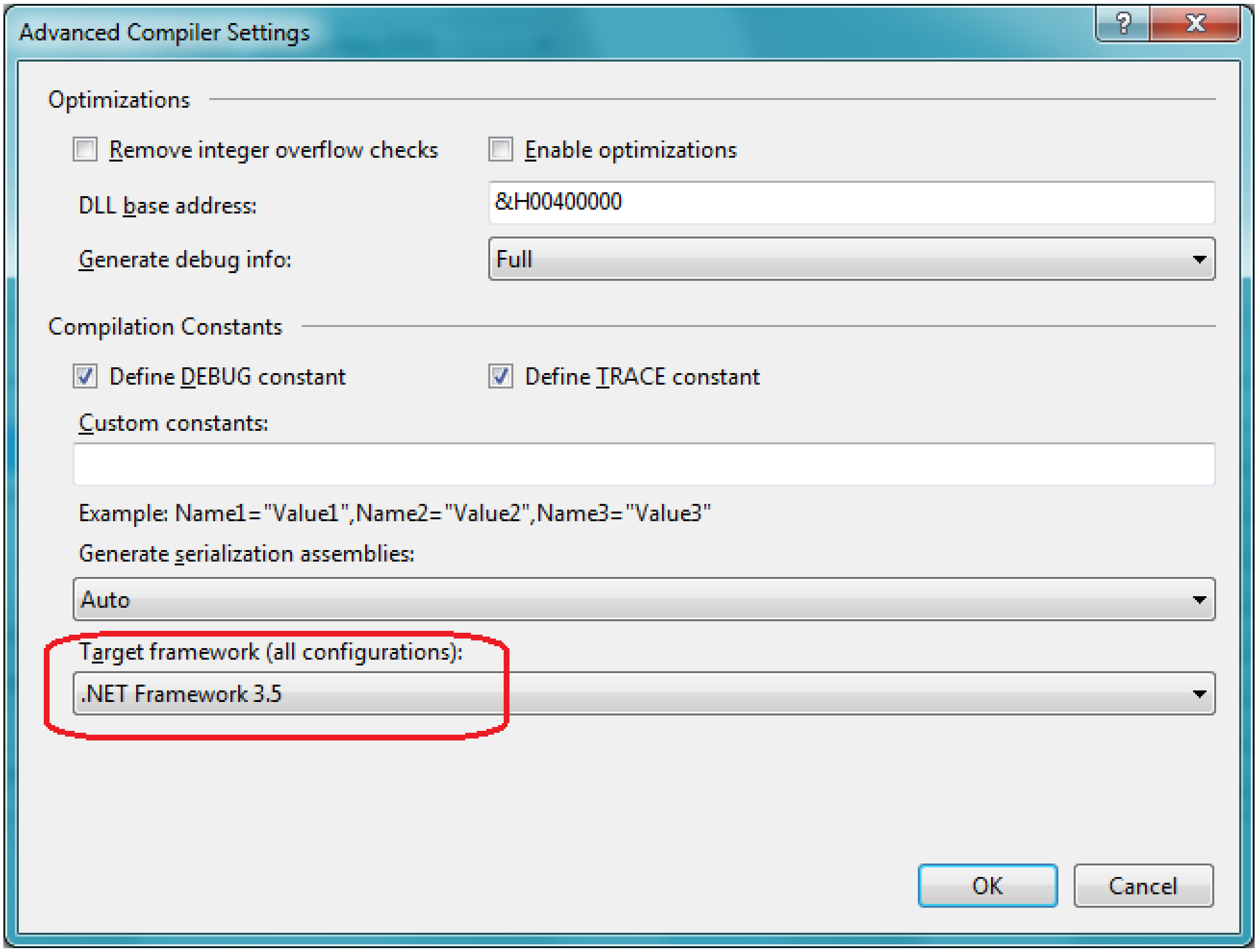
Please specify the DeviceName and DeviceClass of the driver you wish to create:

Device Class: Focuser

Device Name/Model: Acme

ASCOM DeviceId: ASCOM.Acme.Focuser

6) Set Focuser Properties➢Compile➢Advanced Compile Options...➢
Target framework (all configurations): **.NET Framework 3.5**
(The driver must use the same .NET Framework as the server.)



Optimizations

□ Remove integer overflow checks □ Enable optimizations

DLL base address: &H00400000

Generate debug info: Full

Compilation Constants

☑ Define DEBUG constant ☑ Define TRACE constant

Custom constants:

Example: Name1="Value1",Name2="Value2",Name3="Value3"

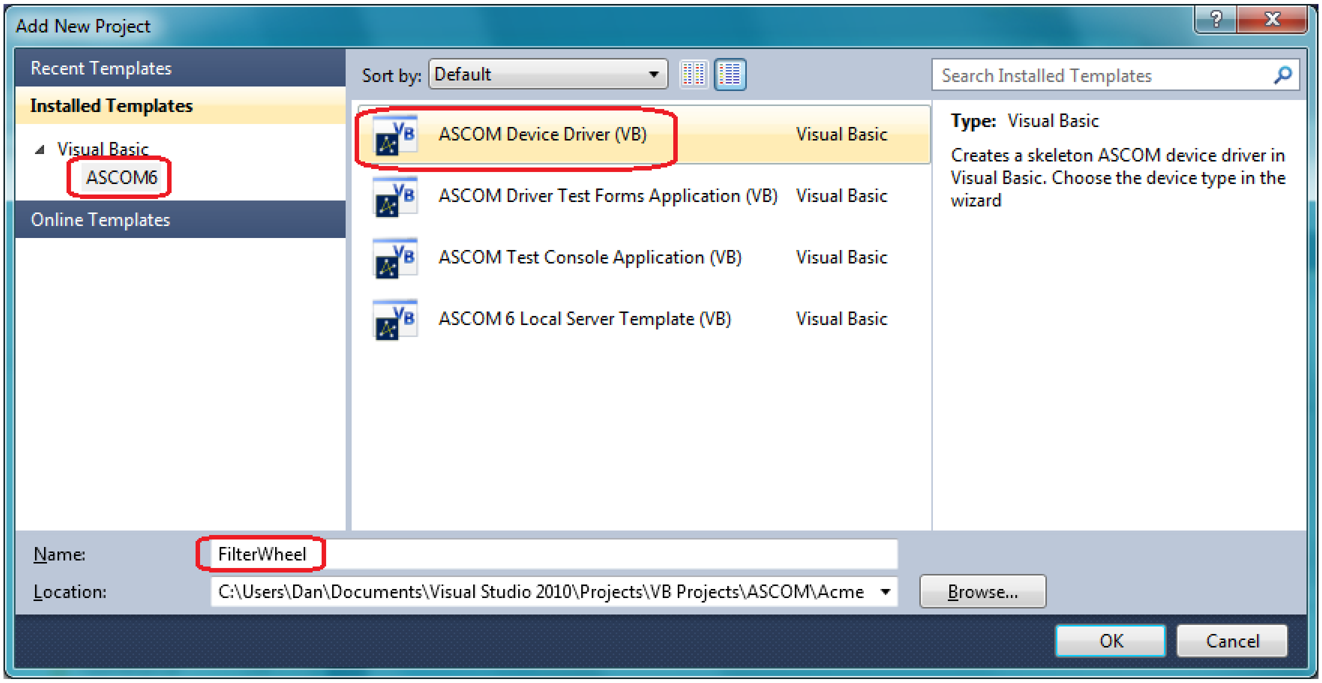
Generate serialization assemblies: Auto

Target framework (all configurations): .NET Framework 3.5

7) Enable only 32-bit code generation for the **Release** configuration of the Focuser driver by modifying (outside of the Visual Basic IDE environment) the <PlatformTarget> tag in the Focuser driver's Focuser.vbproj file to be **x86** located under the following tag
 <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Release|AnyCPU' ">
 (i.e. in Focuser.vbproj: <PlatformTarget>**x86**</PlatformTarget>)
(The Debug and Release configurations are not consistent in the template.)

8) Enable XML documentation for the Focuser driver by modifying (outside of the Visual Basic IDE environment) the empty <DocumentationFile> tags in the Focuser driver's Focuser.vbproj file to be the project's name **Focuser.xml** located under the following tags:
 <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Debug|AnyCPU' ">
 <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Release|AnyCPU' ">
 (i.e. in Focuser.vbproj: <DocumentationFile>**Focuser.xml**</DocumentationFile>)
 (no "Generate XML documentation file" option in Visual Basic 2010 Express Edition's Properties➢Compile)

9) Add New Project for FilterWheel driver
 - File➢Add➢New Project...
 - **ASCOM Device Driver (VB)**
 - Name: **FilterWheel**



Add New Project

Type: Visual Basic

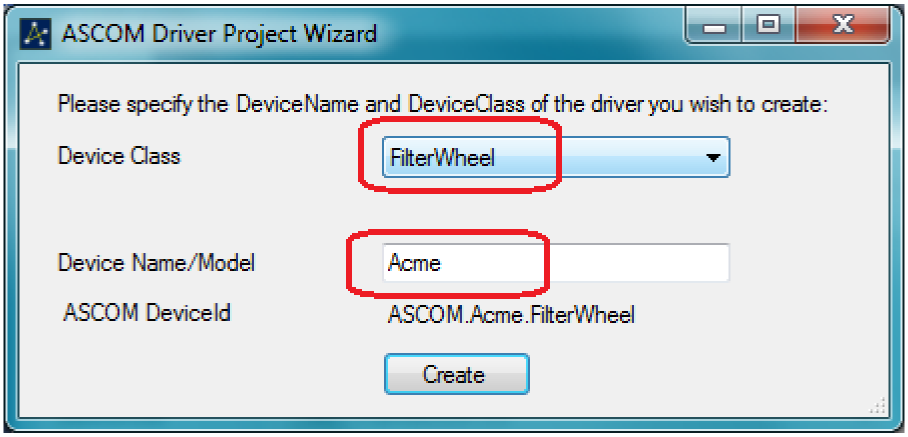
Creates a skeleton ASCOM device driver in Visual Basic. Choose the device type in the wizard

Name: FilterWheel

Location: C:\Users\Dan\Documents\Visual Studio 2010\Projects\VB Projects\ASCOM\Acme

Recent Templates
Installed Templates
Visual Basic
ASCOM6
Online Templates
Sort by: Default
Search Installed Templates
ASCOM Device Driver (VB) Visual Basic
ASCOM Driver Test Forms Application (VB) Visual Basic
ASCOM Test Console Application (VB) Visual Basic
ASCOM 6 Local Server Template (VB) Visual Basic
Browse...
OK
Cancel

10) ASCOM Driver Project Wizard
- Device Class - **FilterWheel**
- Device Name/Model - **Acme**
- Create



ASCOM Driver Project Wizard

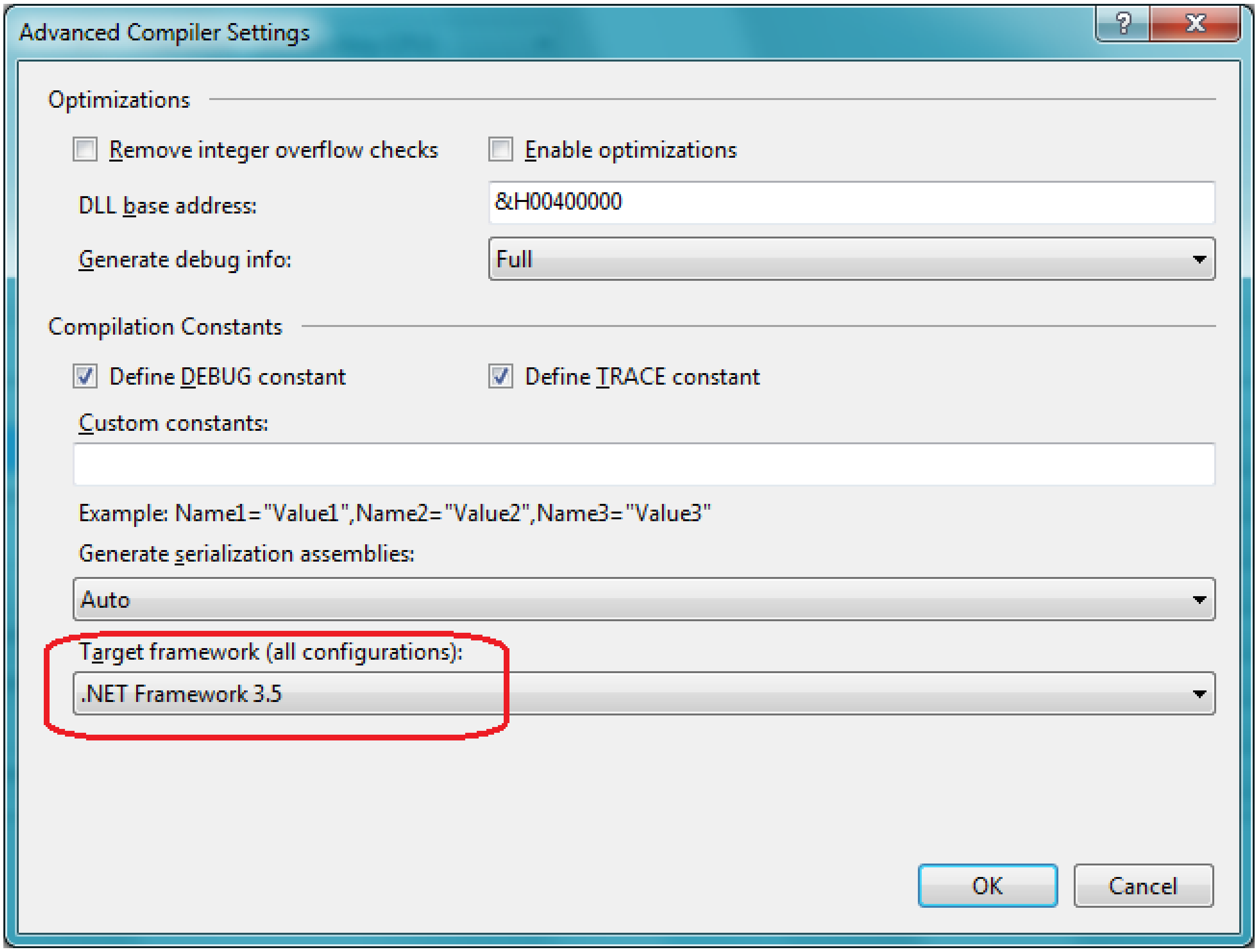
Please specify the DeviceName and DeviceClass of the driver you wish to create:

Device Class: FilterWheel

Device Name/Model: Acme

ASCOM DeviceId: ASCOM.Acme.FilterWheel

11) Set FilterWheel Properties➢Compile➢Advanced Compile Options...➢
Target framework (all configurations): **.NET Framework 3.5**
(The driver must use the same .NET Framework as the server.)



Advanced Compiler Settings

Optimizations

□ Remove integer overflow checks □ Enable optimizations

DLL base address: &H00400000

Generate debug info: Full

Compilation Constants

☑ Define DEBUG constant ☑ Define TRACE constant

Custom constants:

Example: Name1="Value1",Name2="Value2",Name3="Value3"

Generate serialization assemblies: Auto

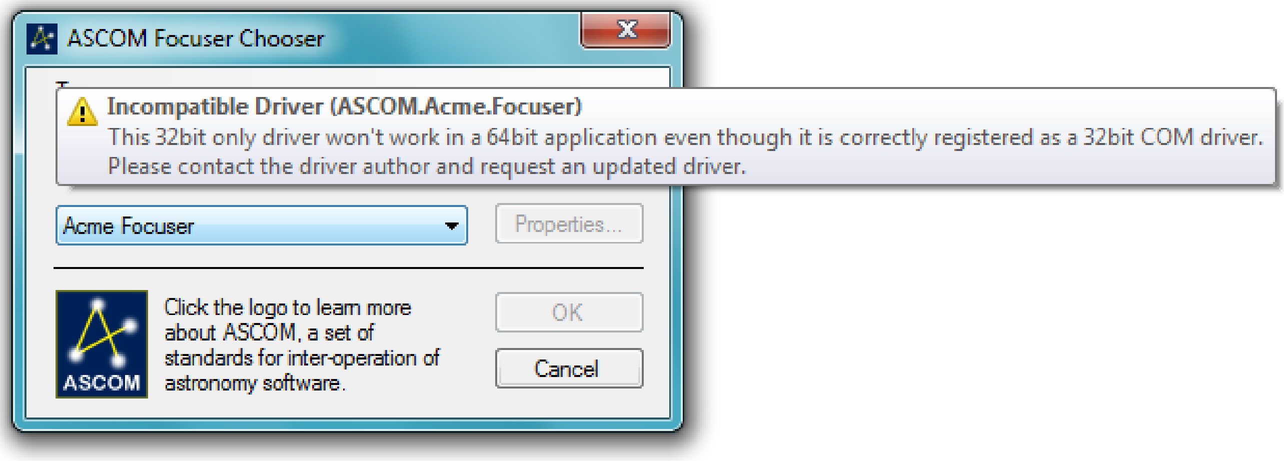
Target framework (all configurations): .NET Framework 3.5

12) Enable only 32-bit code generation for the **Release** configuration of the FilterWheel driver by modifying (outside of the Visual Basic IDE environment) the <PlatformTarget> tag in the FilterWheel driver's FilterWheel.vbproj file to be **x86** located under the following tag
 <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Release|AnyCPU' ">
 (i.e. in FilterWheel.vbproj: <PlatformTarget>**x86**</PlatformTarget>)
 (The Debug and Release configurations are not consistent in the template.)

13) Enable XML documentation for the FilterWheel driver by modifying (outside of the Visual Basic IDE environment) the empty <DocumentationFile> tags in the FilterWheel driver's FilterWheel.vbproj file to be the project's name **FilterWheel.xml** located under the following tags:
 <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Debug|AnyCPU' ">
 <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Release|AnyCPU' ">
 (i.e. in Focuser.vbproj: <DocumentationFile>**FilterWheel.xml**</DocumentationFile>)
 (no "Generate XML documentation file" option in Visual Basic 2010 Express Edition's Properties➢Compile)

14) Build the solution (F6)

15) Run the ASCOM **Conform** tool. If it is running in 64 bit mode, change it to run in 32 bit mode with:
 Options➢Conformance Options➢General➢Conform Settings: [✓] Run as 32bit on a 64bit OS
 (this is needed because in order to be served by the 32-bit LocalServer, the driver is only a 32-bit driver)

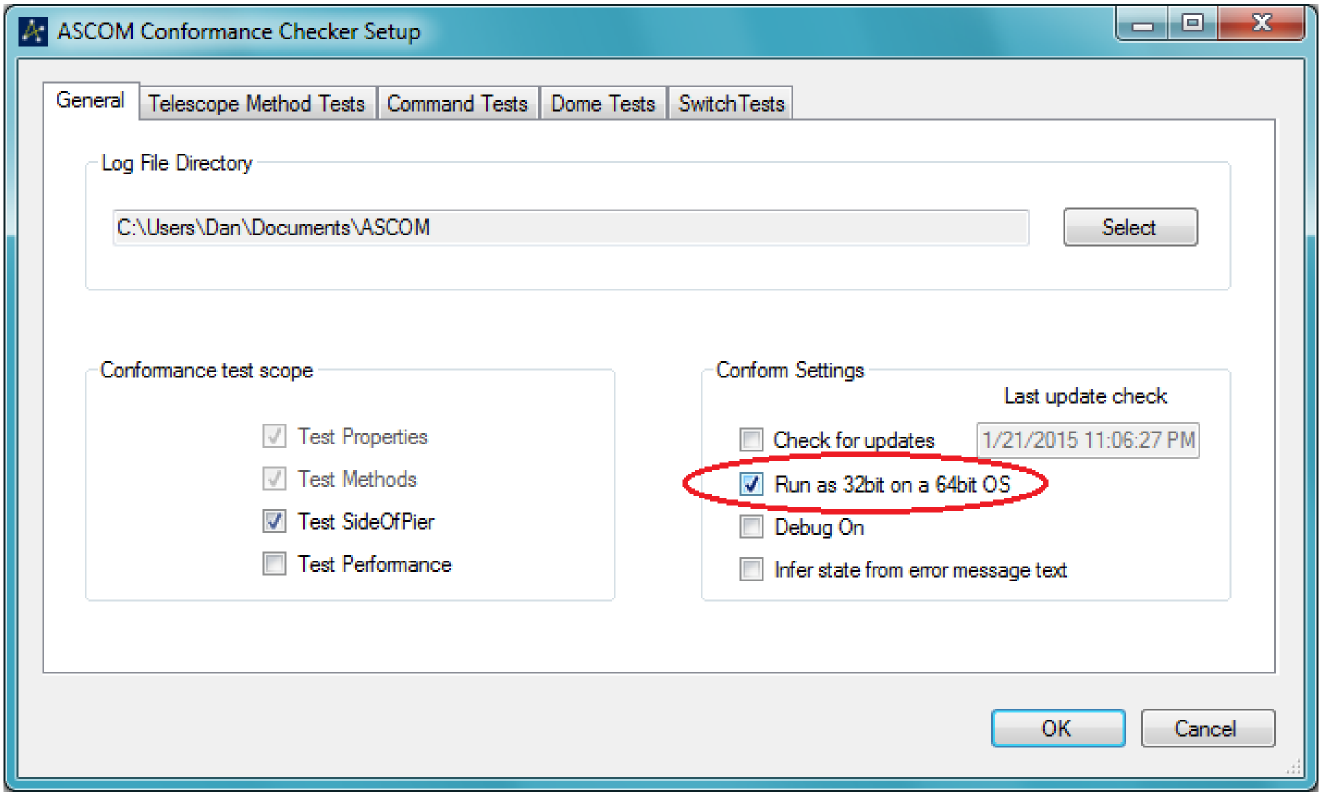


ASCOM Focuser Chooser

Incompatible Driver (ASCOM.Acme.Focuser)
This 32bit only driver won't work in a 64bit application even though it is correctly registered as a 32bit COM driver. Please contact the driver author and request an updated driver.

Acme Focuser

Click the logo to learn more about ASCOM, a set of standards for inter-operation of astronomy software.



ASCOM Conformance Checker Setup

Log File Directory

C:\Users\Dan\Documents\ASCOM

Conformance test scope

☑ Test Properties

☑ Test Methods

☑ Test SideOfPier

☐ Test Performance

Conform Settings

Last update check

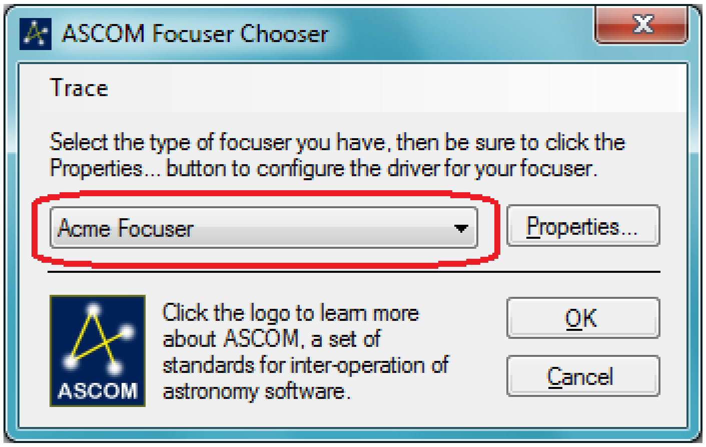
☐ Check for updates 1/21/2015 11:06:27 PM

☑ Run as 32bit on a 64bit OS

☐ Debug On

☐ Infer state from error message text

16) Using the ASCOM **Conform** tool, Options➢Check Focuser, Options➢Select Driver, select the *Acme Focuser*



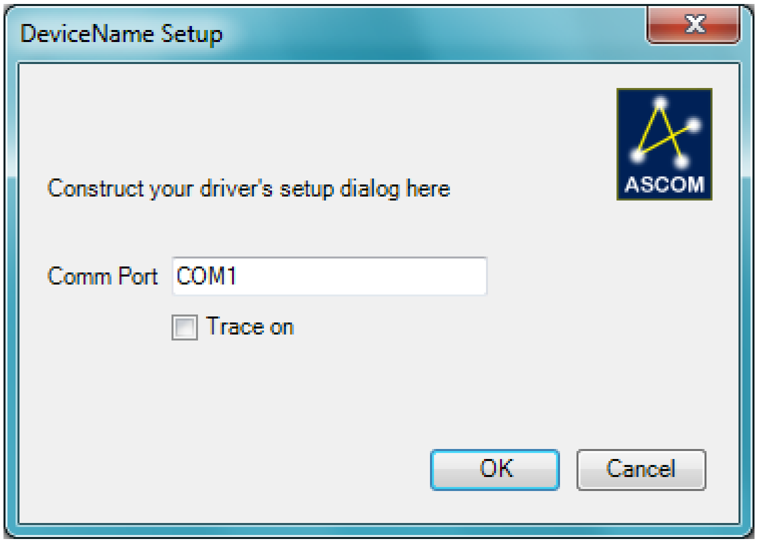
ASCOM Focuser Chooser

Select the type of focuser you have, then be sure to click the Properties... button to configure the driver for your focuser.

Acme Focuser

Click the logo to learn more about ASCOM, a set of standards for inter-operation of astronomy software.

17) Select ASCOM Focuser Chooser➢Properties... to get the Acme DeviceName dialog for the Focuser

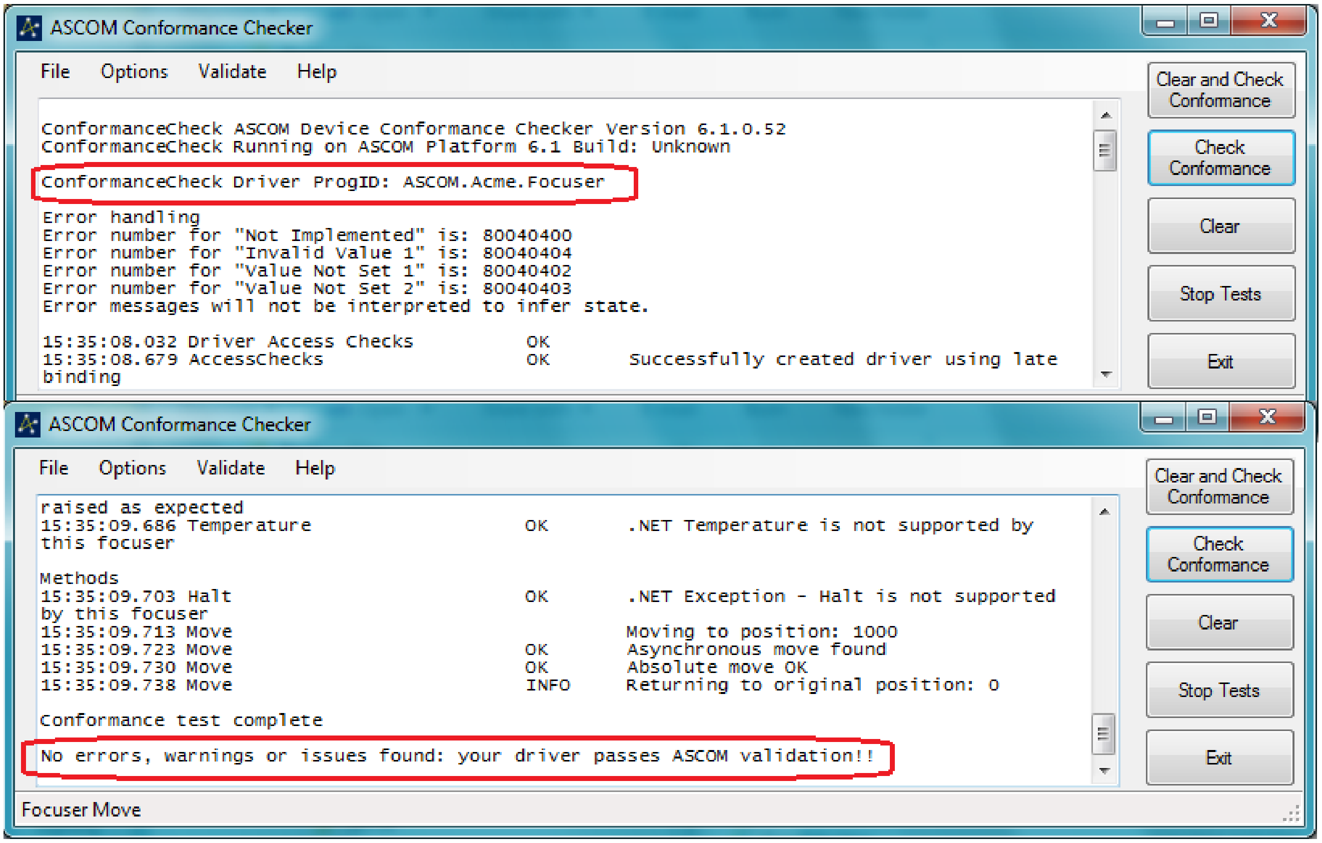


Construct your driver's setup dialog here

Comm Port COM1

□ Trace on

18) Run the Check Conformance and verify that no errors, warnings or issues are found and the Focuser driver passes ASCOM validation!!

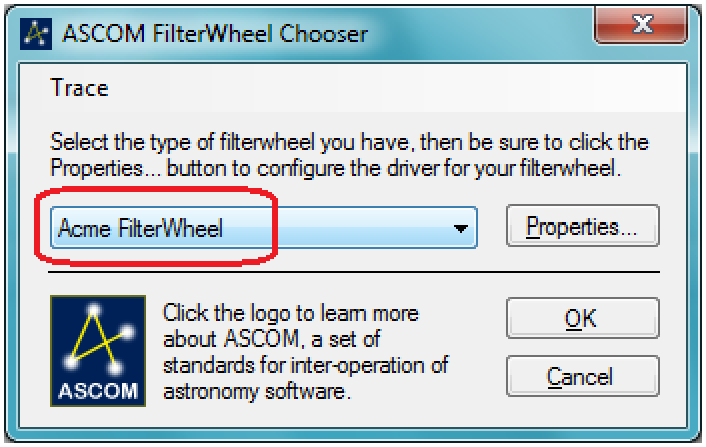


```
ConformanceCheck ASCOM Device Conformance Checker Version 6.1.0.52
ConformanceCheck Running on ASCOM Platform 6.1 Build: Unknown

ConformanceCheck Driver ProgID: ASCOM.Acme.Focuser

Error handling
Error number for "Not Implemented" is: 80040400
Error number for "Invalid Value 1" is: 80040404
Error number for "Value Not Set 1" is: 80040402
Error number for "Value Not Set 2" is: 80040403
Error messages will not be interpreted to infer state.

15:35:08.032 Driver Access Checks          OK
15:35:08.679 AccessChecks                  OK       Successfully created driver using late
binding
```

```
raised as expected
15:35:09.686 Temperature                   OK       .NET Temperature is not supported by
this focuser

Methods
15:35:09.703 Halt                          OK       .NET Exception - Halt is not supported
by this focuser
15:35:09.713 Move                                   Moving to position: 1000
15:35:09.723 Move                          OK       Asynchronous move found
15:35:09.730 Move                          OK       Absolute move OK
15:35:09.738 Move                          INFO     Returning to original position: 0

Conformance test complete

No errors, warnings or issues found: your driver passes ASCOM validation!!
```

Focuser Move

19) Using the ASCOM **Conform** tool, Options➢Check Filter Wheel, Options➢Select Driver, select the *Acme FilterWheel*



ASCOM FilterWheel Chooser

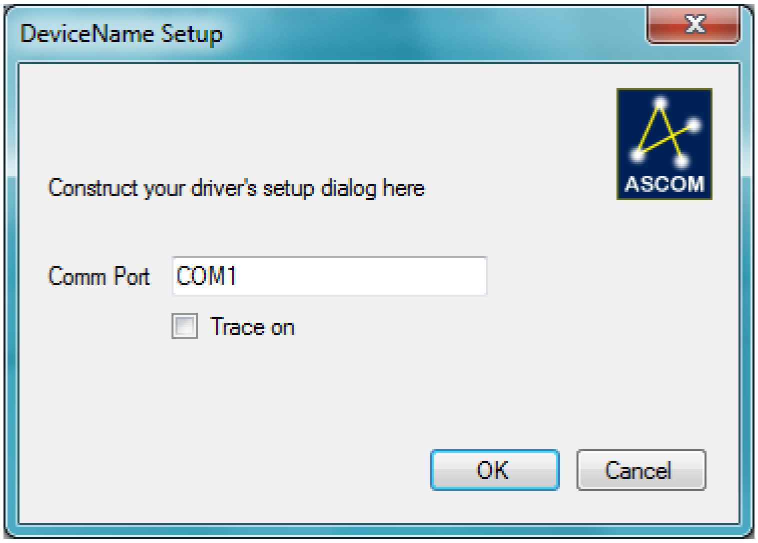
Trace

Select the type of filterwheel you have, then be sure to click the Properties... button to configure the driver for your filterwheel.

Acme FilterWheel

Click the logo to learn more about ASCOM, a set of standards for inter-operation of astronomy software.

20) Select ASCOM FilterWheel Chooser➢Properties... to get the DeviceName Setup dialog for the FilterWheel



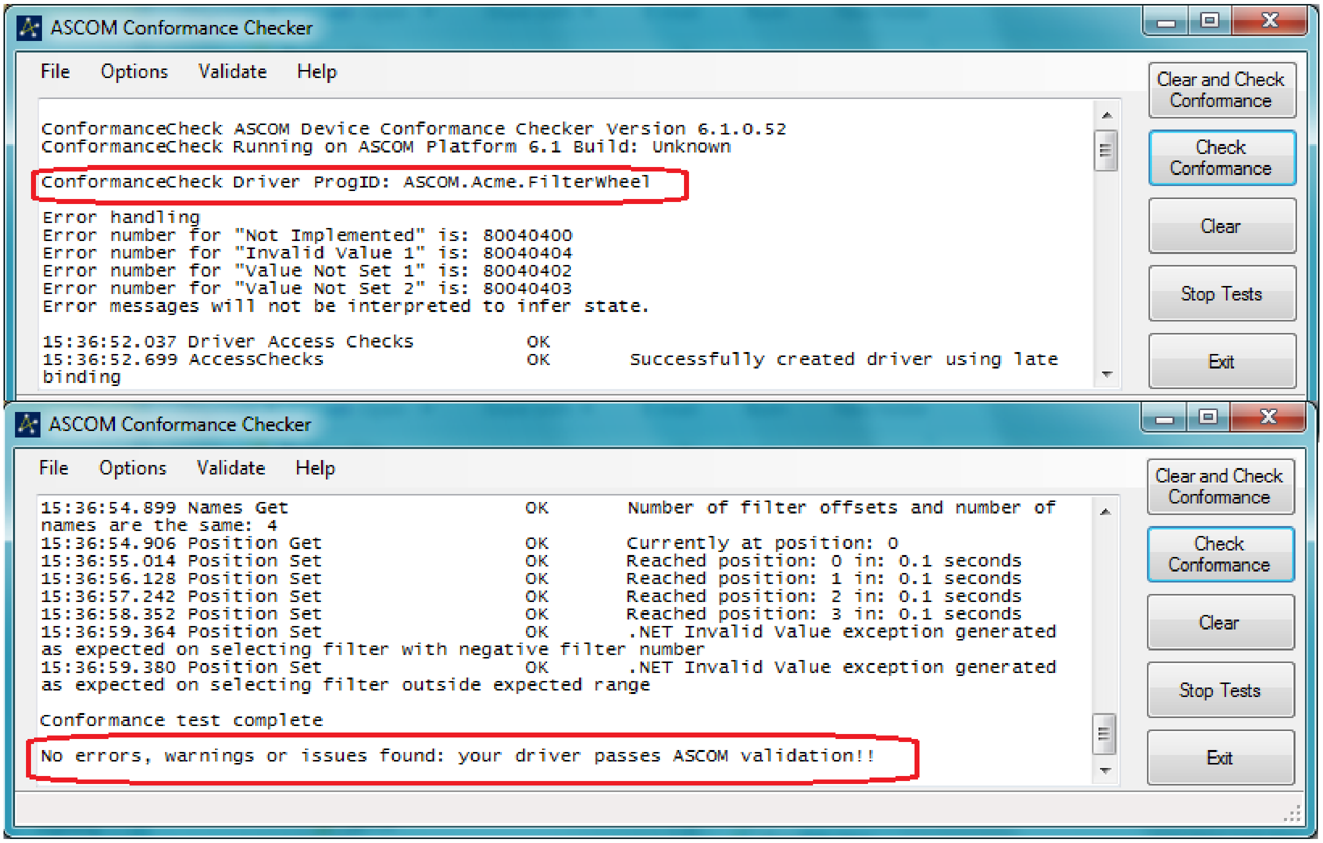
DeviceName Setup

Construct your driver's setup dialog here

Comm Port COM1

□ Trace on

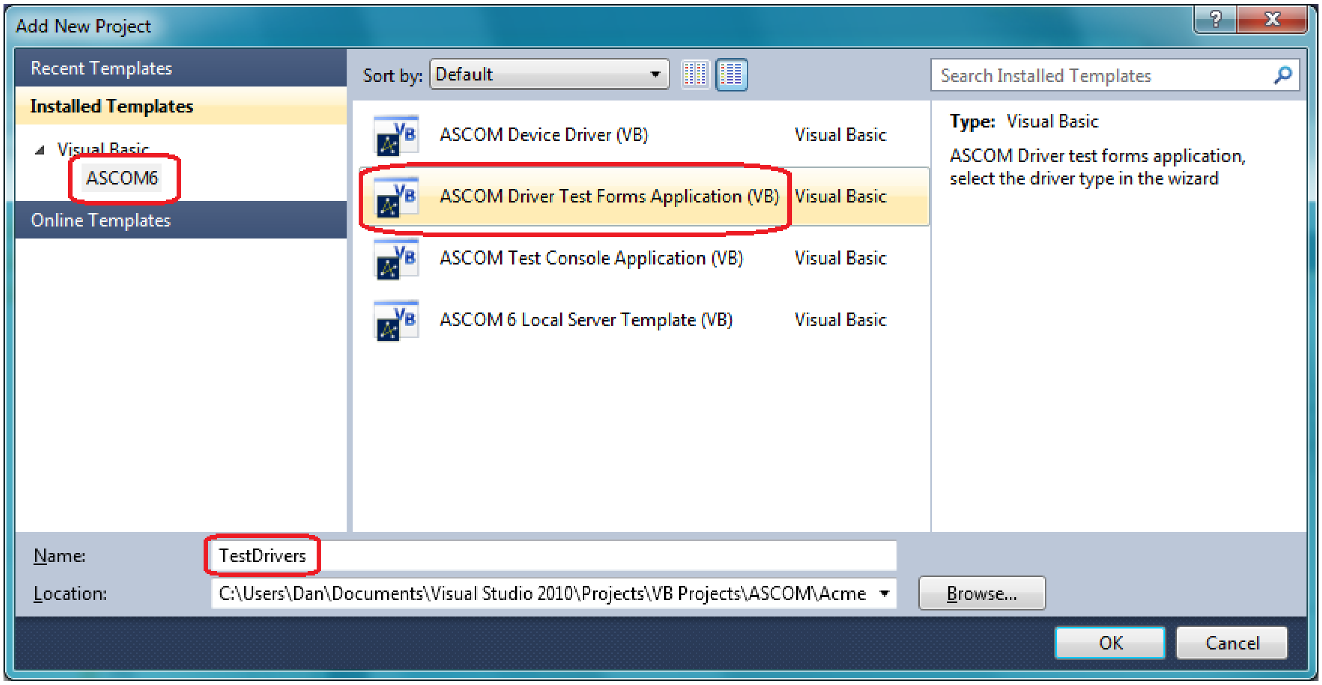
21) Run the Check Conformance and verify that no errors, warnings or issues are found and the FilterWheel driver passes ASCOM validation!!



```
ConformanceCheck ASCOM Device Conformance Checker Version 6.1.0.52
ConformanceCheck Running on ASCOM Platform 6.1 Build: Unknown
ConformanceCheck Driver ProgID: ASCOM.Acme.FilterWheel

Error handling
Error number for "Not Implemented" is: 80040400
Error number for "Invalid Value 1" is: 80040404
Error number for "Value Not Set 1" is: 80040402
Error number for "Value Not Set 2" is: 80040403
Error messages will not be interpreted to infer state.

15:36:52.037 Driver Access Checks          OK
15:36:52.699 AccessChecks                  OK       Successfully created driver using late
binding
```

```
15:36:54.899 Names Get                     OK       Number of filter offsets and number of
names are the same: 4
15:36:54.906 Position Get                  OK       Currently at position: 0
15:36:55.014 Position Set                  OK       Reached position: 0 in: 0.1 seconds
15:36:56.128 Position Set                  OK       Reached position: 1 in: 0.1 seconds
15:36:57.242 Position Set                  OK       Reached position: 2 in: 0.1 seconds
15:36:58.352 Position Set                  OK       Reached position: 3 in: 0.1 seconds
15:36:59.364 Position Set                  OK       .NET Invalid Value exception generated
as expected on selecting filter with negative filter number
15:36:59.380 Position Set                  OK       .NET Invalid Value exception generated
as expected on selecting filter outside expected range

Conformance test complete
No errors, warnings or issues found: your driver passes ASCOM validation!!
```

22) Add New Project for application for testing the drivers
- File➢Add➢New Project...
- **ASCOM Driver Test Forms Application (VB)**
- Name: **TestDrivers**



ASCOM Device Driver (VB)	Visual Basic
ASCOM Driver Test Forms Application (VB)	Visual Basic
ASCOM Test Console Application (VB)	Visual Basic
ASCOM 6 Local Server Template (VB)	Visual Basic

Type: Visual Basic

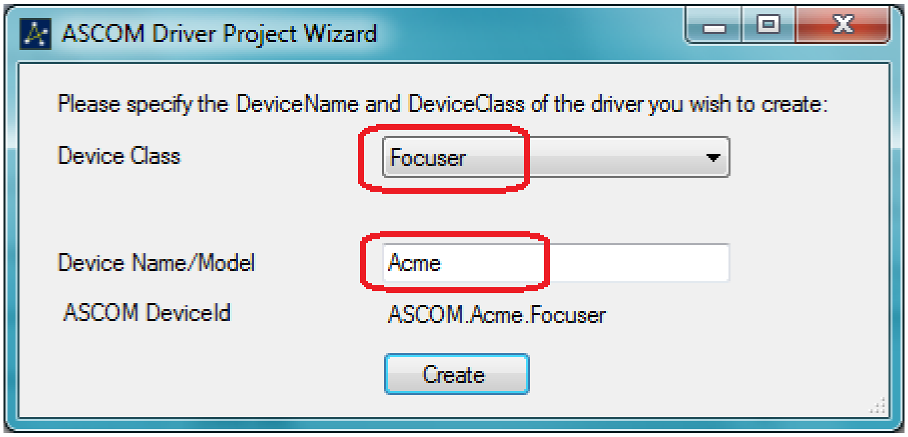
ASCOM Driver test forms application, select the driver type in the wizard

Name: TestDrivers

Location: C:\Users\Dan\Documents\Visual Studio 2010\Projects\VB Projects\ASCOM\Acme

23) ASCOM Driver Project Wizard

- Device Class - **Focuser**
- Device Name/Model - **Acme**
- Create



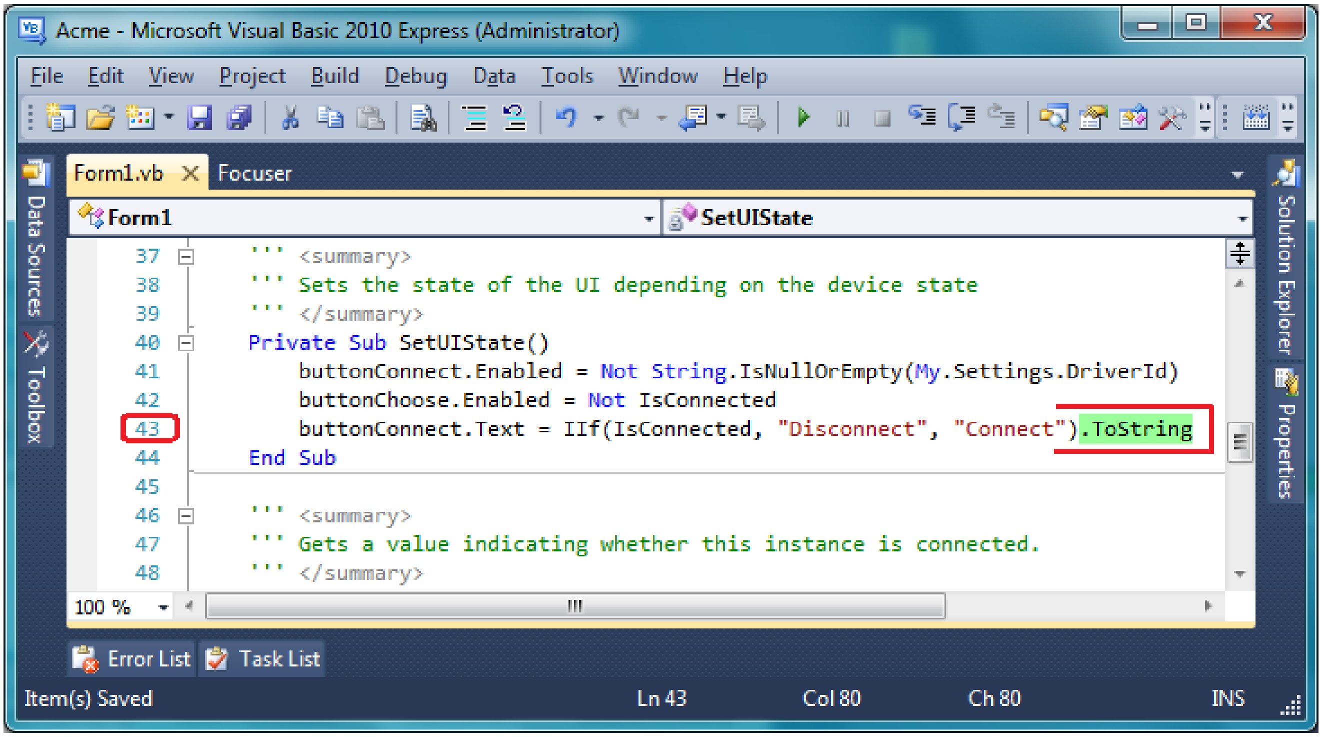
Please specify the DeviceName and DeviceClass of the driver you wish to create:

Device Class: Focuser

Device Name/Model: Acme

ASCOM DeviceId: ASCOM.Acme.Focuser

24) If Option Strict is turned on, remove the following error:
On Form1.vb, line 43, add **.ToString** to the end of the line
(This is an issue with the VB Driver Test Forms Application template.)

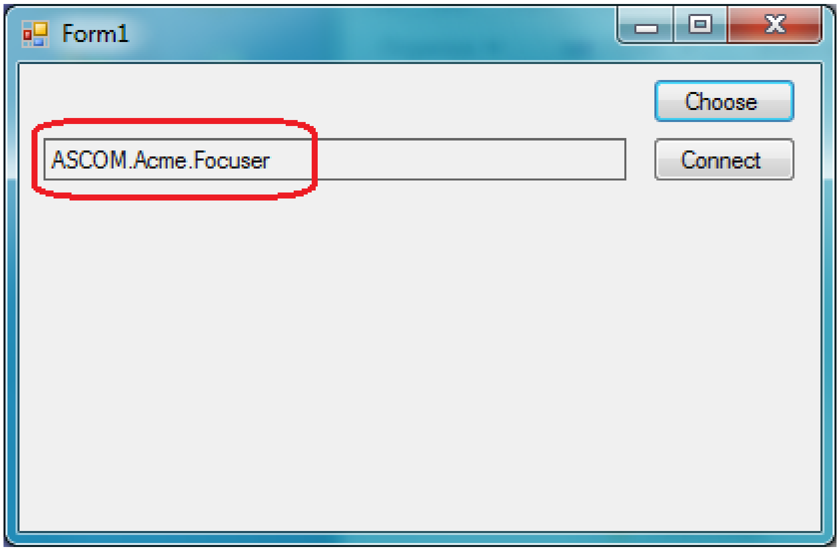


```
37      ''' <summary>
38      ''' Sets the state of the UI depending on the device state
39      ''' </summary>
40      Private Sub SetUIState()
41          buttonConnect.Enabled = Not String.IsNullOrEmpty(My.Settings.DriverId)
42          buttonChoose.Enabled = Not IsConnected
43          buttonConnect.Text = IIf(IsConnected, "Disconnect", "Connect").ToString
44      End Sub
45
46      ''' <summary>
47      ''' Gets a value indicating whether this instance is connected.
48      ''' </summary>
```

25) Right-Click Solution Explorer➢TestDrivers project➢Set as Startup Project to set the **TestDrivers** project as the startup project

26) Build the solution

27) Run the code (F5), click the test form's Choose button, select the *Acme Focuser*, select ASCOM Focuser Chooser➢Properties... to get the DeviceName Setup dialog for the Focuser, OK those dialogs and verify the ASCOM.Acme.Focuser is shown on the test form.



ASCOM.Acme.Focuser

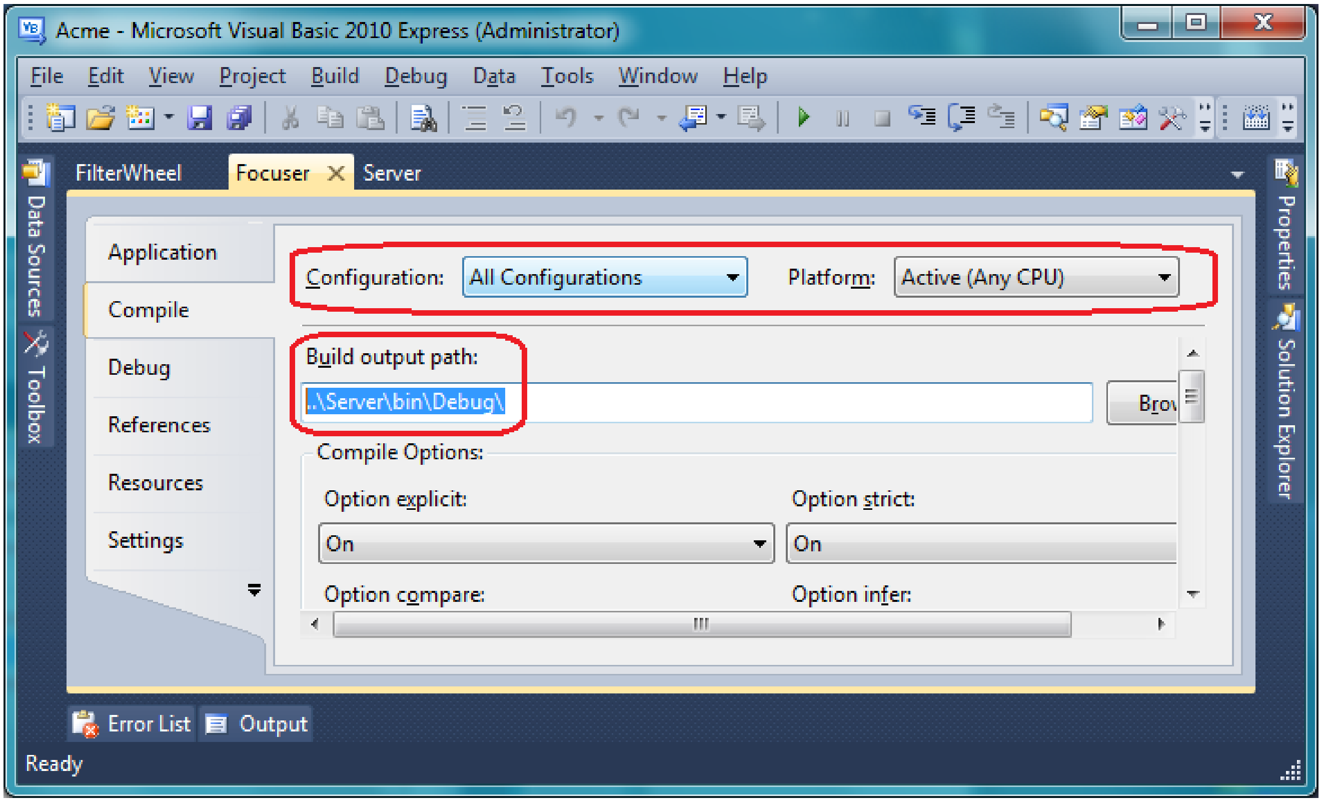
At this point, additional code can be added to the separate Focuser and FilterWheel drivers to independently control the Focuser and FilterWheel hardware and additional code and controls can be added to the TestDrivers project to exercise and debug the features of the in-proc DLL Focuser and FilterWheel drivers.

28) Clean the solution with Build➢Clean Solution so that the driver will be automatically unregistered from COM and ASCOM
(if menu Build➢Clean Solution is not shown, use Tools➢Customize➢Commands➢Menu bar: Build➢Add Command...➢Categories: Build, Commands: Clean Solution, OK, Close to add that menu item)

At this point, the Acme Focuser and FilterWheel should no longer be available in **Conform**'s Select Driver.

Now, make the changes to incorporate the **LocalServer** functionality.

29) In Focuser Properties➢Compile➢
 - Set Configuration: **All Configurations**, Platform: **Active (Any CPU)**
 - Set Output➢Output path: **..\Server\bin\Debug\** or use **..\Server\bin\Debug**
 - Save project



Configuration: All Configurations   Platform: Active (Any CPU)

Build output path:
..\Server\bin\Debug\

Compile Options:

Option explicit: On   Option strict: On

Option compare: Option infer:

30) Disable Register for COM interop for the Focuser driver by modifying (outside of the Visual Basic IDE environment) the <RegisterForComInterop> tag in the Focuser driver's Focuser.vbproj file to be **false** located under the following tag: (default was "true")
 <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Debug|AnyCPU' ">
 (i.e. in Focuser.vbproj: <RegisterForComInterop>**false**</RegisterForComInterop>)

31) Right-Click Focuser project➢Add Reference...➢Projects➢Server to add a reference to the Server project to the Focuser Project

32) Add the following class to the Focuser project's Driver.vb file just before the Focuser class definition:

```
Friend Class FocuserLocalServerConstants
    Friend Const DRIVER_ID As String = "ASCOM.Acme.Focuser";
    Friend Const DRIVER_DESCRIPTION As String = "Acme Focuser";
End Class
```

This provides a single instance of Focuser constants to decorate the Focuser class and for use inside the Focuser class following the DRY principle (Don't Repeat Yourself - ref. Tim Long).

33) Add the following attribute declarations to the Focuser project's Driver.vb Focuser class definition:

```
<ProgId(FocuserLocalServerConstants.DRIVER_ID)> _
<ServedClassName(FocuserLocalServerConstants.DRIVER_DESCRIPTION)> _
```

(The server uses this to identify this driver as a driver to be served.)

34) Change the Focuser project's Driver.vb Focuser class definition to inherit *ReferenceCountedObjectBase*:

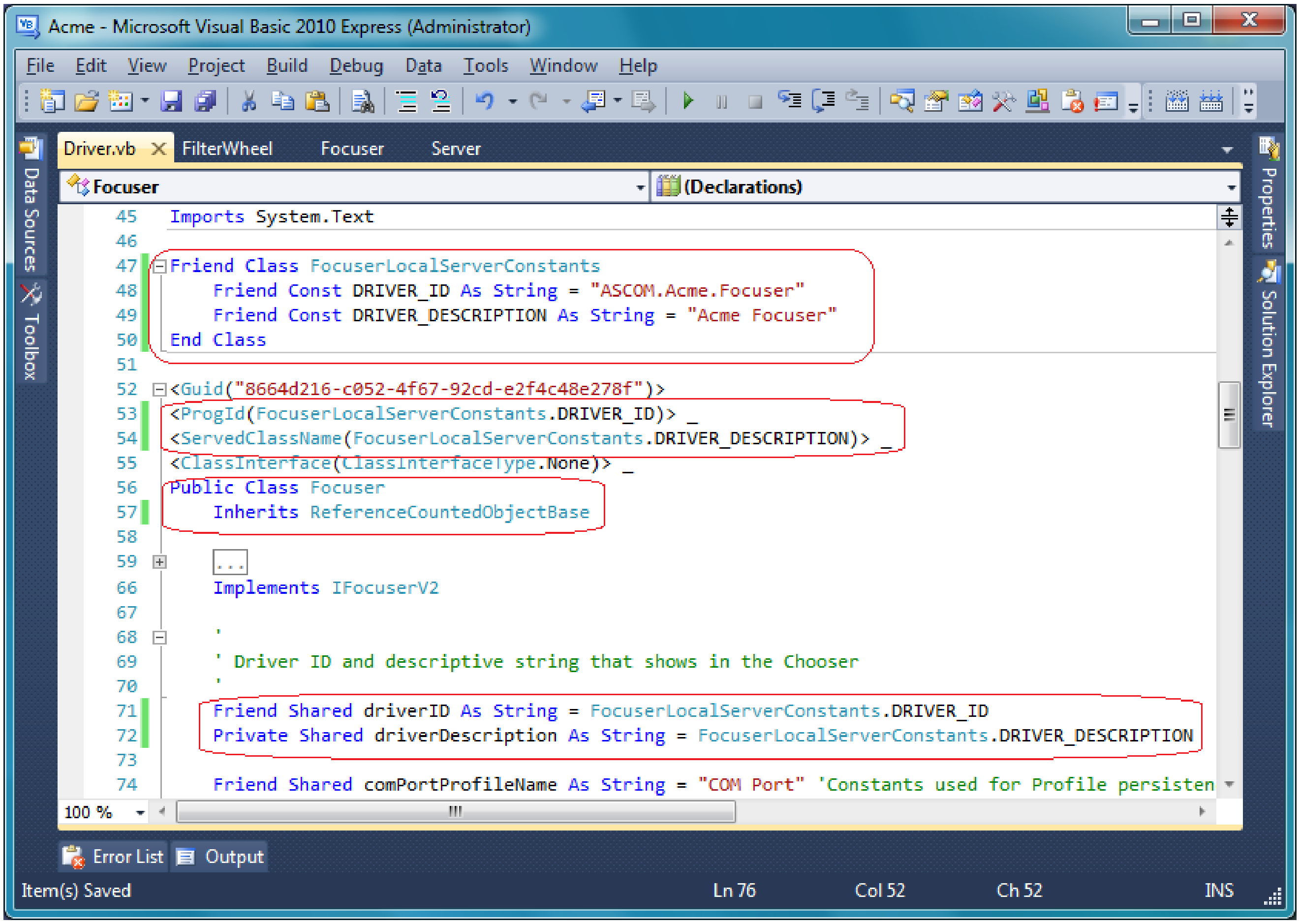
```
Public Class Focuser
       Inherits ReferenceCountedObjectBase
```

35) Change the Focuser project's Driver.vb *driverID* definition to:

```
    Friend Shared driverID As String = FocuserLocalServerConstants.DRIVER_ID;
```

36) Change the Focuser project's Driver.vb *driverDescription* definition to:

```
  Private Shared driverDescription As String =
                                    FocuserLocalServerConstants.DRIVER_DESCRIPTION;
```



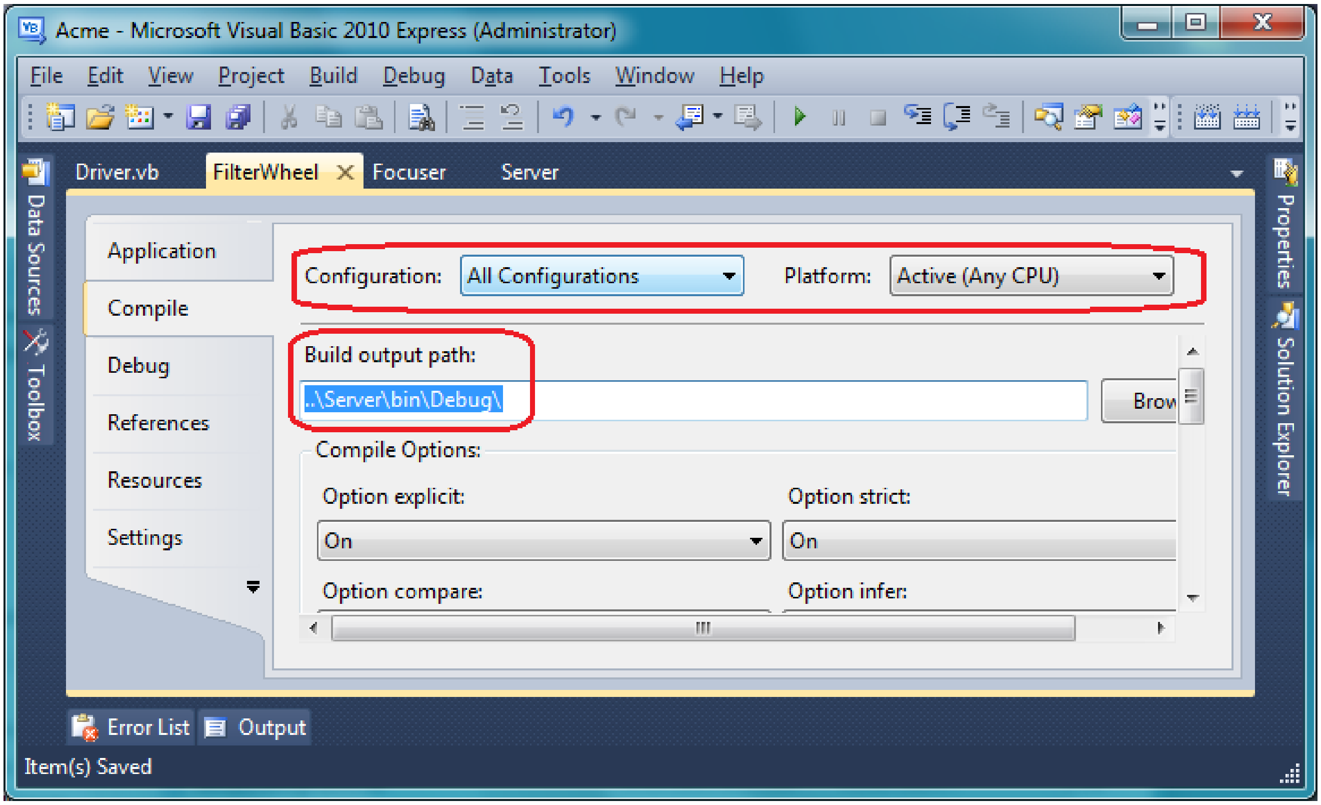
```
Imports System.Text

Friend Class FocuserLocalServerConstants
    Friend Const DRIVER_ID As String = "ASCOM.Acme.Focuser"
    Friend Const DRIVER_DESCRIPTION As String = "Acme Focuser"
End Class

<Guid("8664d216-c052-4f67-92cd-e2f4c48e278f")>
<ProgId(FocuserLocalServerConstants.DRIVER_ID)> _
<ServedClassName(FocuserLocalServerConstants.DRIVER_DESCRIPTION)> _
<ClassInterface(ClassInterfaceType.None)> _
Public Class Focuser
    Inherits ReferenceCountedObjectBase

    ...
    Implements IFocuserV2

    '
    ' Driver ID and descriptive string that shows in the Chooser
    '
    Friend Shared driverID As String = FocuserLocalServerConstants.DRIVER_ID
    Private Shared driverDescription As String = FocuserLocalServerConstants.DRIVER_DESCRIPTION

    Friend Shared comPortProfileName As String = "COM Port" 'Constants used for Profile persisten
```

37) Remove the Focuser project's Driver.vb ASCOM registration region code

This completes the basic changes to the Focuser driver to be able to be served by an ASCOM LocalServer.

38) In FilterWheel Properties➢Compile➢

- Set Configuration: **All Configurations**, Platform: **Active (Any CPU)**
- Set Output➢Output path: **..\Server\bin\Debug\**   or use **..\Server\bin\Debug**
- Save project



Configuration: All Configurations   Platform: Active (Any CPU)

Build output path:

..\Server\bin\Debug\

Compile Options:

Option explicit: On   Option strict: On

Option compare: Option infer:

Item(s) Saved

39) Disable Register for COM interop for the Focuser driver by modifying (outside of the Visual Basic IDE environment) the <RegisterForComInterop> tag in the Focuser driver's Focuser.vbproj file to be **false** located under the following tag: (default was "true")
  <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Debug|AnyCPU' ">
 (i.e. in Focuser.vbproj: <RegisterForComInterop>**false**</RegisterForComInterop>)

40) Right-Click FilterWheel Project➢Add Reference...➢Projects➢Server to add a reference to the Server project to the FilterWheel project

41) Add the following class to the FilterWheel project's Driver.vb file just before the FilterWheel class definition:

```
Friend Class FilterWheelLocalServerConstants
    Friend Const DRIVER_ID As String = "ASCOM.Acme.FilterWheel";
    Friend Const DRIVER_DESCRIPTION As String = "Acme FilterWheel";
End Class
```

This provides a single instance of FilterWheel constants to decorate the FilterWheel class and for use inside the FilterWheel class following the DRY principle (Don't Repeat Yourself - ref. Tim Long).

42) Add the following attribute declarations to the FilterWheel project's Driver.vb FilterWheel class definition:

```
<ProgId(FilterWheelLocalServerConstants.DRIVER_ID)> _
<ServedClassName(FilterWheelLocalServerConstants.DRIVER_DESCRIPTION)> _
```

(The server uses this to identify this driver as a driver to be served.)

43) Change the FilterWheel project's Driver.vb FilterWheel class definition to inherit *ReferenceCountedObjectBase*:

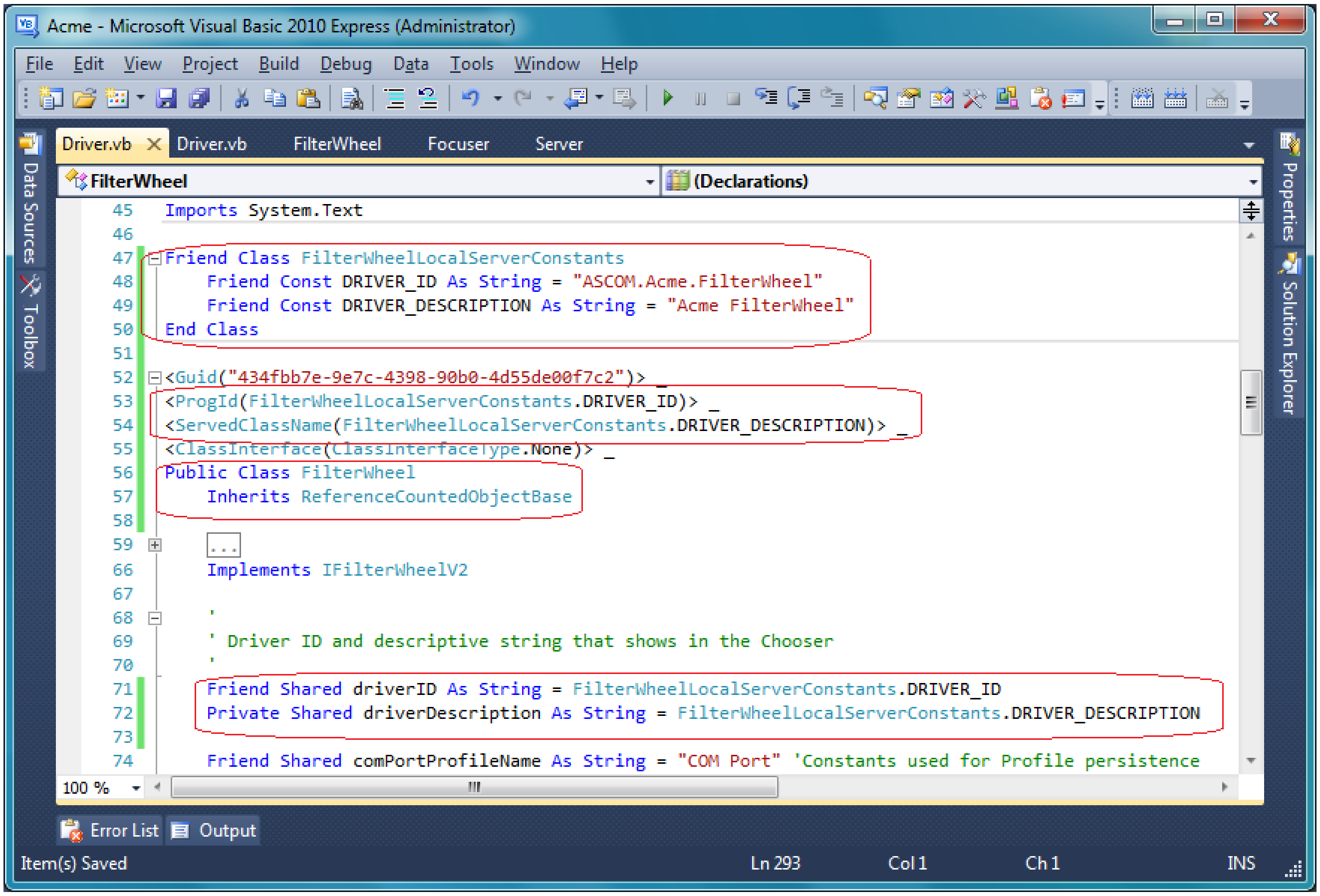
```
Public Class FilterWheel
     Inherits ReferenceCountedObjectBase
```

44) Change the FilterWheel project's Driver.vb *driverID* definition to:

```
Friend Shared driverID As String = FilterWheelLocalServerConstants.DRIVER_ID;
```

45) Change the FilterWheel project's Driver.vb *driverDescription* definition to:

```
Private Shared driverDescription As String =
                                        FilterWheelLocalServerConstants.DRIVER_DESCRIPTION;
```

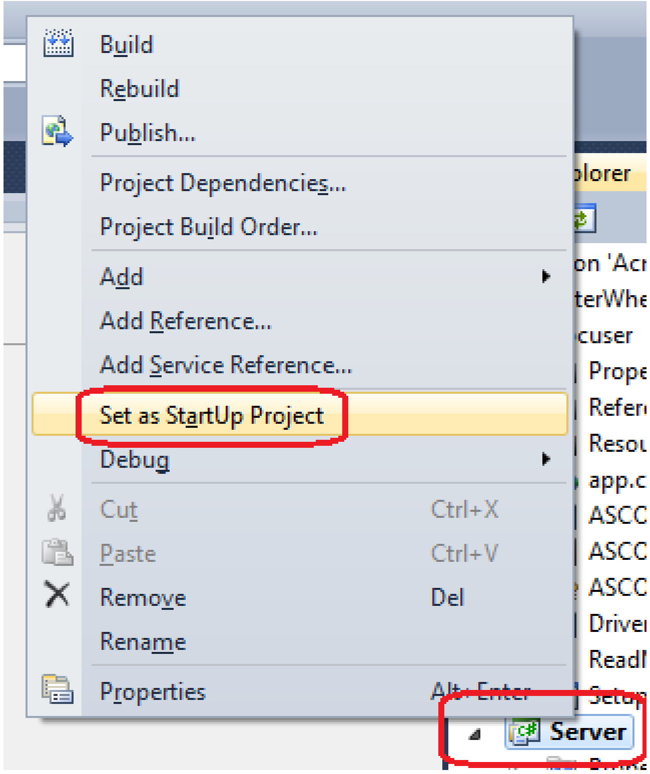


```
Imports System.Text

Friend Class FilterWheelLocalServerConstants
    Friend Const DRIVER_ID As String = "ASCOM.Acme.FilterWheel"
    Friend Const DRIVER_DESCRIPTION As String = "Acme FilterWheel"
End Class

<Guid("434fbb7e-9e7c-4398-90b0-4d55de00f7c2")> _
<ProgId(FilterWheelLocalServerConstants.DRIVER_ID)> _
<ServedClassName(FilterWheelLocalServerConstants.DRIVER_DESCRIPTION)> _
<ClassInterface(ClassInterfaceType.None)> _
Public Class FilterWheel
    Inherits ReferenceCountedObjectBase

    ...
    Implements IFilterWheelV2

    '
    ' Driver ID and descriptive string that shows in the Chooser
    '
    Friend Shared driverID As String = FilterWheelLocalServerConstants.DRIVER_ID
    Private Shared driverDescription As String = FilterWheelLocalServerConstants.DRIVER_DESCRIPTION

    Friend Shared comPortProfileName As String = "COM Port" 'Constants used for Profile persistence
```

46) Remove the FilterWheel project's Driver.vb ASCOM registration region code

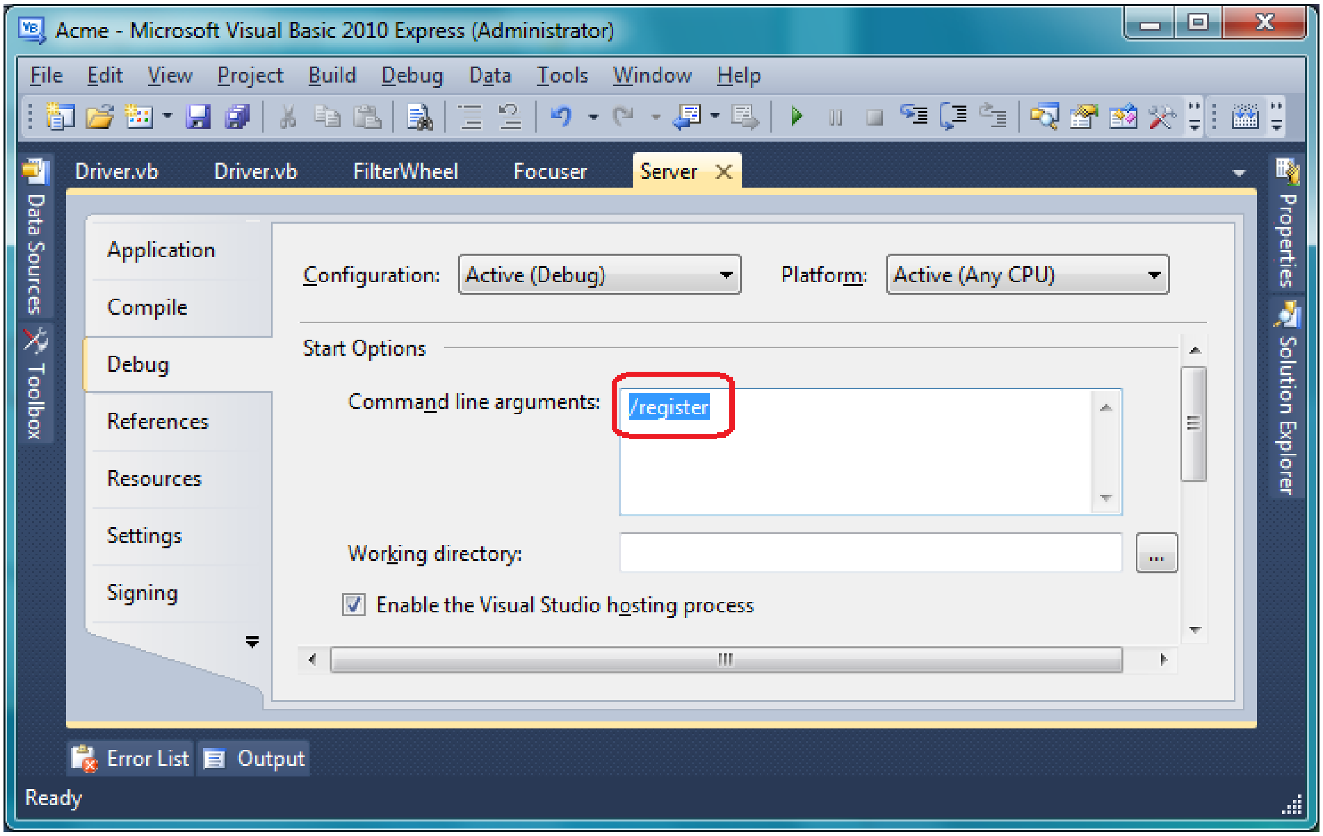
This completes the basic changes to the FilterWheel driver to be able to be served by an ASCOM LocalServer.

47) Right-Click Solution Explorer➢Server project➢Set as Startup Project to set the local server as the startup project



48) Build the solution

49) Add Server Properties➢Debug➢Start Options➢Command line arguments: **/register**


Acme - Microsoft Visual Basic 2010 Express (Administrator)
File Edit View Project Build Debug Data Tools Window Help
Driver.vb Driver.vb FilterWheel Focuser Server
Application
Compile
Debug
References
Resources
Settings
Signing
Data Sources
Toolbox
Properties
Solution Explorer
Error List
Output
Ready

Configuration: Active (Debug)   Platform: Active (Any CPU)

Start Options

Command line arguments: /register

Working directory:

☑ Enable the Visual Studio hosting process

50) Run the project (to have the local server register the drivers with COM and ASCOM)
(this registers the drivers for both 32 bit and 64 bit {on a 64-bit machine}, so the ASCOM Conform tool can now be run as 64 bits without problems)

51) Using the ASCOM Conform tool, Options➢Check Focuser, Options➢Select Driver, select the Acme Focuser

52) Select ASCOM Focuser Chooser➢Properties... to get the DeviceName Setup dialog for the Focuser

53) Run the Check Conformance and verify that no errors, warnings or issues are found and the ASCOM.Acme.Focuser driver passes ASCOM validation!!

54) Using the ASCOM Conform tool, Options➢Check Filter Wheel, Options➢Select Driver, select the Acme FilterWheel

55) Select ASCOM FilterWheel Chooser➢Properties... to get the DeviceName Setup dialog for the FilterWheel

56) Run the Check Conformance and verify that no errors, warnings or issues are found and the ASCOM.Acme.FilterWheel driver passes ASCOM validation!!

57) Change Server Properties➢Debug➢Start Options➢Command line arguments: **/unregister**

58) Run the project (to have the local server unregister the drivers with COM and ASCOM)

At this point, code can be changed in the Focuser and FilterWheel drivers to appropriately work with the common Focuser and FilterWheel hardware and additional code and controls can be added to the TestDrivers project to exercise and debug the features of the LocalServer-served Focuser and FilterWheel drivers.

When the ASCOM local server and drivers development is complete, the ASCOM Driver Install Script Generator an be used to generate an Inno Setup script to generate a Windows setup executable that can be used to distribute the server and drivers just developed.
Note: The server/driver solution should be closed in the IDE before running the Inno Setup compiler.