Documenting your project using the Eclipse help system

Build easy-to-use and searchable help documentation

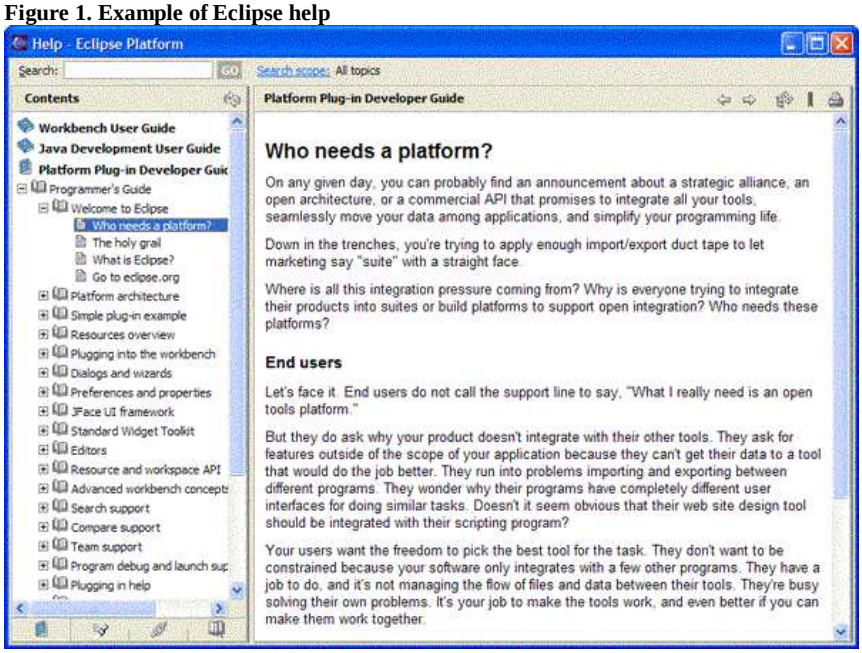
Arthur Barr, Software engineer, IBM

Summary: The Eclipse Platform, which provides a very powerful IDE, includes its own help system based on an XML table of contents referencing HTML files. What isn't immediately obvious is that you don't have to write Eclipse plug-ins to use it. Any project can use a cut-down version of the platform to provide professional, easy-to-use, and searchable documentation. This documentation system has been successfully used on a number of IBM projects, including those as large as the IBM® WebSphere® Application Server.

Date: 29 Jan 2004
Level: Introductory
Also available in: Russian Japanese

When you access the Eclipse help system (through **Help > Help Contents**), you are actually starting up an embedded Apache Tomcat server. A window based on a Web browser is then opened, pointing to the correct page on that server (see Figure 1). Documentation is provided with a collapsible index on the left side and HTML documentation on the right, and can be searched (thanks to the Apache Lucene search engine). Since Tomcat is used, you are not limited to HTML. For example, you can use JSPs to make your documentation change dynamically (though we will discuss later a possible reason to avoid doing this).

Figure 1. Example of Eclipse help



The Hello, World of documentation plug-ins

Documentation is split into "books," and you can have as many books as you like in one instance of the help system. Each book is written as an Eclipse plug-in, but thankfully, the work involved here is minimal. To write a simple plug-in, you will need a plugin.xml file to describe your plug-in, which should look like Listing 1.

Listing 1. Plug-in definition

```
<plugin name="Sample Documentation Plug-in" id="com.ibm.sample.doc"
   version="1.0.0" provider-name="IBM">
   <extension point="org.eclipse.help.toc">
      <toc file="toc.xml" primary="true" />
   </extension>
</plugin>
```

Change the plug-in's `name`, `id`, `version`, and `provider-name` to values appropriate to your project. The extension point of `org.eclipse.help.toc` identifies this as a plug-in to the help system. The file toc.xml is referenced as being the table of contents for this plug-in. This file will provide the data for the hierarchical information in the left pane of the Eclipse help window. A simple file contains something like that shown below.

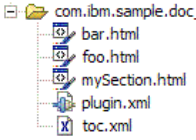
Listing 2. Table of contents definition

```
<toc label="Sample Documentation">
    <topic label="My Section" href="mySection.html">
        <topic label="Foo" href="foo.html"/>
        <topic label="Bar" href="bar.html"/>
    </topic>
</toc>
```

Packaging the plug-in

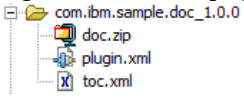
Each topic element is represented in the final documentation by an entry in the navigation list. These topics can be nested (they can contain more topics), and each one points to an HTML or JSP file. Once you've done this, all you need to do is package everything in the structure shown in Figure 2 (notice that the plug-in directory name matches the `id` and `version` attributes of the plug-in defined in the plugin.xml).

Figure 2. Plug-in directory structure



As a convenience, and to reduce file size, Eclipse allows you to keep all your actual documentation (the HTML files) in a ZIP file called doc.zip, so you could use the directory structure shown in Figure 3.

Figure 3. Alternative plug-in directory structure

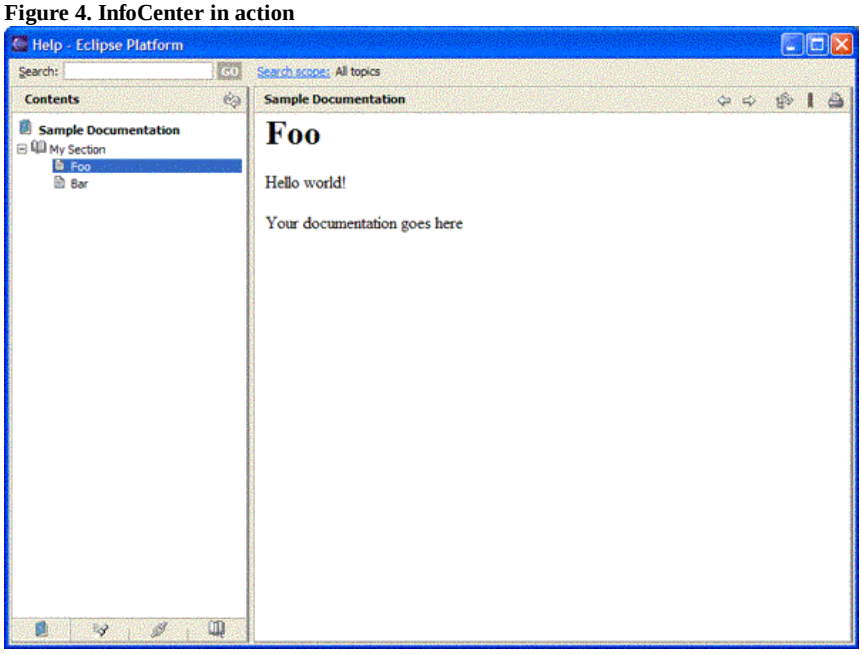


Viewing your documentation

The easiest way to test your plug-in is to simply drop the entire directory (as above) into the plugins directory of an installed Eclipse Platform, then launch Eclipse and select **Help > Help Contents**. You will get a help window with your plug-in added (similar to the one in Figure 1).

Using the IDE is all very well for testing, but to be useful without the IDE, the documentation needs to be more accessible, so what we really want is to run a process in the background that lets us connect to it with a browser. This mode of operation is known as an InfoCenter (see Figure 4). Instructions for starting an InfoCenter process (basically Apache Tomcat) are included with the Eclipse help system documentation (see Resources). Note that there also instructions on how to pare down the Eclipse system to give you just the bits you need.

Figure 4. InfoCenter in action



Handling large tables of contents

If your project has more than a few people working on it or has a large documentation set, updating a single table of contents (toc.xml) file can become impractical. You can change this by adding a `link` element into your topic in the main toc.xml file (see Listing 3 for an example).

Listing 3. Table of contents definition

```
<toc label="Sample Documentation">
    <topic label="My Section" href="mySection.html">
        <topic label="Foo" href="foo.html"/>
        <topic label="Bar" href="bar.html">
            <link toc="bar-toc.xml" />
        </topic>
    </topic>
</toc>
```

The file bar-toc.xml is just another table of contents, and should take exactly the same format as any other toc.xml file. When the documentation is viewed, there will be no difference between using this method and simply including the additional `topic` elements directly.

Generating a stand-alone documentation set

Of course, using the Eclipse help system is all well and good if you don't mind distributing the 20-plus MB of code required, but this isn't realistic for smaller projects. Hosting an InfoCenter on a central server allows people to connect remotely. People receive all the benefits of using the Eclipse help system (such as searching), but people without connectivity are left stranded. So, in addition to using a hosted InfoCenter, it's useful to include the plain HTML in a downloadable package. As long as you haven't used any server-side technologies such as JSPs, you can easily generate an HTML table of contents to replace the XML one used by Eclipse. Which is why we have eXtensible Stylesheet Language Transformations (XSLT).

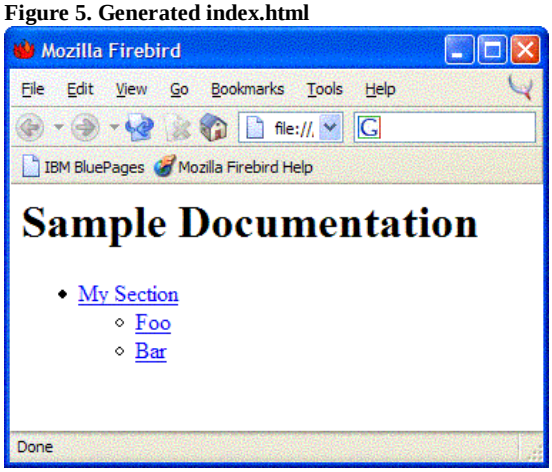
XSLT is a technology used to transform one form of XML to another, such as XHTML (a stricter, XML version of HTML). XSLT provides a rich and powerful language to perform transformations, and is the topic of many books and articles on its own, so we won't go into detail here. Listing 4 shows an example of a simple transformation of a toc.xml file, rendering the entries as nested HTML lists. Note that this particular transformation creates a single HTML file for the contents of the whole documentation set, which will be unwieldy for large numbers of files. Therefore, this XSLT will not work if you have split your table of contents across multiple files.

Listing 4. Sample XSLT to generate HTML table of contents

```
<?xml version="1.0"?>
<xsl:stylesheet
   version="1.1"
   xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:output method="html" indent="no" encoding="ISO-8859-1" />

<xsl:template match="toc">
   <html>
      <head />
      <body>
         <h1><xsl:value-of select="@label" /></h1>
         <ul>
            <xsl:apply-templates />
         </ul>
      </body>
   </html>
</xsl:template>

<xsl:template match="topic">
   <li>
      <xsl:choose>
         <xsl:when test="@href">
            <!-- Only add a hyperlink when there is something to link to ->
            <xsl:element name="a">
               <xsl:attribute name="href">
                  <xsl:value-of select="@href" />
                  </xsl:attribute>
               <xsl:value-of select="@label" />
            </xsl:element>
         </xsl:when>
         <xsl:otherwise>
            <xsl:value-of select="@label" />
         </xsl:otherwise>
      </xsl:choose>

      <!-- If there are any nested topics, then start a new sub-list ->
      <xsl:if test="descendant::topic">
         <ul>
            <xsl:apply-templates/>
         </ul>
      </xsl:if>
   </li>
</xsl:template>

</xsl:stylesheet>
```

Processing the toc.xml file through an XSLT processor, such as Apache Xalan using the above XSLT, yields an HTML file that looks something like Figure 5, when viewed with a browser.

Figure 5. Generated index.html



Conclusion

Using the Eclipse help system is a fairly painless way to develop professional-looking, searchable documentation that will amaze your friends and colleagues. If you don't have a requirement for a stand-alone documentation set, then you don't even need to go near XSLT; you can write just two simple XML files and be on the road to documentation happiness. Off you go.

Download

Name	Size	Download method
source.zip		HTTP

Information about download methods

Resources

Learn

- Check out the latest Find more information about the Eclipse help project at the Eclipse help project Web site.
- See Installing the help system as an InfoCenter. Note that these instructions are for the Eclipse V2.1 platform.
- Read "XSL Formatting Objects (XSL-FO) basics" to learn the basics of XSL Formatting Objects (XSL-FO), a powerful, flexible XML vocabulary for formatting data.
- Read "XSL-FO advanced techniques" to learn advanced techniques for working with XSL Formatting Objects (XSL-FO) for formatting data.
- Take a look at the IBM WebSphere Application Server V5.1 InfoCenter to see the help system in action on a large project.
- Check out the "Recommended Eclipse reading list."
- Browse all the Eclipse content on developerWorks.
- New to Eclipse? Read the developerWorks article "Get started with Eclipse Platform" to learn its origin and architecture, and how to extend Eclipse with plug-ins.
- Expand your Eclipse skills by checking out IBM developerWorks' Eclipse project resources.
- To listen to interesting interviews and discussions for software developers, check out check out developerWorks podcasts.
- For an introduction to the Eclipse platform, see "Getting started with the Eclipse Platform."
- Stay current with developerWorks' Technical events and webcasts.
- Watch and learn about IBM and open source technologies and product functions with the no-cost developerWorks On demand demos.
- 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.

Get products and technologies

- Download the Apache Xalan XSLT processor (there are Java and C versions, but if you are using a Java 2 V1.4 runtime environment, an XSLT processor is included as standard).
- Check out the latest Eclipse technology downloads at IBM alphaWorks.
- Download Eclipse Platform and other projects from the Eclipse Foundation.
- Download IBM product evaluation versions, and get your hands on application development tools and middleware products from DB2®, Lotus®, Rational®, Tivoli®, and WebSphere®.
- 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 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

Arthur Barr is a software engineer working at the IBM Hursley development labs in the UK. He has put the musings of this article into use on the Business Integration for Games project, on which he should probably be working at the moment. You can contact Arthur at arthur.barr at uk.ibm.com.

Close [x]

developerWorks: Sign in

IBM ID:
Need an IBM ID?
Forgot your IBM ID?

Password:
Forgot your password?
Change your password

Keep me signed in.

By clicking **Submit**, you agree to the developerWorks terms of use.

Submit Cancel

The first time you sign into developerWorks, a **profile** is created for you. **Select information in your developerWorks profile is displayed to the public, but you may edit the information at any time**. Your first name, last name (unless you choose to hide them), and display name will accompany the content that you post.

All information submitted is secure.

Close [x]

Choose your display name

The first time you sign in to developerWorks, a profile is created for you, so you need to choose a display name. Your display name accompanies the content you post on developerWorks.

Please choose a display name between 3-31 characters. Your display name must be unique in the developerWorks community and should not be your email address for privacy reasons.

Display name: (Must be between 3 – 31 characters.)

By clicking **Submit**, you agree to the developerWorks terms of use.

Submit Cancel

All information submitted is secure.

Error: Submission failed. Please try again.

Average rating (51 votes)

1 star 1 star
2 stars 2 stars
3 stars 3 stars
4 stars 4 stars
5 stars 5 stars

Submit

Add comment:

Sign in or register to leave a comment.

Note: HTML elements are not supported within comments.

Notify me when a comment is added1000 characters left

Post

Be the first to add a comment