CreatingAlgPlugin

Your Visited Pages

CreatingAlgPlugin
PluginsForPrompt
Prompt

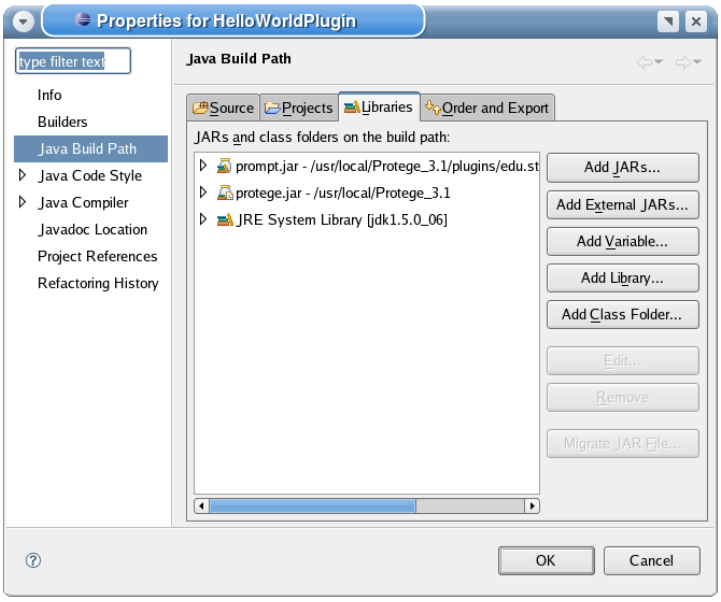
View Backlinks

Search

Creating an Algorithm Plugin (78S)

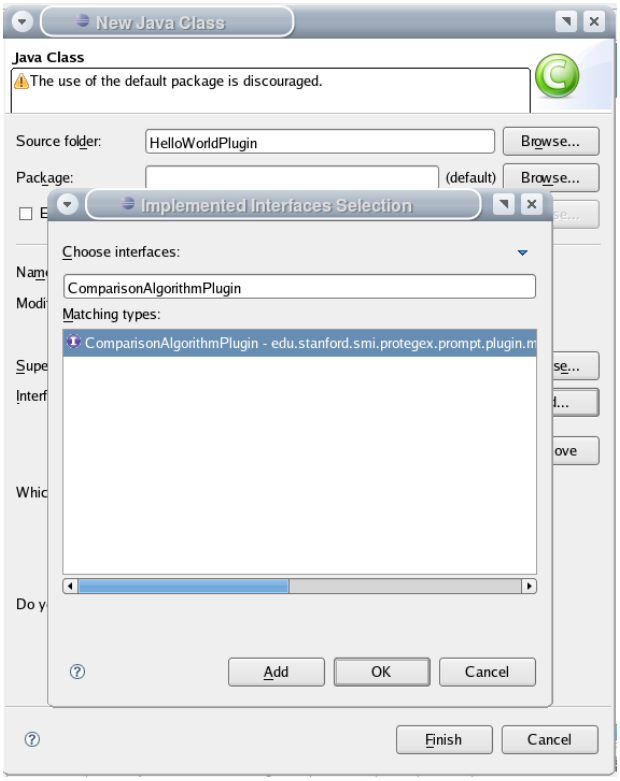
These instructions are written for Eclipse users, but someone not using Eclipse should be able to use them to create their own plugin. You can download the source code here: http://protege.cim3.net/file/pub/files/prompt/HelloWorldPlugin.zip. (78T)

- First begin by downloading Prompt, if you do not already have the latest version. (78U)
- Open Eclipse and create a new Java Project, by clicking File - New - Project. Give the project the name, HelloWorldPlugin. (78V)
- Right click on your HelloWorldPlugin in the Package Exlorer and click Properties. In the dialog that opens, click the Libraries tab, then click Add External JARs. In the file open dialog, find where you have prompt.jar, select it, and click Ok. Do the same for your protege.jar file. (7DF)



(7DG)

- Right click on your HelloWorldPlugin in the Package Exlorer and create a new Class file. Call it, HelloWorldAlgorithm and in the Interfaces section click the Add button. In the Choose interfaces text box, type in ComparisonAlgorithmPlugin and click Add, then click Ok. Finally, click the Finish button. (7DH)



(7DI)

- In your HelloWorldAlgorithm.java file, add the following methods if they were not created automatically by Eclipse. (7DJ)

```
public String getPluginName() {nid 790}
public void invokePlugin() {nid 791}
public Collection performAlignment(AlgorithmProgressMonitor progress, KnowledgeBase sourceKnowledgeBase, KnowledgeBase targetKnowledgeBase, KnowledgeBase mappingKnowledgeBase) {nid 792}
public JComponent getConfigurationComponent() {nid 793}
public boolean validateConfigSettings()    (794)
```

Implementing the extension (7DK)

- Modify getPluginName. This function must return a valid string in order for your plugin to show up in Prompt. (7DL)

```
public String getPluginName() {
   return "Hello World Algorithm";
}    (7DM)
```

- Modify performAlignment. This function is called when Prompt attempts to align two ontologies. The first parameter is a progress monitor that can be used by an algorithm plugin to communicate with Prompt about its current progress. The next two parameters are the two ontologies being aligned, and the final parameter is the mapping ontology. (79X)

```
public Collection performAlignment(AlgorithmProgressMonitor arg0,
                   KnowledgeBase arg1, KnowledgeBase arg2, KnowledgeBase arg3) {
   return null;
}    (7DN)
```

- Modify getConfigurationComponent. This function is called when the user in Prompt selects to use your algorithm. If your algorithm has configuration parameters that a user can set, this is where you can provide your own configuration panel. (7A2)

```
public JComponent getConfigurationComponent() {
   return null;
}    (7DO)
```

- Modify validateConfigSettings. This function is called prior to Prompt running the algorithm in order for the plugin to verify that its input configuration is correct. For this simple example, we'll always return true. (7A5)

```
public boolean validateConfigSettings() {
   return true;
}    (7DP)
```

- Finally, the invokePlugin function can remain empty. This function is called when your plugin is first activated. (7A8)

Creating a manifest file (7DQ)

- The manifest file is used by Prompt in order to load your plugin. If this file does not exist, the plugin will not be loaded. (79B)
- In Eclipse, right click on your HelloWorldPlugin project in the Package Explorer and create a new folder called META-INF. (79C)
- Right click on the newly created META-INF folder, and create a new file called MANIFEST.MF. (79D)
- Open MANIFEST.MF and add the following lines: (79E)

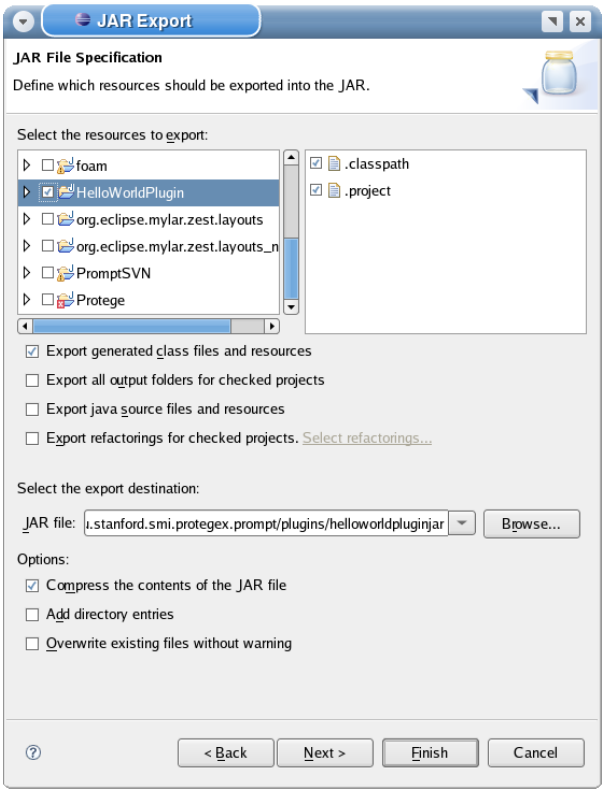
```
Manifest-Version: 1.0 {nid 79F}

Name: HelloWorldAlgorithm.class
Comparison-Extension: True    (7DR)
```

- This will tell Prompt the name of your plugin class and also the type of extension this plugin is. (79I)

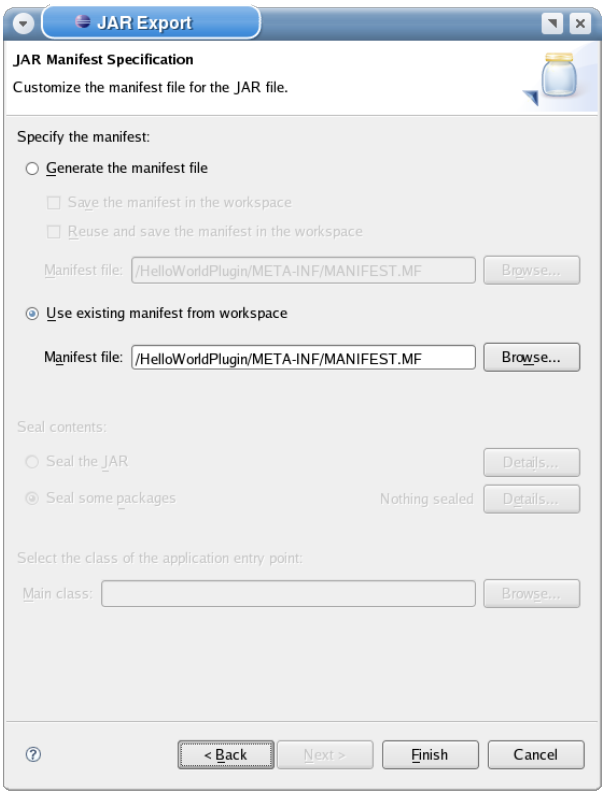
Installing the plugin (7DS)

- Right click on your HelloWorldPlugin project in the Package Explorer and choose Export. (79K)
- Select under Java the JAR file option. (79L)
- Select the check mark by the HelloWorldPlugin project if it is not already selected. (79M)
- Click Browse, and navigate to your Prompt install directory. Create a new directory and call it plugins if this directory does not already exist. In the Name field, call your jar file helloworldplugin.jar. {nid 79N} (79G)



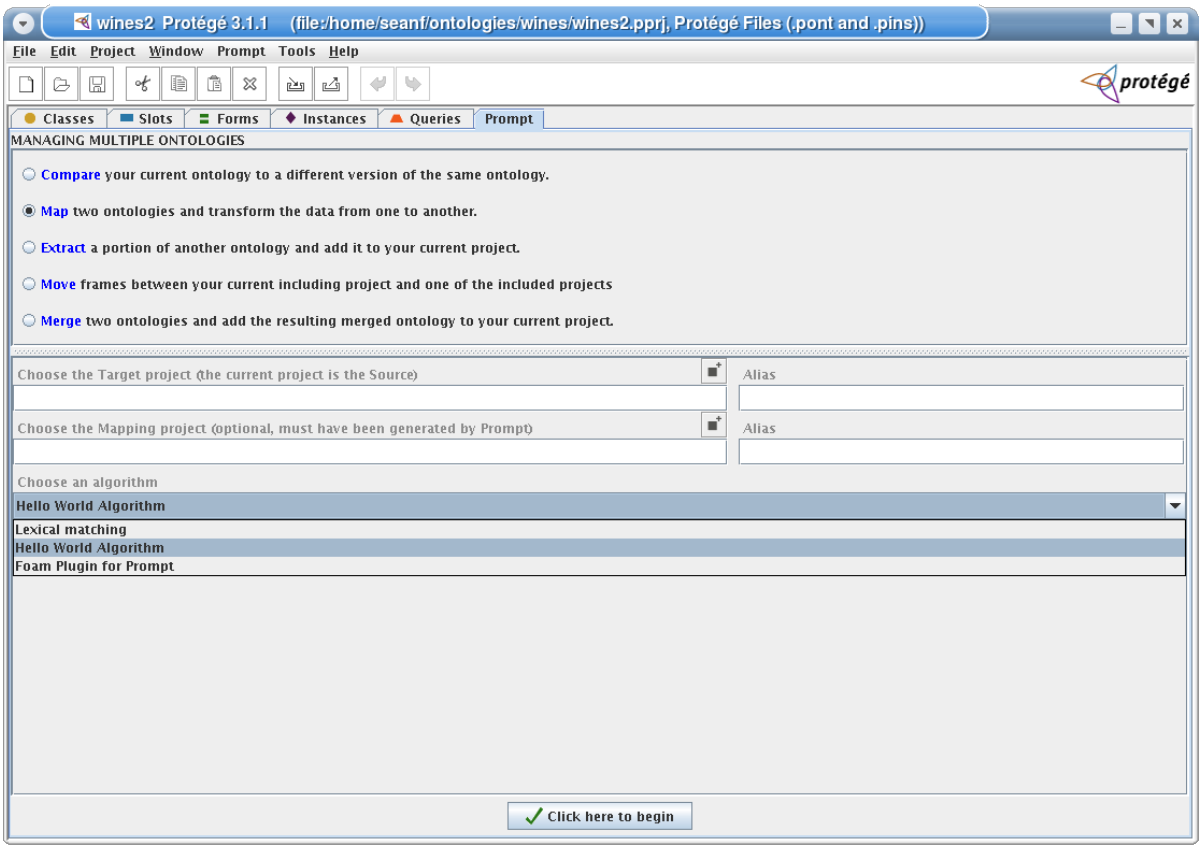
(7DT)

- Click Next twice on the JAR Export dialog. (79O)
- You should now see a JAR Manifest Specification screen. Select the Use existing manifestor from workspace radio button and Browse to your MANIFEST.MF file. (79P)
- Finally click Finish. (79Q)



(7DU)

- You should now be able to run Protege, open a project, open Prompt and select the Map or Merge option. You should see a drop down box displaying Hello World Algorithm. {nid 79R} (7AC)



(7DV)

View other revisions
Last edited August 1, 2006 9:51 (diff)

RSS