# Module 5 Lab B

# Packaging Custom Plug-In

##### Objective: Modify the plugin to read configuration data

**Step 1: Creating a new Dynamics Solution**

|  |  |
| --- | --- |
| 1 | Copy the your solution of Module 5 Lab A to the Module 5 Lab B starter folder and open the copied solution in Visual studio.  Build the solution, it should compile with no errors. |

|  |  |
| --- | --- |
| 2 | Declare a private string field in the AddressFromZipCodePlugin class named webServiceUrl  private string webServiceUrl; |
| 3 | Currently the plug-in is hard coded to use <http://webservicex.net/uszip.asmx> as the webservice endpoint address  We need to remove the hard coded url we copied from the LAB A solution to LAB B.  Find the statement that assigns a value to the httpClient variable and replace it with the code below.  var httpClient = HttpWebRequest.Create(webServiceUrl + "/GetInfoByZIP?USZip=" + strZipcode); |
| 4 | Build the solution and fix any errors. |

**Step 2: Exporting the Solution**

|  |  |
| --- | --- |
| **1** | Back in Dynamics 365 click on the export solution button for the Module 5 Solution |
| **2** | You will be prompted to publish all customizations again but there is no need to since nothing in this solution has changed since it was last published. So click on the Next button. |
| **3** | Click on next again (we do not need to export any settings configuration) |
| **4** | Select to export an unmanaged package ( the default ) this will allow the unmanaged packaged to be editable in an systems that have imported the “unmanaged package”. You can also subsequently export a managed package that isn’t editable. |
| **5** | In the next dialog target the highest version available. |
| **6** | Eventually a zip file (Module5\_1\_0\_0\_0 .zip) will be downloaded to your downloads folder |

|  |  |
| --- | --- |
| **7** | Now unregister the existing plugin by removing the Assembly that references it with the plugin registration tool |
|  |  |

**Step 3 : Deploying and Configuring the new version**

|  |  |
| --- | --- |
| 1 | Deploy and register the new version of your plugin following the same steps as in Lab A remembering to register the “step” for the Create message of the contact entity. The plugin should execute post operation asynchronously, server side.  Notice that we can pass unsecure and secure configuration to the plugin    Add the url of the webservice endpoint to the unsecure configuration. As shown above |
| 2 | Test that the plug-in works as it did before |

**Step 4 : Debugging**

|  |  |
| --- | --- |
| 1 | Using the plugin registration tool right click on the AddressResolver plug-in and enable profiling    Accept the default configuration as shown below |
| 2 | Trigger the execution of the plug-in code by creating another Contact entity |
| 3 | Click on the debug menu of the PluginRegistration tool and then select the profile by using the download button. It may take some time for the file to appear ! |
| 4 | Select the local copy of the assembly but do not click on the start execution button yet. |
| 5 | Go back to the Visual Studio Project that you used to develop the plug-in and place a breakpoint in the code as shown below |
| 6 | Attach the debugger to the pluginregistration tool exe as shown below |
| 7 | Now go back to the plug-in registration tool and click on the start execution button. Visual studio should go into break mode and stop execution at the break point placed in code and you should be able to step through and investigate values of variables using all the usual techniques. |
|  |  |