**MADCAP FLARE 2024 r2**

Eclipse Help

CONTENTS

Introduction

The Eclipse Help output type is designed specifically for content developers who want to use Flare to author an Eclipse Help plug-in. A plug-in lets you extend the existing Help contents that come with the Eclipse Integrated Development Environment (IDE). It requires an additional installation of the latest Java Runtime Environment (JRE) and Java version. This output type is particularly useful when you want to create dynamic, searchable content that integrates into your Eclipse environment as a platform plug-in.

Once the components are in place, you must also set up the Eclipse Help options in the Target Editor in Flare. See "Creating an Eclipse Help Plug-In" on page 8 and "Specifying Eclipse Options" on page 21.

The Eclipse Help output lets you generate Eclipse Help plug-ins using your Flare content as its source material. It displays your output content in the Eclipse Help Viewer.
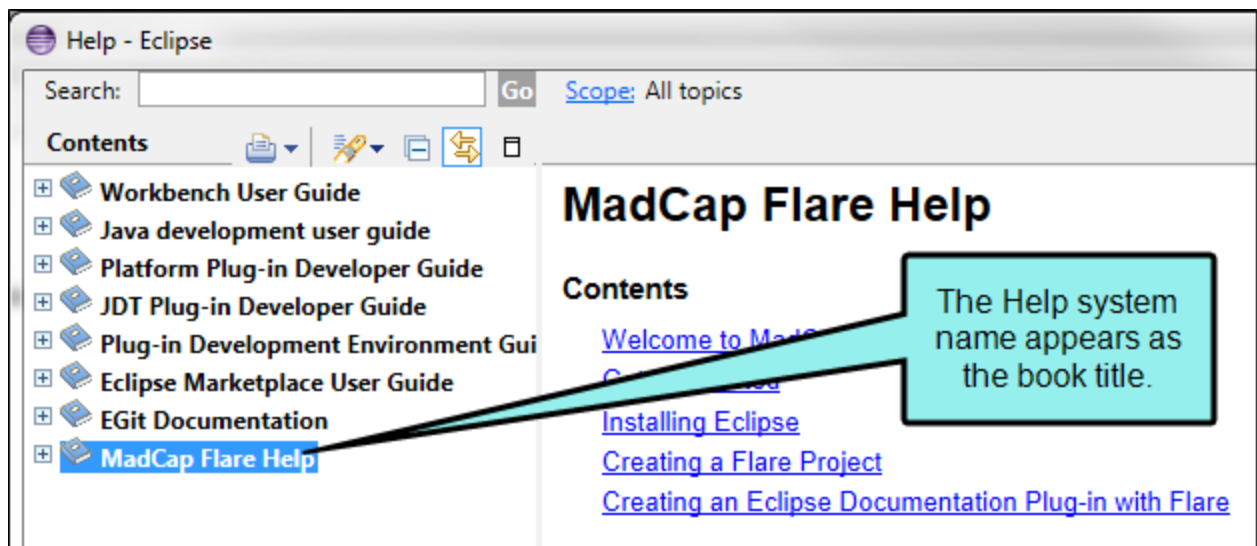
The Eclipse Help Viewer uses an embedded Jetty server to provide your users with navigation features similar to other online viewers. Jetty is a Java-based web server implementation that tightly integrates with Eclipse and other open source projects.
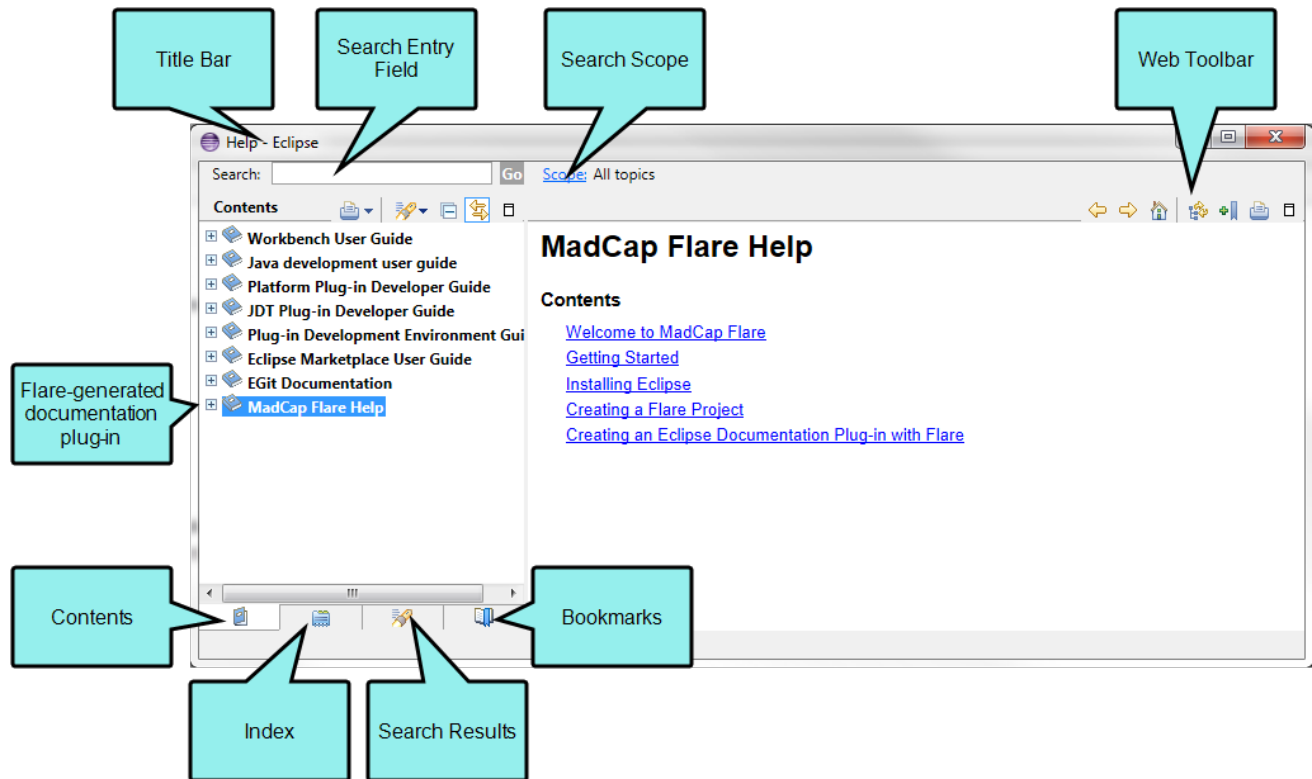
The viewer's features and user interface must be customized in Eclipse. To learn about your customization options, visit help.eclipse.org.

❙ Key Features

Following are some of the key features of the Eclipse Help format:

- **Eclipse Help Plug-In and Extension Points** This output type is particularly useful when you want to create dynamic, searchable content that integrates into the Eclipse workbench as a platform plug-in. Flare generates an XML-based plug-in file (plugin.xml) that specifies the context Help (csh.xml), index (index.xml), and table of contents (toc.xml) files that Eclipse needs to load your Flare content in the Eclipse Help system.

- **Eclipse Help Viewer and Dynamic Help Window** You can generate the Eclipse Help output for viewing with the Eclipse Help Viewer and the Eclipse Dynamic Help window, which displays context-sensitive Help and search results.

- **Distribute Server-Based or Local Content** Your Eclipse content can be distributed over the web via an Eclipse InfoCenter server or on local systems with a pared down version of Eclipse, which provides the search capabilities and skin for the Eclipse Help Viewer.

- **Publish Content as a Collection of Files or Single JAR File** This format lets you publish the output for the Eclipse Help plug-in as a series of files in a folder or bundled in a single Java archive (JAR) file.

> 📝 **NOTE** Eclipse provides its own breadcrumb and topic toolbar support. See the Platform Plug-in Developer Guide from Eclipse at help.eclipse.org.

> 📝 **NOTE** If your Eclipse Help target contains non-web safe images (e.g., WMF, EMF, BMP, TIF, TIFF), be sure to select **Generate "web-safe"images** in the Advanced tab of the Target Editor.

> 📝 **NOTE** The Eclipse Viewer cannot display PDFs opened from links such as those in TOCs.

> 📝 **NOTE** When you create a TOC intended for Eclipse output, you should make sure all items are linked. Unlinked items may not display in the output.

NOTE Eclipse does not use the Flare skin, so there will be no glossary pane in the output. However, you can insert a Glossary proxy (which is normally used for print-based outputs) as an alternative.

NOTE Java bits and Eclipse bits must match in order to view Eclipse output in Flare. For example, Java 64-bit and Eclipse 64-bit must both be used/installed.

Creating an Eclipse Help Plug-In

The Eclipse Help output type gives you the ability to create a plug-in for the Eclipse Help system. A plug-in lets you extend the existing Help contents of the Eclipse IDE by including your own documentation.

If you plan to create an Eclipse plug-in with Flare, you must install some additional Eclipse components before getting started. Once the components are in place, you must also set up the Eclipse Help options in the Target Editor in Flare.

This chapter discusses the following:

❙ Installing Eclipse

Before you begin, you must install the required Java Runtime Environment (JRE) and Java version on the system where you will be building and publishing the Eclipse Help output.

How to Download and Install the JRE

The JRE for Eclipse is part of the Eclipse IDE for Java EE Developers package distribution and can be downloaded from the eclipse.org website. This distribution contains all of the features that you need to build an Eclipse plug-in with Flare.

1. In Flare, add or open an Eclipse Help target.

2. In the Target Editor, select the **Eclipse Options** tab.

3. Under **Viewing Options**, click **Download**.
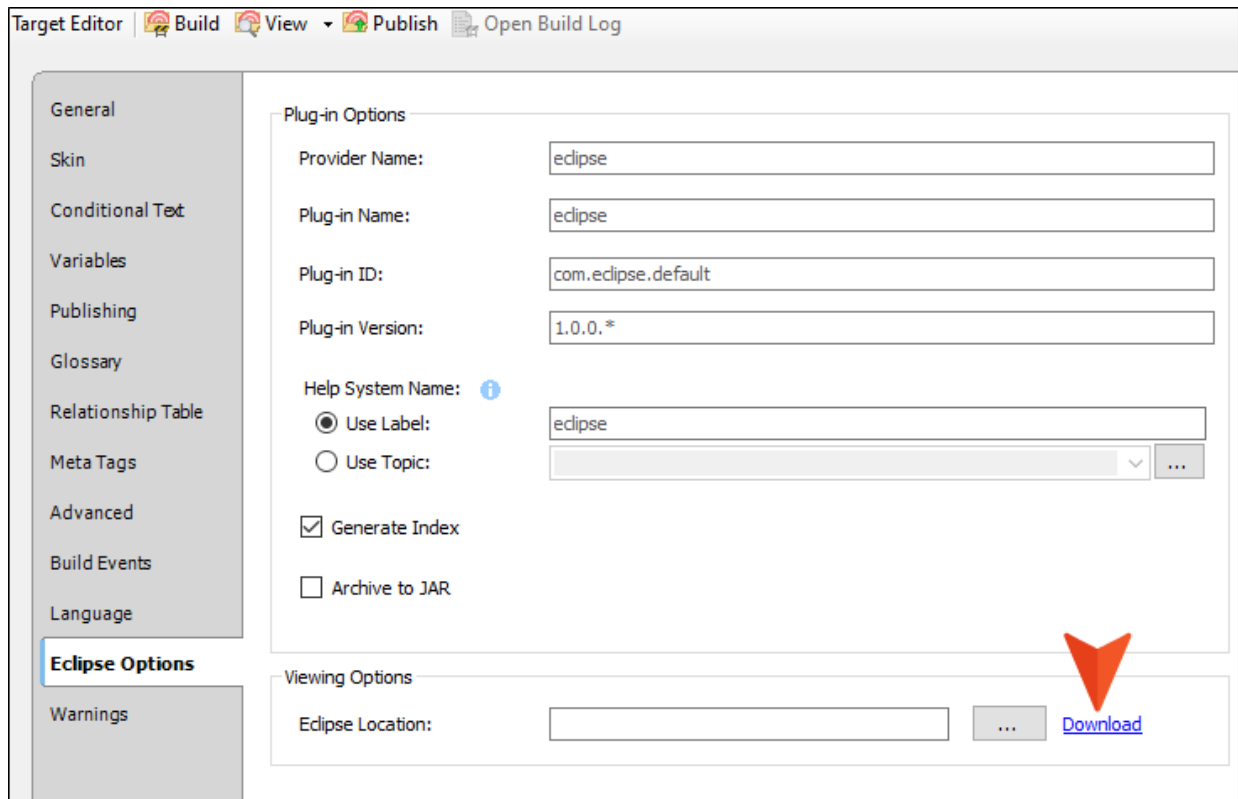
This opens the Eclipse IDE for Java EE Developers page on the eclipse.org website.

4. Under the **Download Links** area, download the appropriate Eclipse IDE package for your operating system:

- **Windows 32-bit** Select this link if your computer is running a 32-bit operating system.

- **Windows 64-bit** Select this link if your computer is running a 64-bit operating system.

5. On the Eclipse downloads page for your operating system, click the **Download** link.

   This downloads a ZIP file to your computer. Depending on your internet bandwidth and connection speed, you should plan for the download to take a minimum of several minutes.

6. When the download is complete, create a folder on your computer where you can decompress the download in Windows.

NOTE For best results, create the folder in a root directory, such as C:\Eclipse. Users have reported Eclipse issues when extracting the files to a mapped network drive or to a location with spaces in the path.
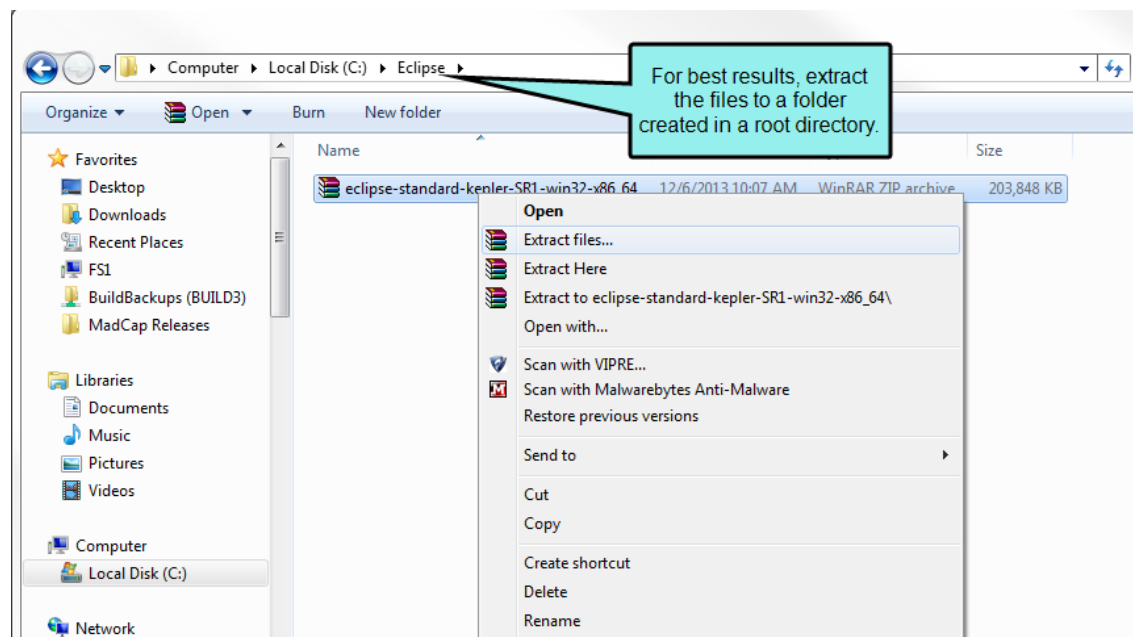
7. Use a decompression utility to extract the ZIP file that you downloaded into the folder you created.

> ☐ **NOTE** The Eclipse installation documentation details known issues when using the built-in decompression utility that is included with all versions of Windows. To learn more, visit this link: http://wiki.eclipse.org/Eclipse/Installation.

8. After extracting the zipped files to the folder you created, you can move the files to a more appropriate location such as Program Files\Eclipse or Program Files (x86)\Eclipse. Please note that these are just examples. You might use a different location than Program Files or Program Files (x86).
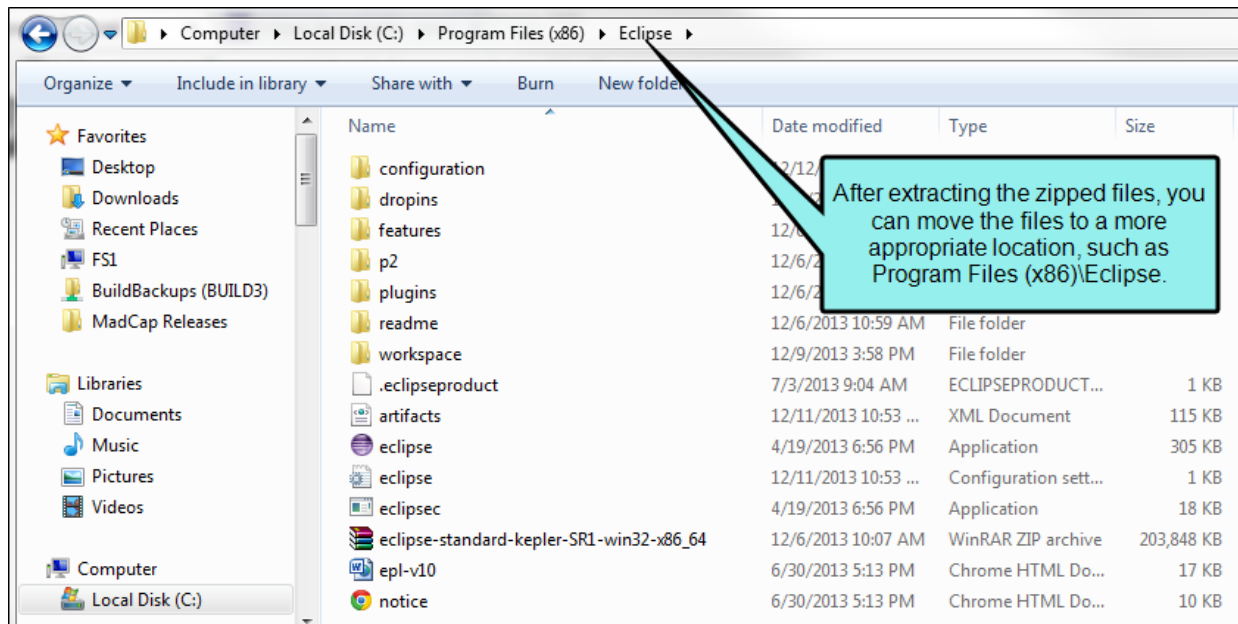
9. Continue by downloading and installing Java on your system.

How to Download and Install Java

To generate the Eclipse Help output type, you must have Java Version 7 (Update 45) or later installed on your system.

1. In a web browser, navigate to the Java downloads page at http://java.com/en/download/manual.jsp.

2. Under the Windows area, download the appropriate Windows Offline file for the 32- or 64-bit operating system:

   - **Windows 64-bit** Download the file named "jre-7u45-windows-zx64.exe."

   - **Windows 32-bit** Download the file named "jre-7u45-windows-i586.exe."

3. Double-click the EXE file. The Open File-Security Warning dialog appears.

4. In the dialog, click **Run**. In a few moments, the Java Setup - Welcome dialog opens.

5. At the Welcome to Java message, click **Install**.

6. At the installation complete message, click **OK**. The Verify Java Version page should now be open in your web browser.

7. In the web page, click the **Verify Java Version** button.

8. If your web browser needs permissions to run the verification process, click the **Run this time** button. In a few moments, a Java Detection message appears.

9. In the Java Detection message, click **Run**.

10. If the verification is successful, close your web browser.

Continue by starting the Eclipse platform for the first time.

How to Start Eclipse for the First Time

After installing the required JRE and Java version on your computer, do the following:

1. In Windows, navigate to the folder where you moved the decompressed Eclipse files.

2. Double-click the file named **eclipse.exe**. The Open File - Security Warning dialog appears.

3. In the dialog, click **Run**. In a few moments, the Workspace Launcher appears.

4. In the Select a Workspace dialog, choose a workspace folder. Then click **OK**.

5. When the Java - Eclipse window appears, click **Help > Check for Updates**.

6. If any updates are found, it is recommended that you install them. If no updates are found, click **OK**.

❚ Creating an Eclipse Plug-In With Flare

To create your Eclipse plug-in with Flare, add a new Eclipse Help targetand make sure you complete the Eclipse options (see "Specifying Eclipse Options" on page 21). Then build and publish your output.

The Eclipse Help output type is viewed in the Eclipse Help Viewer (or if you are including context-sensitive Help, the Eclipse Dynamic Help window).

Eclipse Help Configurations

The viewer is powered by a java-based web server instance that ships with the Eclipse IDE. This makes it possible for you to distribute the published Eclipse Help output using these Eclipse Help configurations:

Eclipse Help Configuration	Eclipse Plug-In Reference	Description
Standalone Help Application	org.eclipse.help.standalone	This is a standalone Help system, which contains a pared down version of the Eclipse IDE. It includes all of the Eclipse Help content, as well as most of the features included in the integrated Eclipse Help system. This is best used when developing a standalone Help system for applications not based on the Eclipse framework.
Infocenter	org.eclipse.help.Infocenter	This is a server-based Help system, where the Eclipse plug-in that you create is installed on a server. This gives your users the ability to access your output over the internet (or over an intranet).

| Eclipse Help Configuration	Eclipse Plug-In Reference	Description
Extension Points	org.eclipse.help.toc org.eclipse.help.index org.eclipse.help.contexts	If your programmers are developing a rich client application, the Eclipse plug-in that you generate with Flare lets you integrate your content using the Eclipse Help extension points (i.e., table of contents, keyword index, and HTML content pages) .

How to Build and View an Eclipse Help Target

1. After specifying the options in your Eclipse Help target (see "Specifying Eclipse Options" on page 21), click the **Build** button in the Target Editor.

2. When prompted if you want to view the generated output, click **Yes**.

3. When the message appears to notify you that the Help Viewer extension site must be refreshed, click **OK**.

> 🗒 **NOTE** If you do not want to see the notification message when building Eclipse Help in the future, clear the "Show this dialog" check box before clicking **OK**.

This launches the Help - Eclipse window.

Understanding the Eclipse Help Output

When you build and publish your Eclipse Help output, Flare creates a collection of output files. The files provide Eclipse with the instructions it needs to load your Flare content in the Eclipse Help Viewer. Your plug-ins files can be distributed as a collection of files in a folder.
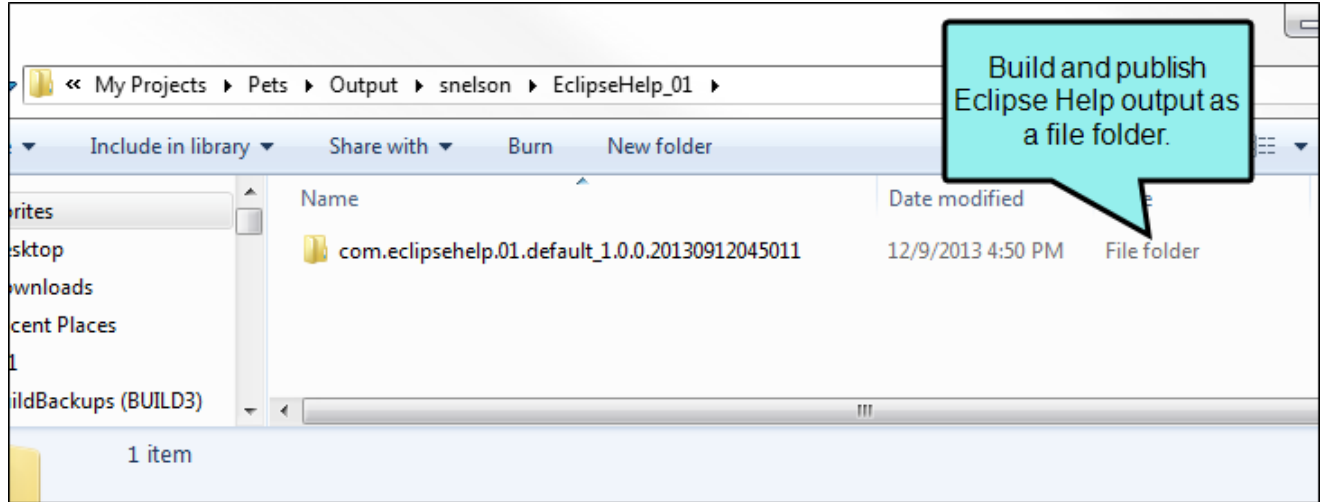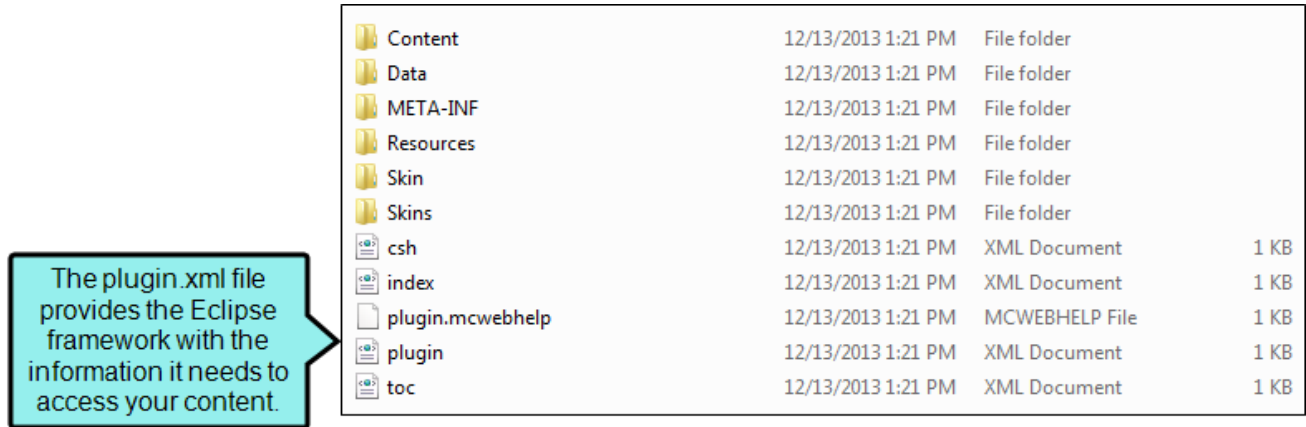

They can also be bundled in a single, executable Java archive (JAR) file.
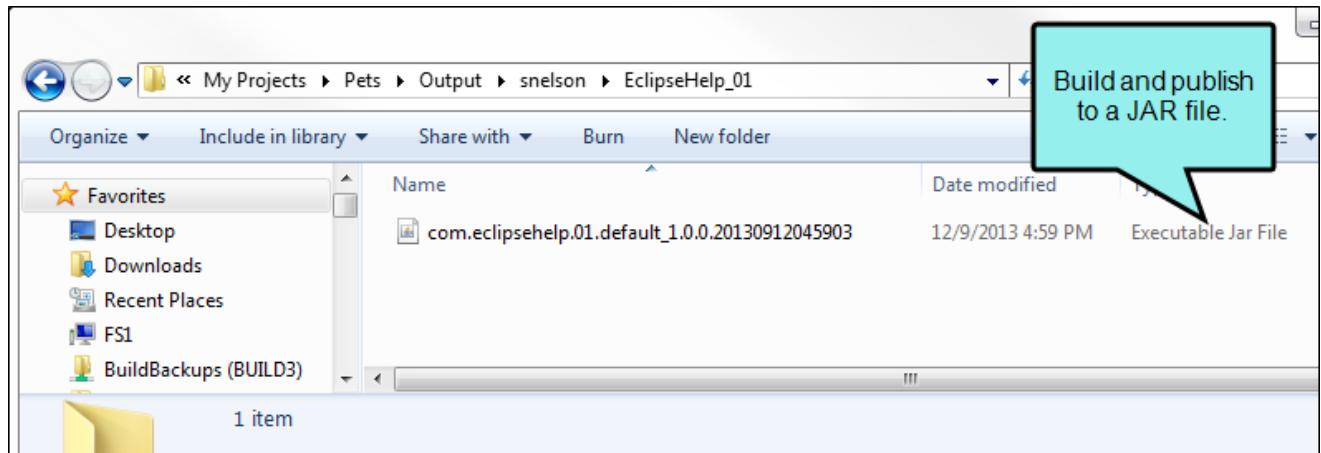
The Eclipse Help output contains the following files and resources:

XML-Based Plug-In File and Extension Points

The plug-in file (plugin.xml) generated by the Eclipse Help output type provides the Eclipse framework with the XML-based instructions it needs to load and access the files you published with Flare.

The content of the file specifies the different extension points for your plug-in. These extension points are required for your output to load in Eclipse and include the following:

- **org.eclipse.help.contexts** If you have created context-sensitive Help, this extension point registers the Flare-generated context file (i.e., csh.xml) for Eclipse.

- **org.eclipse.help.index** If you have included index keywords in your target, this extension point registers the Flare keyword index file (i.e., index.xml) for Eclipse.

- **org.eclipse.help.toc** If you have created a table of contents in your target, this extension point registers the Flare-generated table of contents file (i.e., toc.xml) for Eclipse.

Following is an example of a plugin.xml file generated for the Eclipse Help output:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<?eclipse version="3.0"?>
<plugin provider-name="EclipseHelp_01" name="EclipseHelp_01" ver
  <extension point="org.eclipse.help.contexts">
      <contexts file="csh.xml"/>
  </extension>
  <extension point="org.eclipse.help.index">
      <index file="index.xml"/>
  </extension>
  <extension point="org.eclipse.help.toc">
      <toc file="toc.xml" primary="true"/>
  </extension>
  <extension point="org.eclipse.help.toc"/>
</plugin>
```

Eclipse extension point definitions.

Content and Resource Folders

The output also includes the following content and resource folders:

- **Content** This folder contains the HTML content files that you authored in Flare. If you do not want your content files in a separate Content folder, you can omit it when publishing your target.

- **Data** This folder contains XML and JavaScript instructions for your Flare output.

- **META-INF** When you build your output and generate a JAR file (see "Specifying Eclipse Options" on page 21), the Eclipse Help output creates a file named "Manifest.MF." This file contains information about the files that are packaged in your archive.

- **Resources** This folder contains JavaScript instructions for your Flare output.

- **Skin** This folder contains components related to the user interface of the Eclipse Help Viewer.

- **Skins** This folder also contains components related to the user interface of the Eclipse Help Viewer.

▌Distributing the Eclipse Help Output

You can choose whether to publish the Eclipse Help output that you generate with Flare as a collection of files in a folder or as a single executable file. Depending on your development environment, you will use the Eclipse framework to deliver your output in one of three modes: Infocenter, Standalone, or Rich Client Platform (RCP).

Infocenter (Information Center)

In Eclipse, Infocenter mode lets you deliver server-based documentation via the Help-Eclipse window. This mode is typically selected when developing Eclipse-based applications (e.g., client- and web-based applications).

With this mode, your developers must install an Eclipse information center on a server. Once the Eclipse server is installed, you can upload the Eclipse Help output that you generated in Flare to the server. There are a variety of ways that you can connect the plug-in that you generated in Flare. To learn more, see the Eclipse documentation at http://help.eclipse.org.

Standalone Help System

In Eclipse, you can deliver documentation via the Eclipse Help Viewer as a standalone application. This application gets packaged and installed locally on a user's system and includes a light version of Eclipse that contains only the Eclipse Help system.

Standalone mode is typically selected when developing applications that are not Eclipse-based. To learn more, see the Eclipse documentation at http://help.eclipse.org.

RCP

In Eclipse, this mode lets you deliver documentation as part of an Eclipse RCP application. Typically, documentation is accessed using a Help menu. Unlike the Infocenter and standalone mode, the optional RCP workbench supports Eclipse context Help and the Eclipse Help Viewer.

This mode is typically selected when developing applications that are not Eclipse-based. To learn more, see the Eclipse documentation at http://help.eclipse.org.

Specifying Eclipse Options

If you are sending output to the Eclipse Help format, you can access Eclipse options in the Target Editor.

How to Specify Eclipse Options

1. Open a target that is using the Eclipse Help format.

2. Select the **Eclipse Options** tab.

3. Complete the options as necessary:

SET THE PLUG-IN OPTIONS

Use the **Plug-in Options** area to define information for the org.eclipse.help API, which provides the core Help system for the Eclipse platform.

- **Provider Name** Type the desired provider name. For example, type `MadCap Software`.

- **Plug-in Name** Type a name for your plug-in. For example, name it `MadCap Flare`.

- **Plug-in ID** Type the desired identifier for your plug-in. Use the `com.<your plug-in name>.<default>` naming convention. For example, type `com.madcapflare.help`.

- **Plug-in Version** Set the desired version number for the plug-in. The default value is `1.0.0.*`. This corresponds to the version of Eclipse that your plug-in is targeted for. In the following example, the plug-in is being targeted for Eclipse Standard SDK version `2.0.1.20130919`.

- **Help System Name** Choose one of these options:
  - **Use Label** Choose this option to type a label for your plug-in. This corresponds to the label for the book icon in the left-navigation pane of the Help-Eclipse window.
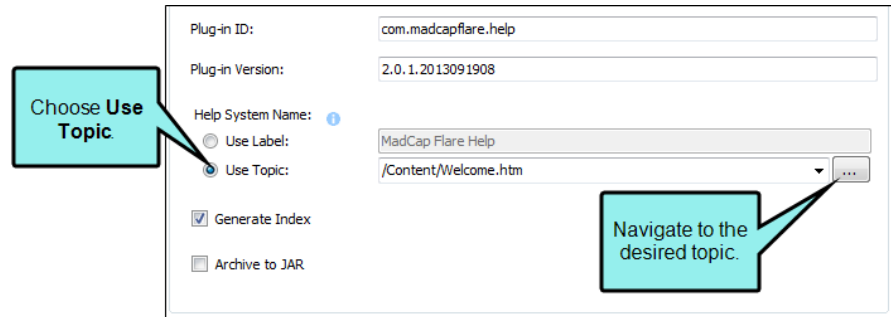
  ⭐ **EXAMPLE** If you choose the **Use Label** option…

  Choose **Label**.

  Type the desired label.

  … your Help system name appears as the label:
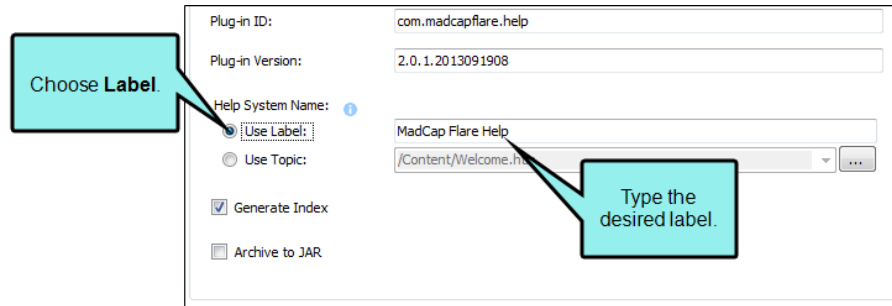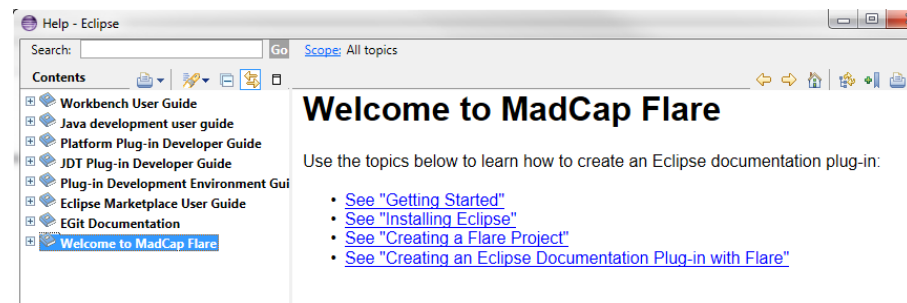
  The Help system name appears as the book title.
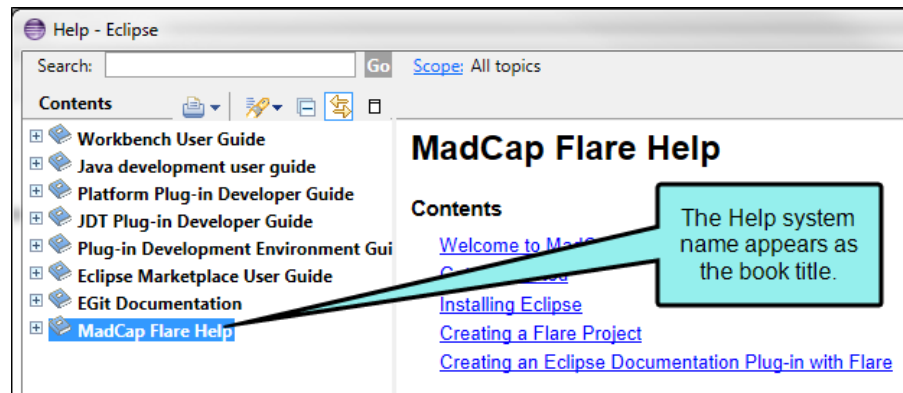
- **Use Topic** Choose this option. In the Link to Topic dialog, navigate to the topic that you want to use as the Help system label.


... your Help system name appears as the label:

- ▪ **(Optional) Generate Index** This check box is selected by default.

- ■ **(Optional) Archive to JAR** This packages the published output as a single, executable file with the Java archive (JAR) file name extension. This check box is blank by default.

☆ **EXAMPLE** If the **Archive to JAR** check box is selected, the Eclipse Help output publishes the plug-in as a single JAR file.
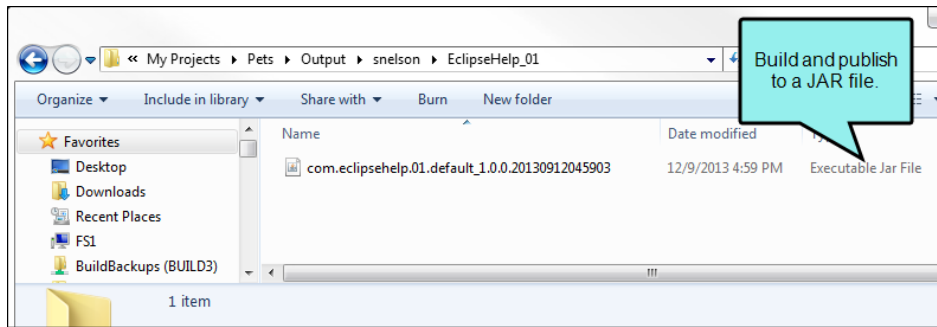
If the check box is not selected, the Eclipse Help output packages the plug-in as a collection of files in a folder.
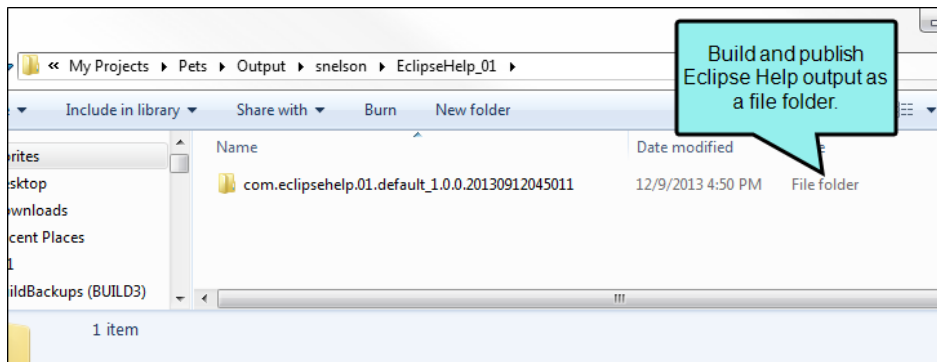
SET THE VIEWING OPTIONS

In the **Viewing Options** area, do the following:

a. Make sure you have completed the steps for installing Eclipse. See "Creating an Eclipse Help Plug-In" on page 8.

b. After downloading the ZIP file to your computer, click the browse button. Then navigate to the Eclipse application file that you downloaded to your computer.

4. Click ⊞ to save your work.

PDFs

The following PDFs are available for download from the online Help.

▎Tutorials

Autonumbers Tutorial

Back-to-Top Button Tutorial

Context-Sensitive Help Tutorial

Custom Toolbar Tutorial

eLearning Tutorial—Basic

eLearning Tutorial—Advanced

Getting Started Tutorial

Image Tooltips Tutorial

Lists Tutorial

Meta Tags Tutorial

Micro Content Tutorial—Basic

Micro Content Tutorial—Advanced

Responsive Output Tutorial

Single-Sourcing Tutorial

Snippet Conditions Tutorial

Styles Tutorials

Tables Tutorial

Word Import Tutorial

Cheat Sheets

Context-Sensitive Help Cheat Sheet

Folders and Files Cheat Sheet

Learning & Development Cheat Sheet

Lists Cheat Sheet

Micro Content Cheat Sheet

Print-Based Output Cheat Sheet

Search Cheat Sheet

Shortcuts Cheat Sheet

Structure Bars Cheat Sheet

Styles Cheat Sheet

User Guides

Accessibility Guide

Analysis and Reports Guide

Architecture Guide

Autonumbers Guide

Branding Guide

Condition Tags Guide

Context-Sensitive Help Guide

Eclipse Help Guide

eLearning Guide

Getting Started Guide

Global Project Linking Guide

HTML5 Guide

Images Guide

Import Guide

Indexing Guide

Key Features Guide

Lists Guide

MadCap Central Integration Guide

Meta Tags Guide

Micro Content Guide

Navigation Links Guide

Plug-In API Guide

Print-Based Output Guide

Project Creation Guide

QR Codes Guide

Reviews & Contributions With Contributor Guide

Scripting Guide

Search Guide

SharePoint Guide

Skins Guide

Snippets Guide

Source Control Guide: Git

Source Control Guide: Perforce Helix Core

Source Control Guide: Subversion

Source Control Guide: Team Foundation Server

Styles Guide

Tables Guide

Tables of Contents Guide

Targets Guide

Template Pages Guide

Templates Guide

Topics Guide

Touring the Workspace Guide

Transition From FrameMaker Guide

Translation and Localization Guide

Variables Guide

Videos Guide

What's New Guide