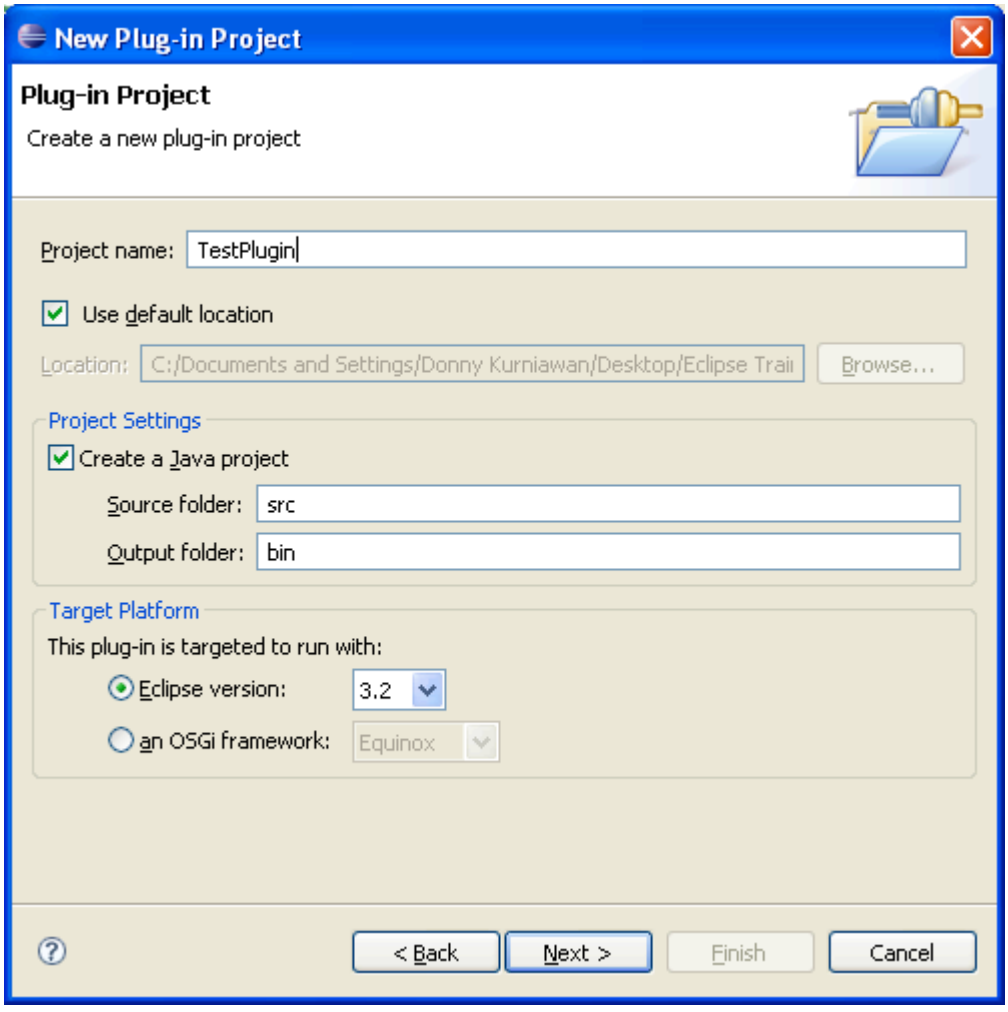
Tutorial 07: Plug-in: Preference Page, Editor, View

Contents:

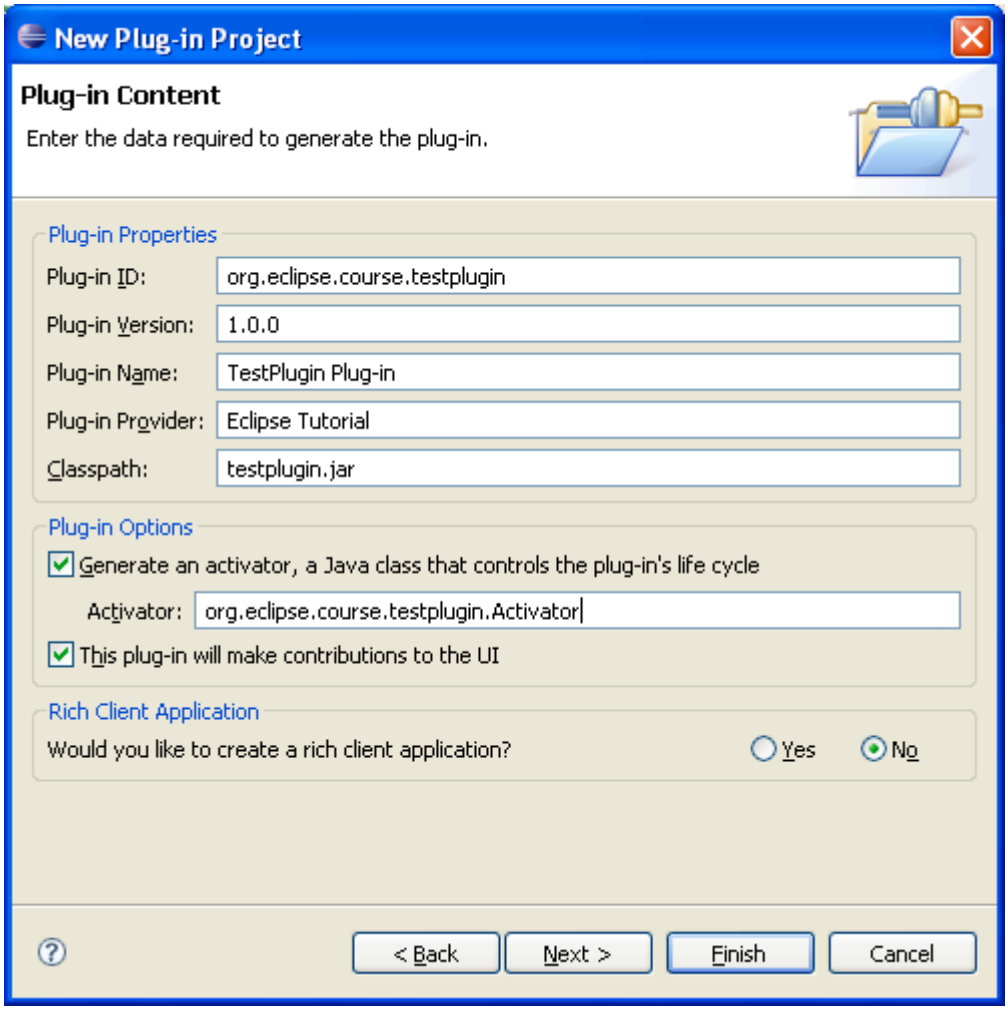
1. Creating the plug-in project.
2. Testing the plug-in.
3. Inspecting the plug-in code.

Creating the plug-in project

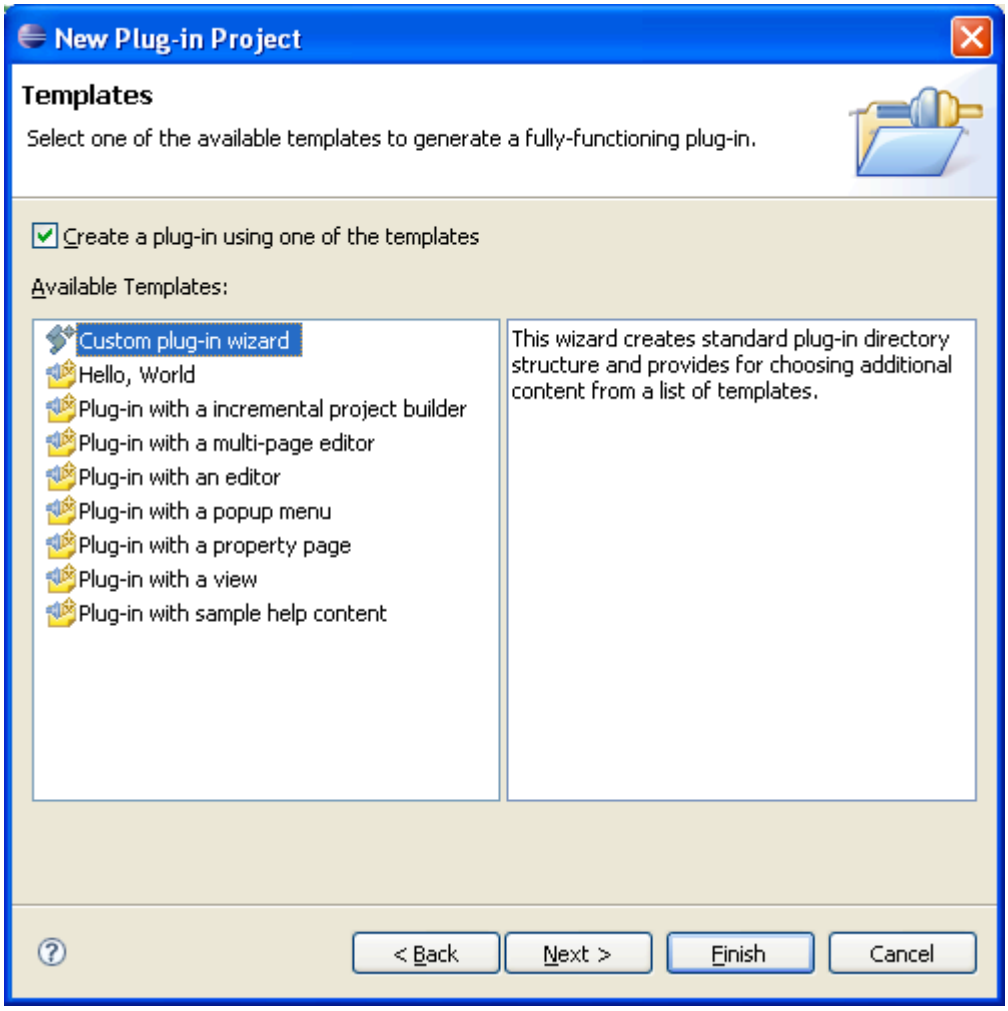
1. Switch to the Plug-in Development perspective.
2. Click File > New > Project…
3. Select Plug-in Development > Plug-in Project.
4. Name the new project, TestPlugin.



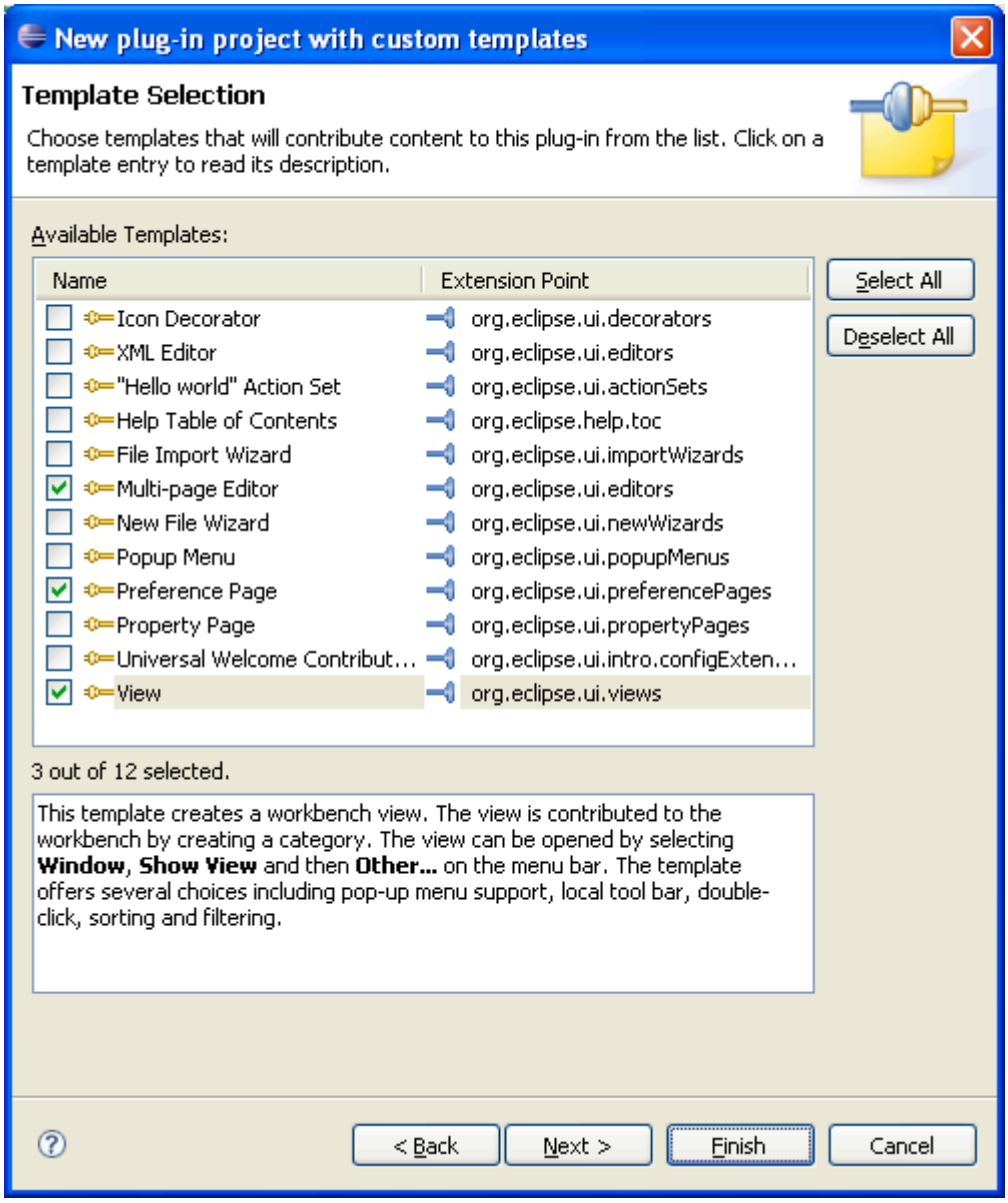
5. Enter the following properties, and click Next.

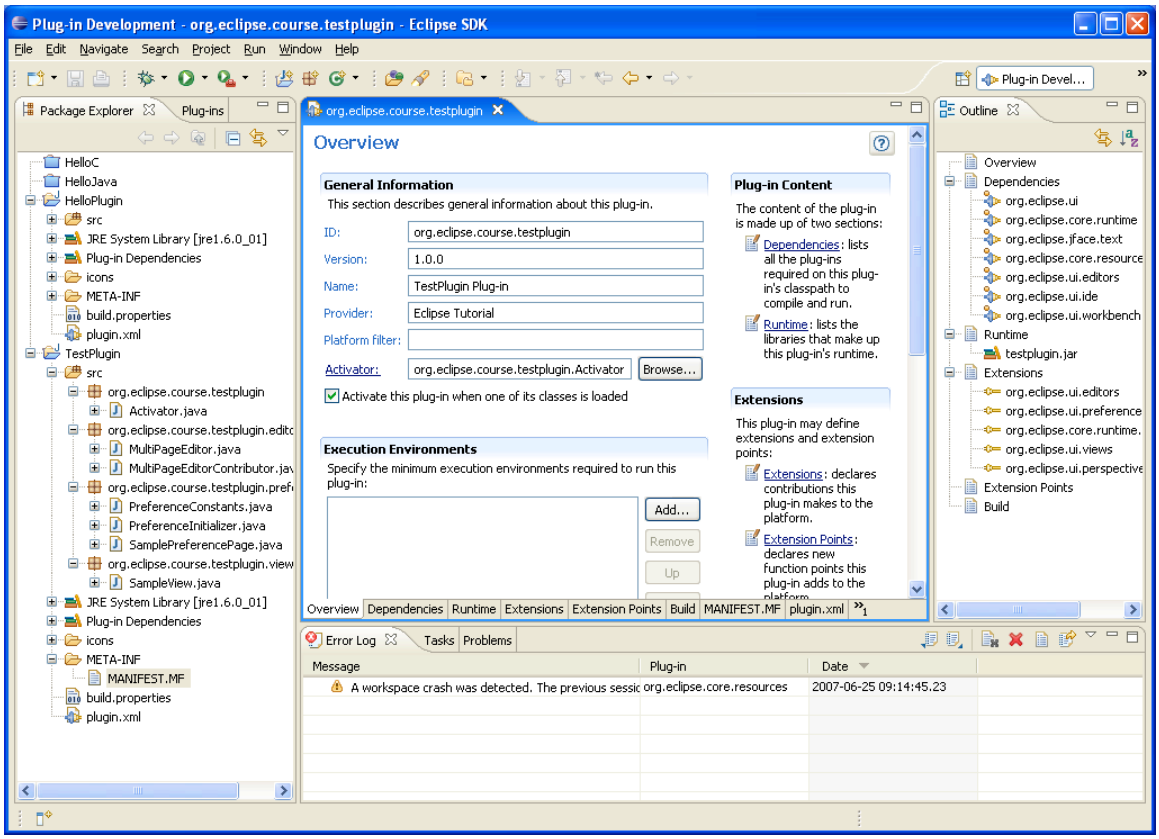


6. Select “Custom plug-in wizard”, and click Next.



7. Choose the following templates and click Finish.

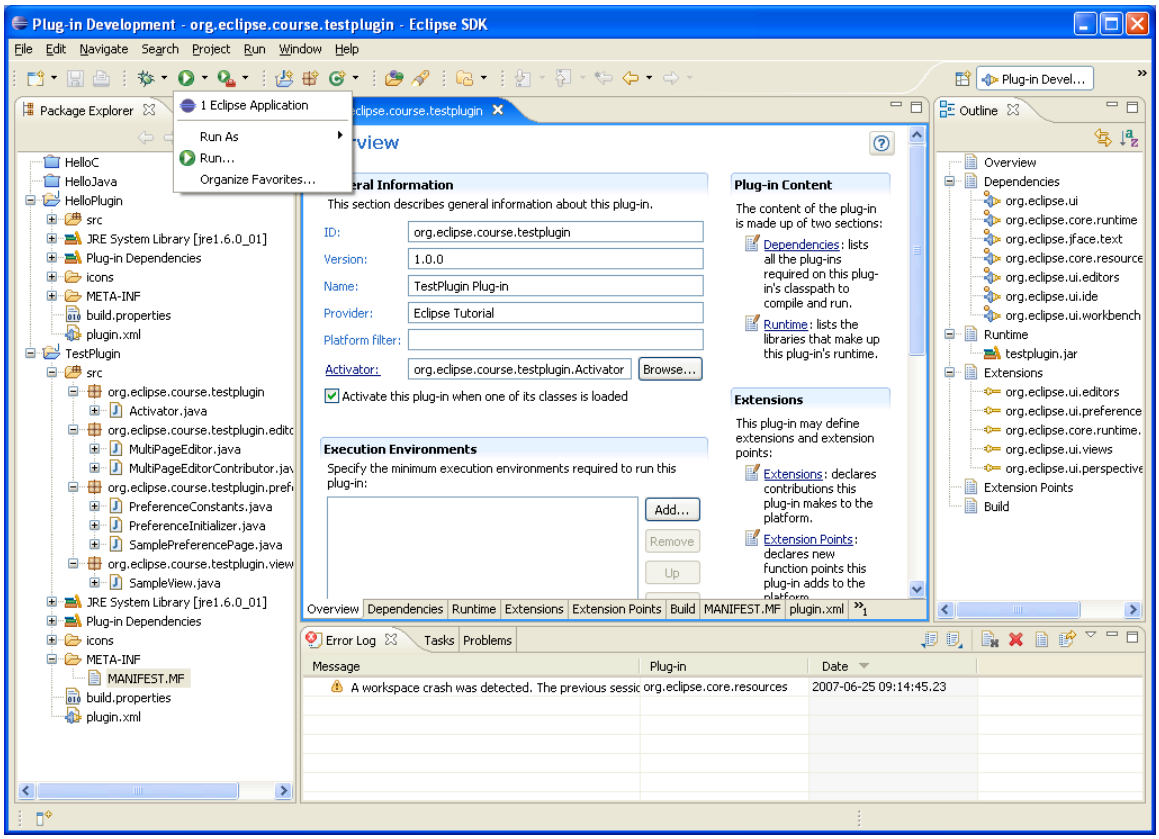

New plug-in project with custom templates
Template Selection
Choose templates that will contribute content to this plug-in from the list. Click on a template entry to read its description.
Available Templates:
Name
Extension Point
Icon Decorator
org.eclipse.ui.decorators
XML Editor
org.eclipse.ui.editors
"Hello world" Action Set
org.eclipse.ui.actionSets
Help Table of Contents
org.eclipse.help.toc
File Import Wizard
org.eclipse.ui.importWizards
Multi-page Editor
org.eclipse.ui.editors
New File Wizard
org.eclipse.ui.newWizards
Popup Menu
org.eclipse.ui.popupMenus
Preference Page
org.eclipse.ui.preferencePages
Property Page
org.eclipse.ui.propertyPages
Universal Welcome Contribut...
org.eclipse.ui.intro.configExten...
View
org.eclipse.ui.views
Select All
Deselect All
3 out of 12 selected.
This template creates a workbench view. The view is contributed to the workbench by creating a category. The view can be opened by selecting Window, Show View and then Other... on the menu bar. The template offers several choices including pop-up menu support, local tool bar, double-click, sorting and filtering.
< Back
Next >
Finish
Cancel



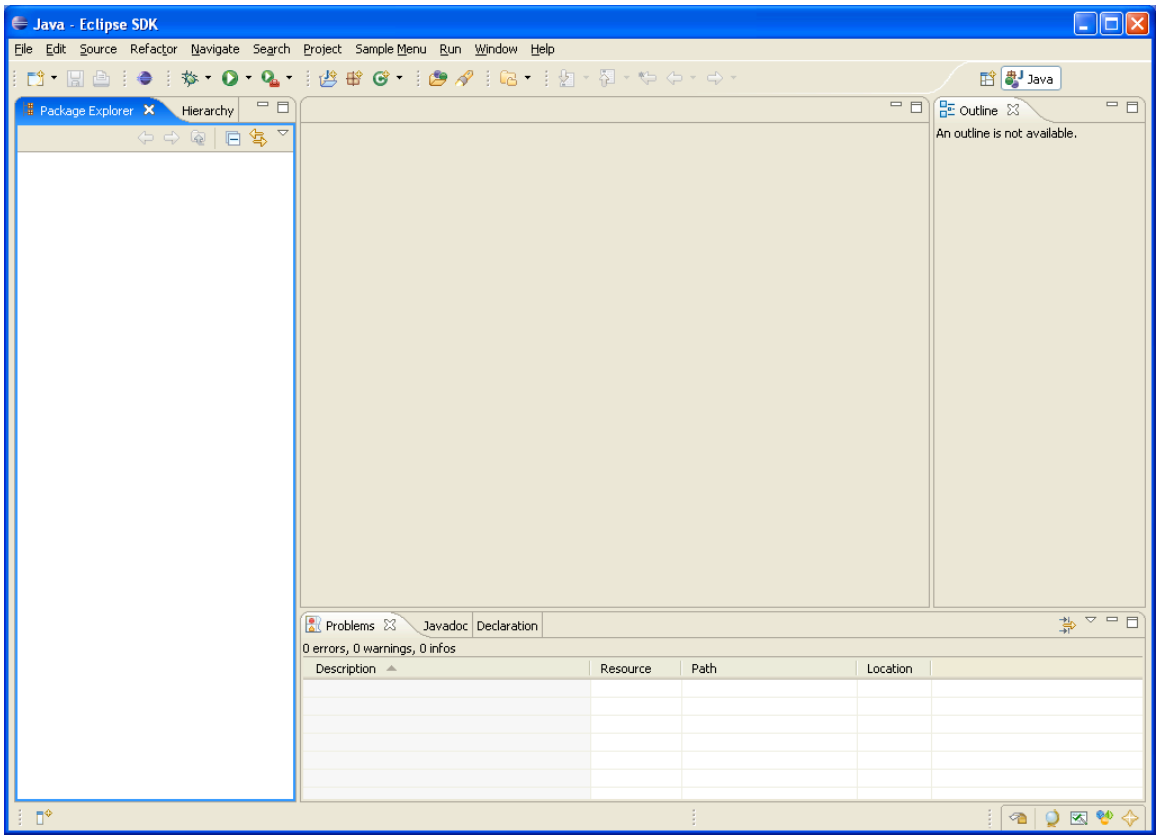
Testing the Plug-in

1. Click the arrow next to the Run button on the toolbar and click Eclipse Application.

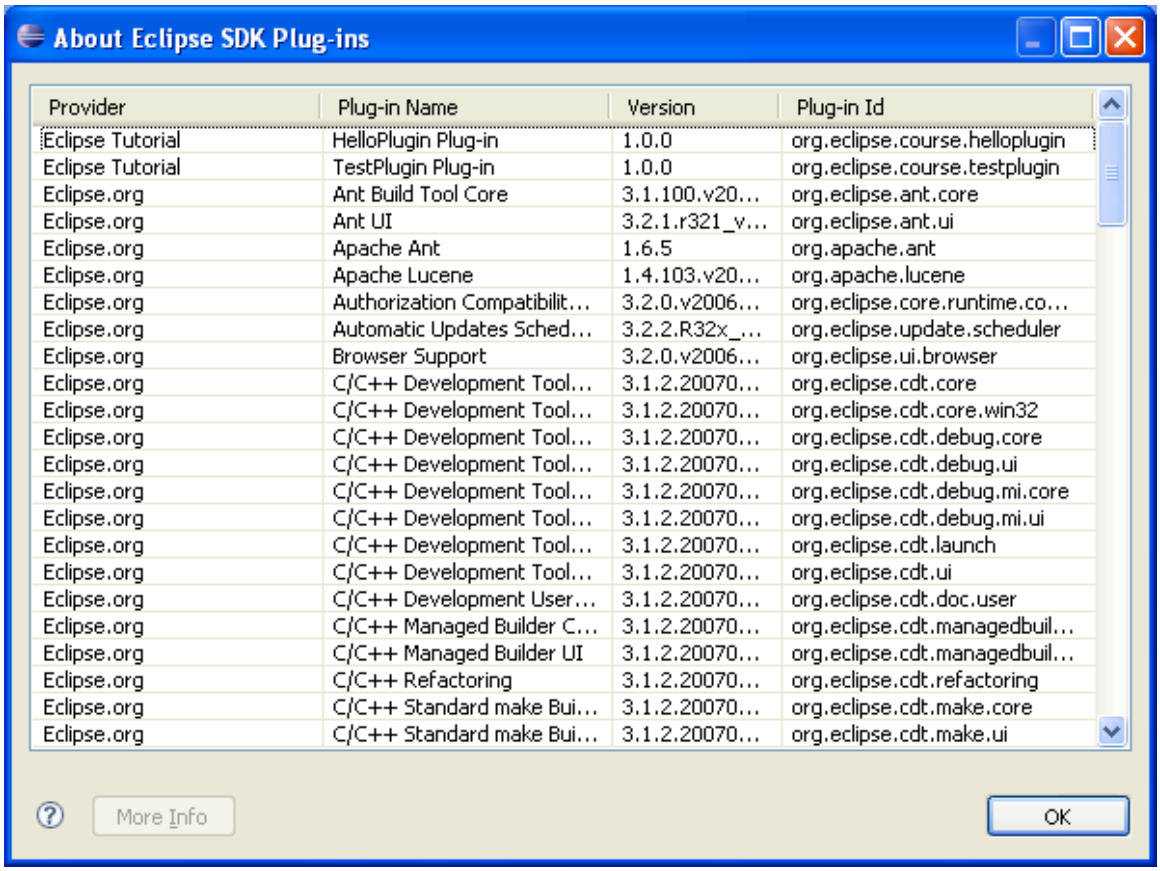
> You can also click the Run button directly if Eclipse Application is the first option.



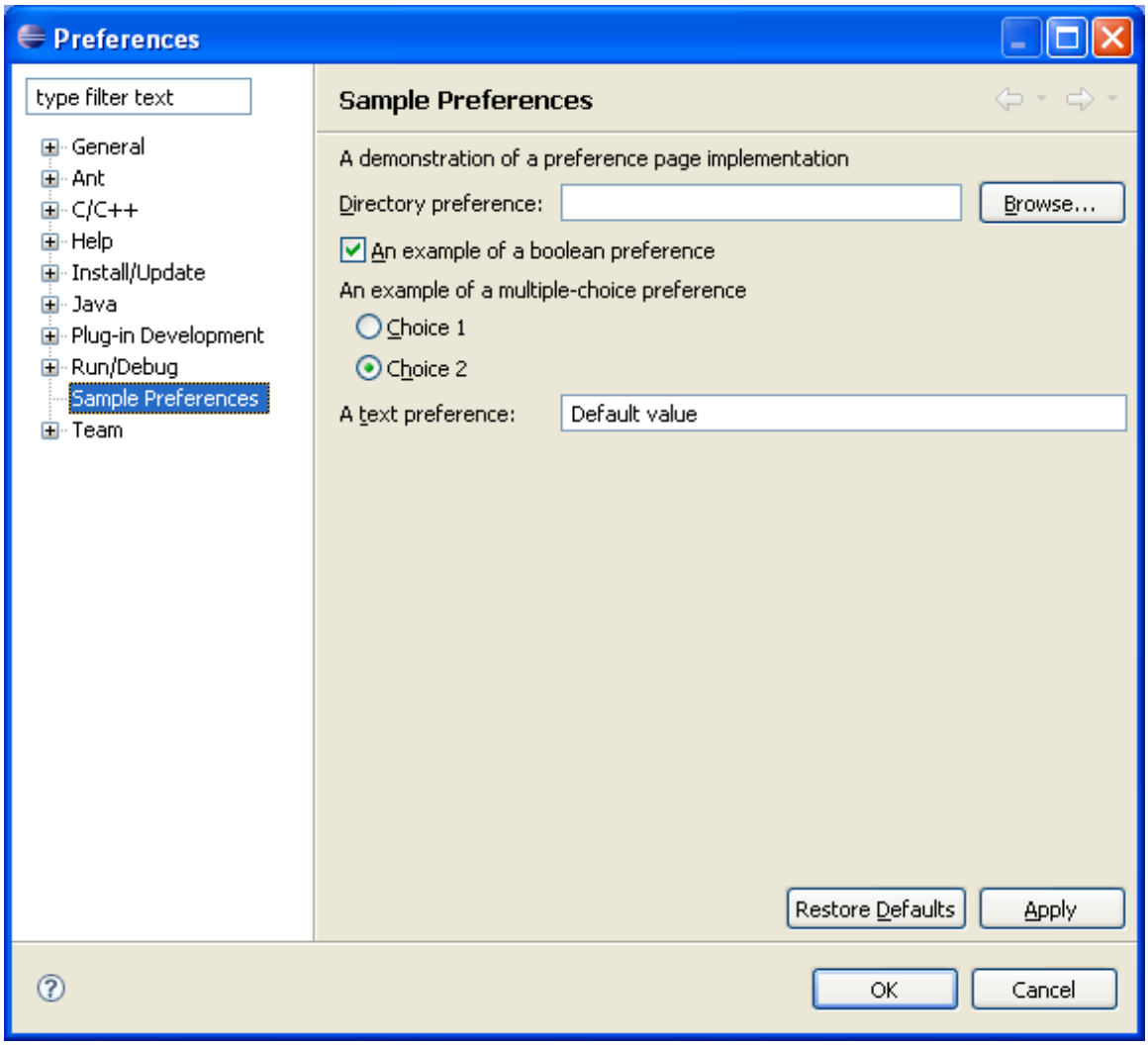
By default, Eclipse loads all applicable plug-ins (i.e. the HelloPlugin and the TestPlugin).



2. To check the plug-in information, click Help > About Eclipse SDK. Click the Plug-in Details.

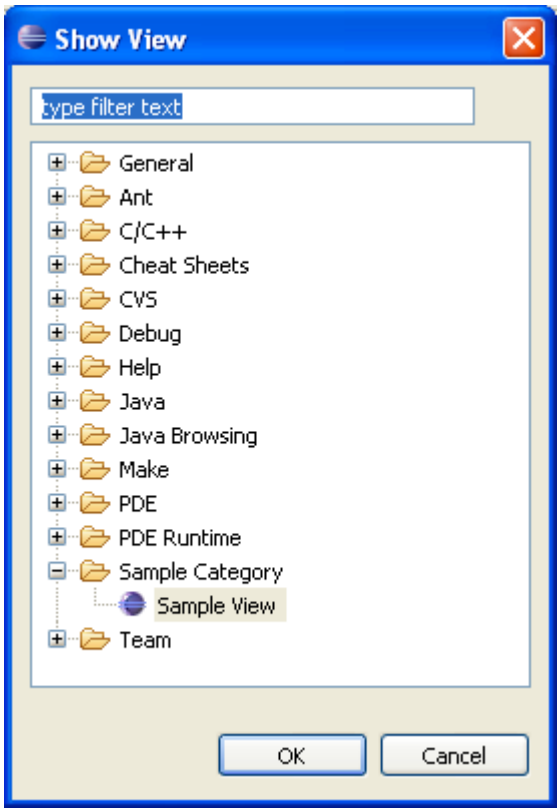


3. The TestPlugin adds 3 contributions: a preference page, an editor, and a view. Click Window > Preferences…

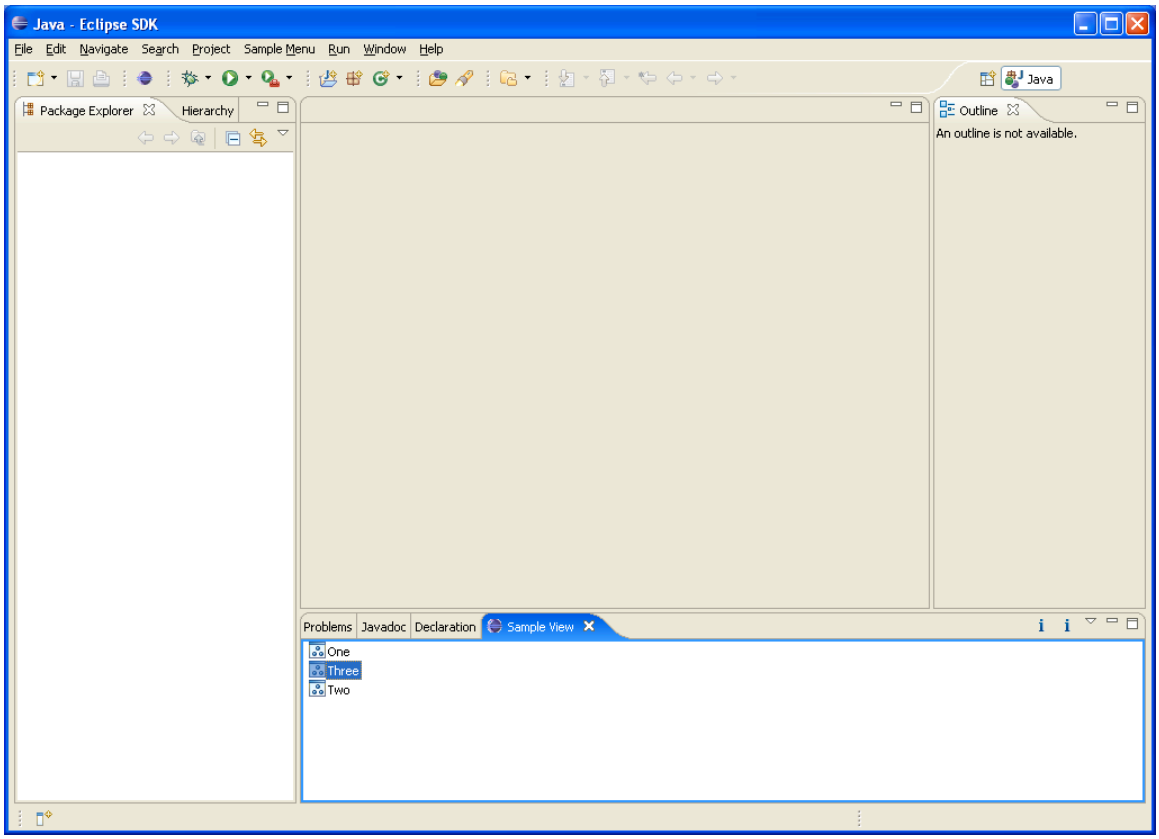


> The Sample Preferences page shows some preferences but they are non-functional. However, it can retain its values between Eclipse Application invocations.

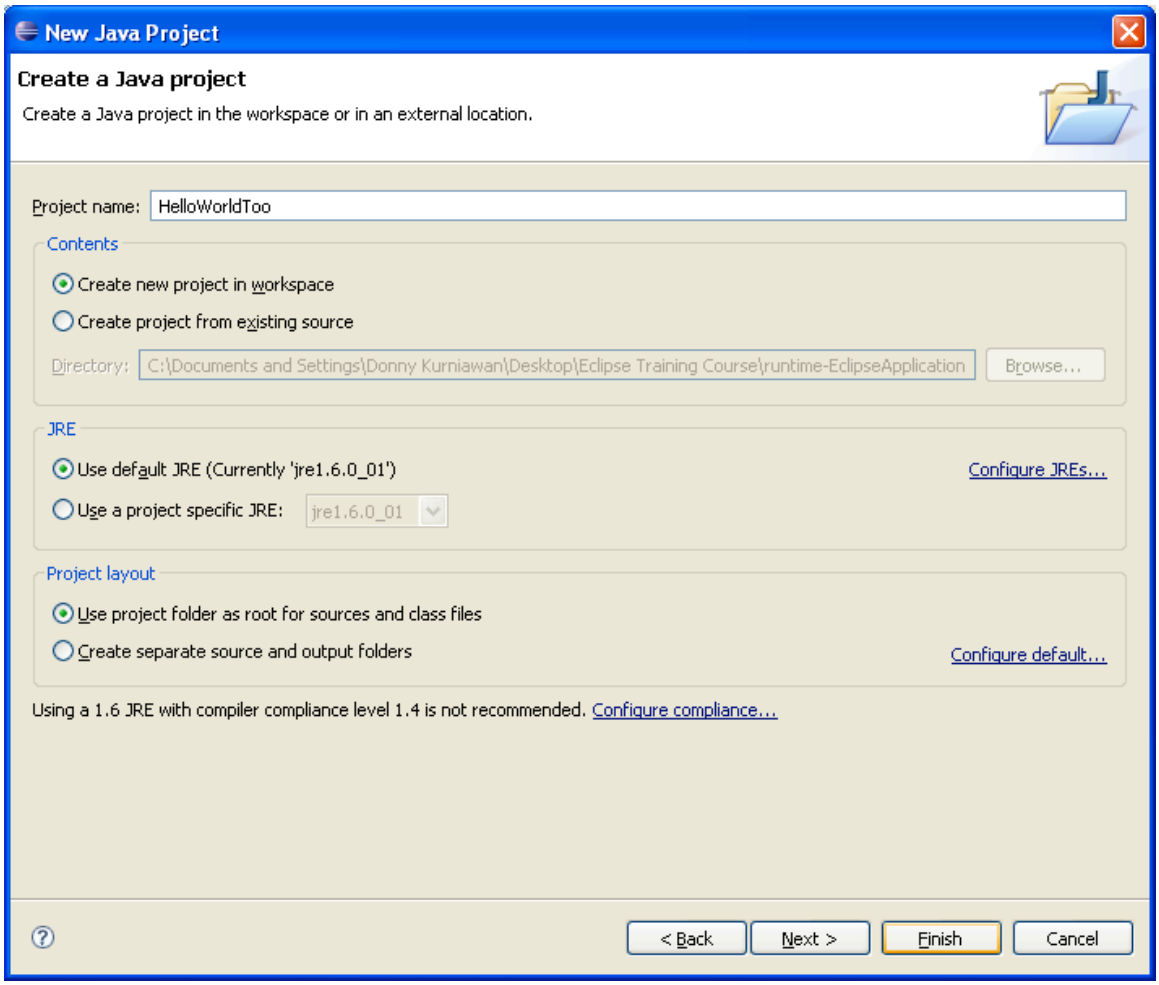
4. Change some values and restart Eclipse Application.
5. In the Eclipse Application, click Window > Show View > Other…



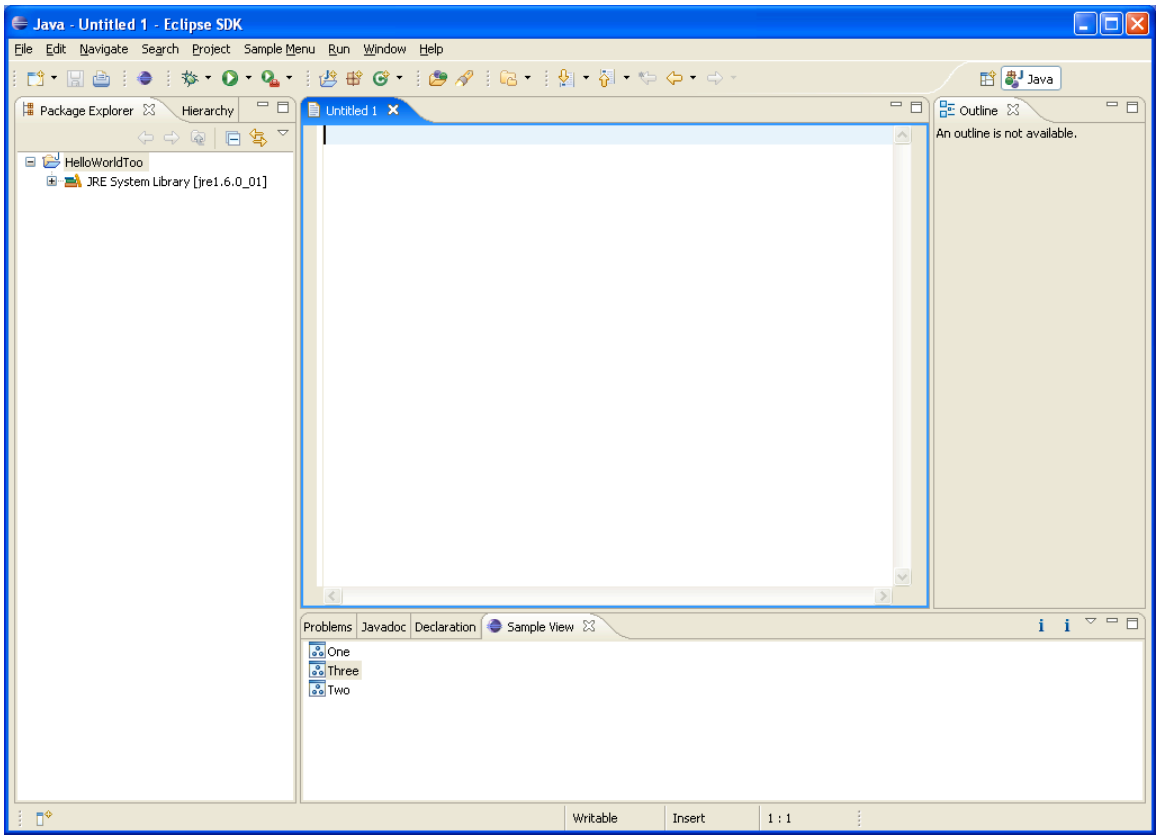
6. Click Sample View and click OK to show the Sample View.



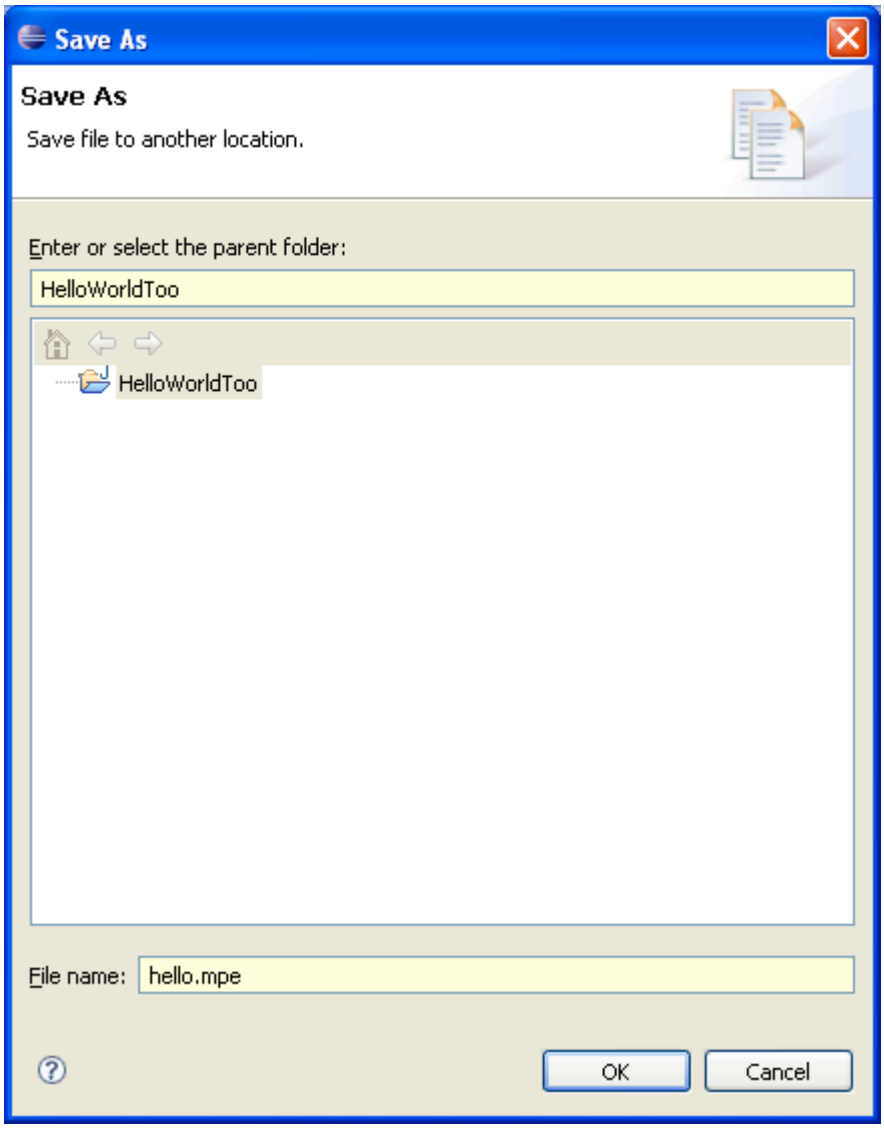
7. Try to click, double-click, and right-click the three items. You can also click the two actions (represented by two i icons) and the pop-up menu (the one with the arrow).
8. Create a new Java project.



9. Click Finish.
10. Select and right-click HelloWorldToo in the Package Explorer view, and select New > Untitled Text File.

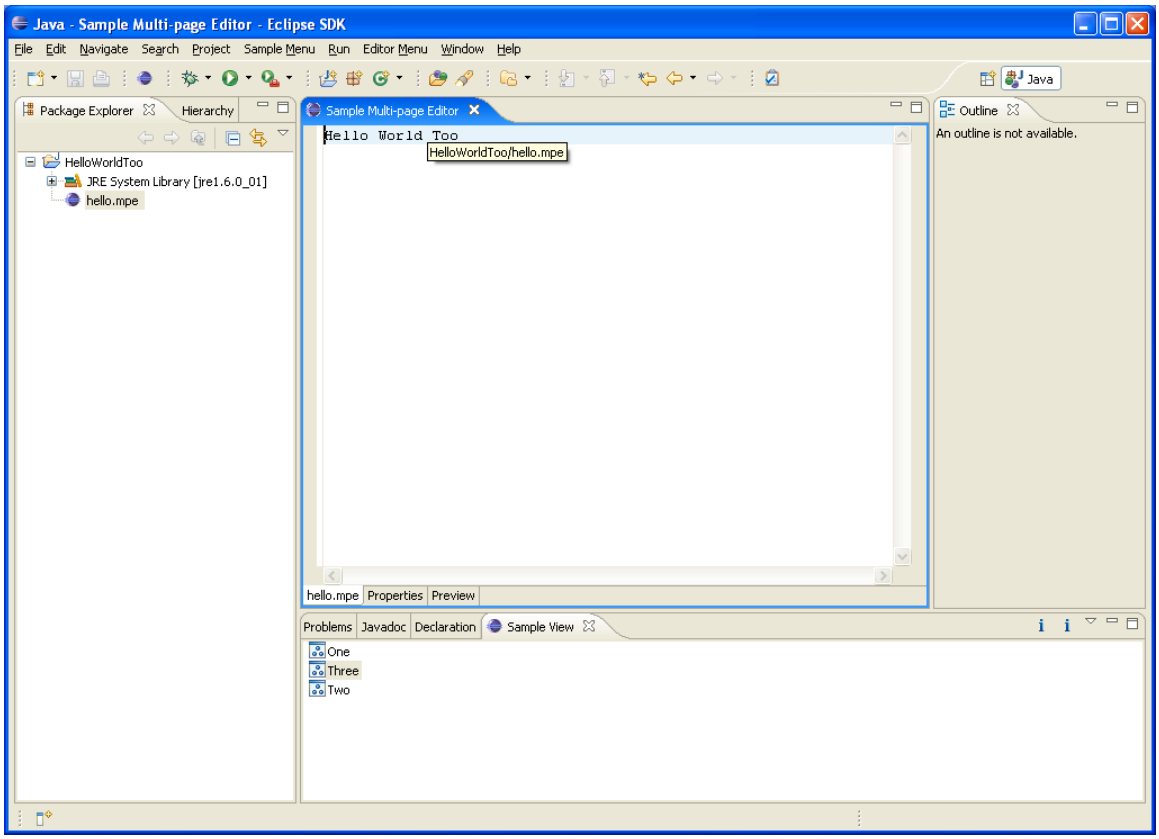


11. Enter the text “Hello World Too” and click File > Save. Name the file hello.mpe.



> The TestPlugin's editor is the default handler for .mpe files. So, if you double-click hello.mpe, Eclipse opens it with the TestPlugin's editor.

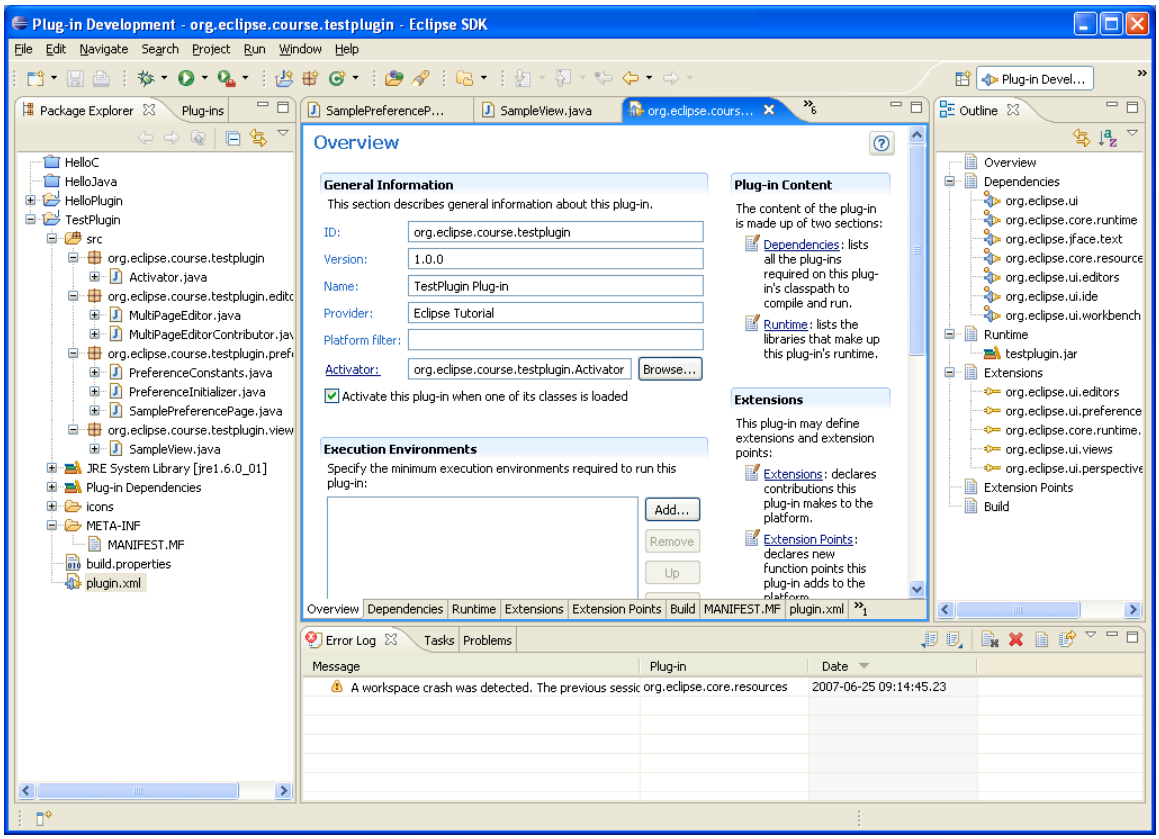
12. Double-click hello.mpe.



13. Click the tabs at the bottom of the editor (hello.mpe, Properties, and Preview). Change the font (on the Properties page), change the text (on the hello.mpe page), and click the Preview tab.

Inspecting the plug-in code

1. Go back to the Eclipse IDE.



2. The wizard-generated code is fully documented. Take a look at the file, make changes, and observe the changes by running the Eclipse Application. Double-click the SampleView.java file, and examine the content.
3. Double-click the MultiPageEditor.java and MultiPageEditorContributor.java, and examine the content.
4. Double-click the PreferenceConstants.java, PreferenceInitializer.java, and SamplePreferencePage.java, and examine the content.
5. Double-click plugin.xml and take a look at the raw text to show the extension point being used.

> Eclipse Help Contents (Help > Help Contents) contains a wealth of information and reference.