17

Integrating Eclipse Plug-ins

17.1 Introduction

Wind River Workbench is based on the Eclipse Integrated Development Environment (IDE). This means that you can incorporate third-party Eclipse-based modules into the Workbench development suite without having to recompile or reinstall it. Such modules are called *plug-ins*, and they can add new capabilities and tools to your copy of Workbench.

Many developers enjoy creating new plug-ins and sharing them with other Eclipse users. There are many Web sites with useful free or open source tools and applications that you can download and incorporate into your Workbench installation.

17.2 Finding New Plug-ins

In addition to the Eclipse Web site, **http://www.eclipse.org**, there are several sites that offer a wide variety of Eclipse plug-ins. The following URLs provide links to two such Web sites:

- **http://marketplace.eclipse.org/**
- **http://www.sourceforge.net/**

If you plan to install plug-ins that are dependent on the Eclipse Java Development Tools (JDT), see *17.4.1 Installing JDT for Third-Party Plug-ins and Debugging*, p.247 for information about downloading and installing the JDT.

If you want to install plug-ins that integrate version control capabilities into Workbench, see *17.4.2 Installing the IBM Rational ClearCase Plug-in*, p. 248 and *17.4.3 Installing the Subclipse SCM Plug-in*, p. 250.

17.3 Incorporating New Plug-ins into Workbench

You can incorporate new plug-ins into Workbench using the following methods:

- Basic (single user) installation

  Use the Eclipse New Software Install Wizard to incorporate the plug-in automatically.

- Advanced (manual or semiautomatic) installation

  Manage third-party plug-ins in a location outside the Workbench installation.

Before you install third-party plug-ins into Workbench, consider the advantages and disadvantages of each installation method. Your working environment—whether you are part of a team or work alone, and how often you need to update or reinstall Workbench—usually determines which method to use.

17.3.1 Installation Methods — Advantages and Disadvantages

This section describes the basic and advanced plug-in installation methods and includes advice about managing multiple plug-in configurations and disabling plugins, when necessary.

Basic Installation

When you use the Eclipse New Software Install Wizard, Workbench manages the installation of your chosen plug-ins. The wizard lets you select software to download from a site that it already knows about, or you can provide a URL and add a repository to the list of available software sites. The wizard then guides you through several steps to validate your selections, check for dependencies, accept licenses, and install the plug-ins. To complete the installation, you must restart Workbench.

Advanced installation

If you choose to install third-party plug-ins manually outside the Workbench installation directory, you will be able to update or reinstall Workbench without having to reinstall your third-party plug-ins. This method of installation also ensures that the default Workbench installation remains unchanged. If you work as part of a team, you can choose a location for your plug-in installation that is accessible to all team members and save each team member the time needed to install plug-ins individually.

This method of installation requires that you create directories, subdirectories, and marker files for you plug-ins, and that you tell Workbench where to find them.

17.3.2 Basic (SIngle User) Installation

The plug-ins you install using this method become part of the Workbench installation — that is, they reside in your user-private and version-specific Workbench configuration directory (**HOME/.workbench-version/plugins**).

Installing software using the Eclipse New Software Install Wizard is a base Eclipse capability. This section points you to Eclipse documentation that shows you how to add a software update site to the list of available software sites in Workbench, and how to use the Eclipse New Software Install Wizard to perform a basic installation of third-party plug-ins in Workbench.

To add a software update site and to install third-party plug-ins, do the following:

1. In Workbench, change to the Advanced Device Development perspective.

   If you have not opened this perspective before, in the Workbench menu click **Window > Open Perspective > Advanced Device Development**.

2. In the Workbench menu, click Help > Install New Software, to open the Eclipse New Software Install Wizard.

3. On the Available Software page, press F1 (in Windows), CTRL-F1 (in Linux), or click the help icon for assistance.

17.3.3 Advanced Installation

If you want to install third-party plug-ins so that they remain unchanged when you update or reinstall Workbench, you must do so in a location that is not affected by the update and install operations. This section shows you how to create such a directory structure. It also describes how to let Workbench know where to find your plug-ins, manage multiple plug-in configurations, and disable plug-ins when necessary.

Manual installation

If your plug-ins are available in archive files that you have to download and unpack, you can create the directory structure and set up marker files and links manually. To do this, see *Creating a Directory Structure for Manual Plug-in Installations*, p.244.

Semiautomatic installation

You can start Workbench with execution options that set up Eclipse configuration files and directories automatically at a location you specify. With Workbench running, you can use the Eclipse New Software Install Wizard to install plug-ins from an update site (locally or on the Web) into the new location. Once you have prepared the location and installed the software, you must tell Workbench where to find your new plug-ins.

You can manage multiple plug-in configurations, and, at any time, you can disable the plug-ins without uninstalling them.

The following tasks describe each of these operations:

- Installing a Plug-in into an Independent Directory Structure
- Managing Multiple Plug-in Configurations

- Disabling Plug-in Capabilities

Creating a Directory Structure for Manual Plug-in Installations

The tasks in this subsection describe how to install third-party plug-ins that you have downloaded in an archive file, and how to make Workbench aware of your new plug-in. If the plug-in you want to install is available only on the plug-in provider's update site, see *Installing a Plug-in into an Independent Directory Structure*, p.245.

To prepare a plug-in directory structure, do the following:

1. In your file system, navigate to a location outside your Workbench installation directory (*installDir*) and create a new directory for your plug-ins using a unique name, such as **eclipseplugins**.

2. Open the new directory and create a subdirectory for each new plug-in.

   Use descriptive names for these directories.

NOTE: Before continuing, download the archive file that contains the plug-in and examine its contents. Some plug-ins provide an **eclipse** directory structure and a **.eclipseextension** file for you, others do not.

- If the destination path for the files begins with **eclipse**, and you see a **.eclipseextension** file in the list, you can skip the rest of this section and extract the plug-in files into the directory you created in step 2.

- If the destination path begins with **plugins** and **features**, then you must complete the rest of the steps in this section.

3. Inside each plug-in subdirectory, create a directory named **eclipse**.

   This directory *must* be named **eclipse**. A separate **eclipse** directory is required inside each plug-in directory.

4. Inside the eclipse directory, create an empty marker file named **.eclipseextension**.

   This marker file *must* be named **.eclipseextension** (with no **.txt** or any other file name extension). A separate **.eclipseextension** file is required inside each **eclipse** directory.

   Your new directory structure (on a Windows operating system) should be similar to the following example:

```
C:.
|
\---eclipseplugins
    \---pluginname
        \---eclipse
                .eclipseextension
```

5. Extract your plug-in into the **eclipse** directory.

   Two directories, called **features** and **plugins**, appear in the directory alongside the **.eclipseextension** marker file. For example:

```
C:.
|
\---eclipseplugins
    \---pluginname
        \---eclipse
            |   .eclipseextension
```

```
|
+---features
\---plugins
```

NOTE: For the new plug-in to work properly, its **features** and **plugins** directories, as well as an empty marker file called **.eclipseextension,** *must* be located inside a directory called **eclipse**.

To make Workbench aware of your new plug-in, do the following:

1. In your file system, navigate to the Workbench installation directory.
2. Locate the **dropins** directory:
   *installDir***\workbench-3.x\wrwb\platform\x86-win32\eclipse\dropins**

   If the **dropins** directory does not exist, create it.
3. In the **dropins** directory, create a file called *pluginname***.link**, where **pluginname** is the name of the plug-in directory you created under **eclipseplugins**. The **.link** file must contain only the absolute or relative path to the location in your file system of the new plug-in directory under **eclipseplugins**. For example:

   ```
   path=../../../../../../../eclipseplugins/pluginname
   ```
4. Restart Workbench.

   Workbench now includes the capabilities that the new plug-in introduces to your working environment.

Installing a Plug-in into an Independent Directory Structure

This task shows you how you can use the Eclipse New Software Install Wizard to install plug-ins into an independent directory structure (*extensionsDir*) that will not be affected by Workbench updates and reinstallations.

To install a plug-in into an independent directory structure, do the following:

1. Modify your Workbench run command as follows; then start Workbench:

   On Windows:

   ```
   wrwb-x86-win32.exe -configuration c:\extensionsDir\eclipse\configuration
   -data c:\extensions\workspace -vmargs -DUSE_ECLIPSE_CONFIG_DIR_STYLE
   ```

   On Linux and Solaris:

   ```
   startWorkbench.sh -configuration /extensionsDir/eclipse/configuration -data
   extensions/workspace -vmargs -DUSE_ECLIPSE_CONFIG_DIR_STYLE
   ```

   You can find the Workbench executable file in the Workbench installation directory.

   On Windows:

   ```
   installDir\workbench-3.x\wrwb\platform\x86-win32\eclipse\wrwb-x86-win32.exe
   ```

   On Linux and Solaris:

   ```
   installDir/startWorkbench.sh.
   ```

2. Install your chosen plug-in from the plug-in provider’s update site using the Eclipse New Software Install Wizard, see *17.3.2 Basic (SIngle User) Installation*, p.243.

3. Create a marker file called **.eclipseextension** in the *extensionsDir*/**eclipse** directory.

   On Windows:

   a. At a command prompt, type:
   ```
   % cd extensionsDir\eclipse
   % echo. > .eclipseextension
   ```
   b. In the MS-DOS editor, save the empty marker file and exit the editor (type **Alt > S > Alt > X).**

   On Linux or Solaris:

   At a terminal prompt, type:

   ```
   % touch extensionsDir/eclipse/.eclipseextension
   ```
4. Locate the **dropins** directory:
   *installDir***workbench-3.x\wrwb\platform\x86-win32\eclipse\dropins**

   If the **dropins** directory does not exist, create it.
5. Create a **.link** file in the Workbench **dropins** directory.

   NOTE: You can also use the **links** directory to store your ***.link** files. If you want to separate ***.link** files for Wind River Workbench and your own extensions, create a **dropins** directory for ***.link** files that point to Wind River Workbench-related plug-ins.

   On Windows:

   a. At a command prompt, type:
   ```
   % cd installDir\workbench-3.x\wrwb\platform\x86-win32\eclipse\dropins
   % edit extensions.link
   ```
   b. In the MS-DOS editor, type the absolute or relative path to the extensions directory.
   ```
   path=c:/extensionsDir
   or
   path=../../../../../../extensionsDir
   ```
   c. Save the **.link** file and exit the editor, type **Alt+F > S > Alt+F > X.**

   On Linux or Solaris:

   At a terminal prompt, type:

   ```
   echo "path=../../../../../extensions" >
   installDir/wrwb/platform/x86-linux2/eclipse/dropins/extensions.link
   ```
6. Restart Workbench.

Managing Multiple Plug-in Configurations

When you make a plug-in available to Workbench using the **.eclipseextension** and **.link** files, you can also make the extensions available only to certain configurations.

When you start Workbench, specify which configuration you want to launch by using the **-configuration** *path* option, where *path* represents your Eclipse configuration directory. Inside this configuration directory, create a **dropins** folder that contains the ***.link** files for those extensions that you want to pick up in this configuration.

NOTE: For more information about **-configuration** and other Eclipse startup parameters, refer to the online help documentation for Eclipse. In the Workbench menu, click **Help > Help Contents**; then click **Workbench User Guide > Tasks > Running Eclipse**.

On Windows:

From a shell, type the following:

```
% cd installdir\workbench-3.x\wrwb\platform\x86-win32\eclipse
% .\wrwb-x86-win32.exe -configuration path
```

Or, copy and modify the desktop shortcut that starts Workbench:

1. Right-click the desktop shortcut; then click **Properties**.
2. In the Properties dialog, click the Shortcut tab.
3. On the Shortcut page, in the Target field, append **-configuration** to the path. For example:

```
C:\installDir\workbench-3.x\wrwb\platform\x86-win32\eclipse\wrwb-x86-win32.exe
-configuration path
```

On Linux and Solaris:

Use the option as a parameter to the **startWorkbench.sh** script, as follows:

```
% ./startWorkbench.sh -configuration path &
```

Disabling Plug-in Capabilities

You can disable plug-in capabilities without uninstalling your manually installed files. This gives you the opportunity to re-enable the capabilities at a later time.

To disable the capabilities of a plug-in, do the following:

1. In your file system, navigate to the Workbench installation directory.
2. Locate the **dropins** directory:
   *installDir***\workbench-3.x\wrwb\platform\x86-win32\eclipse\dropins**
3. In the **dropins** directory, move away or delete the **.link** file that contains the path to the plug-ins whose capabilities you want to disable.
4. Restart Workbench.

17.4 Example scenarios

17.4.1 Installing JDT for Third-Party Plug-ins and Debugging

Previous versions of Workbench included a modified version of the Java Development Tools (JDT), but, for better integration with Eclipse and certain third party packages, Workbench no longer includes this modified JDT. Some third party plug-ins have a dependency on the JDT.

This procedure shows you how to install the JDT in a location unaffected by Workbench updates and reinstallations.

To install the JDT, do the following:

1. Follow step 1 of the instructions for starting Workbench in *Installing a Plug-in into an Independent Directory Structure*, p.245.
2. In the Workbench menu, click **Help > Install New Software**.
3. In the Install dialog, in the Work with drop-down list, type **http://download.eclipse.org/releases/helios**.
4. In the Name list, expand **Programming Languages**; then select **Eclipse Java Development Tools**.
5. Ensure that **Show only the latest versions of available software** and **Group items by category** are selected.
6. (Optional) Click **Contact all update sites during install to find required software**.
7. Click **Next**.
8. On the Install Details page, ensure that Eclipse Java Development Tools is listed; then click **Next**.
9. On the Review Licenses page, accept the license; then click **Finish**.
10. Return to *Installing a Plug-in into an Independent Directory Structure*, p.245, and follow the rest of the procedure from step 3.

17.4.2 Installing the IBM Rational ClearCase Plug-in

The procedures in this section show you where to obtain the IBM Rational ClearCase plug-in, how to install it in a location unaffected by Workbench updates and reinstallations, and how to activate the plug-in.

If you want to share the capabilities of the plug-in among the members of your team, make sure you choose an installation location that is accessible to all team members.

Testing the Readiness of a Linux or Solaris Environment

If you are working in a Linux or Solaris environment, run a sanity check to verify that all of the packages required for the ClearCase visualization tools are installed on your system.

At the command line, run **xclearcase**

This command invokes the ClearCase graphical user interface (GUI).

If you need to troubleshoot your system, see the IBM Rational Support Web site: **https://www-304.ibm.com/support/docview.wss?rs=0&uid=swg21247644**.

Installing the ClearCase Plug-in

To obtain and install the ClearCase plug-in, do the following:

1. Follow step 1 of the instructions for starting Workbench in *Installing a Plug-in into an Independent Directory Structure*, p.245.
2. Visit the IBM Rational ClearCase SCM Adaptor (for Eclipse) Web site:
   http://www-01.ibm.com/support/docview.wss?rs=984&uid=swg24024213
3. Scroll down the page and follow the plug-in installation instructions entitled, **For Eclipse versions 3.4.x and later**.As of adaptor version 7.0.0.2, the following instructions apply:
   a. Download and extract the adaptor archive file to a location on your local hard disk.
   b. In the Workbench menu, click **Help > Install New Software**.
   c. On the Available Software page, click **Add**.
   d. In the Add Repository dialog, click **Local**, and browse to the adaptor archive you extracted earlier; then click **OK**.
   e. (Optional) Type a name for the new local repository; then click **OK**.
   f. On the Available Software page, clear **Group items by category**; then click the adaptor and proceed with the installation.
4. Return to *Installing a Plug-in into an Independent Directory Structure*, p.245, and follow the rest of the procedure from step 3.

To activate the IBM Rational ClearCase plug-in, do the following:

1. In the Workbench menu, click **Window > Preferences > Wind River > Capabilities**.
2. In the Capabilities pane, expand **Team**, select **ClearCase SCM Adapter** and click **Apply**; then click **OK**.
3. In the Workbench menu, click **Window > Customize Perspective**.
4. In the **Customize Perspective** dialog, switch to the **Command Groups Availability** tab.
5. In the **Available command groups** column, select the **ClearCase** option; then click **OK**.

   A new **ClearCase** menu and icons appear on the main Workbench toolbar.
6. In the **ClearCase** menu, select **Connect to Rational ClearCase** to activate ClearCase capabilities.
7. After activating the plug-in, do the following as necessary:
   - Select **Window > Preferences > Team > ClearCase SCM Adapter** to configure the plug-in.
   - For information about using the plug-in, press the help key for your host, select **All Topics**, then see the **Rational ClearCase SCM Adapter** documentation.

For information about the capabilities of the ClearCase SCM adaptor, refer to your ClearCase product documentation.

17.4.3 Installing the Subclipse SCM Plug-in

To use the Subclipse Eclipse Team Provider plug-in to provide support for Subversion version control in Workbench, follow the instructions in this section. These instructions direct you to install the plug-in in a location unaffected by Workbench updates and reinstallations.

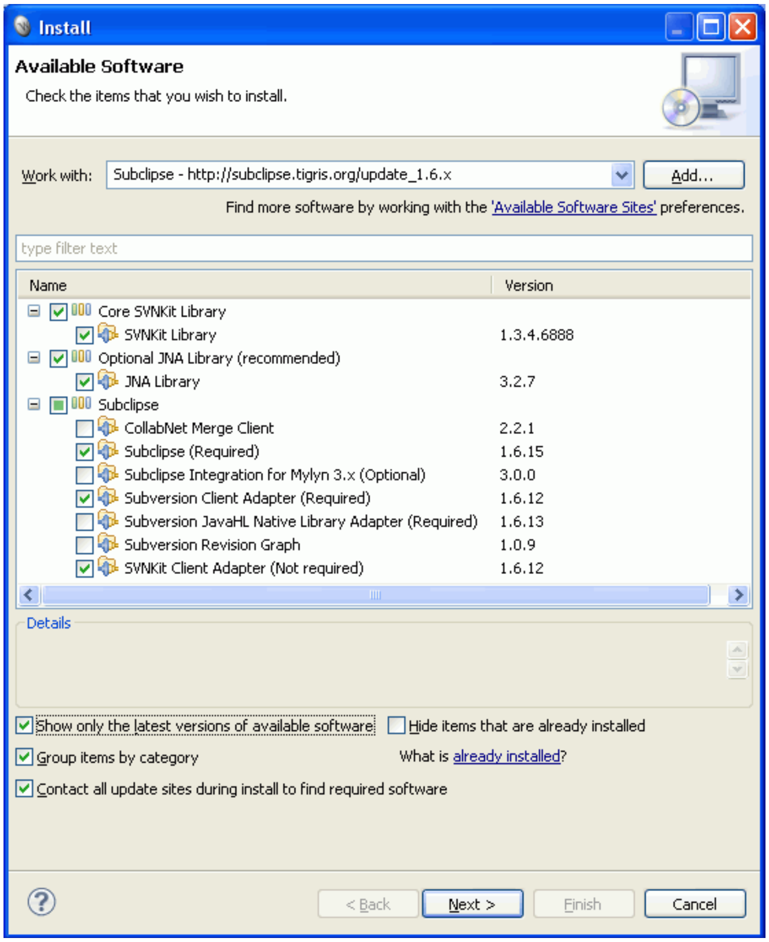
If you want to share the capabilities of the plug-in among the members of your team, make sure you choose an installation location that is accessible to all team members.

Subclipse supports a variety of client adapter versions, which must match the Subversion command-line client and native libraries in use on your system. To get started with Subclipse, installing the SVNKit Client Adapter is a safe choice, as it works on all systems.

To obtain and install the Subclipse plug-in, follow the instructions in *Installing a Plug-in into an Independent Directory Structure*, p.245 and use the Subclipse update site URL, **http://subclipse.tigris.org/update_1.6.x/**, to install Subclipse version 1.6.x.

To use SVNKit, make sure your selections in the Eclipse New Software Install Wizard match those in Figure 17-1.

Figure 17-1 **SVN Kit Settings**



Available Software

Check the items that you wish to install.

Work with: Subclipse - http://subclipse.tigris.org/update_1.6.x

Find more software by working with the 'Available Software Sites' preferences.

type filter text

Name	Version
⊟ ☑ Core SVNKit Library	
☑ SVNKit Library	1.3.4.6888
⊟ ☑ Optional JNA Library (recommended)	
☑ JNA Library	3.2.7
⊟ ■ Subclipse	
□ CollabNet Merge Client	2.2.1
☑ Subclipse (Required)	1.6.15
□ Subclipse Integration for Mylyn 3.x (Optional)	3.0.0
☑ Subversion Client Adapter (Required)	1.6.12
□ Subversion JavaHL Native Library Adapter (Required)	1.6.13
□ Subversion Revision Graph	1.0.9
☑ SVNKit Client Adapter (Not required)	1.6.12

☑ Show only the latest versions of available software □ Hide items that are already installed

☑ Group items by category What is already installed?

☑ Contact all update sites during install to find required software

Visit the Subclipse Web site (**http://subclipse.tigris.org/**) for information about the Subclipse plug-in, and see the Subclipse JavaHL FAQ (**http://subclipse.tigris.org/wiki/JavaHL**) for more information on client adapters.