CreatingAlgPlugin



WikiHomePage | RecentChanges | Page Index

Creating an Algorithm Plugin (785)

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. (781)

- 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 <u>HelloWorldPlugin</u> 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)

💌 🤍 🗣 Propertie	s for HelloWorldPlugin	N × 1
type filter text	Java Build Path	
Info Builders Java Build Path ▷ Java Code Style ▷ Java Compiler Javadoc Location Project References Refactoring History	Source Projects Libraries Order and Export JARs and class folders on the build path: A prompt.jar - /usr/local/Protege_3.1/plugins/edu.st A protege.jar - /usr/local/Protege_3.1 A JRE System Library [jdk1.5.0_06]	Add JARs Add External JARs Add Variable Add Library Add Class Folder Edit Remove Migrate JAR Elle
0	0	K Cancel

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

(7DG)

Login (create accou	
Your Visited Pa	
Your visited Pa	ges
CreatingAlgPlugin PluginsForPrompt Prompt	
View Backlinks	
Search	

New Java Class	×
Java Class	
A The use of the default package is discouraged.	
Source folder: HelloWorldPlugin	Browse
Package: (default)	Browse
E C Implemented Interfaces Selection	× se
Choose interfaces:	-
Nami ComparisonAlgorithmPlugin	
Modi Matching types:	_
ComparisonAlgorithmPlugin - edu.stanford.smi.protegex.prompt.plugin	
Interf	s <u>e</u>
	1
	ove
Whic	
Do γ	
Add OK Cancel	
⑦ <u>Fi</u> nish	Cancel
· · · · · · · · · · · · · · · · · · ·	

• In your HelloWorldAlgorithm java file, add the following methods if they were not created automatically by Eclipse. (703)

public String getPluginName() {nid 790} public void invokePlugin() {nid 791} public Collection performAlignment(AlgorithmProgressMonitor progress, <u>KnowledgeBase</u> source<u>KnowledgeBase</u>, <u>KnowledgeBase</u> target<u>KnowledgeBase</u>, <u>KnowledgeBase</u> mapping<u>KnowledgeBase</u>) {nid 792} public JComponent get<u>ConfigurationComponent()</u> {nid 793} public boolean validate<u>ConfigSettings()</u> (794)

(7DI)

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)

```
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. (742)

```
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. (7AB)

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 <u>HelloWorldPlugin</u> 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. (79c)
- (79E)
- Open MANIFEST.MF and add the following lines:

Manifest-Version: 1.0 {nid 79F}

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

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

Installing the plugin (7DS)

- Right click on your <u>HelloWorldPlugin</u> project in the Package Explorer and choose Export. (79K)
 Select under Java the JAR file option. (79L)
 Select the check mark by the <u>HelloWorldPlugin</u> 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} (79C)

JAR Export X					
JAR File Specification					
Define which resources should be exported into the JAR.					
Select the resources to export:					
▶ □ 💕 foam					
▶ ☑ 🖉 HelloWorldPlugin					
▷ □ 😂 org.eclipse.mylar.zest.layouts					
▷ □ □ □ □ ▷ □ □ □ □					
✓ Export generated class files and resources					
Export all output folders for checked projects					
Export java source files and resources					
Export refactorings for checked projects. <u>Select refactorings</u>					
Select the export destination:					
JAR file: L.stanford.smi.protegex.prompt/plugins/helloworldpluginjar 💌 Browse					
Options:					
✓ Compress the contents of the JAR file					
Add directory entries					
Overwrite existing files without warning					
⑦ < <u>Back</u> <u>Next</u> > <u>Finish</u> Cancel					

- Click Next twice on the JAR Export dialog. (790)
 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)

● JAR Export	×					
JAR Manifest Specification Customize the manifest file for the JAR file.						
Specify the manifest:						
○ <u>G</u> enerate the manifest file						
Manifest file: //HelloWorldPlugin/META-INF/MANIFEST.MF	Browse					
⊚ Use existing manifest from workspace						
Manifest file: /HelloWorldPlugin/META-INF/MANIFEST.MF	Browse					
Seal contents: O Seal the JAR (a) Seal some packages Nothing sealer	Details					
Select the class of the application entry point: Main class:	Browse					
() < Back Next > Finish	Cancel					
Image: Constraint of the sector of the se	Cancel					

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)

💽 🛛 🝕 wines2 Protégé 3.1.1 (file:/home/seanf/ontologies/wines/wines2.pprj, Protégé Files (.pont and .pinsj)						
Eile Edit Project Window Prompt Tools Help						
			🔶 protégé			
Classes Slots Forms Instances Queries Prompt						
MANAGING MULTIPLE ONTOLOGIES						
O Compare your current ontology to a different version of the same ontology.						
Map two ontologies and transform the data from one to another.						
O Extract a portion of another ontology and add it to your current project.						
O Move frames between your current including project and one of the included projects						
O Merge two ontologies and add the resulting merged ontology to your current project.						
Choose the Target project (the current project is the Source)	• ⁺	Alias				
choose the ranget project (the current project is the source)		741143				
Channel at a Manufact much at dealers I are a have been an and due Dearway	•	A11				
Choose the Mapping project (optional, must have been generated by Prompt)	-	Alias				
Choose an algorithm						
Hello World Algorithm			•			
Lexical matching Hello World Algorithm						
Foam Plugin for Prompt						
Click here to begin						

(7DV)

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