Get to know Eclipse User Assistance

Improve the usability of your Eclipse-based apps

Chris Aniszczyk, Software Engineer, IBM, Software Group

Summary: This article introduces the Eclipse User Assistance project and discusses the many ways to deliver quality user assistance for your Eclipse-based applications.

Date: 27 Feb 2007
Level: Introductory
Activity: 774 views
Comments: 0 (Add comments)

★★☆☆☆ Average rating

For any piece of software, the ability to short-circuit a user's learning curve is critical to the adoption and success of that piece of software. Thankfully, Eclipse has the User Assistance project (see Resources), which includes tools for Eclipse-based applications to improve the end users' experience. According to the Eclipse User Assistance project Web site, the project's mission "is to provide for assisting users of Eclipse applications in all phases of the usage cycle. It is not a single workbench artifact, but rather a collection artifacts tailored to provide a particular flavor of assistance."

This article explains some of these artifacts through examples and relevant listings of extension points. Our purpose is not to go into depth into each of these artifacts (see Resources), but rather to serve as a gentle introduction to developers wishing to get started in helping their users.

Tour de Eclipse User Assistance

User Assistance components

Besides the three components discussed here (welcome framework, cheat sheets, and help), the Eclipse User Assistance project is responsible for Eclipse Forms, Samples Gallery, and Eclipse Automation. More information is available in the Eclipse User Assistance Overview.

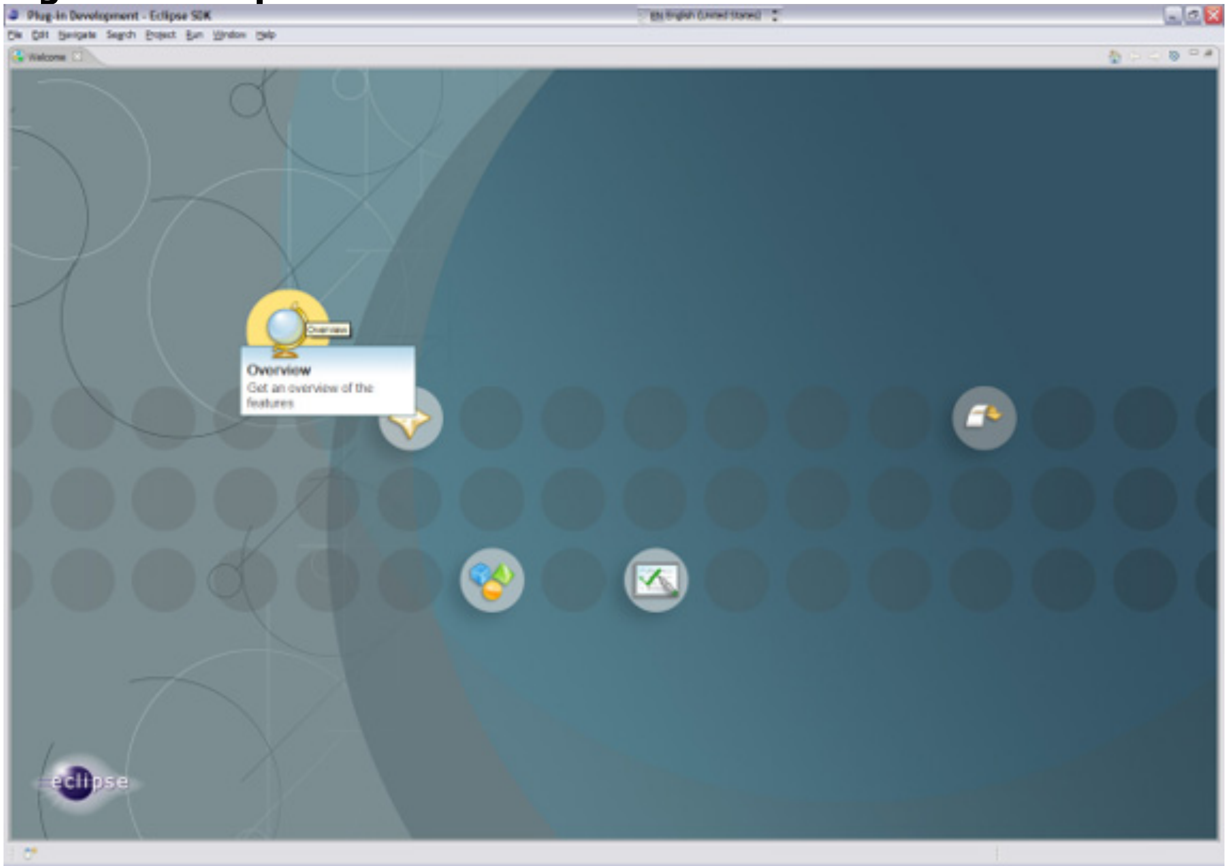
The Eclipse User Assistance project comprises many components, but provides three main tools to assist users:

- Welcome framework
- Cheat sheets
- Help

Welcome framework

Users launching Eclipse are greeted with a welcome screen.

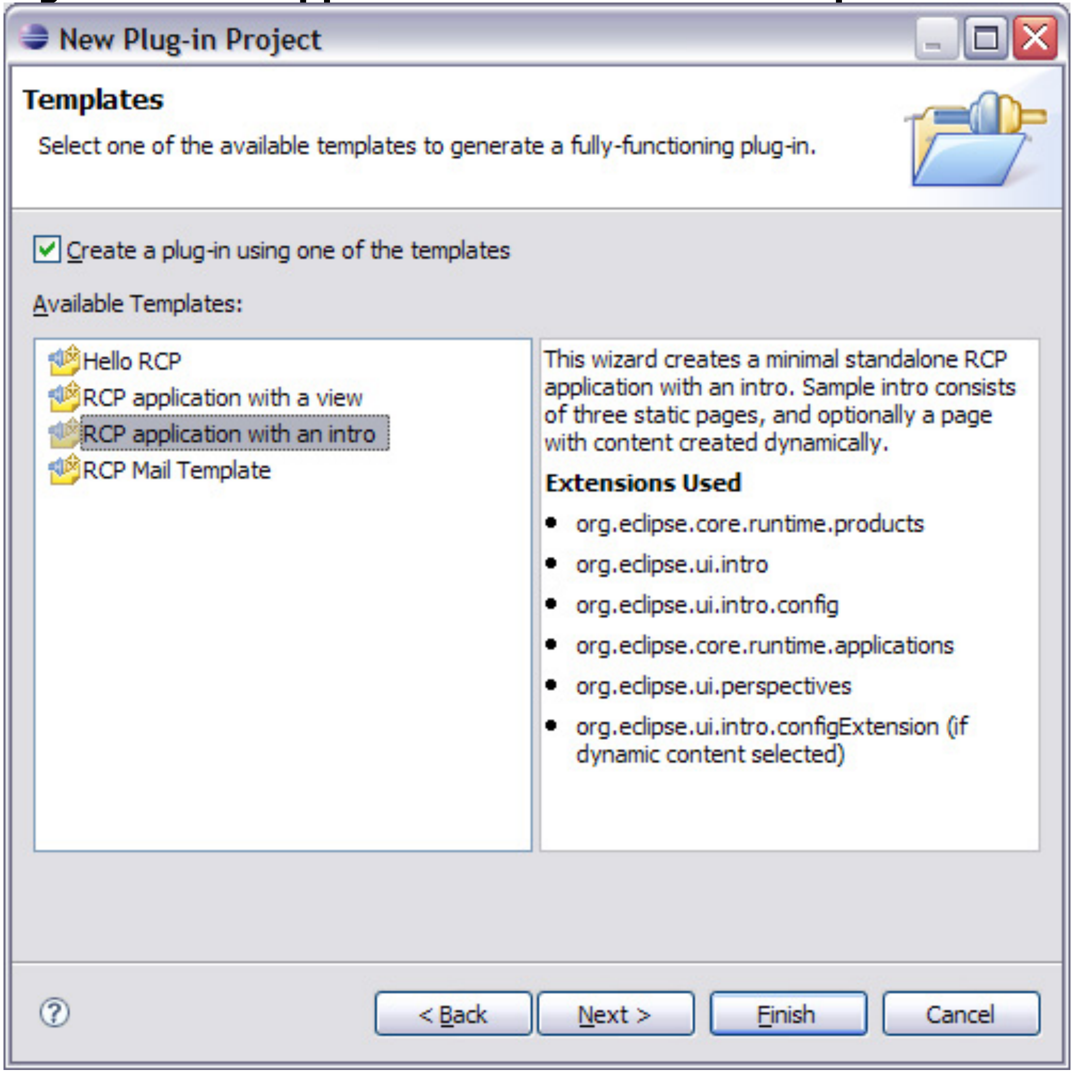
Figure 1. Eclipse welcome


Plug-in Development - Eclipse SDK
File Edit Navigate Search Project Run Window Help
Welcome
Overview
Overview
Get an overview of the features
eclipse

The Eclipse welcome screen provides easy access to tutorials and samples for users. If you want something similar for the users of your application, Eclipse provides the welcome framework. The welcome framework provides the facilities (by way of welcome content) for you to help guide users through tasks like initial setup or provide easy access to online resources. Welcome content can be specified using raw SWT widgets, or more typically, a series of integrated Web pages. To get a good idea of what typical welcome content looks like, we'll use Plug-in Development Environment (PDE) -- using a template -- to create a sample RCP application that provides welcome content.

To take advantage of the PDE template, we need to create a new plug-in project first (**File > New > Project... > Plug-in Project**). It is important to state that we are creating a Rich Client Application in the new plug-in project wizard. Once this is done and we advance to the next page, we can create a project using a set of templates. In our case, we want to select the **RCP application with an intro** template.

Figure 2. RCP application with an intro template



Templates

Select one of the available templates to generate a fully-functioning plug-in.

☑ Create a plug-in using one of the templates

Available Templates:

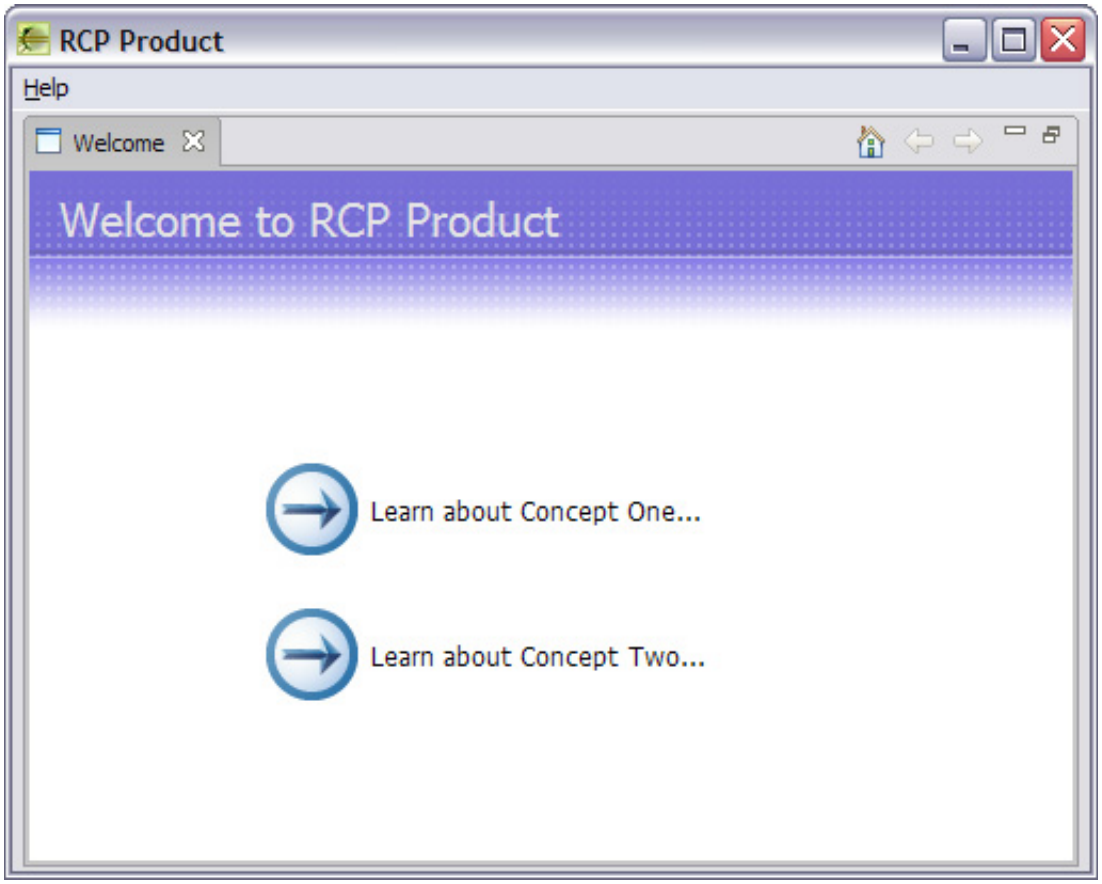
This wizard creates a minimal standalone RCP application with an intro. Sample intro consists of three static pages, and optionally a page with content created dynamically.

Extensions Used

- org.eclipse.core.runtime.products
- org.eclipse.ui.intro
- org.eclipse.ui.intro.config
- org.eclipse.core.runtime.applications
- org.eclipse.ui.perspectives
- org.eclipse.ui.intro.configExtension (if dynamic content selected)

After the plug-in project is created, we can launch the application and see that it pops up to a new welcome screen.

Figure 3. RCP application with welcome screen



If we look at dissecting the generated plug-in, we can see that it uses a few extensions (see Table 1). In particular, we're interested in the use of the org.eclipse.ui.intro.config extension point, whereby the application links to the introContent.xml file. This file contains links to XHTML files, which serve as the welcome content. It's that simple to start building welcome content.

Table 1. Relevant welcome framework extension points

Extension point	Description
org.eclipse.ui.intro	This is used to associate an introduction screen with a specific product.
org.eclipse.ui.intro.config	This is used to create an introductory configuration. Intro configurations are the fundamental building block of Eclipse's welcome framework.
org.eclipse.ui.intro.configExtension	This is used to extend an existing introduction configuration with more content. It also provides a way to separate content from specific configurations so welcome content is more reusable.

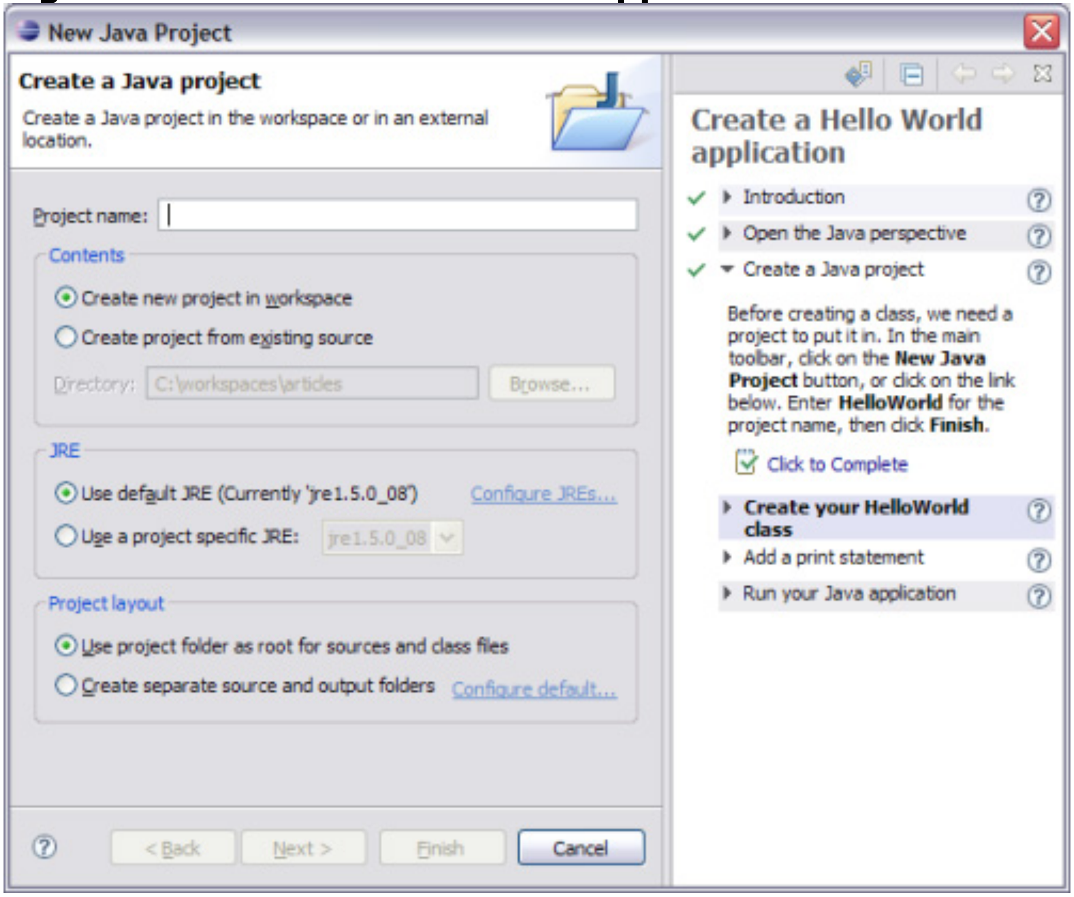
Cheat sheets

A look inside cheat sheets

For a more in-depth understanding of how cheat sheets work, Philipp Tiedt wrote an excellent article titled "Building cheat sheets in Eclipse V3.2" and an accompanying tutorial titled "Building cheat sheets in Eclipse" that includes a thorough example.

Cheat sheets can be thought of as an interactive set of instructions to complete a particular task. They provide a way to guide users through a series of steps with the possibility of scripting interactive actions to simplify what a user needs to do. For example, say you have an application that requires a user to create a special type of project (for example, a photo project) to start working with your application. Cheat sheets allow you to launch your new project wizard automatically, instead of having the user do it manually. A good example of this behavior is the *Create a Hello World application* cheat sheet.

Figure 4. Create a Hello World application cheat sheet



New Java Project

Create a Java project

Create a Java project in the workspace or in an external location.

Project name:

Contents

● Create new project in workspace

○ Create project from existing source

Directory: C:\workspaces\articles

JRE

● Use default JRE (Currently 'jre1.5.0_08')

○ Use a project specific JRE: jre1.5.0_08

Project layout

● Use project folder as root for sources and class files

○ Create separate source and output folders

Create a Hello World application

✓ Introduction

✓ Open the Java perspective

✓ Create a Java project

Before creating a class, we need a project to put it in. In the main toolbar, click on the **New Java Project** button, or click on the link below. Enter **HelloWorld** for the project name, then click **Finish**.

Click to Complete

Create your HelloWorld class

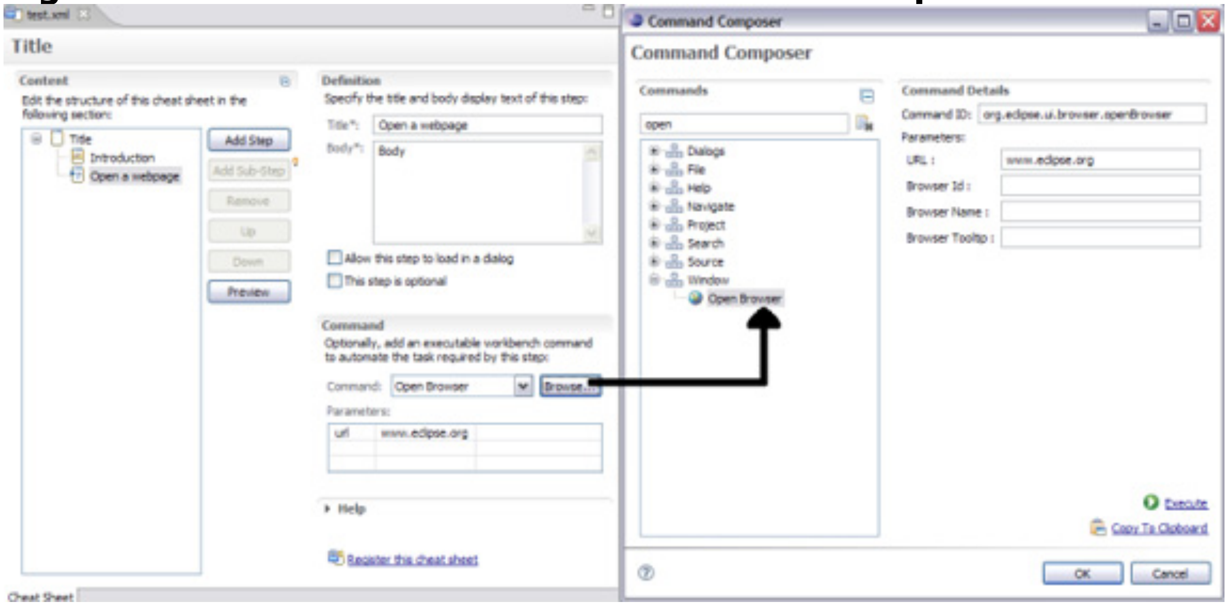
Add a print statement

Run your Java application

Eclipse provides an easy way to author these cheat sheets through a new editor (since Eclipse V3.3) that can be seen in Figure 5 and accessed easily using **File >**

New > Other... > User Assistance > Cheat Sheet. This is much easier for those who don't want to work directly with the XML format of Eclipse cheat sheet files. On top of this new editor, there's a command composer that allows you to easily link with existing Eclipse commands. This effectively removes the guesswork and error-prone nature of working with commands by hand in Eclipse.

Figure 5. Cheat sheet editor and command composer



Edit the structure of this cheat sheet in the following section:

Specify the title and body display text of this step:

Title*: Open a webpage

Body*: Body

□ Allow this step to load in a dialog

□ This step is optional

Optionally, add an executable workbench command to automate the task required by this step:

Command: Open Browser Browse...

Parameters:

url	www.eclipse.org	

open

Command ID: org.eclipse.ui.browser.openBrowser

Parameters:

URL : www.eclipse.org

Browser Id :

Browser Name :

Browser Tooltip :

The relevant extension points for cheat sheets are listed below.

Table 2. Relevant cheat sheet extension points

Extension point	**Description**
org.eclipse.ui.cheatsheets.cheatsheetContent	This is used to register cheat sheet content contributions so they appear in Eclipse's Help menu.
org.eclipse.ui.cheatsheets.cheatSheetItemExtension	This is used to create extensions for cheat sheet items. Effectively, this allows you to introduce new attributes to the cheat sheet file format and have them handled in a desired manner.

Help

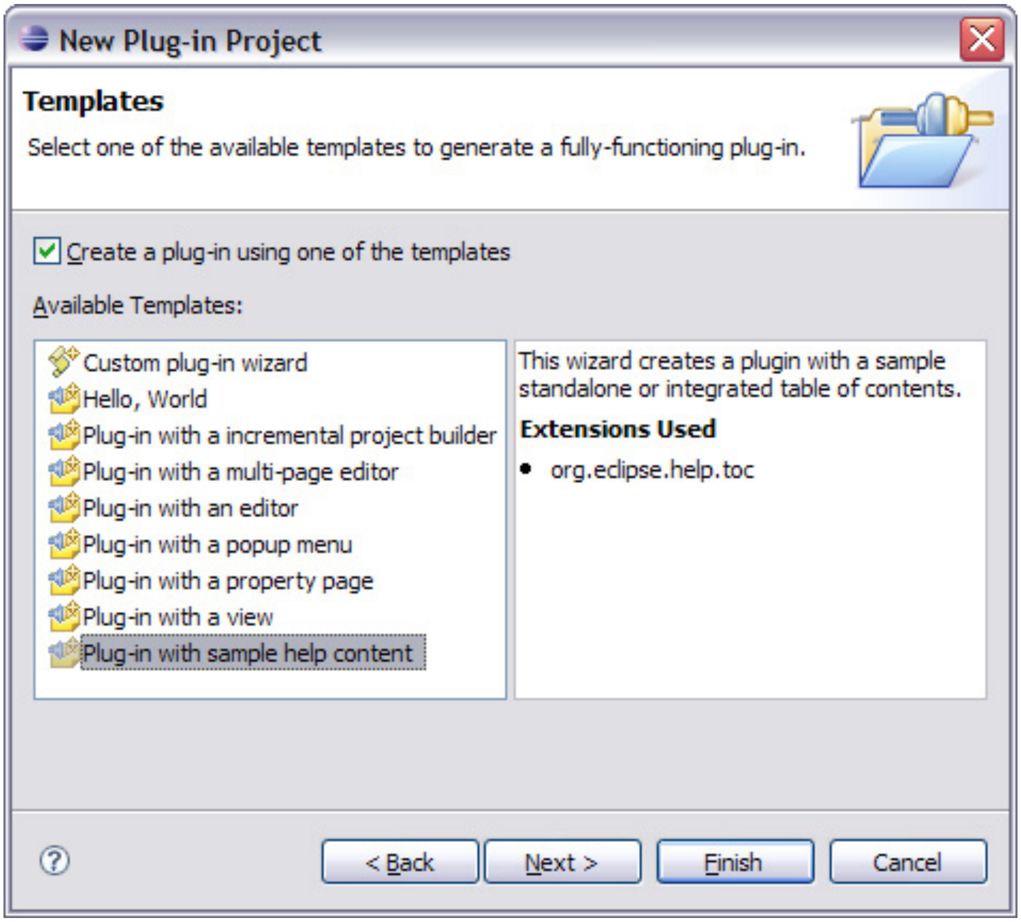
If you worked with Eclipse or IBM® products before, you have most likely stumbled along things known as infocenters. Infocenters are part of the Eclipse help system that offers the ability for you to contribute and organize content and allow for users to search that content quickly. The content can be accessed within Eclipse itself (**Help > Help Contents...**) or via a browser using infocenters as mentioned. Generally, help content is organized into a hierarchy of XML files that link to HTML files. If you already have help content in one of the popular documentation formats like Darwin Information Typing Architecture (DITA) or DocBook, there are transforms available to turn your content into a consumable Eclipse help plug-in.

Help content producers

As of Eclipse V3.3, there is a notion of a help content producer (see the `org.eclipse.help.contentProducer` extension point) that effectively allows other sources of documentation (like DITA or DocBook) to be plugged into the Eclipse help system.

PDE has an excellent template to get you started with what sample help content should look like. By selecting the *Plug-in with sample help content* template (see Figure 6), PDE will create an initial project that has an extension of the `org.eclipse.help.toc` extension point. After the project has been created, simply launch a new runtime workbench and select help (**Help > Help Contents...**) to see your help contribution in action.

Figure 6. Sample help content template



New Plug-in Project

Templates

Select one of the available templates to generate a fully-functioning plug-in.

☑ Create a plug-in using one of the templates

Available Templates:

- Custom plug-in wizard
- Hello, World
- Plug-in with a incremental project builder
- Plug-in with a multi-page editor
- Plug-in with an editor
- Plug-in with a popup menu
- Plug-in with a property page
- Plug-in with a view
- Plug-in with sample help content

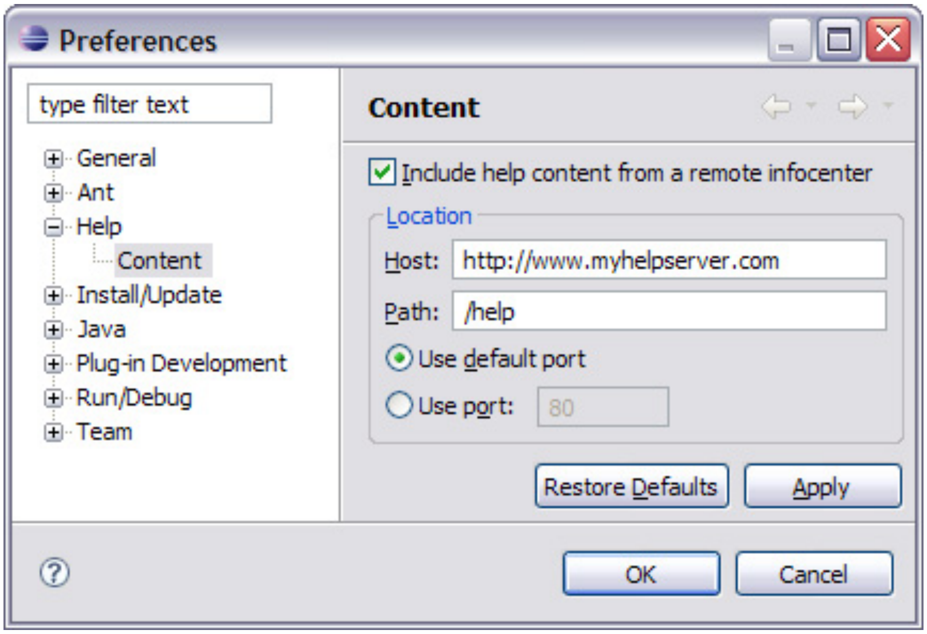
This wizard creates a plugin with a sample standalone or integrated table of contents.

Extensions Used

- org.eclipse.help.toc

Remote help is now an option as of Eclipse V3.3. What this means is you are now able to ship help content separately from your actual application. Instead, you can have a server hosting help content and your application pointing to the help server (see Figure 7). This way, when a user requests help, it will be downloaded when needed. As a developer, you have to be careful about this because if your users are working in an offline environment, this could be detrimental as they won't be able to get help. It's a balance of reducing the initial size of your application download vs. the convenience of working totally in an offline environment.

Figure 7. Sample remote help settings



☑ Include help content from a remote infocenter

Location

Host: http://www.myhelpserver.com

Path: /help

● Use default port

○ Use port: 80

The relevant extension points for Eclipse help are listed below.

Table 3. Relevant cheat sheet extension points

Extension point	Description
org.eclipse.help.toc	This is used to help content by individual plug-ins.
org.eclipse.help.index	This is used to generate an index to be used by help content.
org.eclipse.help.contexts	This is used to help provide context-sensitive help by individual plug-ins.
org.eclipse.help.contentProducer	This is used to provide help content dynamically generated at runtime.
org.eclipse.help.contentExtension	This allows for reusable help content to be reused by the help system.

Conclusion

This article introduced the user assistance facilities of Eclipse and entices you, as a developer, to consider using these facilities to benefit your end users. In the end, good -- and integrated -- documentation can go a long way in reducing the learning curve of your users and the Eclipse User Assistance project provides the tools to get you started.

Acknowledgements

The author would like to thank Dejan Glozic, Curtis D'Entremont, Lee Anne Kowalski, and Chris Goldthorpe for their UA expertise in reviewing this article.

Resources

Learn

- Check out the Eclipse User Assistance project.
- Learn more about the Eclipse Plug-in Development Environment (PDE).
- Read about developing cheat sheets using Phillip Tiedt's developerWorks article "Building cheat sheets in Eclipse V3.2" and the tutorial "Building cheat sheets in Eclipse."
- Learn how to work with the Eclipse help system by reading "Writing Documentation and Help for Eclipse Projects and Plug-ins."
- Check out the Eclipse documentation wiki.
- Visit DocBook.org and DITA Web sites to learn more about the two popular documentation formats.
- Learn about DITA by reading "Introduction to the Darwin Information Typing Architecture" and "Specializing topic types in DITA."
- See the DITA "Frequently Asked Questions about the Darwin Information Typing Architecture," published by developerWorks.
- Learn about authoring with Eclipse from an Eclipse Foundation article titled "Authoring with Eclipse."
- Learn about Eclipse User Assistance by attending EclipseCon 2007.
- Stay up to date on what's happening in the Eclipse community by visiting Planet Eclipse.
- For an excellent introduction to the Eclipse platform, see "Getting started with the Eclipse Platform."
- Check out the "Recommended Eclipse reading list."
- Browse all the Eclipse content on developerWorks.

- Expand your Eclipse skills by checking out IBM developerWorks' Eclipse project resources.
- To listen to interesting interviews and discussions for software developers, check out developerWorks' podcasts.
- Stay current with developerWorks' technical events and webcasts.
- Check out upcoming conferences, trade shows, webcasts, and other Events around the world that are of interest to IBM open source developers.
- Visit the developerWorks Open source zone for extensive how-to information, tools, and project updates to help you develop with open source technologies and use them with IBM's products.
- Visit Safari Books Online for a wealth of resources for open source technologies.

Get products and technologies

- Download the Eclipse Platform and get started with Eclipse now.
- Check out the latest Eclipse technology downloads at IBM alphaWorks.
- Innovate your next open source development project with IBM trial software, available for download or on DVD.

Discuss

- The Eclipse Platform newsgroups should be your first stop to discuss questions regarding Eclipse. (Selecting this link will launch your default Usenet news reader application and open eclipse.platform.)
- The Eclipse newsgroups has many resources for people interested in using and extending Eclipse.
- Participate in developerWorks blogs and get involved in the developerWorks community.

About the author

Chris Aniszczyk is a software engineer at IBM Lotus focusing on OSGi related development. He is an open source enthusiast at heart, and he works on the Gentoo Linux distribution and is a committer on a few Eclipse projects (PDE, ECF, EMFT). He's always available to discuss open source and Eclipse over a frosty beverage.