ArcLogistics Tutorials: How to create a Page plugin

This document explains how to create a *Page* type of plugin for ArcLogistics. Before starting plugin development, please read the document titled **ArcLogistics Plugins: An Introduction**, to have a firm understanding of the basic concepts, and the document titled **ArcLogistics Plugins: Getting Started**, for a step-by-step guide for creating a simple "Hello World" plugin.

Setting up the project environment

1. Start Visual Studio, and create a new C# project of *WPF User Control Library* type. 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. Add references to the following two ArcLogistics files from the install directory: **ESRI.ArcLogistics.App.exe** and **ESRI.ArcLogisticsNG.dll**. You may need to add more references as your development progresses.

2. Use the following block of code as a guide to modify the xaml file in your project. The idea is to create a page with a grid control having a label and a textbox in it. You can add more content inside the Grid control to make the page more useful.

```
<pages:PageBase
      x:Class="PagePluginTutorial.MyCustomPage"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
      xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
      xmlns:pages="clr-
      namespace:ESRI.ArcLogistics.App.Pages;assembly=ESRI.ArcLogistics.App"
      xmlns:app_controls="clr-
      namespace:ESRI.ArcLogistics.App.Controls;assembly=ESRI.ArcLogistics.App"
      mc:Ignorable="d" d:DesignHeight="300" d:DesignWidth="300">
      <Grid HorizontalAlignment="Stretch" VerticalAlignment="Stretch">
              <Label Content="Name:" Height="28" HorizontalAlignment="Left"
                     Margin="60,124,0,0" Name="label1" VerticalAlignment="Top" />
              <TextBox Height="23" HorizontalAlignment="Left"
                       Margin="110,126,0,0" Name="textBox1"
                       VerticalAlignment="Top" Width="120" />
      </Grid>
</pages:PageBase>
```

3. Add the following lines of code to the top of the implementation file (.cs):

```
using ESRI.ArcLogistics.App.Help;
using ESRI.ArcLogistics.App.Pages;
```

You may need to add additional usings as your development progresses.

4. Add the following line of code just above the partial class declaration for the plugin:

```
[PagePlugInAttribute("Schedule")]
```

This registers the new Page with ArcLogistics, so that the new page's action panel will appear on the *Schedule* tab. Replace `"Schedule"` by `"Home"`, `"Setup"`, `"Deployment"` or `"Preferences"` to make your action panel appear on those locations.

5. Add the `PageBase` abstract class as a superclass for the page, as shown in the following example:

```
public partial class MyCustomPage: PageBase
```

6. Right-click *PageBase* and choose **Implement Abstract Class**. This adds stubs for five properties.

7. Modify the stubs in a manner similar to the code sample given below. The constructor creates a new help topic for the page. The *IsAllowed* and *IsRequired* booleans make the action panel enabled and bold, respectively. The *HelpTopic* property returns the help topic for the class. The *PageCommandsCategoryName* property specifies the name of the page's *Tasks* widget. This name must be referenced by commands which want to appear on this page's *Tasks* widget. The *Icon* property determines the icon displayed on the action panel. A default icon if used if null is returned. The *Name* property is used to provide the name of the plugin. The *Title* Property specifies the text that appears on the action panel.

```
[PagePlugInAttribute("Schedule")]
public partial class MyCustomPage: PageBase
{
    public MyCustomPage()
    {
        _helpTopic =
              new HelpTopic(null, "This is the custom Help for My Custom Page");
        IsAllowed = true;
        IsRequired = true;
        InitializeComponent();
    }

    public override ESRI.ArcLogistics.App.Help.HelpTopic HelpTopic
    {
        get { return _helpTopic; }
    }

    public override string PageCommandsCategoryName
    {
        get { return null; }
    }

    public override System.Windows.Media.TileBrush Icon
    {
        get { return null; }
    }
```

```
        public override string Name
        {
            get { return "PagePluginTutorial.MyCustomPage"; }
        }

        public override string Title
        {
            get { return "My Custom Page"; }
        }

        private HelpTopic _helpTopic;
    }
```

8. You can add more methods, UI handling logic etc. to this class to make the plugin more useful.

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, a new action panel will appear on the location you specified.

4. Clicking on the action panel will display the page with a label and a textbox.

Wrapping up

Congratulations! You have just learned how to create *Page* plugins for ArcLogistics. To learn how to create other kinds of plugins and adding more functionality to ArcLogistics, go through the other tutorials and various 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.