



Evaluation Guide - Eclipse Integration

Copyright © 1994-2005 Embarcadero Technologies, Inc.

Embarcadero Technologies, Inc.
100 California Street, 12th Floor
San Francisco, CA 94111 U.S.A.
All rights reserved.

All brands and product names are trademarks or registered trademarks of their respective owners. This software/documentation contains proprietary information of Embarcadero Technologies, Inc.; it is provided under a license agreement containing restrictions on use and disclosure and is also protected by copyright law. Reverse engineering of the software is prohibited.

If this software/documentation is delivered to a U.S. Government Agency of the Department of Defense, then it is delivered with Restricted Rights and the following legend is applicable:

Restricted Rights Legend Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of DFARS 252.227-7013, Rights in Technical Data and Computer Software (October 1988).

If this software/documentation is delivered to a U.S. Government Agency not within the Department of Defense, then it is delivered with Restricted Rights, as defined in FAR 552.227-14, Rights in Data-General, including Alternate III (June 1987).

Information in this document is subject to change without notice. Revisions may be issued to advise of such changes and additions. Embarcadero Technologies, Inc. does not warrant that this documentation is error-free.

Contents

- Working within Eclipse 5
 - Configuring Eclipse to Use the Correct Java Runtime Environment..... 6
 - Using the Integration..... 7
 - Opening Describe in Eclipse Platform..... 8
 - Disconnecting and Reconnecting to Describe in Eclipse 8
 - Running the Eclipse/Describe Integration..... 9
 - Eclipse Window..... 11
 - Describe Workspace Pane 11
 - Create a New Project 13
 - Create a New Diagram 14
 - Create a Model Element 17
 - Creating New Classes in Eclipse 20
 - Working With Eclipse Projects 21
 - Connecting Existing Eclipse Projects to Describe..... 21
 - Disconnecting Eclipse Projects from Describe 22
 - Manually Turn Off Auto Compile When Reverse Engineering Projects 23
 - Importing Source Code into Eclipse 23

Describe Enterprise functionality is accessible through an Integrated Development Environment (IDE). This version of the application supports the following IDEs:

- Borland JBuilder 9.0
- IBM WebSphere Studio Application Developer 5.0. and Integrated Edition 5.0
- Eclipse Platform 3.0.1 (WinXP only)

Working within Eclipse

Describe Enterprise offers a seamless, embedded integration within Eclipse Platform to provide a single, integrated modeling and development environment. With superior navigation, wizard-based automation of common tasks, and synchronous, RoundTrip engineering, Describe Enterprise turns your UML into a live tool for understanding and manipulating the source base.

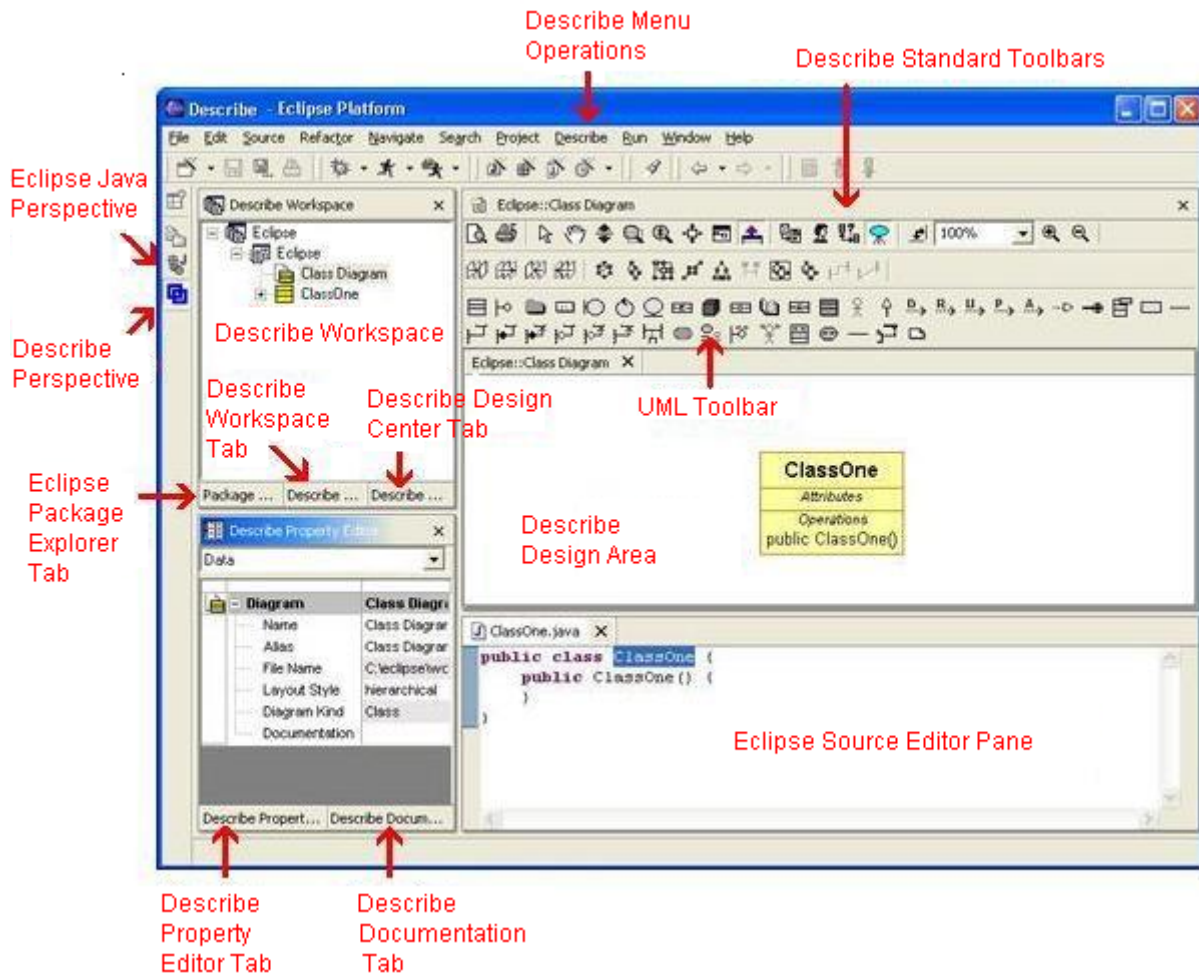
What Is Eclipse?

Eclipse software is an Integrated Development Environment (IDE) for the Java platform providing a fully modular and extensible environment. This environment, called the workspace, consists of several panes and panels for performing the following development functions:

- Editing
- Visual designing
- Navigating
- Browsing
- Compiling
- Debugging

Why Integrate with Describe?

The Eclipse/Describe integration lets Unified Modeling Language tools communicate with Integrated Development Environments. This eliminates the need to switch back and forth from design environment to development environment and maintains synchronization between design elements and the source code as you design systems. The two environments are combined by embedding modeling functionality within Eclipse, letting you view and manipulate the class model and its underlying code simultaneously.



TIP: You can have multiple Projects open simultaneously while working with integrated IDEs. This lets you easily switch back and forth between Projects.

Configuring Eclipse to Use the Correct Java Runtime Environment

Describe's Eclipse 3 integration has been qualified with Sun Microsystems 1.4.2_06 Java Runtime Environment (JRE). While other versions of the JRE may work, we recommend running Eclipse with this JRE when using the Describe integration.

TIP: The runtime JRE used for Eclipse depends on how Eclipse is started. We recommend using the Eclipse shortcut created in the Embarcadero group of the Windows start menu, as that includes a parameter specifying the JRE you selected at install time. If you start Eclipse from Windows Explorer or a different shortcut, a different JRE may be used.

If you are unsure of the JRE Eclipse is using, select the "Help/About Eclipse Platform" menu item and click on "Configuration Details". The java.version parameter lists the JRE version and the java.home parameter lists the location of the JRE.

Describe integration also requires additional files to be installed in the JRE home directory that Eclipse uses at runtime. During installation of the Describe-Eclipse plugin, you were prompted for the location of the JRE used to run Eclipse. Describe installs the following files to the JRE folder specified:

- <jre home directory>\bin\awtresolve.dll
- <jre home directory>\bin\JavaNISupport.dll
- <jre home directory>\classes*.*

If Eclipse is running with a JRE that does not include the files above, you will see a dialog warning you that a detected JRE location does not include the necessary files to correctly run the integration. If you continue to run the integration in this state, you will encounter specific problems such as code generation not working and the inability to create UML diagrams.

You can correct this problem by performing the following steps:

- 1 If you do not have a supported JRE on your computer, install a supported version such as Sun Microsystems JRE 1.4.2_06.
- 2 Determine the correct home directory of the JRE. The JRE home directory contains a \bin and \lib subdirectory. For example, the default JRE home directory for Sun Microsystems 1.4.2_06 JRE is C:\Program Files\Java\j2re1.4.2_06.
- 3 Configure Eclipse to use this JRE at runtime. We recommend always starting Eclipse by including the -vm parameter. Describe also provides a shortcut in the Windows Start Menu, similar to this command line:
C:\eclipse\eclipse.exe -vm "C:\Program Files\Java\j2re1.4.2_06\bin\javaw.exe"
- 4 Install the required Describe files into the JRE folder, using one of the following methods:
 - Use **Add/Remove** programs to "Change" the Describe installation. When changing the Describe installation, elect to "Modify" the installation.
In the Custom Installation dialog, highlight **Client > Integrations > Eclipse**, right-click and select "This feature will not be available" from the menu. "Change" the installation again, modifying it to add Eclipse. When prompted to specify the Eclipse JRE location, specify the JRE home directory. For example:
C:\Program Files\Java\j2re1.4.2_06
 - If you wish to change JREs without reinstalling Describe, use Windows Explorer to copy or move all required Describe JRE files from the location specified at installation time to the JRE location Eclipse is currently using. For example, if the files are copied correctly, your JRE home directory will contain the additional files:
 - C:\Program Files\Java\j2re1.4.2_06\bin\awtresolve.dll
 - C:\Program Files\Java\j2re1.4.2_06\bin\JavaNISupport.dll
 - C:\Program Files\Java\j2re1.4.2_06\classes*.*
- 5 Launch Eclipse.

Using the Integration

There are two important functions that impact how you use the Describe-Eclipse integration: refactoring and real-time model-code synchronization.

Refactoring and renaming using UML Model

The UML model may be used to rename common elements such as classes, attributes and operations. While this is supported and associated source is updated, this renaming mechanism does not invoke Eclipse refactoring. If you need to rename model elements that have associated code, we recommend that you use the Eclipse refactoring mechanism. After you invoke a refactoring, the UML model updates.

Understanding Real-Time Model-Code Synchronization

When Describe integration is enabled with Eclipse, the UML model and Java source code are synchronized. The following issues should be understood for optimal usability and performance.

- When working with source code that has a corresponding UML model element, the model will be updated in real time as you type in source. Depending on the timing and speed of your machine, the model will sometimes update before you are done typing. The model will correct itself as you type, so source and model are in sync.
- When pasting in a large amount of code into the source editor, the integration may need to update several elements in the UML model. This could take several seconds depending on the size of the code. When this is happening, you will see 'updating metadata...' message in the left hand side of the Eclipse status bar.

Opening Describe in Eclipse Platform

When you start Eclipse and create an Eclipse project, the application automatically displays selected Describe windows, toolbars, and menu options in the Describe perspective. Upon opening, Describe automatically embeds itself in Eclipse if you install Eclipse Platform prior to Describe. Once embedded, you can connect and disconnect to Describe as needed.

NOTE: When Describe is successfully installed, a Describe menu will appear in the Eclipse menu bar. If this does not appear, Eclipse 3.x did not recognize the Describe plugin after installation. This occurs because Eclipse 3.x caches plug-in information for a faster startup.

If you experience this problem, include the "-clean" option on the command line used to launch Eclipse. When the `-clean` command line option is specified, any cached data used by the OSGi framework and eclipse runtime will be wiped clean. This will clean the caches used to store bundle dependency resolution and eclipse extension registry data. Using this option will force eclipse to reinitialize these caches. It is only necessary to use the `-clean` option the first time Eclipse is launched after the Describe is installed.

Opening Describe in Eclipse

- 1 On the Windows **Start** button, point to **Programs, Embarcadero Describe, Eclipse**.

Eclipse starts, and automatically detects the integration with Describe. The application then displays its workspace, along with the embedded Describe windows, toolbars, and menu items in the Describe perspective.

Disconnecting and Reconnecting to Describe in Eclipse

Once open, Eclipse Platform lets you disconnect and reconnect to Describe. When you disconnect from Describe, Eclipse Platform removes the Describe toolbar and all related panes. When you reconnect, the application redisplay these workspace components.

Disconnecting from Describe

- 1 On the **Describe** menu, click **Disconnect from Describe**.

Eclipse Platform disconnects from Describe, and removes the toolbar and all related panes.

Reconnecting to Describe

- 1 On the **Describe Tools** menu, click **Connect to Describe**.

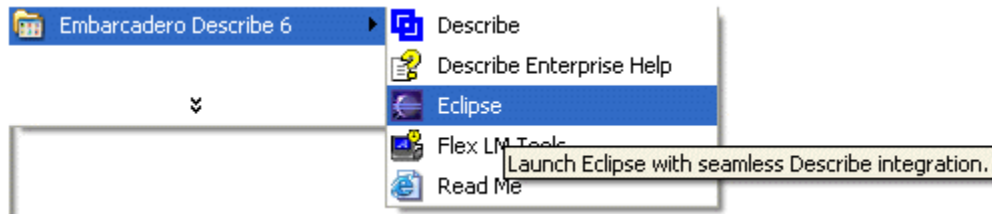
Eclipse Platform reconnects to Describe and displays all Describe windows and toolbars.

Running the Eclipse/Describe Integration

The sections below will walk you through the steps necessary to run the Eclipse/Describe Integration.

Starting Eclipse

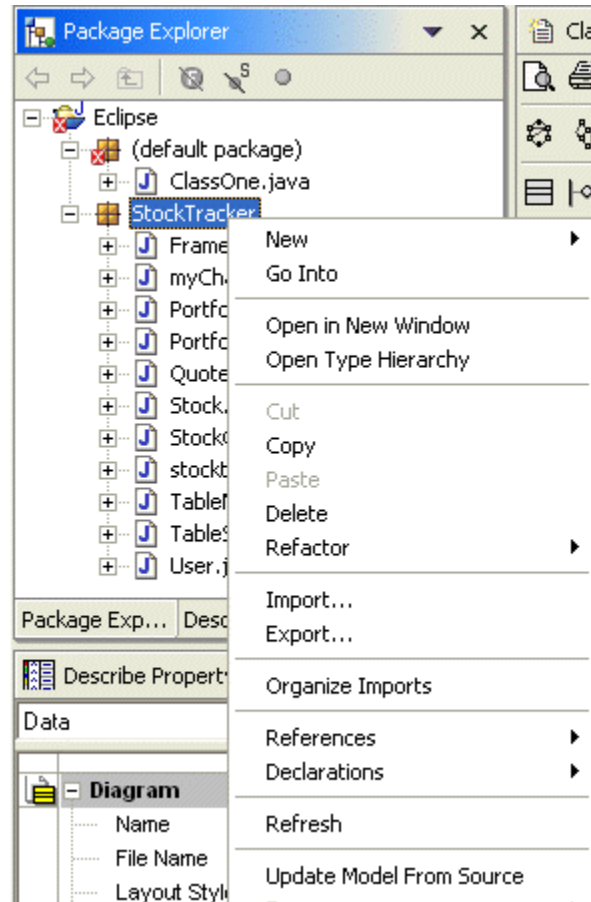
- 1 On the Windows **Start** button, point to **Programs, Embarcadero Describe, Eclipse**.



Eclipse starts, and automatically detects the integration with Describe. Initially, the Eclipse splash screen opens.

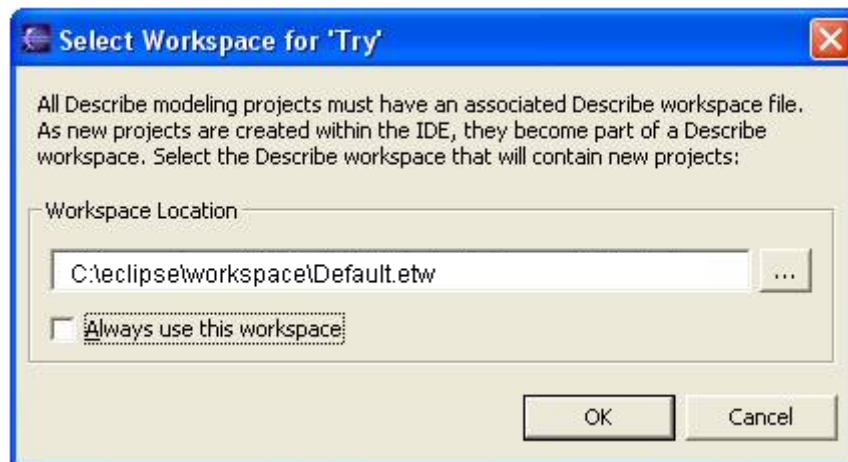
As Describe connects to Eclipse, Describe allows user to reverse engineer existing source code:

- 2 On the **Eclipse Package** tab, right-click Java files or packages.
- 3 Click **Update Model From Source**.



NOTE: Describe also Reverse Engineers Java source code when importing files or file systems into an Eclipse project. For Eclipse 3.x and above, the user must enable the **Advanced ->Integrations ->Reverse Engineer Imported Source Files** in the Describe Preference dialog for automatic reverse engineering of imported source files.

When working with Eclipse, you will be prompted to select a workspace directory where your projects will be created.



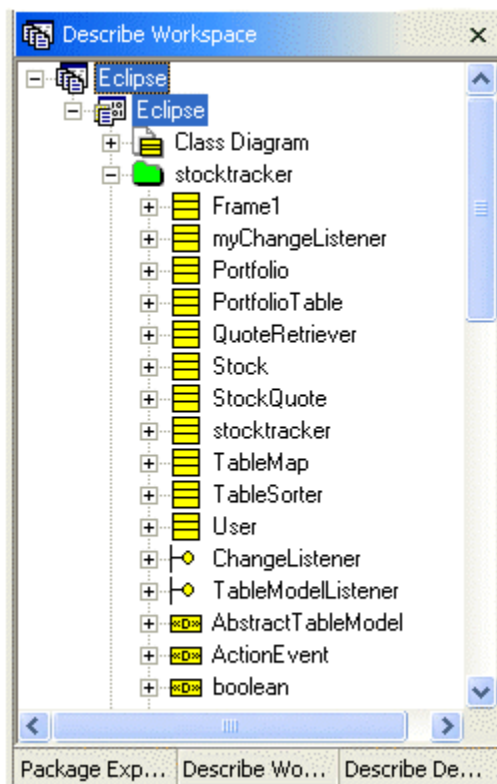
Eclipse Window

The icons below are the Describe modeling icons. You see these when you open the diagram window.



Describe Workspace Pane

Select the Describe Workspace pane tab and the Workspace appears and displays the current Describe system.



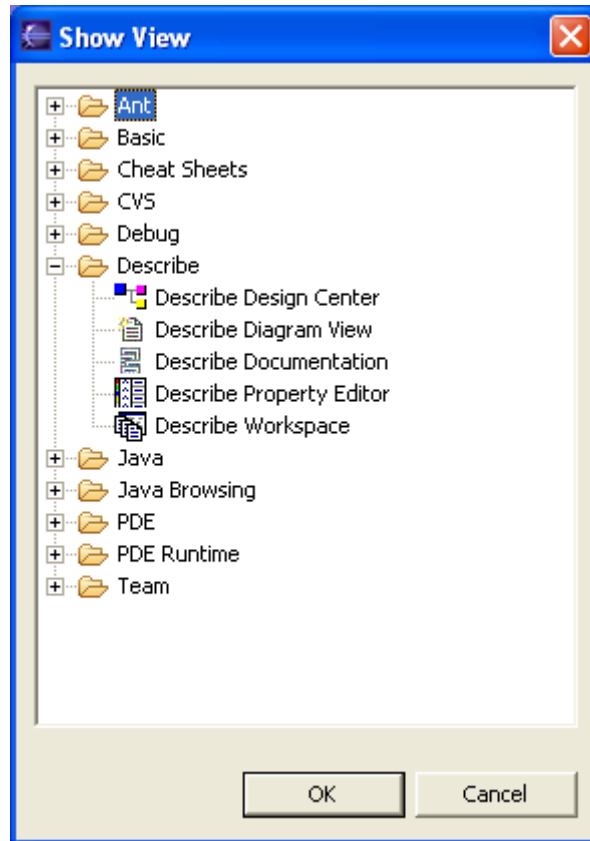
You can access the following Describe functionality from within the IDE:

- Describe Design Center
- Describe Documentation
- Describe Property Editor
- Describe Workspace

To see one of these views:

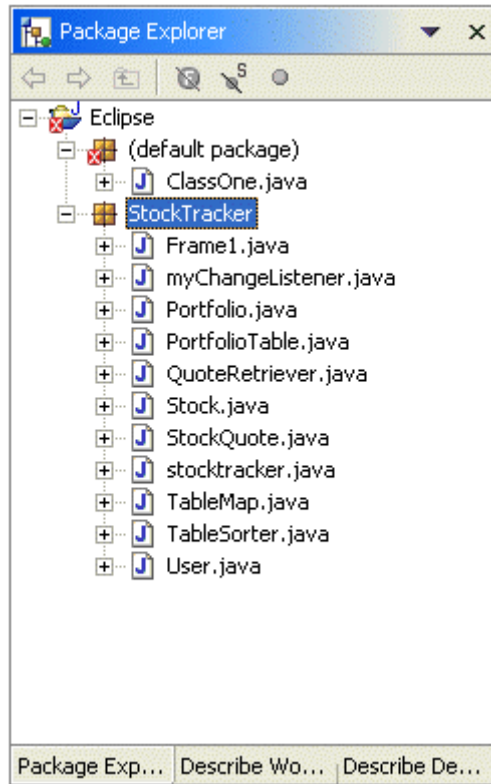
- 1 From the **Eclipse Window** menu, select **Show View** and then click **Other**.

The **Show View** dialog box opens.



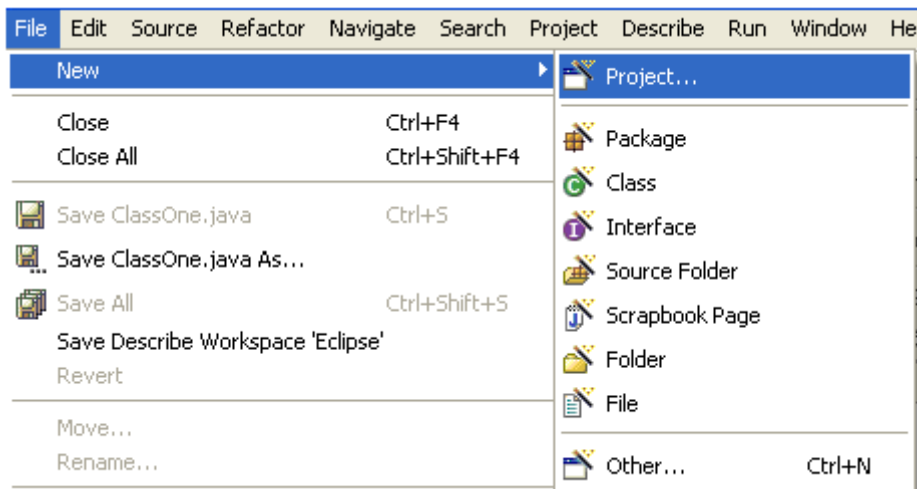
- 2 To work in the Describe Workspace, click the Describe Workspace node and then click **OK**.

An example of how the packages and classes are displayed in the Package Explorer pane of Eclipse is shown below:

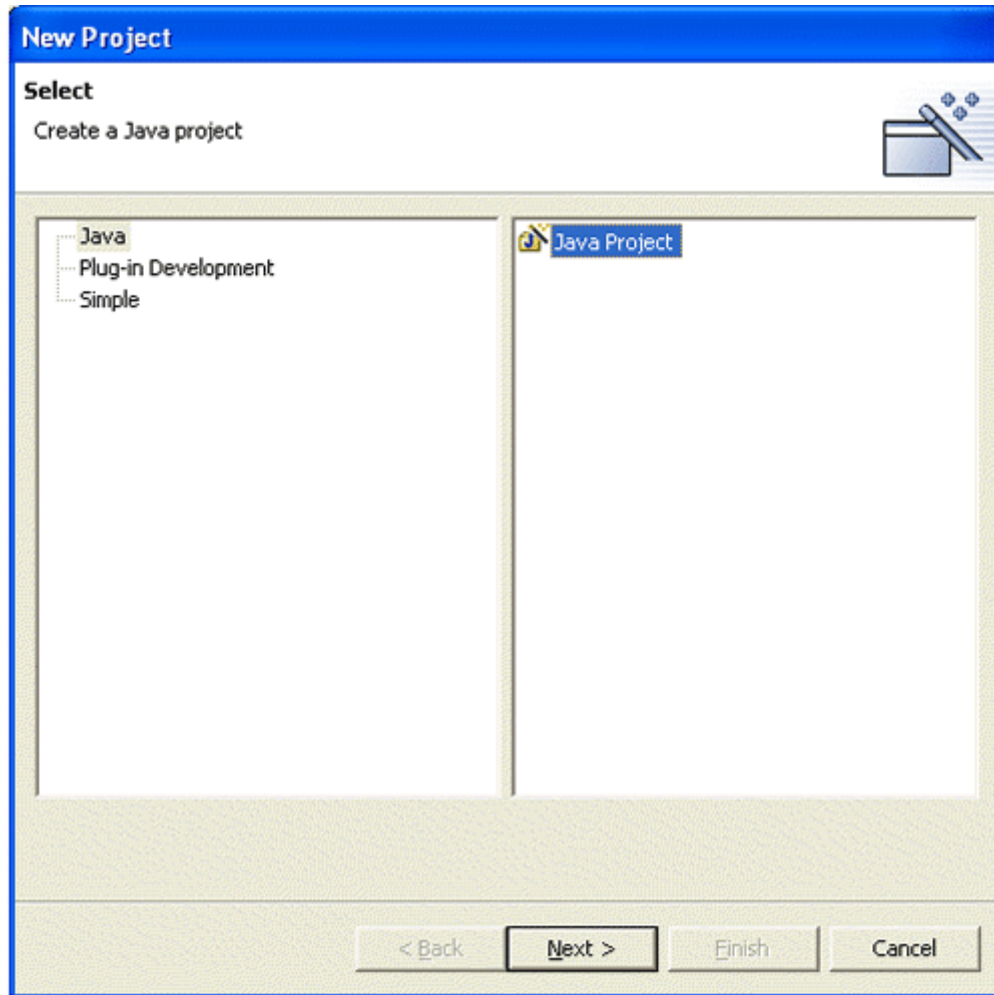


Create a New Project

- 1 On the **Eclipse File** menu, click **New Project**



The New Project dialog box opens.



- 2 **Select Java, Java Project, and then Next.**

The New Java Project dialog box opens.

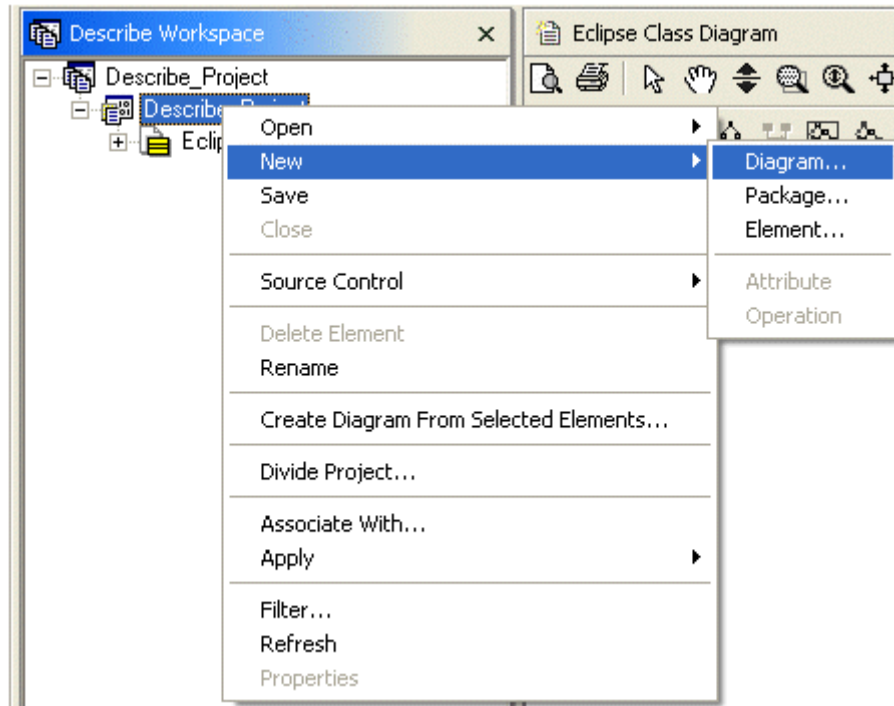
- 3 In the **Project Name** box, type the project's name. For purposes of this tutorial name the project "**Describe_Project.**"
- 4 Click **Finish.**

Describe associates a workspace and project with the newly created Eclipse project.

Create a New Diagram

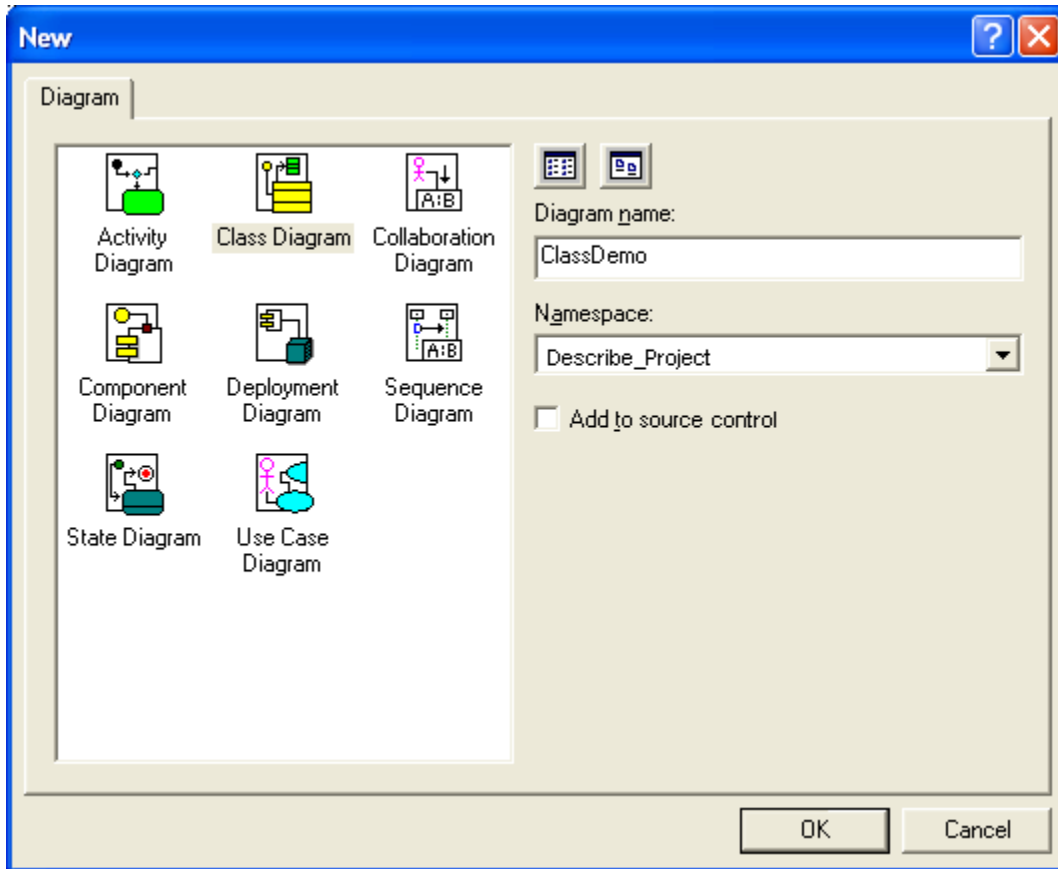
- 1 In the **Describe Workspace** pane, right-click **Describe_Project.**

A background menu opens.



2 Choose **New Diagram** from the menu.

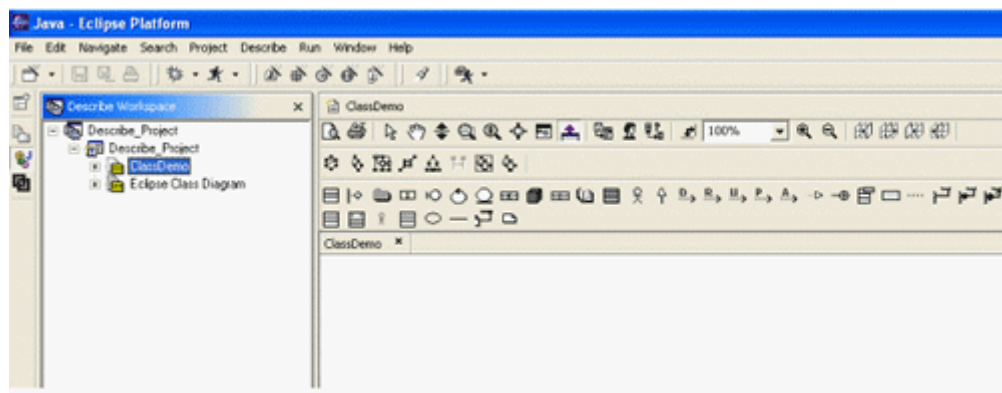
The Create New Diagram dialog box opens.



3 In the **Diagram name** box, type a diagram name.

4 Click **OK**.

The Diagram pane opens.



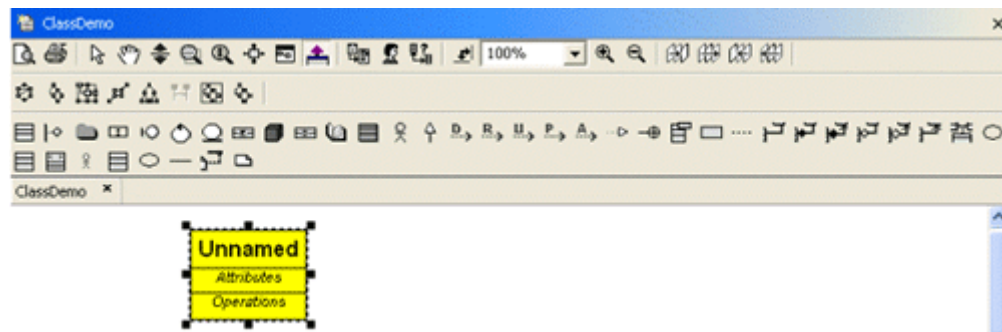
NOTE: Once the diagram is created, the navigation icons and the Class Methodology icons are now active in the Eclipse main window.

Create a Model Element

As a final step, we are going to create a class.

- 1 In the **Eclipse** window, click .
- 2 In the **Diagrams** pane, click once to place the class notation.

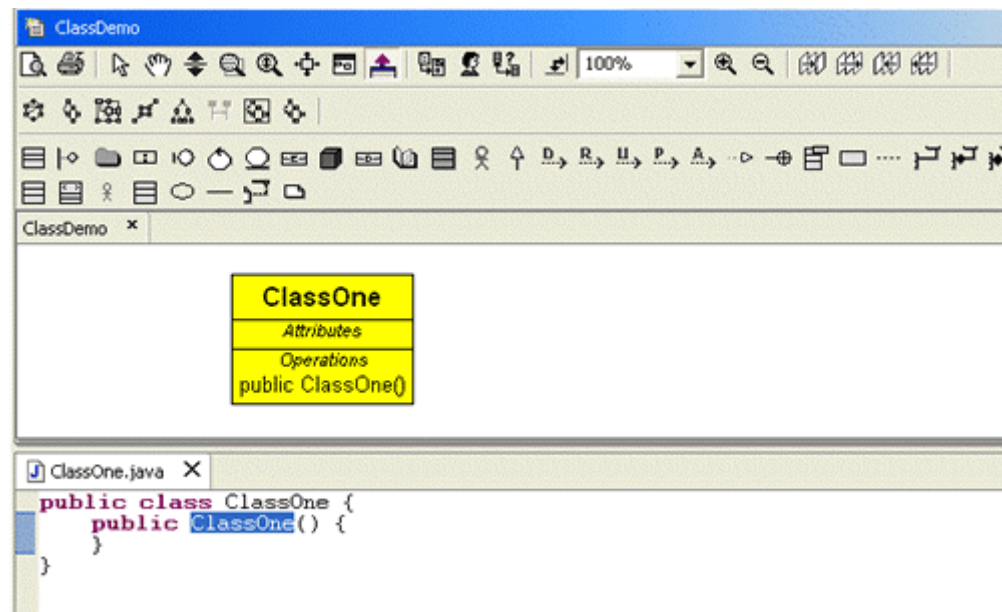
A class notation is placed in pane.



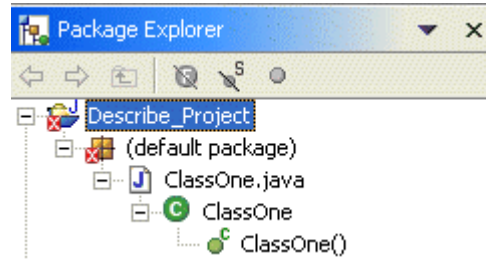
- 3 Label the class.

TIP: Start typing to automatically name it.

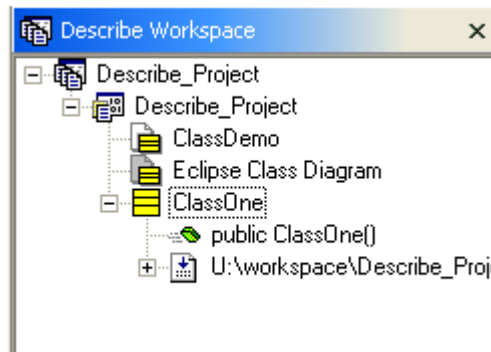
The Source pane opens and you can see the source code.



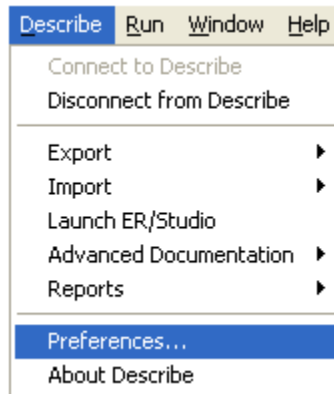
The class you just created appears in the Eclipse Package Explorer.



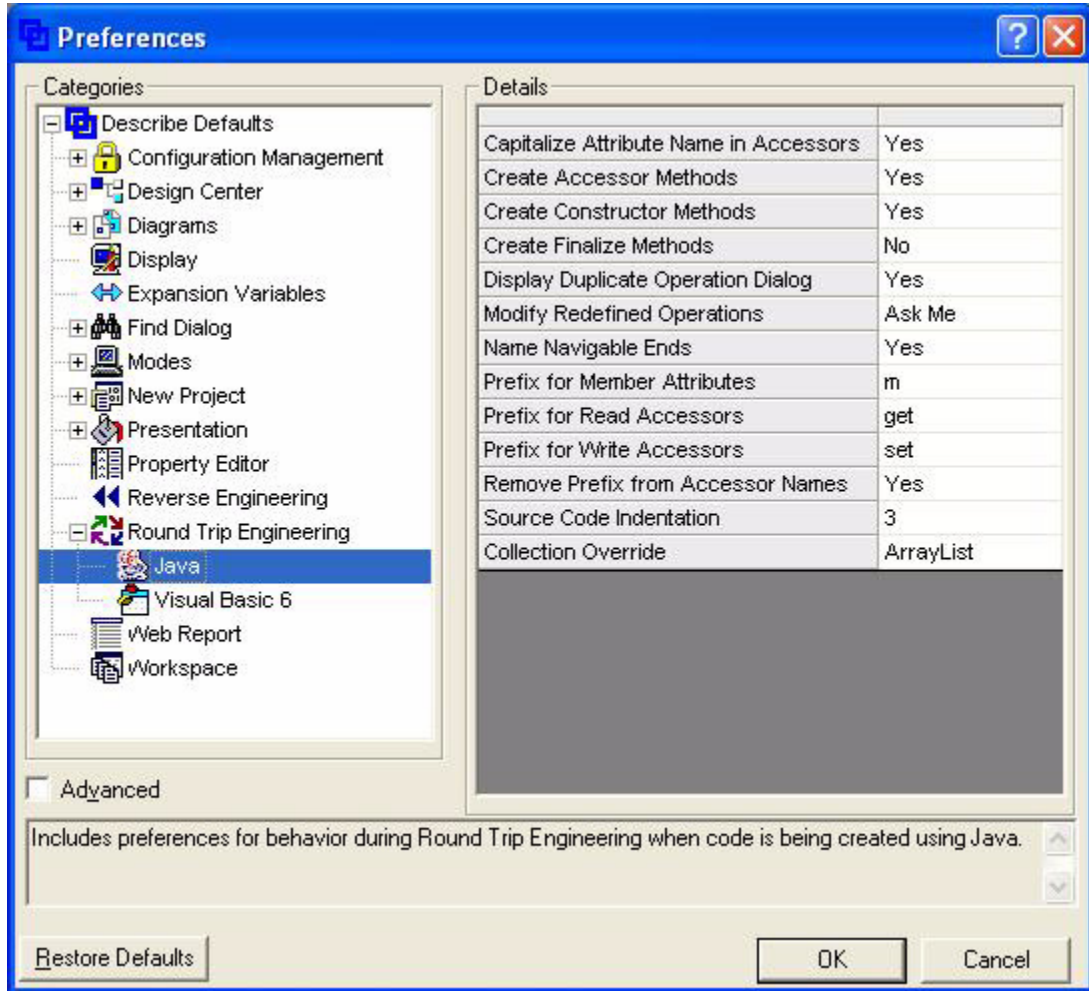
NOTE: The diagram and class you just created also appears in the Describe Workspace pane.



- 4 Using the Describe Preferences dialog box, you can set preferences on modeling elements as well as source code generation. From the **Eclipse** menu select **Describe**, and then **Preferences**.

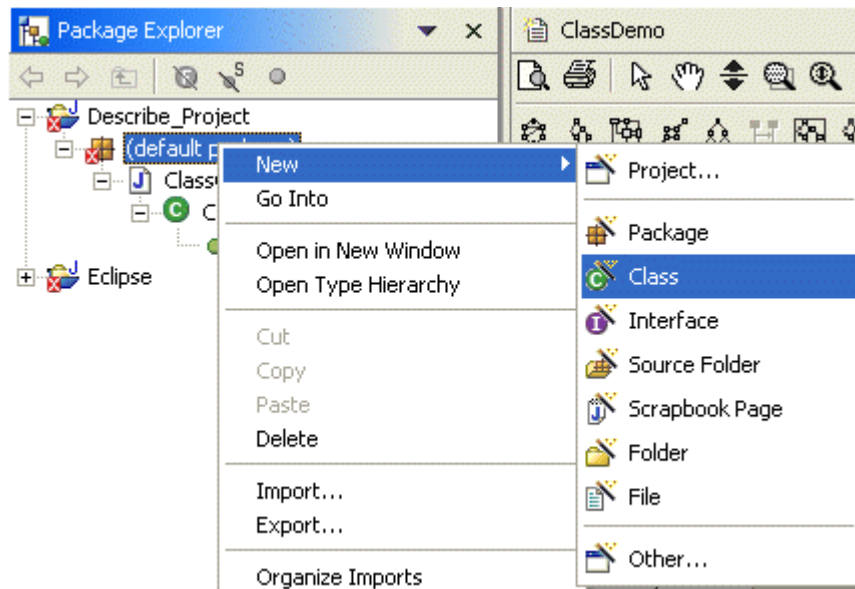


The Preferences dialog box opens.



Creating New Classes in Eclipse

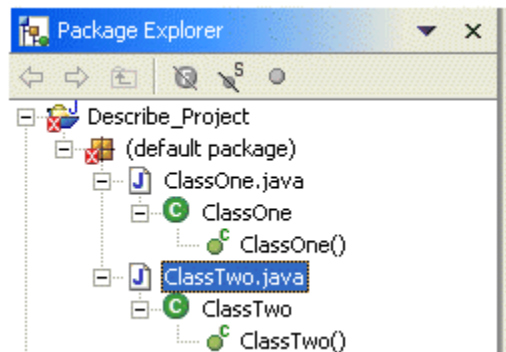
- 1 On the **Eclipse Package Explorer**, select **New Class**.



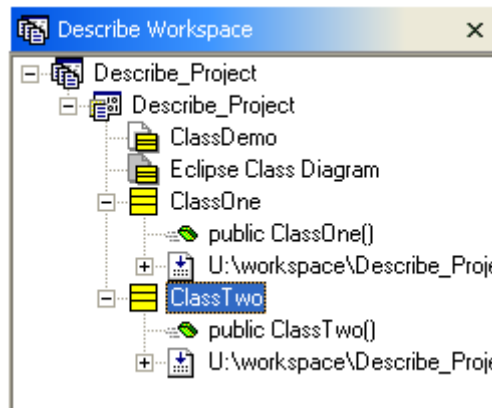
- 2 In the **Class Wizard** dialog box, name the class (in this case, **ClassTwo**).
- 3 Click **Finish**.

NOTE: Any classes that are created or modified in Eclipse are updated in the Describe Workspace Tab.

The new class appears in the Eclipse Package Explorer pane.



The new class also appears in the Describe Workspace Pane.



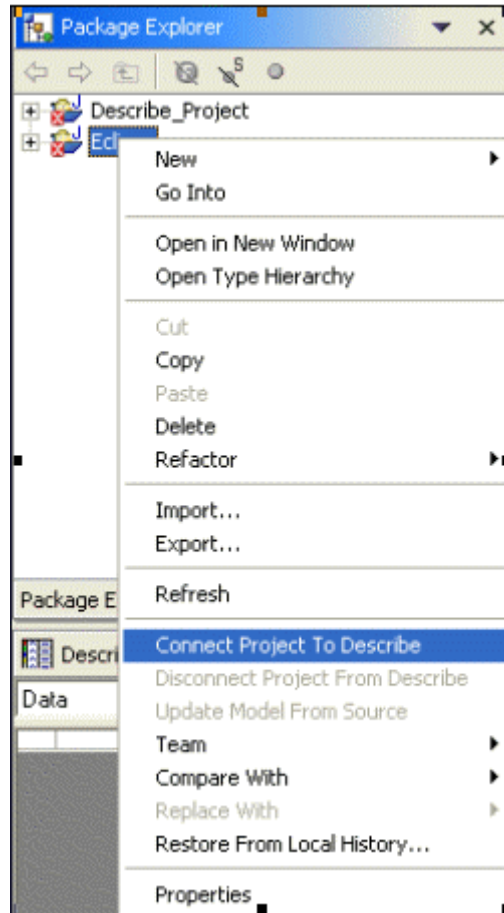
Working With Eclipse Projects

This section includes information on:

- Connecting existing Eclipse Projects to Describe
- Disconnecting Eclipse Projects from Describe
- Manually Turn Off Auto Compile When Reverse Engineering Projects
- Importing Source Code into Eclipse
- Real-Time Model-Code Synchronization

Connecting Existing Eclipse Projects to Describe

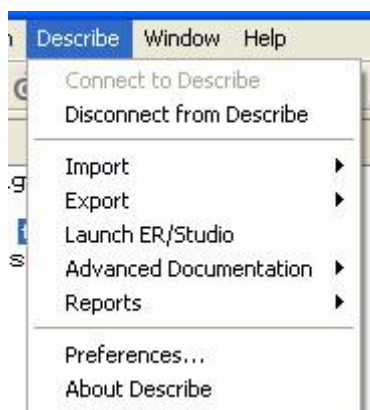
- 1 From the **Package Explorer**, select **Eclipse Project**.
- 2 Right click and select **Connect Project To Describe**.



Eclipse projects are now available in Describe.

Disconnecting Eclipse Projects from Describe

- 1 From the **Describe** menu, select **Disconnect from Describe**.



Manually Turn Off Auto Compile When Reverse Engineering Projects

- 1 On the **Window** menu, click **Preferences**.

The Preferences dialog box opens.

- 2 On the **Workbench** node, clear the **Perform build automatically on resource modification** check box.
- 3 When you want to build, do one of the following:

Press CTRL+B.

OR

On the **Project** menu, click **Build All** (or **Rebuild All**).

TIP: When you turn "auto compile" off and build manually, you may also want to select the Window > Preferences > Workbench > Save all modified resources automatically prior to manual build check box.

NOTE: When reverse engineering a project like DigSim in Eclipse the main project package is seen as a file folder when Importing the Filesystem. To reduce compile errors, create a package (for DigSim create the DigSim source directory) in the Package Explorer. Move all the source files from the "default package" to the DigSim package. This removes errors from DigSim when compiling. The remaining "warnings" generally relate to deprecated code.

Importing Source Code into Eclipse

For Eclipse 3.x and above, when importing a large number of source files, you may see multiple prompts to reverse engineer the source files. If you experience this problem, set the Reverse Engineer Imports Source Files preference to 'Never'.

This preference determines whether or not you are prompted to reverse engineer source files when importing into them Eclipse. When set to 'Ask Me', you are automatically prompted to reverse engineer any imported source files. When set to 'Never', you are never prompted to reverse engineer imported source files. When set to 'Always', imported source files are automatically reverse engineered. To update the Describe model, you must manually select 'Update Model From Source' from the source file's shortcut menu.

NOTE: The default setting for this preference is "Never".

Setting the Prompt to Reverse Engineer Source Files Preference:

- 1 Under the Describe menu, select **Preferences...**
- 2 In the Describe Preferences dialog, click **Advanced**.
- 3 Navigate to the **Integrations** item in the Categories tree.
- 4 Set the **Reverse Engineer Imported Source Files** preference to **Never**.
- 5 Click **OK** to close the Preferences dialog box.

Index

A

Auto Compile 23

D

Disconnecting From Describe Developer in Eclipse 8

E

Eclipse

 Select Workspace 10

I

IDE Integrations 5

O

Opening Describe Developer in Eclipse 8

R

Reconnecting To Describe Developer in Eclipse 8

Running the Forte/Describe Integration 9