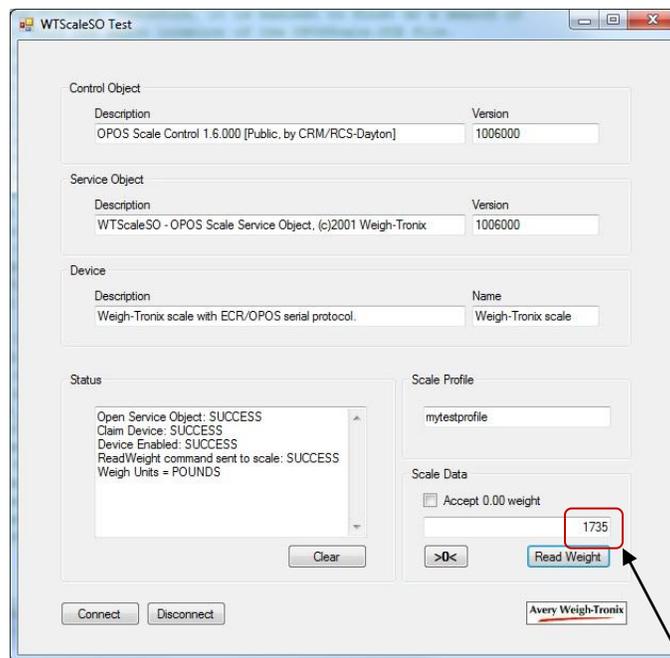


## About WTSaleSO Test Program

The **WTSaleSO Test Program** is a simple test program used to verify the proper installation and operation of the Avery Weigh-Tronix legacy **WTSaleSO.DLL** ActiveX component for OPOS with your scale. As a courtesy to its customers, Avery Weigh-Tronix provides this test program along with the complete source code used to build the test application.

The source code and tips will help developers integrate the legacy ActiveX driver (originally written in VB6 on Windows Me) into new environments using Visual Studio 2008, C#, and .NET Framework on Windows 7 (32/64-bit).



1.735 pounds

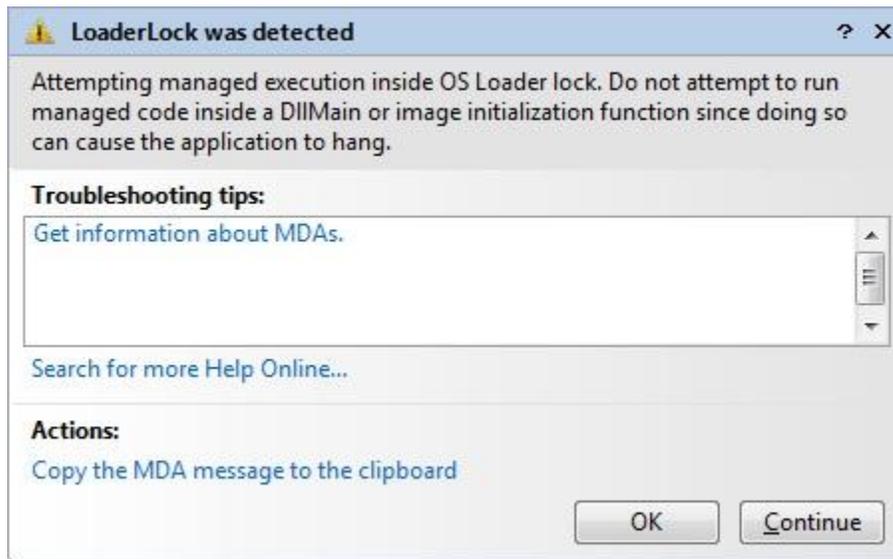
Please see the PROGRAMMERS NOTES section in the MAIN.cs file for more details.

## TIP: "Class not registered..." Error

To prevent the class not registered error, when in the debug mode, make sure to set Platform Target: x86 in Project Properties - Build – General.

## TIP: "LoaderLock Detected" Warning

Under some conditions, the following warning message could be displayed when running the test program in the debug mode:



A **Loader lock** is one of the Managed Debugging Assistants (MDAs) that were added to VS2005 (.NET Framework 2.0) to help find hard to debug runtime issues. It **ONLY** affects code run under the debugger (i.e. it won't happen when users run your EXE).

This warning message is not critical. To prevent this error from stopping your code, you can disable LoaderLock as follows:

From the main **Debug** menu, select the **Exceptions...** sub-menu.

On the Exceptions window, click on the '+' button next to **Managed Debugging Assistants** to open the tree node. Then, uncheck **LoaderLock** and click the **OK** button to save the change.

**Note:** You only want this setting to be disabled when building applications using WTScaleSO. So, please remember to turn this back on when you have other scenarios that may need this warning to be fired.

## Basic Operation

The test program features have been purposely kept to a minimum. In order to run the program you must have previously run the **WTScaleAdmin** program and added a valid *scale profile*. The scale profile name must be entered in the test program. The profile name is used by the CCO to access scale serial communications properties such as COM port, baud rate etc from the system registry.

Then, you can click on the **Connect** button. This button is used to perform the **Open, Claim and Enable** device functions that are necessary before you can access weight data from the scale.

To access the scale weight data click the **Read Weight** button. A single reading will be returned and displayed if the weight on the scale is positive and stable. Otherwise, the read will timeout (3 seconds in the test program).

Similarly, you may zero the scale by clicking on the **>0<** button. If the weight is stable and within the zeroing range of the scale, then the scale will be zeroed. Since, only one command can be sent manually at a time, it is necessary to follow the zero command by a request to ReadWeight to see the updated weight displayed.

Note: the legacy OPOS specification did not support returning a **0.00** weight value. Avery Weigh-Tronix added a 'vendor-specific' optional command that does allow that value to be returned.

## General Development Notes and Tips

### 1).NET Framework 2.0:

This test application originally used .NET Framework v2.0 but you can use any version.

### 2)ClickOnce:

This test application was originally deployed using ClickOnce technology for convenience.

### 3)Desktop Shortcut:

ClickOnce technology (on .NET Framework v2.0) does not support placing a shortcut to the application on the desktop at install time. So, a CreateDesktopShortcut() function was added to do this. If you use newer .NET Frameworks that do support desktop icons, then you can omit this function and related namespaces.

### 4)To Add a Reference to the OPOSScale.ocx CCO:

- \* Right-Click on References in the Solution Explorer.
- \* Click on 'Add Reference...'
- \* Click on the Browse tab.
- \* Browse to find the OPOSScale.OCX file.

On Windows 7 it is usually located at c:\Windows\SysWoW64\OPOSScale.OCX but it could be located in the c:\Program Files (x86)\Avery Weigh-Tronix\.. folder or one of its sub-folders. Therefore, it is easiest to first do a search of

your computer to find the exact location of the OPOSScale.OCX file.

- \* Click OK to finish.

- \* OposScale\_1\_6\_Lib should now be listed in the References list in Solution Explorer.

#### 5) Referencing the CCO in C#:

Add the namespace reference at the top of the program as follows:

```
using OposScale_1_6_Lib;
```

#### 6) Declaring Instances of the CCO:

Since the CCO has a constructor, you may now declare an instance of the OPOSScaleClass as you would any other class.

#### 7) To add the CCO ActiveX control to the Toolbox:

Alternatively, you can add the CCO to the toolbox. This CCO does not have a user interface for setting properties etc, but it can be used as any other control by dragging from the toolbox to a form and then setting properties in the Properties window.

- \* Click on the Toolbox tab to open the Toolbox window.

- \* Right-Click on the 'All Windows Forms' section header of the Toolbox window.

- \* Click on 'Choose Items...'

- \* In the Choose Toolbox Items window, click on the COM Components tab.

- \* Scroll down and check the box: OPOS Scale Control 1.6.000 [Public, by CRM/RCS-Dayton]

- \* Click OK.

- \* The OPOS Scale Control 1.6.000 should now be added to the Toolbox.

Note: If it does not show up, save all files in your project, exit Visual Studio and then open your project again and it should be in the Toolbox.

- \* AxOposScale\_1\_6\_Lib should now be listed in the References list in Solution Explorer.

Note: The first object created by dragging/dropping will be called axOPOSScale1

#### 8) Include Files Ported From VB6 to C#:

To implement this test program, three include files originally written in Visual Basic 6, (i.e. Opos.bas, OposScal.bas, and WTCCommands.bas) have been modified to work in the C# (VS 2008) development environment. Those files have also been renamed to use the .cs file extension.

For more extensive applications requiring other include files, it may be necessary for you to modify the files to be C# compliant.

#### 9) Runtime Executable Error (Scale profile name not found in system registry) :

When your application is built, deployed to a Windows 7 computer and the executable is launched, you might encounter a runtime error when attempting to **Connect** even when you have a proper **Scale Profile** name entered. This error occurs because the application is structured to access the standard registry whereas the legacy CCO needs to get these from the virtualized registry storage area in Windows 7.

Note: this error will not occur in the debug mode as registry access in Win 7 are automatically virtualized for legacy applications.

This error will be indicated when the Connect button is pressed by the following status

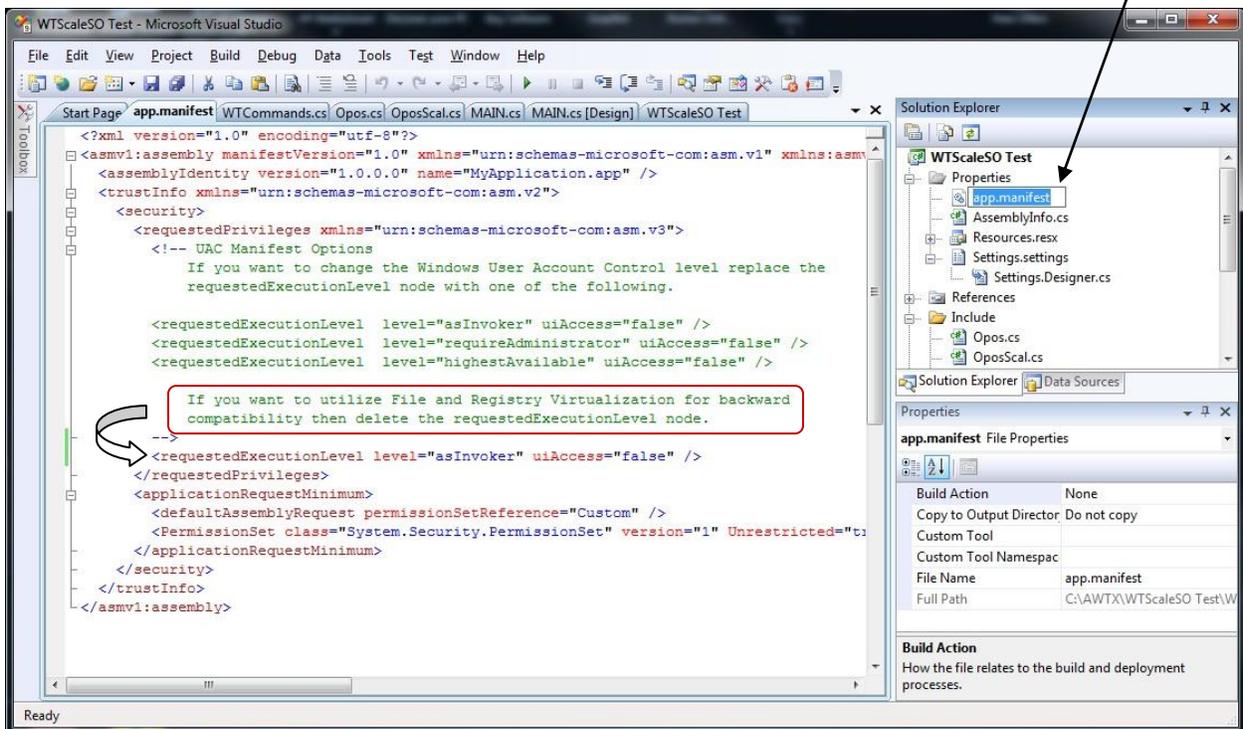
Open returned: 109 OPOS\_E\_NOEXIST

OpenResult code: 302 OPOS\_OR\_REGBADNAME

Open Service Object: FAIL

To remedy this runtime problem, the applications manifest file must be modified slightly so as to let the application (with legacy CCO and SO) access the virtualized registry. A means is provided in the application manifest to use Registry Virtualization to support this backward-compatibility.

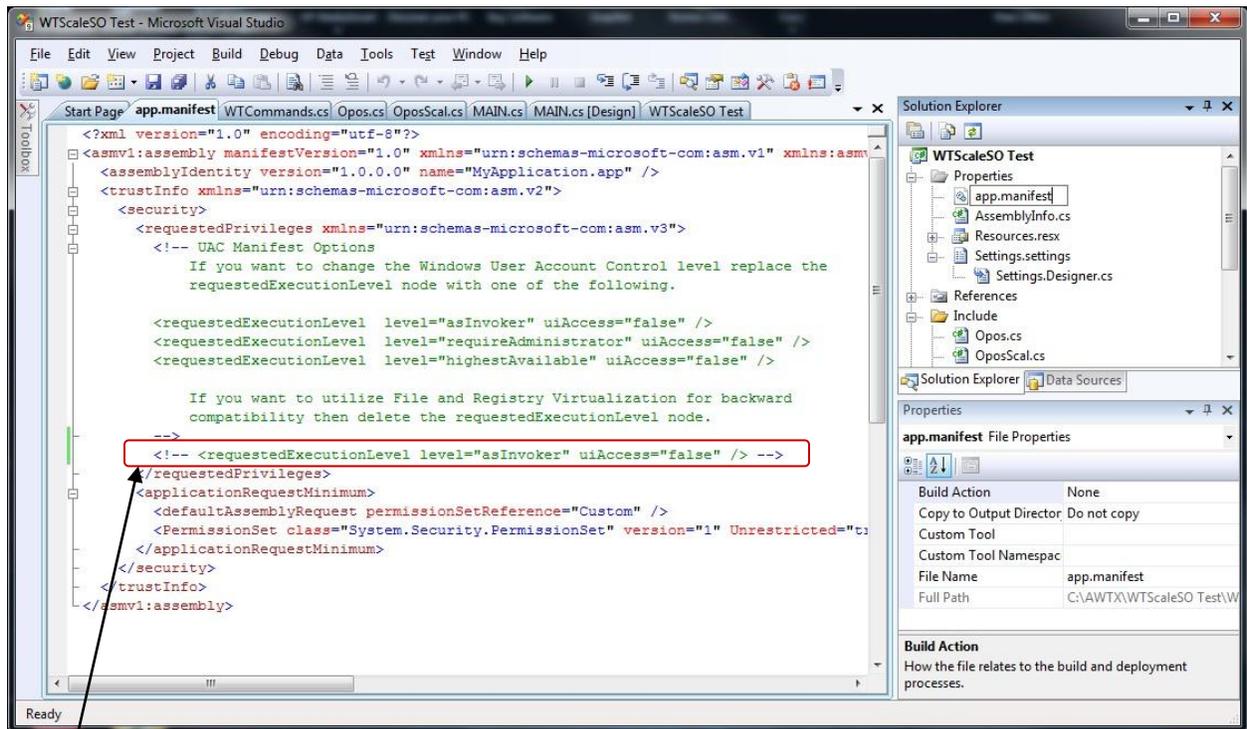
In the **Solution Explorer** double-click on the **app.manifest** file under the **Properties** section.



The `app.manifest` file is an XML file and it includes commented instructions on utilizing Registry Virtualization which is to delete the `requestedExecutionLevel` node.

The next screenshot shows the `requestedExecutionLevel` node line effectively removed by commenting the line out using standard XML comment markers.

Now, when the project is rebuilt and published, the runtime executable will recognize the application as requiring legacy support and access to the virtualized registry storage are.



Line commented out.

Scale drivers are located [here](#) at the Salter Brecknell website.

[sales@salterbrecknell.com](mailto:sales@salterbrecknell.com)  
1-800-637-0529

[service@salterbrecknell.com](mailto:service@salterbrecknell.com)  
1-800-242-2807