

Import Export Data Plugin

This sample plugin is useful for learning how to create a custom action panel in the *Preferences*, and adding two very useful custom Commands to the *Optimize and Edit* action panel on the *Setup* tab. The commands enable the user to import order information from - and export information to - external entities by using custom executables. The file *ImportExportPluginPreferencesPage.xaml* contains xaml code for creating the custom preferences page, *ImportExportPluginPreferencesPage.xaml.cs* contains the support code for the page, *ImportDataCmd.cs* and *ExportDataCmd.cs* contain the code for the commands, and *ImportExportPluginPreferencesPageParams.cs* contains the code for a support class which acts as an intermediate container for the preferences data.

Installation:

1. Copy the plugin DLL (*ImportExportDataPlugin.dll*) into the ArcLogistics install directory.

Usage:

1. A new action panel called *Import/Export* appears in the *Preference*. Two buttons also appear on the *Tasks* widget of the *Optimize and Edit* action panel of the *Schedule* tab, and they are disabled by default.
2. To enable the buttons, go to the *Import/Export* action panel in the *Preferences*.
3. Enter the names to be used with the Import and Export commands, and click the *Browse* buttons to specify the Import and Export executable files.
4. To use the provided sample batch files in place of custom executables, follow these steps:
 - a. Copy the provided *ImportSample.bat*, *ExportSample.bat* and *SampleOrders.csv* files into any directory and note this location.
 - b. Open *ImportSample.bat* using any text editor, and change the full file path to the actual full path of *SampleOrders.csv*.
 - c. Open *ExportSample.bat*, and change the full output file paths to reflect the desired output locations and filenames.
 - d. In the *Import/Export Preferences*, use *ImportSample.bat* and *ExportSample.bat* as the Import and Export executable files.
5. The *Import Data* and *Export Data* buttons should be enabled now, and should have the names which you specified in the preferences.
6. Click *Import Data from <program name>* to kick off the import executable. If you are using the provided sample batch files, then it will copy the provided orders file to a temporary location, and provide the temporary file to ArcLogistics for import. In actual usage, the custom executable should obtain orders information from its source, create a temporary csv file, and provide it to ArcLogistics.
7. Similarly, click *Export Data to <program name>* to kick off the export executable. The plugin first creates temporary csv files containing the routes, stops and orders information, and then provides these temporary files to the executable. If you are using the provided sample batch files, then it will just copy the provided temporary files to the locations specified in the batch file. In actual usage, the custom executable should send the information from the temporary files to the desired destination.

Customization:

1. Open the Visual Studio project *ImportExportDataPlugin*.
2. Modify references and project settings if needed.
3. Modify the code in the *Execute* methods of *ImportDataCmd.cs* and *ExportDataCmd.cs* to change plugin behavior. They contain the logic for reading data from csv files, creating orders, adding them to the project, and utilize the export functionality of ArcLogistics to write the routes, stops and orders information to files.
4. It may also be useful to change some of the other methods or properties, such as *Title* and *Tooltip*.
5. Modify the preference page's xaml code in *ImportExportPluginPreferencesPage.xaml* to change how the page looks, or the support code in *ImportExportPluginPreferencesPage.xaml.cs* to change how the page reacts to user interaction.
6. Rebuild the project to obtain the new dll.
7. Copy the dll into the install directory, and run ArcLogistics to check the results.