ArcLogistics Plugins: Getting Started

This document explains how to create a simple "Hello World" plugin for ArcLogistics. Please be sure to read the document titled **ArcLogistics Plugins: An Introduction**, to have a firm understanding of the basic concepts before starting plugin development.

Setting up the project environment

1. Start Visual Studio, and create a new C# project of **Class Library** type. Name the project **HelloWorldPlugin.** Change the target framework to **.Net Framework 4.0** if it is not already selected.

2. To make debugging easier, change the *Build* settings by providing the **ArcLogistics install path** as the *Output path* for the plugin. This will cause the plugin dll to be built in the install directory, where it is automatically activated and is ready to be used.

3. Also, change the *Start Action* setting under the *Debug* tab to **Start External Program** and select the **ESRI.ArcLogistics.App.exe** file from the install directory. This will launch ArcLogistics with the plugin enabled when you start debugging in Visual Studio.

Implementation

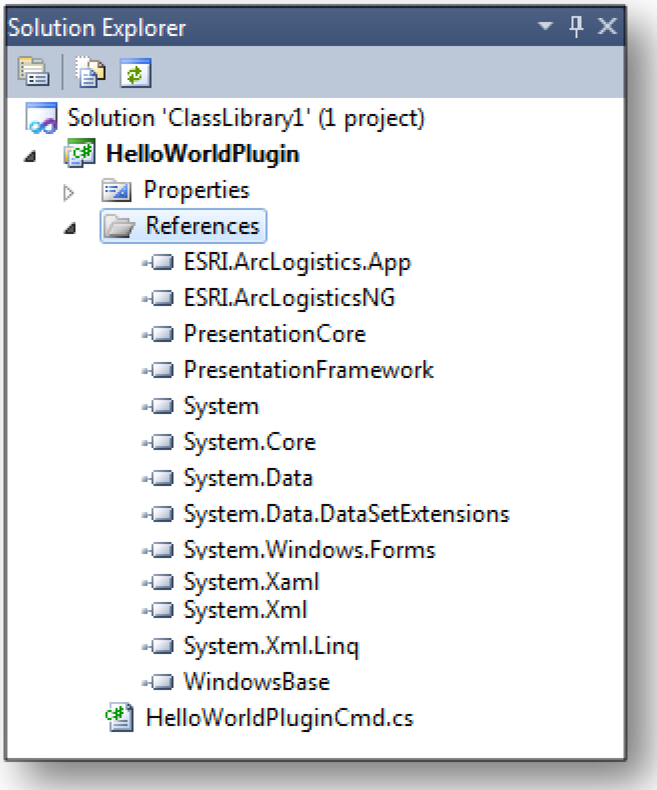
1. First, modify any incompatible references from the Solution Explorer.

2. Add references to the following two ArcLogistics files from the install directory: **ESRI.ArcLogistics.App.exe** and **ESRI.ArcLogisticsNG.dll**.

3. Add other references (PresentationCore, PresentationFramework, System.Wondows.Forms and WindowBase) as shown in Figure 1.

4. Rename Class1 to **HelloWorldPluginCmd**.

Figure 1: Project references



Solution 'ClassLibrary1' (1 project)
- HelloWorldPlugin
  - Properties
  - References
    - ESRI.ArcLogistics.App
    - ESRI.ArcLogisticsNG
    - PresentationCore
    - PresentationFramework
    - System
    - System.Core
    - System.Data
    - System.Data.DataSetExtensions
    - System.Windows.Forms
    - System.Xaml
    - System.Xml
    - System.Xml.Linq
    - WindowsBase
  - HelloWorldPluginCmd.cs

5. Add the following lines of code to the top of the source file (HelloWorldPluginCmd.cs):

   ```
   using System.Windows.Forms;
   using ESRI.ArcLogistics.App.Commands;
   ```

6. Add the following line of code just above the class declaration for HelloWorldPluginCmd:

   ```
   [CommandPlugIn(new string[1] { "ScheduleTaskWidgetCommands" })]
   ```

   This registers the new task we are creating with ArcLogistics, so that the button for this new task will appear in the *Tasks* pane of the *Schedule* tab in ArcLogistics.

7. Modify the class declaration as shown in the following line:

   ```
   public class HelloWorldPluginCmd : ESRI.ArcLogistics.App.Commands.ICommand
   ```

8. Right-click *ICommand* and choose **Implement Interface**. This adds stubs for two methods and five properties.

9. Modify the stubs using the code sample below:

   ```
   public class HelloWorldPluginCmd : ESRI.ArcLogistics.App.Commands.ICommand
   {
       #region ICommand Members

       public void Execute(params object[] args)
       {
           MessageBox.Show("Hello, World!");
       }

       public void Initialize(ESRI.ArcLogistics.App.App app)
       {
       }

       public bool IsEnabled
       {
           get { return true; }
       }

       public System.Windows.Input.KeyGesture KeyGesture
       {
           get { return null; }
       }

       public string Name
       {
           get { return "HelloWorldPlugin.HelloWorldPluginCmd"; }
       }

       public string Title
       {
           get { return "Say Hello"; }
       }

       public string TooltipText
       {
           get { return "Hello World Tooltip"; }
       }

       #endregion
   }
   ```

Testing

1. Build the solution. If build succeeds, the plugin dll should appear in the application install directory.

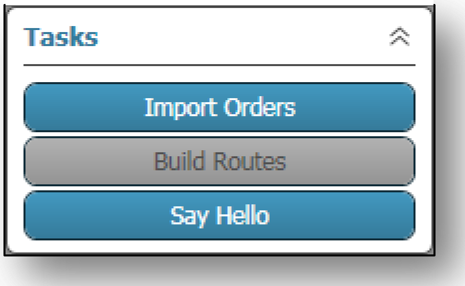
2. Either run ArcLogistics, or start debugging the solution from Visual Studio. In either case, the application will start, load the plugin and activate it automatically.

3. Once the application finishes starting up, go to the *Schedule* tab. You may need to setup a project before the *Schedule* tab is enabled.

4. A new button labeled “Say Hello” will appear on the *Tasks* widget of the *Optimize and Edit* action panel of the *Schedule* tab, as shown in Figure 2.

5. Clicking the *Say Hello* button will show a message box. Click **OK** to dismiss it.

Figure 2: Tasks widget



Wrapping up

Congratulations! You have just created your first ArcLogistics plugin. In this plugin, you added a new task, or *Command*, to the *Tasks* widget of the *Optimize and Edit* action panel on the *Setup* tab. To learn how to create different kinds of plugins and adding more functionality to ArcLogistics, go through the various tutorials and sample projects available on the ArcLogistics Resource Center.

Refer back to the document titled **ArcLogistics Plugins: An Introduction** for a complete list and descriptions of all resources available for you to create your own plugins.